Received February 26, 2021, accepted March 21, 2021, date of publication April 7, 2021, date of current version April 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3071501

Construction and Analysis of SHA-256 Compression Function Based on Chaos S-Box

JUAN WANG[1,2,3], GE LIU[2], YONGQI CHEN[2], AND SHU WANG[4]
[1]College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China
[2]College of Electronic and Information Engineering, Heilongjiang University of Science and Technology, Harbin 150027, China
[3]National Science Park of Harbin Engineering University, Harbin 150001, China
[4]State Grid Anshan Electric Power Supply Company, Anshan 114000, China

Corresponding author: Juan Wang (1131814@s.hlju.edu.cn)

This work was supported in part by the Heilongjiang Fundamental Research Foundation for the Local Universities, in part by the Science and Technology Innovation Foundation of Harbin under Grant 2017RAQXJ031, and in part by the Heilongjiang University of Science and Technology Graduate Innovation Research Foundation under Grant YJSCX2020-223HKD.

ABSTRACT To further improve the security of SHA-256, a compression function construction scheme based on chaotic S-box is proposed. Through the reasonable design of the iteration mode and using the nonlinearity, confusion, and anti-difference of chaotic S-box, the relationship between the plaintext and the hash value becomes more complicated and unpredictable, by the diffusion of the linear transformation P-box, the dependence between the plaintext and the hash value is further improved. The compression function is applied to the SHA-256 Hash Algorithm to test and analyze the distribution, sensitivity, confusion, diffusion, and collision resistance, and compared with the original SHA-256 and the mainstream Hash Algorithms. The results show that the proposed scheme can effectively improve the collision resistance of the SHA-256 Hash Algorithm and enhance the stability of confusion and diffusion with fewer cryptographic components and lower operational consumption to provide more reliable security guarantee for its application.

INDEX TERMS Hash function, compression function, chaotic S-box, P-box.

I. INTRODUCTION

Secure Hash Algorithm 2 (SHA-2) is a Hash Algorithm designed by the National Security Agency (NSA) and published by the National Institute of Standards and Technology (NIST), including a total of six standard algorithms: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 [1]. Generally speaking, the longer the message digest, the more difficult it is to collide, but the slower it will run. Considering the security, efficiency, implementation, and other factors, the SHA-256 algorithm has gradually become an authoritative algorithm in SHA-2, which is widely used in many fields such as information encryption, digital signature, message authentication, and blockchain [2]–[4].

Traditional Hash Algorithms such as SHA-256 generally use relatively simple nonlinear logic operations. With the deepening research on differential attacks, it is easy to calculate the optimal linear approximation function of the original function by linear approximation of nonlinear operations, thus increasing the probability of collision. Handschuh *et al.* found a partial collision through the Chabaud-Joux attack [5]. Flofian Mende *et al.* have given an 18-steps differential attack route through XOR difference [6]. Gilbert has found 9-steps local collision, and the computational complexity is 2^{66}. Hawkes has optimized the attack complexity to 2^{39} on this basis [7]. Krystian *et al.* have used linear function instead of a nonlinear function and identity transformation instead of permutation to find the collision with the complexity of 2^{64} [8]. The above research shows that the SHA-256 Hash Algorithm still has some security risks. The improved algorithms of SHA-3 [9] and SHA-256 are successively proposed, which are considered by NIST as an effective and safe supplement to the Hash Algorithm.

Hash Algorithm is the process of mapping finite length message into fixed-length hash value by compression function. The improvement and analysis of compression function have been the research focus to improve the security performance of SHA-256. Literature [10] used the carry adder

The associate editor coordinating the review of this manuscript and approving it for publication was Di He.

to improve the compression function efficiency of the Hash Algorithm, but the improvement of its collision resistance was not obvious. Literature [11] improved the operation relationship between compression functions, which improved the operation efficiency and collision resistance performance, but its ability to resist birthday attacks did not improve. Literature [12] proposed to improve the compression function and iteration method, which is not ideal for the improvement of safety performance, but leads to the decrease of operation efficiency.

