Security analysis of a one-way hash function based on spatiotemporal chaos*

Wang Shi-Hong(王世红)[a)†] and Shan Peng-Yang(单鹏洋)[b)]

[a)] School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

[b)] International School, Beijing University of Posts and Telecommunications, Beijing 100876, China

(Received 11 January 2011; revised manuscript received 15 May 2011)

The collision and statistical properties of a one-way hash function based on spatiotemporal chaos are investigated. Analysis and simulation results indicate that collisions exist in the original algorithm and, therefore, the original algorithm is insecure and vulnerable. An improved algorithm is proposed to avoid the collisions.

Keywords: collision, spatiotemporal chaos, hash function

PACS: 05.45.–a, 05.45.Gg **DOI:** 10.1088/1674-1056/20/9/090504

1. Introduction

The application of chaos theory to information security is a promising new development. Many cryptographic systems based on chaotic dynamics have been proposed, such as chaos-based encryption ciphers,[1,2] image encryption,[3] and hash functions.[4,5] Hash functions are fundamental functions for compressing an arbitrary length message into a fixed length one, which is called a hash value. Hash functions are usually used in message authentication and digital signatures. Due to Wang's analysis of SHA-1 (SHAs stands for the secure hash algorithm standard),[6] the design and analysis of hash functions has become a challenge. Recently, some chaos-based hash functions were proposed[7−11] and the security analysis of these systems helps to improve them and to accelerate the application of the research.[12−14] In Ref. [8], Ren *et al.* proposed a keyed hash function based on spatiotemporal chaos. Compared to other hash functions,[4,7] Ren's algorithm has higher efficiency. In this paper, we will analyze the collision and statistical properties of Ren's algorithm.

The rest of this paper is arranged as follows. In Section 2, the original hash function based on spatiotemporal chaos is briefly reviewed. Section 3 shows the collision analysis and the simulation results. The statistical analysis of Ren's algorithm is given in Section 4. In Section 5, we propose an improved algorithm and give the corresponding collision analysis. Finally, the conclusion is given in Section 6.

2. Ren's algorithm

In this section, we give a description of Ren's algorithm.[8] The algorithm is a keyed hash function and the length of the key is 128-bits. We define the following operations: $\oplus$ denotes the logical XOR, $\diamondsuit$ refers to the concatenation of two-bit sequences. The $A^{[g\cdots h]}$ means bit sites g to h of variable A. Hexadecimal numbers are prefixed by "0x".

2.1. System of spatiotemporal chaos

In Ren's algorithm, a system of spatiotemporal chaos is used. Figure 1 shows the whole coupled structure of the system, where $x_{j,i}$ and $m_{j,i}$ are the state and the input variables, respectively, i and j are the time and the position indexes, respectively. The spatiotemporal chaos system is defined by the following equations:

$$\begin{aligned}
x_{0,i+1} &= (f_{0,i} + \mathrm{Trim}(f_{7,i}, 16) + \mathrm{Trim}(f_{6,i}, 16))/3,\\
x_{1,i+1} &= (f_{1,i} + \mathrm{Trim}(f_{0,i}, 8) + f_{7,i})/3,\\
x_{2,i+1} &= (f_{2,i} + \mathrm{Trim}(f_{1,i}, 16) + \mathrm{Trim}(f_{0,i}, 16))/3,\\
x_{3,i+1} &= (f_{3,i} + \mathrm{Trim}(f_{2,i}, 8) + f_{1,i})/3,\\
x_{4,i+1} &= (f_{4,i} + \mathrm{Trim}(f_{3,i}, 16) + \mathrm{Trim}(f_{2,i}, 16))/3,\\
x_{5,i+1} &= (f_{5,i} + \mathrm{Trim}(f_{4,i}, 8) + f_{3,i})/3,\\
x_{6,i+1} &= (f_{6,i} + \mathrm{Trim}(f_{5,i}, 16) + \mathrm{Trim}(f_{4,i}, 16))/3,
\end{aligned}$$

*Project supported by the National Natural Science Foundation of China (Grant No. 60973109).

