A Fast New Cryptographic Hash Function Based on Integer Tent Mapping System

Jiandong Liu
Information Engineering College, Beijing Institute of Petrochemical Technology, Beijing, China
Liujiandong@bipt.edu.cn

Xiahui Wang, Kai Yang, Chen Zhao
Information Engineering College, Beijing Institute of Petrochemical Technology, Beijing, China
{Wnagxiahui, Yangkai0212, zhao_chen}@bipt.edu.cn

***Abstract*—This paper proposes a novel one-way Hash function which is based on the Coupled Integer Tent Mapping System and termed as THA (THA-160, THA-256). The THA-160 compresses a message of arbitrary length into a fingerprint of 160 bits, well the THA-256 compresses a message of arbitrary length into a fingerprint of 256 bits. The algorithm adopts a piecewise message expansion scheme. Compared with SHA-1 and SHA-256 message expansion, the message expansion scheme has enhanced the degree of nonlinear diffusion of the message expansion, and thus increased the computation efficiency. In addition, as the major nonlinear component of compression function, the traditional logic functions are replaced by the integer tent map, and so the scheme has ideal properties of diffusion and confusion. Furthermore, the parallel iteration structure is adopted in the compression functions, which is advantageous to high speed parallel operation of software and hardware. Preliminary security testing indicates that, this Hash function has a high degree of security, and it can be realized easily with great rapidity. Therefore, it is an ideal substitution for conventional Hash function.**

***Index Terms*—cryptography, Hash function, tent map, message expansion, diffusion**

I. INTRODUCTION

Hash functions are essentially easy to compute functions that produce a digital fingerprint of messages or data and they are ubiquitous in today's IT systems and have a wide range of applications in security protocols and schemes, such as providing software integrity, digital signatures and password protection. Furthermore, hash function algorithms have been used for constructing pseudo-random number generators, key derivation algorithms, message authentication codes, as well as stream ciphers and block ciphers.

In recent years, an extensive attention has been given to the design and analysis of hash functions within the field of cryptology. The attacks used to find the "breakthrough" collisions presented at the rump session of Crypto'04 by Wang et al.[10]. At that conference, Wang announced the presence of collisions within for MD4, MD5, HAVAL-128 and RIPEMD. "Finding collisions by SHA-1" by Wang and Yu is another breakthrough in the hash functions history [11].

Based on the MD iteration structure [3], the conventional Hash functions (MDx and SHA) have many common design guidelines. The designs of their mixed operations for each round are very similar, and all of them adopt integer modulo addition and logic function; therefore, many Hash functions have been breached successively within a short period of time, which indicates the defects in the design of the Hash functions.

In recent years, in order to obtain more secure hash functions, the research on constructing one-way hash function has been carried out and it has achieved progress by utilizing the sensitivity of the chaotic system to the variation of the variables and parameters [12, 18, 17, 13, 8, 1, 2, 4, 7, 20]. But at present, the Hash function structure scheme, which is based on the Chaos Theory, is not acceptable to the world due to the degradation of dynamic properties of digital chaotic system and the defects of the structure scheme itself [5, 14].

To work out the SHA-3, a new cryptographic Hash algorithm standard, the analysis and assessment of the Hash algorithm Competition, held by National Institute of Standards and Technology (NIST), is quite popular all around the world. Most of the schemes of SHA-3 algorithm, accepted by NIST, are improvements of the conventional algorithm, which are filled with the elements of old cryptographic algorithm. However, the paper combines the research result of the chaotic hash function with the conventional hash function structure method, improves the design standard and the message expansion mode of conventional hash function and eventually proposes a coupled integer tent mapping system-based cryptographic one-way hashing algorithm termed as THA(Tent map-based hash algorithm). THA-160 compresses a message of arbitrary length into a fingerprint of 160 bits; THA-256 compresses a message of arbitrary length into a fingerprint of 256 bits. The algorithm adopts a piece-wise message expansion scheme. Compared with SHA-1 and SHA-256 message expansion, the message expansion scheme has enhanced the degree of nonlinear diffusion of the message expansion, and thus

doi:10.4304/jcp.7.7.1671-1680

increases the computation efficiency. In addition, the integer tent map replaces the traditional logic functions as the major nonlinear component of compression function, and so the scheme has ideal properties of diffusion and confusion. Furthermore, the parallel iteration structure is adopted in the compression functions, which is advantageous to high speed parallel operation of software and hardware.

II. INTEGER TENT MAP

A. The Standard Uniform Distribution Property of the Integer Tent Map

The tent map is 1-D and piecewise-linear map as follows [18]:

$$F_\alpha : x_i = \begin{cases} \dfrac{x_{i-1}}{\alpha}, & 0 \le x_{i-1} < \alpha, \\ \dfrac{1-x_{i-1}}{1-\alpha}, & \alpha \le x_{i-1} \le 1. \end{cases} \quad (1)$$

This mapping is chaotic within the real number field and one of its excellent characteristics is the uniform distribution function. When the parameter α=0.5, real number operations will be turned into integer operations:

$$\mathrm{F} : x_{n+1} = \begin{cases} 2x_n + 1, & x_n \in [0, 2^{k-1}) \\ 2(l - x_n), & x_n \in [2^{k-1}, l] \end{cases} \quad (2)$$

In which, $l=2k-1$ and the map F is termed as Integer Tent Map. The multiplication (division) within the integer fields in the formula (1) is turned into shifting function in the formula (2).

Definition: if the possible values of the random variable X in the set of integer are 0, 1… l and they are distributed as:

$$p_k = P(X = k) = \frac{1}{2^{32}}$$

Then, the random variable X will be submitted to the uniform distribution in the integer [0, 1].

Theorem: The distribution of $y=F(x)$ for integer $x \in [0,l]$ of randomly chosen is the standard uniform distribution $U[0, l]$.