As a nonlinear cryptographic component, S-box will produce corresponding random changes for any change in input [13]. With its powerful confusion and scrambling functions, the S-box plays an irreplaceable role in resisting linear and differential analysis. S-box has not only become an extremely important component in block ciphers, but has also been introduced into the design of compression functions in Hash Algorithms [14], [15]. Literature [16] has used 16 S-boxes in the compression function of MD5 to realize the confusion of information. Literature [17] has introduced 2 kinds of S-boxes in the compression function of SBH to carry out 16 iteration operations. Although the existing scheme can improve the cryptographic performance of the hash function to a certain extent, it will also lead to larger memory and operation consumption.

With the deepening of S-box research and the development of chaos theory, the S-box with excellent performance based on nonlinearity, randomness, initial value sensitivity, and unpredictability of chaos [18], which provides a new idea for constructing the compression function of SHA-256 Hash Algorithm. Based on the above analysis, this paper proposes to introduce nonlinear chaotic S-box and linear transformation P-box into the compression function of SHA-256 to realize the confusion and diffusion of information respectively, and through the reasonable design of iteration mode, it can achieve effective enhancement of cryptographic performance with fewer cryptographic components and lower operational consumption. Theoretical analysis and experimental results show that the application of the compression function based on the chaotic S-box to the SHA-256 Hash Algorithm can effectively balance the security strength and operation efficiency.

II. PREPARATORY KNOWLEDGE

A. SHA-256 HASH ALGORITHM

SHA-256 Hash Algorithm is a compression operation for the message whose length is less than 2^{64} bits, and the length of output hash value is 256 bits. The specific calculation process is described as follows [19], [20]:

1) MESSAGE FILLING

Add a "1" and several "0" after the message. The number of "0" should be just enough to make the remainder result of the number of the entire message bits to 512 is 448, and the 64 bits original message is appended afterward so that after filling, to make the total length of the message block is $512 \times L$ bits.

2) VARIABLE DEFINITION

The initial link variable values, the intermediate link variable values, and the final hash values are all stored in eight registers A, B, C, D, E, F, G, H, where the values of initial link variables are $H_0 = 6a09e667$, $H_1 = bb67ae85$, $H_2 = 3c6ef372$, $H_3 = a54ff53a$, $H_4 = 510e527f$, $H_5 = 9b05688c$, $H_6 = 1f83d9ab$, $H_7 = 5be0cd19$.

3) COMPRESSION FUNCTION OPERATION

The compression function of SHA-256 is shown in Figure 1. 64 rounds of cyclic operations are performed on each message block. The values of registers A, B, C, D, E, F, G, H are used as the input of each round of calculation, and the current values of which come from the output link variable values of the previous round.

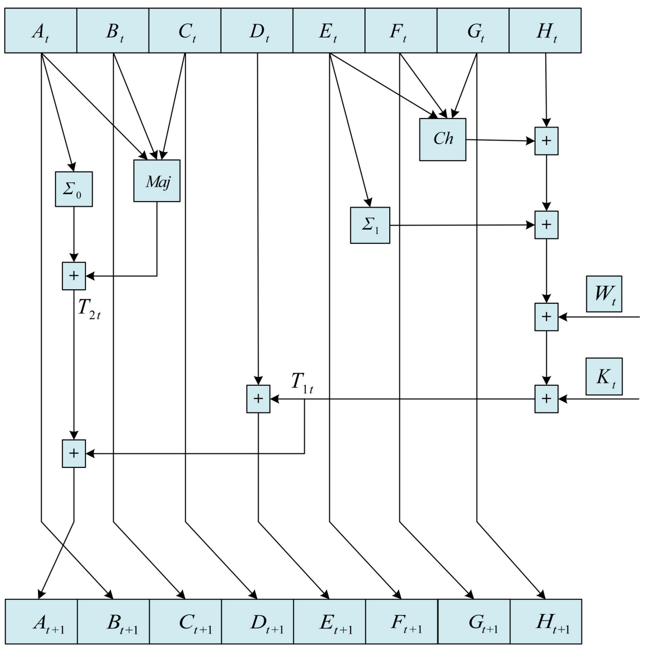


FIGURE 1. The compression function of SHA-256.