†Corresponding author. E-mail: shwang@bupt.edu.cn

© 2011 Chinese Physical Society and IOP Publishing Ltd

$$x_{7,i+1} = (f_{7,i} + \mathrm{Trim}(f_{6,i}, 8) + f_{5,i})/3, \quad (1)$$

$$m_{j,i+1} = \begin{cases} m_{j,i} + a_j, & m_{j,i} + a_j < 1, \\ m_{j,i} + a_j + 1/256 \bmod 1, & m_{j,i} + a_j \geq 1, \end{cases} \quad (2)$$

where $f_{j,i}$ and $\mathrm{Trim}(x, y)$ are defined as

$$f_{j,i} = \begin{cases} y_{j,i}/b, & 0 < y_{j,i} \leq b, \\ (1 - y_{j,i})/(1 - b), & b < y_{j,i} < 1, \end{cases} \quad (3)$$

$$\mathrm{Trim}(x, y) = x \times 2^y - \lfloor x \times 2^y \rfloor, \quad (4)$$

and $\lfloor x \rfloor$ returns the largest integer not greater than x, and b is a parameter. In Eq. (3), $y_{j,i}$ is defined as

$$y_{j,i} = (x_{j,i} + m_{j,i})/2. \quad (5)$$

For Eq. (2), the constants a_j are specified as

$$\begin{aligned} &a_0 = 0\text{x}4\text{D}34\text{D}34\text{D}/2^{32},\ a_1 = 0\text{x}\text{D}34\text{D}34\text{D}3/2^{32}, \\ &a_2 = 0\text{x}34\text{D}34\text{D}34/2^{32},\ a_3 = 0\text{x}4\text{D}34\text{D}34\text{D}/2^{32}, \\ &a_4 = 0\text{x}\text{D}34\text{D}34\text{D}3/2^{32},\ a_5 = 0\text{x}34\text{D}34\text{D}34/2^{32}, \\ &a_6 = 0\text{x}4\text{D}34\text{D}34\text{D}/2^{32},\ a_7 = 0\text{x}\text{D}34\text{D}34\text{D}3/2^{32}. \end{aligned} \quad (6)$$

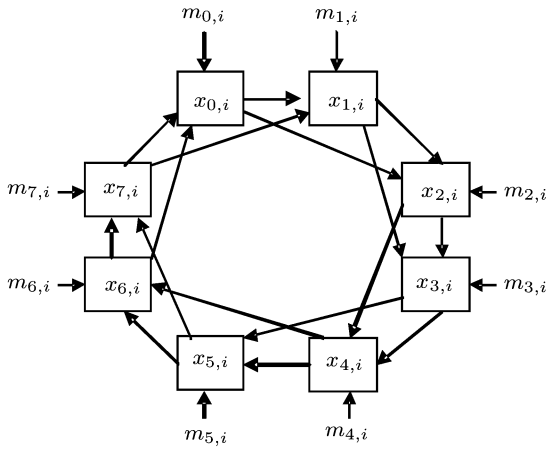


Fig. 1. Whole coupled structure of spatiotemporal chaos.

2.2. Initialization of the algorithm

Initial state $x_{j,0}$ and input $m_{j,0}$ in the spatiotemporal chaos system are computed by using a 128-bit key, $K^{[127\cdots0]}$. The key is divided into eight sub keys, namely, $k_0 = K^{[15\cdots0]}$, $k_1 = K^{[31\cdots16]}$, ..., $k_7 = K^{[127\cdots112]}$. The initial state and the input take the forms

$$x_{j,0} = \begin{cases} (k_{j+1\bmod8} \diamondsuit k_j)/2^{32}, & \text{for } j \text{ even}, \\ (k_{j+5\bmod8} \diamondsuit k_{j+4\bmod8})/2^{32}, & \text{for } j \text{ odd}, \end{cases} \quad (7)$$

$$m_{j,0} = \begin{cases} (k_{j+4\bmod8} \diamondsuit k_{j+5\bmod8})/2^{32}, & \text{for } j \text{ even}, \\ (k_j \diamondsuit k_{j+1\bmod8})/2^{32}, & \text{for } j \text{ odd}. \end{cases} \quad (8)$$

2.3. Details of the hash algorithm

The details of Ren's algorithm are described by the following steps.

Step 1 Initialize the system with a key, as described in Section 2.2, and iterate Eqs. (1) and (2) four times with the results of Eqs. (7) and (8).

Step 2 An original message is padded with zeros so that its length is a multiple of 32 bytes. The padded message is divided into blocks, e.g., $B_0, B_1, \ldots, B_n$, where $B_i = (M_i^0, M_i^1, \ldots, M_i^7)$ and M_i^j are 32-bit integers.

Step 3 For each block B_i, transform M_i^j into the input according to

$$m_{j,4(i+1)} = (M_i^j + 0.8)/2^{32}, \quad j = 0, 1, \ldots, 7, \quad i = 0, 1, \ldots, \quad (9)$$

and iterate Eqs. (1) and (2) four times with the result of Eq. (9).