Proof: Considering that x is randomly chosen from [0, l], x obeys the standard uniform distribution $U[0,l]$, i.e.

$$\forall\, \varepsilon \in (0, l) \qquad P(x_n < \varepsilon) = \frac{\varepsilon}{2^k}$$

Also as $\varepsilon<l$, so $\varepsilon \le 2^k-2$, then:

$P(x_{n+1}<\varepsilon)=P(F(x)<\varepsilon)$
$= P(F(x)<\varepsilon, 0\le x<2^{k-1}) + P(F(x)<\varepsilon, 2^{k-1}\le x\le l)$
$=P(2x+1<\varepsilon, 0\le x<2^{k-1}) + P(2(l-x)<\varepsilon, 2^{k-1}\le x\le l)$
$=P(2x+1<\varepsilon, 0\le 2x\le 2^k-2) + P(x>2^k-\varepsilon/2-1, 2^{k-1}\le x\le 2^k-1)$
$= P(0\le x<\varepsilon/2-1/2) + P(2^k-\varepsilon/2-1<x\le 2^k-1)$

When ε is even:

$$P(0\le x<\varepsilon/2-1/2) = P(0\le x<\varepsilon/2) = \varepsilon/2^{k+1};$$

$$P(2^k-\varepsilon/2-1<x\le 2^k-1) = P(2^k-\varepsilon/2\le x\le 2^k-1) = 1-(1-\varepsilon/2^{k+1});$$

$$P(x_{n+1}<\varepsilon) = \varepsilon/2^{k+1} + 1-(1-\varepsilon/2^{k+1}) = \varepsilon/2^k$$

When ε is odd,

$$P(0\le x<\varepsilon/2-1/2) = P(0\le x<(\varepsilon-1)/2) = (\varepsilon-1)/2^{k+1};$$

$$P(2^k-\varepsilon/2-1<x\le 2^k-1) = P(2^k-(\varepsilon+1)/2\le x\le 2^k-1) = 1-(1-(1-(\varepsilon+1)/2^{k+1}));$$

$$P(x_{n+1}<\varepsilon) = (\varepsilon-1)/2^{k+1} + 1-(1-(1+\varepsilon)/2^{k+1}) = \varepsilon/2^k$$

Therefore, F obeys the standard uniform distribution $U[0, \boldsymbol{l}]$. Integer tent map has the nonlinear nature of rolled-out and folded-over, and its rolled-out characteristic will finally lead to the exponent separation of the adjacent points. While its folded-over characteristic keeps generating sequence with boundary, and makes the mapping irreversible.

B. Realization approach of Integer Tent Map

Suppose $D=2^{31}$, operator (? :) in C language could be used in $GF(2^{32})$ to describe Eq.(2) as follows:

$$x_{n+1}=x_n<D?(x_n<<1)+1:\sim x_n<<1 \quad (3)$$

Obviously, Eq. (3) could be realized through simple logical judgment, logic reverse and shifting operations. If it is realized by assembly language or hardware, then the operation could be further simplified: Testing whether the highest digit is 0 or not, if it is 0, then shift one digit to left and add 1; otherwise, shift one digit left after bitwise complement operation. Simple as the operation is, it will be translated into jump instruction if we come cross *if* block here. Considering the effect of the jump instruction to the pipeline efficiency of modern Super Pipeline CPU, the water flow could be blocked whenever the branch forecast fails and thus prolong the instruction period. To avoid *if* jump, we adopt another equivalent form of Eq. (2) (described by C language operator):

$$x_{n+1}=((x_n \,\hat{}\, (-(x_n>>31)))<<1)|(!(x_n>>31)) \quad (4)$$

It can be seen that, simple instructions are solely used in Eq. (4) which completely avoids the jumping operation.

C. Coupled Integer Tent Mapping System

Since the Integer Tent Maps are defined within the integer set, the iterative sequence generated from the formula must turn to periodical form. In the iteration process, the disturbance is applied to breaking the inherent cycle of the Integer Tent Maps to improve the Ergodicity of the Integer Tent Maps. As a result, the iteration series of the systems are randomized. Thus, we build the following coupled mapping system model:

$$x_j^{(i+1)} = F(x_j^{(i)}) + (F(x_{j-1 \bmod p}^{(i)})<<<21) + (F(x_{j+1 \bmod p}^{(i)})<<<11) \quad (5)$$

The operators in the formula are described in the following text. Fig.1 is the result of the state variable

after 12000 iterative operations in the Eq. (5) when a set of initial values are selected randomly.

III. PIECE-WISE MESSAGE EXPANSION

A. SHA-1 Message Expansion

The iterative hash functions can be divided into MDx and SHA family and the important feature that distinguishes SHA family from MDx is that the message expansion of SHA family is conducted in the manner of recursive expansion. The so-called message expansion refers to a process in which the current input message block is expanded into a number of words. For SHA-1, the message block is represented by 32-bit words, denoted by M_t, with $0 \le t \le 15$. In the message expansion, this input is expanded into 80 32-bit words W_t, also denoted as the 2560-bit expanded message row-vector. The words W_t are defined as follows [6]:

$$W_t = \begin{cases} M_t, & 0 \le t \le 15 \\ (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <<< 1, & 16 \le t \le 79 \end{cases} \quad (6)$$

In the message block space of SHA-1, the searching space generated from 16 32-bits free variables could reach as high as 2^{512}. However, as the message expansion adopted by SHA-1 is a kind of quasi-cyclic codeword with linear features, the searching scope of minimum message weight could be narrowed to 2^{38}. In the message space of 2^{512}, the minimum weight of SHA-1 is 44. To describe the process of the message expansion, we randomly select a message block (512bits), then observe the effects on 2048(64*32) expansion bits in the subsequent recursive process after each change of one single bit. The results are illustrated in Fig.2.

B. SHA-256 Message Expansion

It should be noted that SHA-256, unlike SHA-1, has only 64 steps. Two reasons can be used to explain the safety of SHA-256: firstly, since SHA-256 produces a 256 bit output, its nonlinear block cipher has eight 32 bit registers instead of five possessed by SHA-1. This in turn means that any disturbance introduced using the expanded message words W_i carries on for at least eight rounds(instead of five), and hence the probability of forcing local collisions goes down. Secondly, the SHA-256 message expansion code itself is more involved and possibly has better minimum distance. It is defined as follows:

$$W_t = \begin{cases} M_t, & 0 \le t \le 15; \\ \delta_1(W_{t-2}) + W_{t-7} + \delta_0(W_{t-15}) + W_{t-16}, & 16 \le t \le 63 \\ where & \\ \quad \delta_0(x) = (x >>> 7) \oplus (x >>> 18) \oplus (x >> 3) & \\ \quad \delta_1(x) = (x >>> 17) \oplus (x >>> 19) \oplus (x >> 10) & \end{cases} \quad (7)$$

We randomly select a message block (512bits), then observe the effects on 1536(48*32) expansion bits in the subsequent recursive process after each change of one single bit. The results are illustrated in Fig.3.