In each round of calculation, 32 bits message word W_t and addition constant K_t are needed. K_t takes the first 32 bits of the binary representation of the decimal part of the cube root of the first 64 primes, and W_t is represented as

$$W_t = \begin{cases} M_j[32 \cdot t + 31 : 32 \cdot t] & t < 16 \\ \sigma_0(W_{t-2}) + W_{t-7} & \\ +\sigma_1(W_{t-15}) + W_{t-16} & t \geq 16 \end{cases} \tag{1}$$

In formula (1), M_j represents the j-th message block, $+$ represents 2^{32} modulo addition operation, $\sigma_0(x)$ and $\sigma_1(x)$ are respectively represented as

$$\begin{cases} \sigma_0(x) = x \gg >_r 7 \oplus x \gg >_r 18 \oplus x \ggg 3 \\ \sigma_1(x) = x \gg >_r 17 \oplus x \gg >_r 19 \oplus x \ggg 10 \end{cases} \tag{2}$$

In formula (2), $\gg\!>_r n$ represents the cyclic right shift operation, $\gg\!> n$ represents the right shift operation, and $\oplus$ represents the XOR operation. The following iteration is performed per cycle

$$\begin{cases} A_{t+1} = T_{1t} + T_{2t} & B_{t+1} = A_t \\ C_{t+1} = B_t & D_{t+1} = C_t \\ E_{t+1} = D_t + T_{1t} & F_{t+1} = E_t \\ G_{t+1} = F_t & H_{t+1} = G_t \end{cases} \tag{3}$$

In formula (3), T_{1t} and T_{2t} are defined as

$$\begin{cases} T_{1t} = \Sigma_1(E_t) + Ch(E_t, F_t, G_t) + H_t + W_t + K_t \\ T_{2t} = \Sigma_0(A_t) + Maj(A_t, B_t, C_t) \end{cases} \tag{4}$$

In formula (4), the operational function is Ch, Maj, Σ_0, Σ_1 expressed as follows

$$\begin{cases} Ch(x, y, z) = (x \wedge z) \oplus (\neg x \wedge y) \\ Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \\ \Sigma_0(x) = (x \gg >_r 2) \oplus (x \gg >_r 13) \\ \qquad\qquad \oplus (x \gg >_r 22) \\ \Sigma_1(x) = (x \gg >_r 6) \oplus (x \gg >_r 11) \\ \qquad\qquad \oplus (x \gg >_r 25) \end{cases} \tag{5}$$

4) INTERMEDIATE HASH VALUES

After completing 64 rounds of the compression cycle, the intermediate hash values would be obtained by the modular addition operation of the values of registers A, B, C, D, E, F, G, H and the hash values of the $(j-1)$ block respectively.

$$\begin{cases} H_0^{(j)} = A + H_0^{(j-1)} & H_1^{(j)} = B + H_1^{(j-1)} \\ H_2^{(j)} = C + H_2^{(j-1)} & H_3^{(j)} = D + H_3^{(j-1)} \\ H_4^{(j)} = E + H_4^{(j-1)} & H_5^{(j)} = F + H_5^{(j-1)} \\ H_6^{(j)} = G + H_6^{(j-1)} & H_7^{(j)} = A + H_7^{(j-1)} \end{cases} \tag{6}$$

5) FINAL HASH VALUE

The hash value obtained by iterative operation of the last message block is used as the output result of SHA-256, that is

$$output = H_0 || H_1 || H_2 || H_3 || H_4 || H_5 || H_6 || H_7 \tag{7}$$

In formula (7), || represents connection computing.

B. CHAOTIC S-BOX

For any change in the input, the output of the ideal S-box will produce a corresponding random change, and it is almost impossible to approximate this change with a linear function. Based on the nonlinearity, randomness, ergodicity, initial sensitivity and unpredictability of chaos, constructing S-box by the chaotic system has become the mainstream of S-box generation mechanism [21]. Logistic, Tent and other low-dimensional chaos are widely used due to the simple mathematical model. However, due to the shortcomings of small full-mapping intervals and low complexity, the design of chaotic S-box is relatively single and the performance improvement is limited. To solve this problem, many robust 2D chaotic systems have been developed and these new chaotic systems can show good performance and high implementation efficiency [22],[23], cryptographers turn to study and apply high dimensional chaos to construct S-boxes.

TABLE 1. Chaotic S-box based on fireworks algorithm.