Step 4 After all message blocks have been processed, the algorithm produces a hash value. Extract the 32 bits after the decimal point from the state value of each site, i.e., $X_j^{[31\cdots0]}$, and make the following transformations:

$$\begin{aligned} H^{[15\cdots0]} &= X_0^{[15\cdots0]} \oplus X_5^{[31\cdots16]}, \\ H^{[31\cdots16]} &= X_0^{[31\cdots16]} \oplus X_3^{[15\cdots0]}, \\ H^{[47\cdots32]} &= X_2^{[15\cdots0]} \oplus X_7^{[31\cdots16]}, \\ H^{[63\cdots48]} &= X_2^{[31\cdots16]} \oplus X_5^{[15\cdots0]}, \\ H^{[79\cdots64]} &= X_4^{[15\cdots0]} \oplus X_1^{[31\cdots16]}, \\ H^{[95\cdots80]} &= X_4^{[31\cdots16]} \oplus X_7^{[15\cdots0]}, \\ H^{[111\cdots96]} &= X_6^{[15\cdots0]} \oplus X_3^{[31\cdots16]}, \\ H^{[127\cdots112]} &= X_6^{[31\cdots16]} \oplus X_1^{[15\cdots0]}, \end{aligned} \quad (10)$$

where $H^{[127\cdots0]}$ is the 128-bit hash value of the input message.

3. Collision analysis and simulation results

If two distinct messages are hashed to the same hash value, then collisions exist in the hash function and an adversary could use the collisions to forge a fake message during the communication. A secure hash algorithm must resist collision attacks. In this section, we investigate the collision properties of the algorithm for three situations, a pair of different messages having identical keys, a pair of identical messages having different keys and a pair of different messages having different keys.

3.1. Collisions of a pair of different messages with identical keys

In this section, we aim to find a pair of different messages hashing to the same value with a known secret key. Without loss of generality, we only consider a one-block message, i.e., $B_0 = (M_0^0, M_0^1, \ldots, M_0^7)$. The inputs $m_{j,4}$ for Eqs. (1) and (2) are obtained from Eq. (9) as

$$m_{j,4} = (M_0^j + 0.8)/2^{32}, \quad j = 0, 1, \ldots, 7. \tag{11}$$

We try to find a pair of different inputs, i.e., $m_{j,4}$ and $m'_{j,4}$, $j = 0, 1, \ldots, 7$, having the same output in Eq. (1), i.e., $x'_{j,8} = x_{j,8}$, $j = 0, 1, \ldots, 7$.

Based on Eq. (5), we can write

$$y_{j,4} = (x_{j,4} + m_{j,4})/2, \quad y'_{j,4} = (x_{j,4} + m'_{j,4})/2. \tag{12}$$

To obtain $x'_{j,8} = x_{j,8}$ in Eqs. (1) and (2), we may simply require

$$f(y_{j,4}) = f(y'_{j,4}) \tag{13}$$

and

$$m_{j,5} = m'_{j,5}. \tag{14}$$

Combining Eq. (3) with Eq. (13) and assuming that $y_{j,4} \le b$ and $y'_{j,4} > b$, we have the following equation,

$$\begin{aligned} & y_{j,4}/b = (1 - y'_{j,4})/(1 - b), \\ \Rightarrow\ & y'_{j,4} = 1 - \frac{1-b}{b} y_{j,4}. \end{aligned} \tag{15}$$

Assuming that $m_{j,4} + a_j < 1$ and $m'_{j,4} + a_j \ge 1$, and considering Eqs. (2) and (14), we can give the relation between $m_{j,4}$ and $m'_{j,4}$ as

$$m'_{j,4} = m_{j,4} + \frac{255}{256}. \tag{16}$$

Next we specify the values of $m_{j,4}$ and $m'_{j,4}$. Considering Eqs. (12) and (16), we have

$$y'_{j,4} = y_{j,4} + \frac{255}{512}. \tag{17}$$

Combining Eq. (15) with Eq. (17), we have

$$y_{j,4} = \frac{257b}{512}, \quad y'_{j,4} = \frac{255 + 257b}{512}. \tag{18}$$

Substituting Eq. (18) into Eq. (12) leads to

$$m_{j,4} = \frac{257b}{256} - x_{j,4}, \quad m'_{j,4} = \frac{255 + 257b}{256} - x_{j,4}, \tag{19}$$

where $x_{j,4}$ is defined by the key of the algorithm and b is the parameter of the chaotic piecewise linear map, i.e., the parameter of Eq. (3). In general, $m'_{j,4} \ne m_{j,4}$ in Eq. (19), then we find a pair of different messages corresponding to the same $x_{j,5}$ and $m_{j,5}$ in Eqs. (1) and (2), respectively.