As the modulo addition operation with nonlinear features is firstly adopted in SHA-256, the effect of message expansion of SHA-256 is better than that of SHA-1. However, since too many circular shift operation and bitwise XOR operation (each recursive computation needs four times of circular shift operations, twice of right-shift operations, four times of bitwise XOR operations, and three times of addition modulo operations) have been adopted in the message expansion of the SHA-256, its computation efficiency is rather low. The message expansion modes of the other members of the SHA-2 family (SHA-384, SHA-512) are very similar to that of SHA-256.

The complexity of the differential cryptanalysis of Hash function is directly proportional to the differential diffusion degree of the message in compression functions (a useful heuristic, which is often used in the analysis of SHA-0 and SHA-1, is each bit difference in the key (in the latter 64 rounds) lowers the probability of success on average by a factor of $2^{2.5}$). The larger the minimum weight of the massage expansion is, the more complex the differential cryptanalysis of the compression

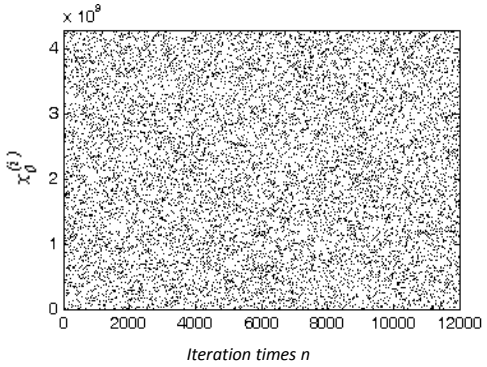

Figure 1. Distribution Diagram of Series for Coupled Integer Tent Mapping System

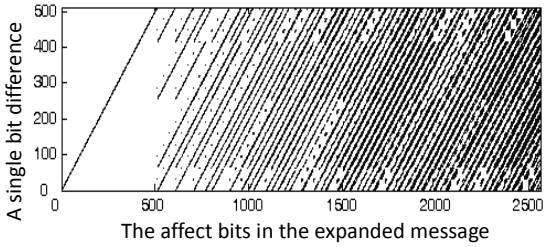

Figure 2. SHA-1message expansion

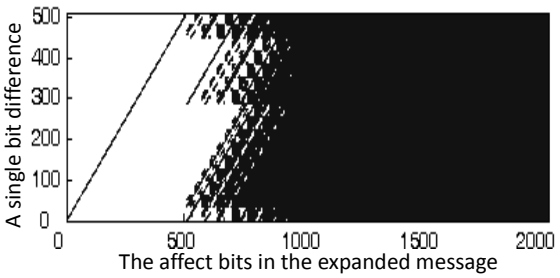

Figure 3. SHA-256 message expansions

functions will be. Therefore, in the next generation of Hash function design, it is necessary to design a better message expansion mode.

C. Piece-wise Message Expansion Scheme of THA-160

The paper [9] improves the message expansion mode of SHA-1, which enhances the degree of differential diffusion and the quantity of computation. With reference to the design idea of the paper [9], we work out a piece-wise message expansion scheme. Firstly, ten times of recursive expansion is conducted through circular shift and addition modulo modes aiming to enhance the relevancy between each bit of expansion message and 512 bits message block. Then process codeword message expansion by a simple addition modulo. The piece-wise message expansion is defined as follows:

$$W_t = \begin{cases} M_{t,} & 0 \le t \le 15; \\ W_{t-3} + W_{t-8} + W_{t-14} + W_{t-16} & \\ \quad + (W_{t-1} \oplus W_{t-2} \oplus W_{t-15}) <<< 13 & \\ \quad + (W_{t-1} \oplus W_{t-4} \oplus W_{t-11}) <<< 23, & 16 \le t \le 25; \\ W_t = W_{t-1} + W_{t-2} + W_{t-9} + W_{t-16}, & 26 \le t \le 80. \end{cases} \quad (8)$$

The author tested the differential diffusion characteristics of the Eq. (8) and observed the influence of each change of one single bit upon 2048(64*32) expansion bits in the subsequent recursive process. The results are illustrated in Fig.4.

We attempt to get an idea about the effect of all the changes in the above message expansion schemes. For the single bit differences, Table 1 illustrates the comparison of the number of affected bits in message expansion when the variants are 36, 64, 80 steps. It must be pointed out that, the expanded message code weight acquired from the experiment is the result of a block of randomly selected message (512bit) and 1 bit change each time. The experimental result given in table 1 sufficiently indicates that the piece-wise message expansion can promote the diffusion degree of message difference. In addition, under the condition of the main frequency of P4, 2.0GHz, the author tested the computation efficiency of SHA-1, SHA-256 and the piece-wise expanded message. When the recursive process is 80 steps, the execution efficiency of the piece-wise message expansion mode is improved by 29% compared with that of SHA-1, and by 48% with that of SHA-256.

TABLE I.
COMPARISON OF THE NUMBER OF AFFECTED BITS FOR A SINGLE BIT DIFFERENCE IN MESSAGE EXPANSION

	SHA-1	SHA-256	Piece-wise Message Expansion
Min(36steps)	13	78	192
Max(36steps)	23	293	304
Min(64steps)	67	503	612
Max(64steps)	102	735	777
Min(80steps)	107	755	852
Max(80steps)	174	963	1042

D. Piece-wise-nonlinear Message Expansion Scheme of THA-256

From the realization process of integer tent map, we can conclude that integer tent map is extremely sensitive to the highest digit of the message. Therefore, ten times of recursive expansion is conducted through circular shift and addition modulo modes before using integer tent maps to make message expansion. The purpose of these operations is to enhance the relevancy between high bit digit of expansion message and 512 bits message block. The piece-wise-nonlinear message expansion is defined as follows:

$$W_t = \begin{cases} \boldsymbol{M}_{t,} & 0 \le \boldsymbol{t} \le 15; \\ \boldsymbol{W}_{t-3} + \boldsymbol{W}_{t-8} + \boldsymbol{W}_{t-14} + \boldsymbol{W}_{t-16} & \\ \quad + (\boldsymbol{W}_{t-1} \oplus \boldsymbol{W}_{t-2} \oplus \boldsymbol{W}_{t-15}) <<< 13 & \\ \quad + (\boldsymbol{W}_{t-1} \oplus \boldsymbol{W}_{t-4} \oplus \boldsymbol{W}_{t-11}) <<< 23, & 16 \le \boldsymbol{t} \le 25; \\ \boldsymbol{W}_{t-1} = ((\boldsymbol{W}_{t-1} \oplus (-(\boldsymbol{W}_{t-1} >> 31))) << 1) \mid (!(\boldsymbol{W}_{t-1} >> 31)), & \\ \boldsymbol{W}_t = \boldsymbol{W}_{t-2} \oplus \boldsymbol{W}_{t-3} \oplus \boldsymbol{W}_{t-9} \oplus \boldsymbol{W}_{t-16}, & 26 \le \boldsymbol{t} \le 63. \end{cases} \quad (9)$$