33	24	79	AD	8F	4B	6	5D	50	23	9D	93	66	3F	5B	1B
C0	F8	8C	C5	82	CB	A7	7C	C	54	73	34	35	F4	44	B1
1A	89	D0	1	18	87	1E	A6	80	E1	3D	91	69	68	12	6C
D9	11	DA	61	46	55	37	DD	1C	B4	C7	F1	70	17	77	A3
67	7E	F0	2F	E2	EA	B	96	39	6E	FF	4A	64	B0	F6	E8
D3	2D	EE	BC	58	B7	A4	7	2B	BF	C1	EB	27	CF	DE	31
AA	81	A8	C4	2	41	F2	28	AE	C6	A2	D1	2E	4D	A9	97
40	49	B2	65	45	FA	56	F9	F3	BE	AF	B5	E4	4C	FE	16
9B	2C	53	6B	30	74	C8	F5	75	0	52	85	FB	94	10	A0
DB	5A	38	E6	ED	21	43	14	5F	8E	1F	C9	5E	84	1D	83
BD	71	99	7A	E5	D6	63	C3	A	7F	9E	E7	5	92	48	4
9	BA	8D	7B	86	4F	32	4E	AC	3C	47	5C	26	CE	FD	B6
FC	EF	36	B8	CC	E3	D5	7D	3E	57	62	D4	6F	D7	3B	3A
F	3	D8	CD	76	60	B3	13	E0	59	C2	19	DC	29	AB	D2
CA	8B	9C	A1	42	90	95	2A	72	22	9F	D	E9	A5	78	20
B9	DF	BB	51	88	6A	F7	8A	15	98	9A	8	25	EC	6D	E

Lorenz is the most classical three-dimensional continuous chaotic system. The solution space composed of three initial values and three parameters is much higher than that of low-dimensional chaos. The higher dimension and complexity make it have more excellent dynamic characteristics. At the same time, the application of three chaos not only makes the design of the S-box more flexible, but also provides reliable security for the cryptography performance of the S-box. The nonlinear differential equation of the Lorenz chaotic system is

$$\begin{cases} \dfrac{dx}{dt} = a(y - x) \\ \dfrac{dy}{dt} = (bx - y - xz) \\ \dfrac{dz}{dt} = (xy - cz) \end{cases} \tag{8}$$

In formula (8), a, b and c are the control parameters of the Lorenz chaotic system, the typical values generally adopted are $a = 10$, $b = 28$ and $c = 8/3$, and the initial values will not affect the chaotic performance.

The firework algorithm has high optimization accuracy and convergence speed, and it is not easy to fall into the local optimum, especially for high-dimensional function optimization with better efficiency and stability. Therefore, this paper adopts the Lorenz chaotic S-box construction method based on the firework algorithm proposed in Literature [24]. The 8×8 S-box generated based on x-axis output of the Lorenz chaotic system is shown in Table 1.

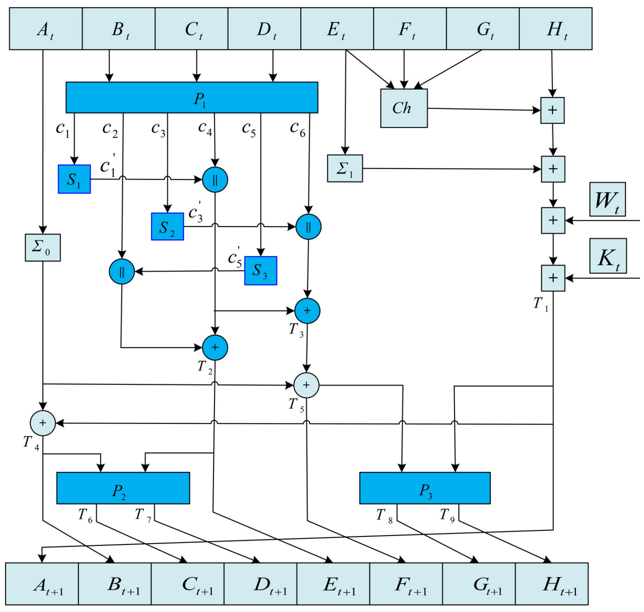


FIGURE 2. Compression function of SHA-256 based on chaotic S-box.