Now we give the simulation results. Without loss of generality, we let $x_{2,4} = 0.3$ and $b = 0.3$ in Eq. (19). We therefore have $m_{2,4} = 0.001171875$ and $m'_{2,4} = 0.997265625$, which correspond to $M_0^2 = 5033164$ and $M_0'^2 = 4283223244$, respectively. By using the above pair, M_0^2 and $M_0'^2$, and $x_{2,4} = 0.3$, $b = 0.3$, we calculate $y_{2,4}$ and $y'_{2,4}$ in Eq. (5), $f_{2,4}$ and $f'_{2,4}$ in Eq. (3), and $m_{2,5}$ and $m'_{2,5}$ in Eq. (2). We obtain the following results:

$$\begin{aligned} & m'_{2,4} \ne m_{2,4}, \quad y'_{2,4} \ne y_{2,4}, \\ & f'_{2,4} = f_{2,4}, \quad m'_{2,5} = m_{2,5}. \end{aligned} \tag{20}$$

As shown in Eq. (20), although we take a pair of different M_0^2 and $M_0'^2$, we still have the equalities $x'_{j,5} = x_{j,5}$, $m'_{j,5} = m_{j,5}$.

3.2. Collisions of a pair of identical messages with different keys

In this section, we shall consider the situation in which a message with different keys may hash to the same hash value. For two distinct keys, K and K', through the initialization of the algorithm (Eqs. (7) and (8)) and the four iterations with Eqs. (1) and (2), we finally have random and different $x_{j,4}$ and $x'_{j,4}$, $j = 0, 1, \ldots, 7$. For an input block $m_{j,4}$, $j = 0, 1, \ldots, 7$, we aim to keep $f'_{j,4} = f_{j,4}$ in Eq. (3), which will result in the identity $x'_{j,5} = x_{j,5}$. Based on Eq. (5), we define

$$\begin{aligned} & y_{j,4} = (x_{j,4} + m_{j,4})/2, \quad y'_{j,4} = (x'_{j,4} + m_{j,4})/2, \\ & j = 0, 1, \ldots, 7. \end{aligned} \tag{21}$$

Assuming $y_{j,4} \le b$, $y'_{j,4} > b$ and $f'_{j,4} = f_{j,4}$, we have Eq. (15). Substituting Eq. (21) into Eq. (15) leads to

$$m_{j,4} = 2b - (1 - b)x_{j,4} - bx'_{j,4}, \quad j = 0, 1, \ldots, 7. \tag{22}$$

For a pair of different K and K', i.e., $x_{j,4}$ and $x'_{j,4}$, if we find $m_{j,4}$ from Eq. (22) and $m_{j,4} \ge 0.8/2^{32}$, we can say that a block message can be hashed to the same hash value with different keys. Table 1 shows one of the numerical results with $k_j = 1$ and $k'_j = 53$, $j = 0, 1, \ldots, 7$, $b = 0.4$. From Table 1, we can observe that $x'_{j,4} \ne x_{j,4}$ for the given different keys after the initialization process. By processing the specified block message obtained from Eq. (22), i.e., $m_{j,4}$ in Table 1, we observe the following equalities: $x'_{j,5} = x_{j,5}$, $j = 0, 1, \ldots, 7$. The listed figures in Table 1 are accurate to the sixth decimal place.

Table 1. Simulation results with $k_j = 1$ and $k'_j = 53$, $j = 0, 1, \ldots, 7$.

j	$x_{j,4}$	$x'_{j,4}$	$m_{j,4}$	$x_{j,5}$	$x'_{j,5}$	$m_{j,5}$
0	0.694616	0.544619	0.195382	0.571967	0.571967	0.496969
1	0.615903	0.626089	0.180023	0.986223	0.986223	0.009326
2	0.533643	0.731357	0.187271	0.658279	0.658279	0.393620
3	0.464749	0.682382	0.248198	0.688348	0.688348	0.549786
4	0.640996	0.813873	0.089854	0.796852	0.796852	0.915250
5	0.583023	0.717762	0.163081	0.901412	0.901412	0.369430
6	0.574674	0.657302	0.192275	0.630710	0.630710	0.493862
7	0.502130	0.572995	0.269524	0.841552	0.841552	0.098827

3.3. Collisions of a pair of different messages with different keys

In this section, we discuss collisions of a pair of different messages with different keys. For two distinct messages with different keys, i.e., $x_{j,4}$ and $m_{j,4}$ (message 1), $x'_{j,4}$ and $m'_{j,4}$ (message 2), $j = 0, 1, \ldots, 7$, we have corresponding $y_{j,4}$ and $y'_{j,4}$. For two situations, $y_{j,4} = y'_{j,4}$ and $y_{j,4} \neq y'_{j,4}$, we consider the collision properties of the algorithm.