The author tested the differential diffusion characteristics of the Eq. (9) and observed the influence of each change of one single bit upon 1536(48*32) expansion bits in the subsequent recursive process. The results are illustrated in Fig.5.

Table 2 illustrates the comparison of SHA-1, SHA-256 and THA-256. Having adopted the modulo addition operation, with nonlinear features, Eq. (9) further introduces integer tent map and makes the message diffusion and propagation very complex. The experimental result given in table 2 sufficiently indicates that the piece-wise-nonlinear message expansion can promote the diffusion degree of message difference. In

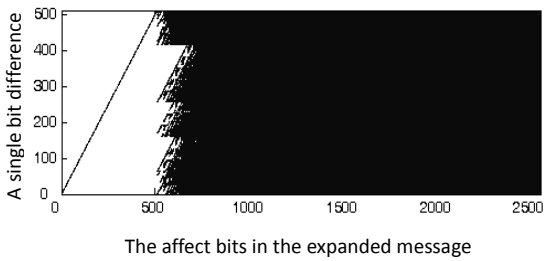

Figure 4. Piece-wise Message Expansion

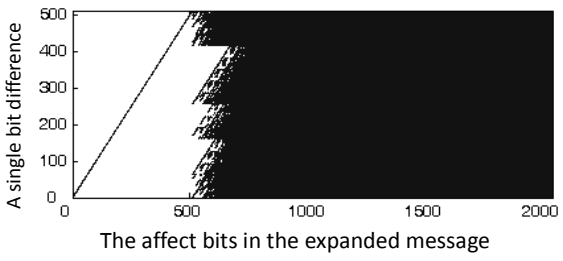

Figure 5. Piece-wise- nonlinear message expansion

TABLE II.
COMPARISON OF THE NUMBER OF AFFECTED BITS FOR A SINGLE BIT DIFFERENCE IN MESSAGE EXPANSIONS

	SHA-1	SHA-256	THA-256
Min	107	503	626
Max	174	735	779

addition, under the condition of the main frequency of P4, 2.0GHz, the computation efficiency of the piece-wise-nonlinear expanded message given by this paper is 10% faster than that of the SHA-256 expanded message.

Ⅳ. THA CONSTRUCTING

The following notations are used: + denotes addition modulo 2^{32}, ~ bitwise complement operation, ∨ bitwise OR, ⊕ bitwise XOR, << and >> left-shift and right-shift operation, <<< left bit-wise rotation.

A. THA-160

(1) Padding the Message: Suppose that the length of the message, M, is l bits. Append the "1" to the end of the message, followed by k zero bits, where k is the smallest, non-negative solution to the equation $l+1+k=448 \bmod 512$. Parse the padded message into N 512-bit message blocks, $M_1 \cdots M_i \cdots M_N$.

(2) Parse the message block M_i into sixteen 32-bit message words: $m_0, m_1 \cdots m_{15}$.

(3) Five 32-bit initial Hash Value:

$x_0^{(0)}$=0x01234567, $x_1^{(0)}$=0x89abcdef,

$x_2^{(0)}$=0x3210fedc, $x_3^{(0)}$=0xba987654,

$x_4^{(0)}$=0x02468ace.

(4) Five 32-bit initial parameter:

k_0=0x5a827999, k_1=0x6ed9eba1,

k_2=0x8f1bbcdc, k_3=0xca62c1d6,

k_4=0x5793c62a.

(5) Piece-wise message expansion scheme (Eq. (8)) is adopted in the message expansion.

(6) Parallel iterated processing based on Coupled Integer Tent Mapping System (Eq. (5)).

1) $x_0=x_0^{(0)}$, $x_1=x_1^{(0)}$, $x_2=x_2^{(0)}$, $x_3=x_3^{(0)}$, $x_4=x_4^{(0)}$;

2) For t=0 to 12

{ $k=t*5+16$;

G_i= ((($x_i \oplus$ (-(x_i>>31)))<<1)

$\vee$(!(x_i >>31)))+W_{i+k} i=0,...,4

x_i= G_i + (($G_{i\text{-}1 \bmod 5} \oplus G_{i+2 \bmod 5}$) <<<21)

+(($G_{i+1 \bmod 5} \oplus G_{i+3 \bmod 5}$) <<<11)+$k_i$ i=0,...,4

}

3) Adding $x_0^{(0)} \cdots x_4^{(0)}$ to $x_0 \cdots x_4$:

$x_0=x_0+x_0^{(0)}$, $x_1=x_1+x_1^{(0)}$, $x_2=x_2+x_2^{(0)}$,

$x_3=x_3+x_3^{(0)}$, $x_4=x_4+x_4^{(0)}$;

4) Repeating step 2) ~3) through to the last message block.

5) The result 160-bits message digest of the message, M, is $x_0 || x_1 || x_2 || x_3 || x_4$.

B. THA-256

(1) Padding the Message: Suppose that the length of the message, M, is l bits. Append the "1" to the end of the message, followed by k zero bits, where k is the smallest, non-negative solution to the equation $l+1+k=960 \bmod 1024$. Parse the padded message into N 1024-bit message blocks, $M_1 \cdots M_i \cdots M_N$.

(2) Parse the message block M_i into two groups, Each one is constituted by sixteen 32-bit message words, respectively is $m_0, m_1, \ldots, m_{15}$ and $\dot{m_0}, \dot{m_1}, \ldots, \dot{m_{15}}$.