If $y_{j,4} = y'_{j,4}$, we obtain $f_{j,4} = f'_{j,4}$ in Eq. (3) and $x_{j,5} = x'_{j,5}$ in Eq. (1), and only require the following relations to be satisfied in Eq. (2),

$$m'_{j,5} = m_{j,5}, \quad j = 0, 1, \ldots, 7. \tag{23}$$

Assuming that $m_{j,4} + a_j < 1$ and $m'_{j,4} + a_j \geq 1$, referring to Eq. (23), and taking $y_{j,4} = y'_{j,4}$, we find messages 1 and 2 need to satisfy the following relations:

$$x'_{j,4} = x_{j,4} - \frac{255}{256}, \quad j = 0, 1, \ldots, 7, \tag{24}$$

$$m'_{j,4} = m_{j,4} + \frac{255}{256}, \quad j = 0, 1, \ldots, 7. \tag{25}$$

We estimate the possibility of finding a pair of messages satisfying Eqs. (24) and (25). We suppose that after the initialization of the algorithm, variables $x_{j,4}$ are randomly distributed in a range from 0 to 1, so the probability is $(\frac{1}{256})^8 = 2^{-64}$, which is the cost of a birthday attack against this algorithm.

If $y_{j,4} \neq y'_{j,4}$, according to Eqs. (1) and (2), we require the following relations to be satisfied,

$$f'_{j,4} = f_{j,4}, \quad m'_{j,5} = m_{j,5}, \quad j = 0, 1, \ldots, 7. \tag{26}$$

Referring to the analysis in Section 3.1, we solve Eqs. (15) and (16) by substituting $y_{j,4} = \frac{x_{j,4}+m_{j,4}}{2}$ and $y'_{j,4} = \frac{x'_{j,4}+m'_{j,4}}{2}$ to satisfy Eq. (26). Then the two messages with the fixed keys are given by

$$m_{j,4} = \frac{257b}{256} - (1-b)x_{j,4} - bx'_{j,4}, \ j = 0, 1, \ldots, 7,$$

$$m'_{j,4} = \frac{255}{256} + \frac{257b}{256} - (1-b)x_{j,4} - bx'_{j,4}. \tag{27}$$

Substituting the restrictions $m_{j,4} > 0$ and $m'_{j,4} \leq 1$ into Eqs. (27) leads to

$$\frac{257b-1}{256} < (1-b)x_{j,4} + bx'_{j,4} \leq \frac{257b}{256}, \quad j = 0, 1, \ldots, 7. \tag{28}$$

The probability of finding a pair of messages that satisfy Eq. (28) is $(\frac{1}{256})^8 = 2^{-64}$.

The above theoretical analysis of the two situations shows that we need to test 2^{64} possibilities to obtain a collision of a pair of different messages with different keys.

4. Statistical analysis

In Ref. [8], the authors had the following statistical test. A text file (120 kB) was randomly chosen and the corresponding changed message was generated by randomly selecting and toggling only one bit of the above message. Comparing the two hash values of the original and the changed messages, the authors concentrated on the number of the changed bits of the two hash values and showed that Ren's algorithm has a strong capability for diffusion and confusion (see Table 2 in Ref. [8]).

In this section, we will use a different statistical test. We concentrate on the statistical properties of every bit of the hash value. For two messages, M and M', and the corresponding hash values, $H^{127\cdots0}$ and $H'^{127\cdots0}$, the average change in the value of every bit of the two hash values is given by

$$\left\langle \Delta H^{[i]} \right\rangle = \frac{1}{T} \sum_{1}^{T} \Delta H^{[i]}, \ i = 0, 1, \ldots, 127, \tag{29}$$

where $\Delta H^{[i]} = H^{[i]} \oplus H'^{[i]}$ and T is the test number of the two messages.

Figure 2 shows the results of Eq. (29) for $T = 10^6$. Without loss of generality, we only consider messages of one block, i.e., M_0^i and $M_0'^i$, $i = 0, 1, \ldots, 7$. In Fig. 2(a), we randomly choose M_0^i, $i = 0, 1, \ldots, 7$ and $M_0'^1$, and let $M_0'^i = M_0^i$, $i = 0, 2, 3, \ldots, 7$. Figure 2(a) shows satisfactory average values for the hash value of 128 bits, namely close to 0.5, and the fluctuation around 0.5 is inversely proportional to $\sqrt{T}$. In Fig. 2(b), we also randomly choose M_0^i, $i = 0, 1, \ldots, 7$ and let $M_0'^i = M_0^i$, $i = 0, 2, 3, \ldots, 7$, while the less significant bit of M_0^1 is chosen and toggled, $M_0'^1 = M_0^1 \oplus 1$. In Fig. 2(b), we can see that the average values of some bit sites deviate from 0.5, e.g., the 47^{th}, 126^{th} and 127^{th} bit site, and the magnitudes of these deviations are much larger than that of normal fluctuation. The deviations indicate that the distribution of the hash value is not random. In Figs. 2(c) and 2(d), we also randomly choose M_0^i and let $M_0'^i = M_0^i$ except that $M_0'^2 = M_0^2 \oplus 1$ and $M_0'^3 = M_0^3 \oplus 1$. As a result, we also observe an unsatisfactory distribution of the average values of Eq. (29).