(3) Eight 32-bit initial Hash Value:

$\boldsymbol{x_0^{(0)}}$=0x6a09e667, $x_1^{(0)}$=0xbb67ae85,

$x_2^{(0)}$=0x3c6ef372, $x_3^{(0)}$=0xa54ff53a,

$x_4^{(0)}$=0x510e527f, $\boldsymbol{x_5^{(0)}}$=0x9b05688c,

$\boldsymbol{x_6^{(0)}}$=0x1f83d9ab, $\boldsymbol{x_7^{(0)}}$=0x5be0cd19.

(4) Eight 32-bit initial parameter:

k_0=0x5a827999, k_1=0x6ed9eba1,

k_2=0x8f1bbcdc, k_3=0xca62c1d6,

k_4=0x999728a5a,k_5=0x1abe9de6,

k_6=0xcdcbb1f8, k_7=0x6d1c26ac.

(5) Piece-wise- nonlinear message expansion scheme (Eq. (9)) is adopted in the message expansion for m_0, m_1, ..., m_{15} *and* $\dot{m_0}, \dot{m_1}, \ldots, \dot{m_{15}}$, which obtain message words sequences W_0, W_1..., W_{63} and $\dot{W_0}, \dot{W_1}, \ldots, \dot{W_{63}}$ respectively.

(6) Parallel iterated processing based on Coupled Integer Tent Mapping System (Eq. (5)).

1) $x_0=x_0^{(0)}$, $x_1=x_1^{(0)}$, $x_2=x_2^{(0)}$, $x_3=x_3^{(0)}$,

$x_4=x_4^{(0)}$, $x_5=\boldsymbol{x_5^{(0)}}$, $x_6=\boldsymbol{x_6^{(0)}}$, $x_7=\boldsymbol{x_7^{(0)}}$;

2) For t=0 to 11

{When t *is* even,

k=(t/2)*8+16;

G_i= ((($x_i \oplus$ (-(x_i>>31)))<<1)

$\vee$(!(x_i >>31)))+W_{i+k} i=0,...,7

$$x_i = G_i + ((G_{i+1 \bmod 8} \oplus G_{i+2 \bmod 8} \oplus G_{i+3 \bmod 8} \oplus G_{i+4 \bmod 8}) <<<11) + ((G_{i-1 \bmod 8} \oplus (G_{i-2 \bmod 8} \vee (\sim G_{i-3 \bmod 8}))) <<< 21) + k_i \quad i=0,\ldots,7$$

When t is odd,

$k=((t-1)/2)*8+16;$

$$G_i = (((x_i \oplus (-(x_i >> 31))) << 1) \vee (!(x_i >> 31))) + \boldsymbol{W}'_{i+k} \quad i=0,\ldots,7$$

$$x_i = G_i + ((G_{i+1 \bmod 8} \oplus (G_{i+2 \bmod 8} \vee (\sim G_{i+3 \bmod 8}))) <<< 21) + ((G_{i-1 \bmod 8} \oplus G_{i-2 \bmod 8} \oplus G_{i-3 \bmod 8} \oplus G_{i-4 \bmod 8}) <<< 11) + k_i \quad i=0,\ldots,7$$

}

3) Adding $x_0^{(0)} \ldots \boldsymbol{x}_7^{(0)}$ to $x_0 \ldots x_7$:

$$x_0=x_0+x_0^{(0)},\ x_1=x_1+x_1^{(0)},\ x_2=x_2+x_2^{(0)},$$
$$x_3=x_3+x_3^{(0)},\ x_4=x_4+x_4^{(0)}\ x_5=x_5+\boldsymbol{x}_5^{(0)},$$
$$x_6=x_6+\boldsymbol{x}_6^{(0)},\ x_7=x_7+\boldsymbol{x}_7^{(0)};$$

4) Repeating step 2) ~3) through to the last message block.

5) The result 256-bit message digest of the message, M, is $x_0||\ x_1||\ x_2||\ x_3||\ x_4||\ x_5||\ x_6||\ x_7$.

C. Characteristics Analysis of THA

(1) Substitute Integer Tent Map for conventional logic functions. In the conventional Hash functions design, logic function (round function) is taken as the major nonlinear component of mixed operation. HAVAL adopts logical functions that have high non-linearity, and, is 0-1 balance and satisfy the Strict Avalanche Criterion [19]. However, adopting modular differential method could find one collision of HAVAL-128 within the period of 2^7 times of HAVAL-128 operations [15]. It indicates that the selection of ideal logic function does not make the Hash functions safe enough. In logic function, various inputs could generate the same outputs, which we call as value collision of logic function. Due to the value collision of logic function, the interior of compress function could generate differential convergence. Wang et al. just use this feature of logic function to control differential convergence, and find the collision of the MD5 and HAVAL-128 by combining the technologies such as diversity of formula of integer mode reduction difference, transfer characteristics of left shift difference, and bit modification, etc.

Integer tent maps are uniformly distributed and have good nonlinear features, and they are easily realized with fast computation speed, but they are 1-1 mapping between the single variables and have no value collision problems. Diffusion mechanisms of conventional Hash functions (MD5, SHA-1, etc) are realized by the mixed operation of modular 32 addition, and bitwise Boolean Operation. The change of each bit is realized by shifting or carrying to affect other bits. THA still has logic shifting and diffusion mechanism of addition modulo 2^{32}. For THA, the important thing is that, whenever the mappings are transformed, besides moving the state variable 1 bit to the left, the difference characteristics of the highest bit could also lead this state variable to generate the maximum extended codeword weight. Furthermore, each message bit difference has chance to trigger dynamic differential diffusion through circular shift operation and coupling diffusion of coupled integer tent mapping system.

(2) Message expansion mode has been improved, and the diffusion and confusion of message in mixed operation is accelerated in mixed function.

(3) The operation steps of conventional Hash functions can only be realized by serial mode. The iteration structure of internal compression function of THA is different from MD5 and SHA, etc. Its realization is suitable to the parallel operation, and can avoid the problem of local collision existing in the Hash functions of SHA family [11].

(4) THA algorithm inherits the design idea of the conventional Hash algorithm featured by simple description and easy realization, borrows and improves the tent map model which is widely applied in the research of chaotic cryptography. With these merits, it transforms the tent map model from real number field to integer set and realizes the diffusion and confusion of message by adopting rolled-out and folded-over nonlinear nature of the tent map as well as its feature of uniform distribution. All of the algorithms adopt simple operations based on 32 bit operands, which is helpful to achieve the high speed operation of both software and hardware.