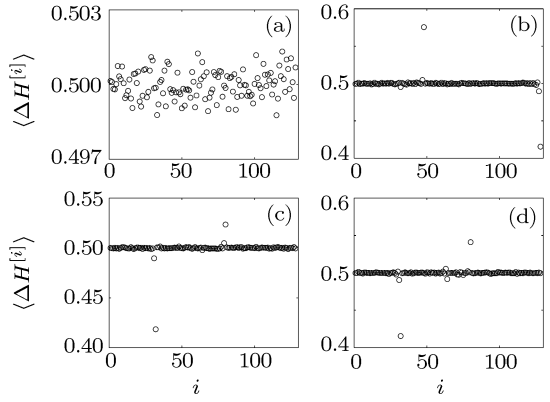


Fig. 2. Average values of Eq. (29) for $T = 10^6$. The $M_0^i, i = 0, 1, \ldots, 7$, are randomly chosen. (a) $M_0'^i = M_0^i$, $i = 0, 2, 3, \ldots, 7$, and $M_0'^1$ is randomly chosen with $M_0'^1 \neq M_0^1$. (b) $M_0'^1 = M_0^1 \oplus 1$ and $M_0'^i = M_0^i$, $i = 0, 2, 3, \ldots, 7$. (c) $M_0'^2 = M_0^2 \oplus 1$ and $M_0'^i = M_0^i$, $i = 0, 1, 3, \ldots, 7$. (d) $M_0'^3 = M_0^3 \oplus 1$ and $M_0'^i = M_0^i$, $i = 0, 1, 2, 4, 5, 6, 7$.

5. An improved algorithm and collision analysis

In Section 3, we analyzed three types of collisions under different conditions. In our analysis, above, we concentrate on the specific dynamics, namely, Eqs. (2) and (3). This kind of piecewise map may result in local collisions. For example, if $0 < y \leq b$, we can find another y', $y' > b$, that satisfies $f(y') = f(y)$ in Eq. (3). Although the system of spatiotemporal chaos has complex couplings (three nodes are coupled) and strong nonlinear operations (the function $\text{Trim}(x, y)$), the local collisions may also lead to the same output of the system. Therefore, both the whole structure and the local dynamics are important for designing a secure hash function. To avoid the collision problems and to improve the statistical properties of Ren's algorithm, we suggest an improved algorithm below. In this section, we will introduce and analyze the algorithm briefly.

5.1. The improved algorithm

To avoid collision problems and statistical weakness, we suggest the following modifications to Ren's algorithm:

(i) Modifying the second term on the right-hand side of Eq. (1). We obtain

$$\begin{aligned}
x_{0,i+1} &= (f_{0,i} + \text{Trim}(y_{7,i}, 16) + \text{Trim}(f_{6,i}, 16))/3,\\
x_{1,i+1} &= (f_{1,i} + \text{Trim}(y_{0,i}, 8) + f_{7,i})/3,\\
x_{2,i+1} &= (f_{2,i} + \text{Trim}(y_{1,i}, 16) + \text{Trim}(f_{0,i}, 16))/3,\\
x_{3,i+1} &= (f_{3,i} + \text{Trim}(y_{2,i}, 8) + f_{1,i})/3,\\
x_{4,i+1} &= (f_{4,i} + \text{Trim}(y_{3,i}, 16) + \text{Trim}(f_{2,i}, 16))/3,\\
x_{5,i+1} &= (f_{5,i} + \text{Trim}(y_{4,i}, 8) + f_{3,i})/3,\\
x_{6,i+1} &= (f_{6,i} + \text{Trim}(y_{5,i}, 16) + \text{Trim}(f_{4,i}, 16))/3,\\
x_{7,i+1} &= (f_{7,i} + \text{Trim}(y_{6,i}, 8) + f_{5,i})/3. \qquad (30)
\end{aligned}$$

(ii) Modifying Eq. (9) to

$$\begin{aligned}
m_{j,4} &= (M_0^j \oplus (M_0^j \gg 16) + 0.8)/2^{32},\\
&j = 0, 1, \ldots, 7, \qquad (31)
\end{aligned}$$

where $x \gg y$ stands for a rotation shift of y bits of x.