Ⅴ. STATISTIC ANALYSIS ON DIFFUSION AND CONFUSION

A dependence test method is introduced in the statistical analysis of the security of block cipher [16]. In this paper, the author adopted the test method to analyze statistically the security of Hash algorithm on diffusion and confusion.

For H is a Hash of n input bits and m output bits, the input vector $\boldsymbol{x}=(x_1,\ldots,x_n) \in (0,1)^n$, the vector $\boldsymbol{x}^{(i)} \in (0,1)^n$ denotes the vector obtained by complementing the i-th bit of $\boldsymbol{x}$(for $i=1,\ldots,n$). The output vector is $H(x)$, $H(x^{(i)}) \in (0,1)^m$ corresponding to the input vector $\boldsymbol{x}$, $\boldsymbol{x}^{(i)}$.

The Hamming weight $w(x)$ of x is defined as the number of nonzero components of x.

A Hash function H of n input bits and m output bits is said to be complete, if each output bit depends on each input bit.

A Hash function H has the avalanche effect, if an average of 1/2 of the output bits changes whenever a single input bit is complemented.

A Hash function H satisfies the strict avalanche criterion, if each output bit changes with a probability of 1/2 whenever a single input bit is complemented.

For X is a set of input bits of Hash function. $a_{ij}=\#\{x \in X | (H(x))_j \neq (H(x^{(i)}))_j\}$(for $i=1,2,\ldots,n$ and $j=1,2,\ldots,m$) denotes the number of inputs for which complementing

the *i*-th input bit results in a change of the *j*-th output bit; $b_{ij}=\#\{x \in X|\ w(H(x^{(i)}) \oplus H(x)) =j\}$(for i=1,2, ...,n and j=1, 2, ...,m) denotes the number of inputs for which complementing the *i*-th input bit results in a change of *j* output bits.

The degree of completeness is defined as:

$$d_c = 1 - \frac{\#\{(i,j)|a_{ij} = 0\}}{nm}$$

The degree of avalanche effect is defined as:

$$d_a = 1 - \frac{\sum_{i=1}^{n}\left|\frac{1}{\#X}\sum_{j=1}^{m} 2jb_{ij} - m\right|}{nm}$$

And the degree of strict avalanche criterion is defined as:

$$d_{sa} = 1 - \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\left|\frac{2a_{ij}}{\#X} - 1\right|}{nm}$$

If *H (.)* is stochastic transformation, the paper gives the following conclusion [21]:

1) The tested sample amount X should be at least $nm\times(z_{\alpha/2})^2$;

2) $p(d_c)=1-2^{-\#X}\approx 1.0$;

3) $E\{d_a\}=1.0-\sqrt{2/(\pi\times m\times \#X)}$, confidence interval:

$(E\{d_a\}-z_{\alpha/2}\sqrt{1/(n\times m\times \#X)}, E\{d_a\}+z_{\alpha/2}\sqrt{1/(n\times m\times \#X)})$;

4) $E\{d_{sa}\}=1.0-\sqrt{2/(\pi\times \#X)}$, confidence interval:

$(E\{d_{sa}\}-z_{\alpha/2}\sqrt{1/(n\times m\times \#X)}, E\{d_{sa}\}+z_{\alpha/2}\sqrt{1/(n\times m\times \#X)})$.

Where $z_{\alpha/2}$ is the *z*-value with an area $\alpha/2$ in the right tail of a standard normal distribution.

It is necessary to point out that the ideal Hash functions should be a random mapping from all possible input values to limited possible output values. Strictly speaking, the Hash function like random mapping does not exist, because Hash function is deterministic, and the certainty and uniform output characteristics mean that the entropy of the output is larger than the input entropy. However, according to the Shannon Entropy Theory, it is impossible that a deterministic function could amplify entropy. Our design aims at making it impossible to distinguish the probability distribution of Hash functions and random mapping from each other. If the tested values of d_c, d_a, d_{sa} of a real hash function fall into the confidence interval, it indicates that the Hash algorithm satisfies the basic requirements of nonlinear diffusion.

A. Statistic Analysis of THA-160

Given input length n=512 bits, output length m=160 bits, under the significance level of α=0.05, we can get the following result theoretically:

1) $z_{\alpha/2}$=1.96, sample capacity X is 320000;

2) d_c=1.000000;

3)$E\{d_a\}$=0.999888, the confidence interval is approximated as (0.999876, 0.999900);

4)$E\{d_{sa}\}$=0.998589, the confidence interval is approximated as (0.998577, 0.998601).

We randomly select 320000 blocks of 512bit (taken Rand( ) from Visual C) sample set X as input message words of THA-*160*. The actual test results are shown in Table 3.

The same test on SHA-1 has been carried out and the actual test results are shown in Table 4.

It can be seen from Table 3 that, under the significance level of α=0.05, the values of d_c, d_a, and d_{sa} remain surprisingly stable, and fall into their own confidence interval after iterative 1 step, and the basic requirements of nonlinear diffusion are satisfied, which is notably better than the experimental results of SHA-1 given in Table 4 (Since the THA-160 algorithm adopts parallel iteration structure, and its iterative 1 step could roughly be corresponding to the iterative 5 steps of SHA-1, successively, iterative 2 steps of THA-160 could be corresponding to the iterative 10 steps of SHA-1, ……).

B. Statistic Analysis of THA-256

In order to compare with SHA-256, given input length n=512 bits, output length m=256 bits, under the significance level of α=0.05, we can get the following result theoretically:

TABLE III.
THE DEGREES OF COMPLETENESS, OF AVALANCHE EFFECT, AND OF STRICT AVALANCHE CRITERION FOR THA-160 WITH INCREASED NUMBER OF ITERATIONS

number of Iterations	d_c	d_a	d_{sa}
1	0.982336	0.860203	0.796704
2	1.000000	0.999884	0.998579
3	1.000000	0.999891	0.998586
4	1.000000	0.999887	0.998587
5	1.000000	0.999888	0.998584
6	1.000000	0.999889	0.998586
7	1.000000	0.999894	0.998587
8	1.000000	0.999881	0.998587
9	1.000000	0.999885	0.998589
10	1.000000	0.999884	0.998587
11	1.000000	0.999890	0.998596
12	1.000000	0.999888	0.998587
13	1.000000	0.999900	0.998585