(iii) Modifying the iteration number of Eqs. (1) and (2). For each block message, we iterate five times with Eqs. (1) and (2).

(iv) Modifying the padding rule. The last padded 64 bits denote the length of the original message.

The above modifications achieve the following purposes: (i) is used to avoid the collision problems; both (ii) and (iii) are used to improve the statistical properties of the algorithm; (iv) is used to avoid the change of the message blocks.

5.2. Sensitivity of the hash value to changes of messages

To study the sensitivity of the hash value of the improved algorithm to changes of message bits, we compute Eq. (29) by using a pair of different messages

with the same key. We concentrate on the relation between the fluctuation of Eq. (29) and the test number T. We define

$$E^{[i]}(T) = |\left\langle \Delta H^{[i]} \right\rangle - 0.5|, \quad i = 0, 1, \ldots, 127. \quad (32)$$

The minimum and the maximum values of $E^{[i]}(T)$ are given by $E_{\min}(T)$ and $E_{\max}(T)$, respectively.

Figure 3 shows the test number dependence of $E_{\min}(T)$ and $E_{\max}(T)$, where the dots and the circles correspond to $E_{\max}(T)$ and $E_{\min}(T)$, respectively. In Fig. 3(a), we randomly choose M_0^i, $i = 0, 1, \ldots, 7$ and let $M_0'^0 = M_0^0 \oplus 1$, $M_0'^i = M_0^i$, $i = 1, 2, \ldots, 7$. We calculate Eq. (29) by using the improved algorithm and plot the curves of $E_{\max}(T)$ and $E_{\min}(T)$. From Fig. 3(a), we can see that the minimum and the maximum fluctuation is proportional to $1/\sqrt{T}$, which implies good randomness. In Figs. 3(b)–3(d), we also randomly choose M_0^i, $i = 0, 1, \ldots, 7$, and $M_0'^i = M_0^i$ except that $M_0'^1 = M_0^1 \oplus 1$, $M_0'^2 = M_0^2 \oplus 1$ and $M_0'^3 = M_0^3 \oplus 1$. The curves in Figs. 3(b)–3(d) also show good randomness. Figure 3 shows that compared with the original algorithm, the improved algorithm enhances the sensitivity of the hash value to changes of message bits.

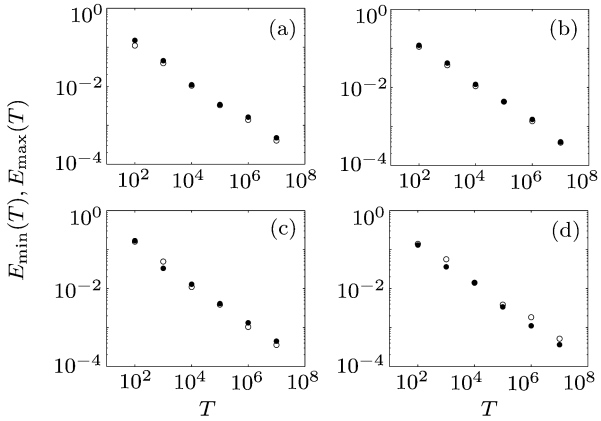


Fig. 3. Test number dependence of $E_{\min}(T)$ and $E_{\max}(T)$. The dots and the circles correspond to $E_{\max}(T)$ and $E_{\min}(T)$, respectively. We randomly choose M_0^i, $i = 0, 1, \ldots, 7$. The $M_0'^i = M_0^i$ except that $M_0'^0 = M_0^0 \oplus 1$, $M_0'^1 = M_0^1 \oplus 1$, $M_0'^2 = M_0^2 \oplus 1$ and $M_0'^3 = M_0^3 \oplus 1$ in (a) to (d), respectively.

5.3. Sensitivity of hash value to changes of secret keys

To study the sensitivity of the hash value of the improved algorithm to changes of secret keys, we compute Eqs. (29) and (32) by using a pair of identical messages with different keys. Figure 4 shows the test number dependence of $E_{\min}(T)$ and $E_{\max}(T)$. Without loss of generality, we have $M_0^i = 0$, $i = 0, 1, \ldots, 7$. In Fig. 4, we randomly choose k_i, $i = 0, 1, \ldots, 7$, and let $k_i' = k_i$ except that $k_0' = k_0 \oplus 1$, $k_1' = k_1 \oplus 1$, $k_2' = k_2 \oplus 1$ and $k_3' = k_4 \oplus 1$ in (a) to (d), respectively. Figure 4 shows that the outputs of Eq. (32) for different keys are clearly random, therefore the hash value of the improved algorithm is sensitive to the secret key. Our simulations show that due to the initialization process in step 1, both the original algorithm and its improved version show a good sensitivity of the hash value to changes of the keys.

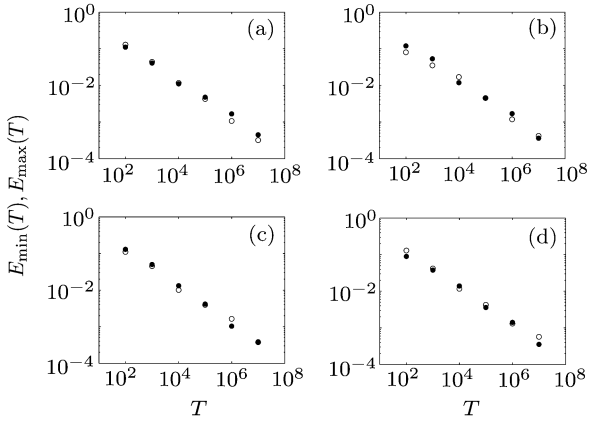


Fig. 4. Test number dependence of $E_{\min}(T)$ and $E_{\max}(T)$. The dots and the circles correspond to $E_{\max}(T)$ and $E_{\min}(T)$, respectively. We randomly choose k_i, $i = 0, 1, \ldots, 7$. The $k_i' = k_i$ except that $k_0' = k_0 \oplus 1$, $k_1' = k_1 \oplus 1$, $k_2' = k_2 \oplus 1$ and $k_3' = k_3 \oplus 1$ in (a) to (d), respectively.

5.4. Resistance against collision attacks

Compared with the original algorithm, the improved algorithm not only has the characteristics of the original algorithm with respect to the collision attacks investigated in Ref. [8], but also can resist the collision attacks described in Section 3. In Eq. (30), we use variables of both $f_{j,i}$ and $y_{j,i}$, $j = 0, 1, \ldots, 7$. If $m'_{j,4} \neq m_{j,4}$ and $x'_{j,4} = x_{j,4}$, which correspond to the collision of a pair of different messages with identical keys, we have $y'_{j,4} \neq y_{j,4}$. Even if $f'_{j,4} = f_{j,4}$, we still have $x'_{j,5} \neq x_{j,5}$ in Eq. (30). For $m'_{j,4} = m_{j,4}$ and $x'_{j,4} \neq x_{j,4}$, which correspond to the collision of a pair of identical messages with different keys, we also have the same results as that of a pair of different messages with identical keys. Thus the improved algorithm resists the collisions described in Subsections 3.1 and 3.2.

6. Conclusion

In this paper, the collision and statistical properties of a keyed hash function based on spatiotemporal chaos are analyzed. We propose an improved algorithm to overcome the flaws of the original algorithm. Although the security of the improved algorithm is enhanced, its performance slows down slightly. Therefore, for chaotic hash functions, balancing the security and the efficiency remains a challenge.

References

[1] Cuomo L M and Oppenheim A V 1993 *Phys. Rev. Lett.* **71** 65

[2] Lu H P, Wang S H, Li X W, Tang G N, Kuang J Y, Ye W P and Hu G 2004 *Chaos* **14** 617

[3] Chen G, Mao Y and Chui C K A 2003 *Chaos Soliton. Fract.* **21** 749

[4] Xiao D, Liao X and Deng S 2005 *Chaos Soliton. Fract.* **24** 65

[5] Wang S and Hu G 2007 *Chaos* **17** 023119

[6] Wang X, Yin Y and Yu H *Finding Collisions in the Full SHA-1*, https://www.infosec.sdu.edu.cn/paper

[7] Wang Y, Liao X F, Xiao D and Wong K W 2008 *Information Sciences* **178** 1391

[8] Ren H J, Wang Y, Xie Q and Yang H Q 2009 *Chaos Soliton. Fract.* **42** 2014

[9] Xiao D, Liao X F and Deng S J 2008 *Phys. Lett.* A **372** 4682

[10] Amin M, Faragallah O S and El-Latif A A 2009 *Chaos Soliton. Fract.* **42** 767

[11] Xiao D, Liao X F and Wang Y 2009 *Neurocomputing* **72** 2288

[12] Wang J Z, Wang Y L and Wang M Q 2006 *Acta Phys. Sin.* **55** 5048 (in Chinese)

[13] Wang J Z, Wang M Q and Wang Y L 2008 *Acta Phys. Sin.* **57** 2737 (in Chinese)

[14] Guo W, Wang X M, He D K and Cao Y 2009 *Phys. Lett.* A **373** 3201