TABLE IV.
THE DEGREES OF COMPLETENESS, OF AVALANCHE EFFECT, AND OF STRICT AVALANCHE CRITERION FOR SHA-1

number of Iterations	d_c	d_a	d_{sa}
1	0.006396	0.001559	0.000574
3	0.051648	0.009894	0.004823
5	0.145032	0.036693	0.024572
7	0.267920	0.092986	0.073287
10	0.456213	0.237795	0.212311
15	0.768115	0.544870	0.518823
20	0.987427	0.845407	0.823081
25	1.000000	0.993809	0.989917
30	1.000000	0.999892	0.998594
40	1.000000	0.999885	0.998594
60	1.000000	0.999887	0.998587
80	1.000000	0.999896	0.998589

1) $z_{\alpha/2}$=1.96, sample capacity X is 503526;
2) d_c=1.000000;
3)E{d_a}=0.999930, the confidence interval is approximated as (0.9999221, 0.9999373);
4)E{d_{sa}}=0.998876, the confidence interval is approximated as (0.9988679, 0.9988831).

We randomly select 503526 blocks of 512-bit (taken Rand( ) from Visual C) sample set X as input message words of THA-256. The actual test results are shown in Table 5.

The same test on SHA-256 has been carried out and the actual test results are shown in Table 6.

Ⅵ. Analysis of Collision Resistance

Collision attack refers to the process of finding out two different messages and making them produce the same Hash results. In this paper, the author intends to test the collision resistance ability of the algorithm quantitatively. First, randomly select a section of plaintext is in the plaintext space, and work out its Hash value which will be expressed in single byte character. Then another new Hash result will be available by randomly selecting and changing the value of one-bit in the plaintext. The distance between the two Hash values is defined as:

$$d = \sum_{i=1}^{S} \left| t(e_i) - t(e_i') \right|$$

Among which, e_i and e_i' stand for the i-th character of the initial and the new Hash values respectively. S stands for the number of single byte character expressed in Hash value. The function $t(\ )$ transform e_i and e_i' into corresponding numbers in decimal system. If the two Hash values are respectively formed by two independent and uniformly distributed random sequences, theoretically, the average distance of per character between the two Hash values should be 85.33[8].

To THA-160, the average distance per character of the Hash values is tested through random selecting of input samples with an input length of n=512 bits. After statistical tests of 100,000 times, the actual test results are shown in Table 7.

It can be seen from Table 4 that after iterative one-step the average distance per character of Hash values of the THA-160 algorithm tends to be steady and much closer to its theoretical value. The test result indicates that the two Hash values derived from two plaintexts that only have a difference of one-bit are statistically equal to two independent, uniform and random sequences. In this sense, it is impossible to distinguish the THA-160 from random mapping through this probability model.

For comparison, Table 8 illustrates the test results of SHA-1 collision resistance. It can be seen that just after iterative 30 steps the average distance of per character of the Hash values of SHA-1 algorithm tends to be steady.

To THA-256, the average distance of per character of the Hash values is tested by random selecting of input samples with an input length of n=1024 bits. After 100,000 times of statistical tests, the actual test results are shown in Table 9. It can be seen that after iterative two-step the average distance per character of Hash values of THA-256 tends to be steady and much close to its theoretical value.

For easy comparison, Table 10 illustrates the test results of SHA-256 collision resistance. It can be seen that just after iterative 24 steps the average distance per character of the Hash values of SHA-256 tends to be steady.

TABLE V.
THE DEGREES OF COMPLETENESS, OF AVALANCHE EFFECT, AND OF STRICT AVALANCHE CRITERION FOR THA-256

number of Iterations	d_c	d_a	d_{sa}
1	1.000000	0.994594	0.987220
2	1.000000	0.999930	0.998876
3	1.000000	0.999931	0.998875
4	1.000000	0.999928	0.998872
5	1.000000	0.999932	0.998876
6	1.000000	0.999930	0.998873
7	1.000000	0.999928	0.998881
8	1.000000	0.999928	0.998870
9	1.000000	0.999926	0.998880
10	1.000000	0.999930	0.998875
11	1.000000	0.999933	0.998874
12	1.000000	0.999930	0.998875

TABLE VI.
THE DEGREES OF COMPLETENESS, OF AVALANCHE EFFECT, AND OF STRICT AVALANCHE CRITERION FOR SHA-256

number of Iterations	d_c	d_a	d_{sa}
5	0.1904602	0.1404253	0.1190314
10	0.5031509	0.4521524	0.4298077
15	0.8159256	0.7646300	0.7419542
20	1.0000000	0.9956154	0.9890846
25	**1.0000000**	**0.9999270**	**0.9988740**
30	1.000000	0.9999302	0.9988722
35	1.0000000	0.9999284	0.9988763
40	1.0000000	0.9999263	0.9988766
45	1.0000000	0.9999290	0.9988762
50	1.0000000	0.9999309	0.9988752
55	1.0000000	0.9999307	0.9988745
64	1.0000000	0.9999287	0.9988738

TABLE VII.
ANALYSIS ON COLLISION RESISTANCE FOR THA-160

number of Iterations	1	2	3	4	5	6	7	8	9	10	11	12	13
Average distances /character	73.06	85.30	85.27	85.29	85.30	85.38	85.36	85.37	85.34	85.28	85.29	85.29	85.32

TABLE VIII.
ANALYSIS ON COLLISION RESISTANCE FOR SHA-1

number of Iterations	1	3	5	7	10	15	20	25	30	40	60	80
Average distances /character	0.006	0.244	0.714	3.885	21.22	47.33	73.22	85.14	85.34	85.40	85.35	85.36

TABLE IX.
ANALYSIS ON COLLISION RESISTANCE FOR THA-256

number of Iterations	1	2	3	4	5	6	7	8	9	10	11	12
Average distances /character	42.25	84.90	85.33	85.31	85.29	85.32	85.29	85.37	85.35	85.29	85.35	85.33

TABLE X.
ANALYSIS ON COLLISION RESISTANCE FOR SHA-256

number of Iterations	1	4	8	16	20	24	32	40	48	56	64
Average distances /character	0.01	7.09	28.05	70.71	84.92	85.37	85.33	85.43	85.31	85.27	85.27

Ⅶ. SPEED OF THA

Table 11 demonstrates the speed test results of three kinds of Hash Functions (THA-160, THA-256, SHA-256, SHA-1), which are realized by C Language, under P4, 2.0GHz condition. It can be seen from Table 11 that THA is obviously faster than conventional Hash algorithm. Moreover, its internal iteration structure makes it easy to realize the parallel operation of THA and provides a larger space to improve its time performance.

Ⅷ. CONCLUSION

Considering the severe threat to the current security of the conventional Hash functions, this paper inherits the advantages of the conventional Hash functions and designs a new Hash algorithm by improving message expansion, introducing new nonlinear components and adopting the structure of parallel iteration. The THA Hash function divides this process into two steps: the first step realizes the self-diffusion of message codeword by adopting recursive mode; the second step accomplishes the confusion and diffusion of message code words and link variables. The analyses of nonlinear diffusion and collision resistance show that THA *(*THA-160,THA-256*)* provided in this paper has a high degree of security. Statistical method cannot distinguish the THA from the actual random function. And if necessary, its security can be further enhanced by simply adding variants. The THA can be easily realized, and thus it has evident advantages in execution efficiency. We will optimize this algorithm in future advanced version, introduce message diffusion mechanism and increase round times to further improve the security of this algorithm.

TABLE XI.
COMPARISON OF SPEED FOR THA-256, SHA-256, THA-160 AND SHA-1

Data Length (B)	THA-256	SHA-256	THA-160	SHA-1
240	233.53	136.09	265.63	158.79
2048	286.65	165.96	321.58	180.84

ACKNOWLEDGMENT

This work described in this paper was supported by the Beijing Natural Science Foundation Project, China (No.4112018) and by the National Natural Science Foundation of China (41101311/D0106).

REFERENCES

[1] A Akhshani, S. Behnia, A. Akhavan, M.A. Jafarizadeh, H. Abu Hassan, Z. Hassan, Hash function based on hierarchy of 2D piecewise nonlinear chaotic maps, Chaos, Solitons & Fractals, 42(2009) 2405-2412.

[2] A Akhavan, A. Samsudin, A. Akhshani, Hash function based on piecewise nonlinear chaotic map , Chaos, Solitons & Fractals, 42(2009)1046-1053.

[3] I Damgard, A design principle for Hash functions. In Gilles Brassard, editor, Advances in Cryptology: CRYPTO89,volume 435 of Lecture Notes in Computer Science. Springer-Verlag.1989:416-427.

[4] S. Deng, Y. Li, D. Xiao, Analysis and improvement of a chaos-basedHash function construction, Communications in Nonlinear Science and Numerical Simulation, 15(2010)1338-1347.

[5] S. Li, X. Mou, et al., On the security of a chaotic encryption scheme :problems with computerized chaos in finite computing precision, Computer Physics Communications 153 (2003) 52-58.

[6] National Institute of Standards and technology Secure Hash Standard, Federal Information Processing Standards(FIPS) Publication 180-2, 2004.

[7] H. Ren, Y. Wang, Q. Xie, H. Yang, A novel method for one-way hash function construction based on spatiotemporal chaos, Chaos, Solitons & Fractals, 42(2009)2014-2022.

[8] L. Sheng , G. Li , Z. Li, One-way Hash function construction based on tangent-delay ellipse reflecting

cavity-map system, Acta Physica Sinica 55 (2006) 5700-5706 (in Chinese).
[9] C S. Jutla and A. C. PatTHAk, A Simple and Provably Good Code for SHA Message Expansion, Cryptology ePrint Archive, Report 2005/247, 2005. http://eprint.iacr.org/.
[10] X. Wang, D. Feng, X. Lai, et al., Collisions for hash functions MD4, KD5, HAVAL-128 and RIPEMD, In:Advances in Cryptology — CRYPTO 2004:The 24rd Annual International Cryptology Conference. Berlin: Springer-Verlag, 2004.
[11] X. Wang, Y. Yin, H. Yu, Finding collisions on the Full SHA-1, Advances in Cryptology--Crypto'05,LNCS 3621,pp.17-36, 2005.
[12] K.Wong, A combined chaotic cryptographic and hashing scheme, Phys Lett A 307 (2003) 292-298.
[13] Y. Wang, X. Liao, D. Xiao, et al., One-way Hash function construction based on 2D coupled map lattices,Information Sciences 178(2008) 1391-1406.
[14] J. Wang, M. Wang, Y. Wang , The collision of one keyed hash function based on chaotic map and analysis, Acta Physica Sinica 57 (2008) 2737-2742 (in Chinese).
[15] X. Wang , D. Feng , X. Yu, An attack on hash function HAVAL-128, Science in china(Information sciences) 48 (2005) 545-556.
[16] A.F. Weister, S.E.Tavares, On the design of S-boxes[A], Dvances in Cryptology-CRYPTO'85[C], Berlin:Springer -Verlag, 1986.523-533.
[17] D Xiao, X. Liao, S. Deng, One-way Hash function construction based on the chaotic map with changeable-parameter, Chaos Solitons & Fractals 24 (2005) 65-71.
[18] X.Yi, Hash function based on chaotic tent maps, IEEE transactions on circuits and systems-Ⅱ:Express briefs 52 (2005) 354-357.
[19] Y. Zheng, Josef Pieprzyk and Jennifer Seberry, HAVAL——A one-way Hashing algorithm with variable length of output. Advances in Cryptology AUSCRYPT'92 Proceedings, Springer-Verlag,1993,83-104.
[20] Q. Zhou, X. Liao, K Wong, et al., true random number generator based on mouse movement and chaotic hash function, Information Sciences 179(2009) 3442-3450.
[21] M. Zhu, B. Zhang, S. Lv, A statistical method of blockcipher on diffusion & propagation, Journal of china institute of communications 23 (2002) 122-128 (in Chinese)

Jiandong LIU, born in 1966. He has been professor of Beijing Institute of Petro-chemical Technology since 2008. His main research interests are chaos cryptography and information hiding.

Xiahui WANG, born in 1988. He has been postgraduate student of Beijing University of Chemical Technology since 2010.His main research interest is hash function.

Kai YANG, born in 1985. He has been postgraduate student of Beijing University of Chemical Technology since 2009. His main research interest is chaos cryptography.

Chen ZHAO, born in 1989. He has been postgraduate student of Beijing University of Chemical Technology since 2011.9.His main research interest is hash function.