The firework algorithm is used to search for the optimal solution in the x, y, z solution space of the Lorenz chaotic system respectively, and then three different chaotic S-boxes are constructed. The combination of an efficient fireworks algorithm and a high-dimensional chaotic system can provide a reliable guarantee for the security performance and efficient generation of chaotic S-box. When 8 bits of data are input, the number composed of the first 4 bits is the row of the S-box, and the number composed of the last 4 bits is the column of the S-box, to realize the confusion and scrambling of the input information. At the same time, by changing the initial value slightly, the dynamic iteration of the Lorenz chaotic system can obtain completely different chaotic S-boxes.

III. COMPRESSION FUNCTION OF SHA-256 BASED ON CHAOTIC S-BOX

This paper proposes a compression function based on the chaotic box structure as shown in Figure 2. Compared with the original SHA-256 compression function, three secure and efficient chaotic S-boxes are used to replace the nonlinear function to confuse the outputs of three registers B, C, D, and through the reasonable design of the operation mode, three linear transformation P-boxes are used to diffuse the outputs of the eight registers, thus realizing the effective enhancement of cryptographic performance.

Taking the t round compression operation as an example, the 96 bits initial link variables output by registers B_t, C_t, D_t are sequentially input into the P_1-box. The information is diffused according to the following rules

$$P_1(l) = \begin{cases} l \cdot 24 mod 95 & 0 \le l \le 94 \\ 95 & i = 95 \end{cases} \tag{9}$$

In formula (9), l represents the bit position and mod represents the modulo operation.

The output of P_1-box is first divided into 12 groups equally, denoted as $m_1, \ldots, m_{12}$ respectively, and then divided into 6 groups according to the following rules, denoted as $c_1, \cdots, c_6$ respectively, namely

$$c_{2n+1} = m_{4n+1} \quad n = 0, 1, 2 \tag{10}$$

$$c_{2n} = m_{4n-2} \,||\, m_{4n-1} \,||\, m_{4n} \quad n = 1, 2, 3 \tag{11}$$

c_1, c_3, c_5 are input into three chaotic S-boxes constructed based on fireworks algorithm, the first four bits of each group mean the row of S-box, and the last four bits mean the column of S-box, then the confusion and replacement of information are realized by looking up the table, that is

$$S(c_{2n+1}) = c'_{2n+1} \tag{12}$$

Connect c'_1, c'_3, c'_5 with c_4, c_6, c_2 respectively, and then perform XOR operation to get

$$\begin{cases} T_2 = (c'_1||c_4) \oplus (c'_5||c_2) \\ T_3 = (c'_3||c_6) \oplus (c'_1||c_4) \end{cases} \tag{13}$$

The initial link variables of registers A_t, E_t, F_t, G_t, H_t perform the following operations with message word W_t and constant K_t

$$\begin{cases} T_1 = \Sigma_1(E_t) + Ch(E_t, F_t, G_t) \\ \quad +H_t + W_t + K_\mathrm{t} \\ T_4 = T_1 \oplus \Sigma_0(A_t) \\ T_5 = T_3 \oplus \Sigma_0(A_t) \end{cases} \tag{14}$$

T_2, T_4 and T_1, T_5 are input into P_2-box and P_3-box respectively, and the outputs of which are divided into four groups, denoted as $T_6, \ldots, T_9$. The diffusion rules are

$$P_2(l) = P(C_l \ggg 10) \tag{15}$$

$$P_3(l) = \begin{cases} l \cdot 16 \bmod 63 & 0 \le l \le 62 \\ 63 & l = 63 \end{cases} \tag{16}$$

The values of this round operation are assigned to 8 registers for the next round of compression operation, the specific corresponding relationship is

$$\begin{array}{llll} T_1 \to A_{t+1} & T_4 \to B_{t+1} & T_6 \to C_{t+1} & T_7 \to D_{t+1} \\ T_2 \to E_{t+1} & T_5 \to F_{t+1} & T_8 \to G_{t+1} & T_9 \to H_{t+1} \end{array} \tag{17}$$

The above operations are carried out for 64 rounds, the compression processing of a message block can be completed.

IV. EXPERIMENTAL ANALYSIS

As the core component to achieve compression mapping, the compression function will directly determine the performance of the Hash Algorithm. The compression function based on chaotic S-box is applied to the SHA-256 Hash Algorithm, and its security performance is evaluated by testing and analyzing the hash value distribution, sensitivity, diffusion, confusion, and collision resistance.

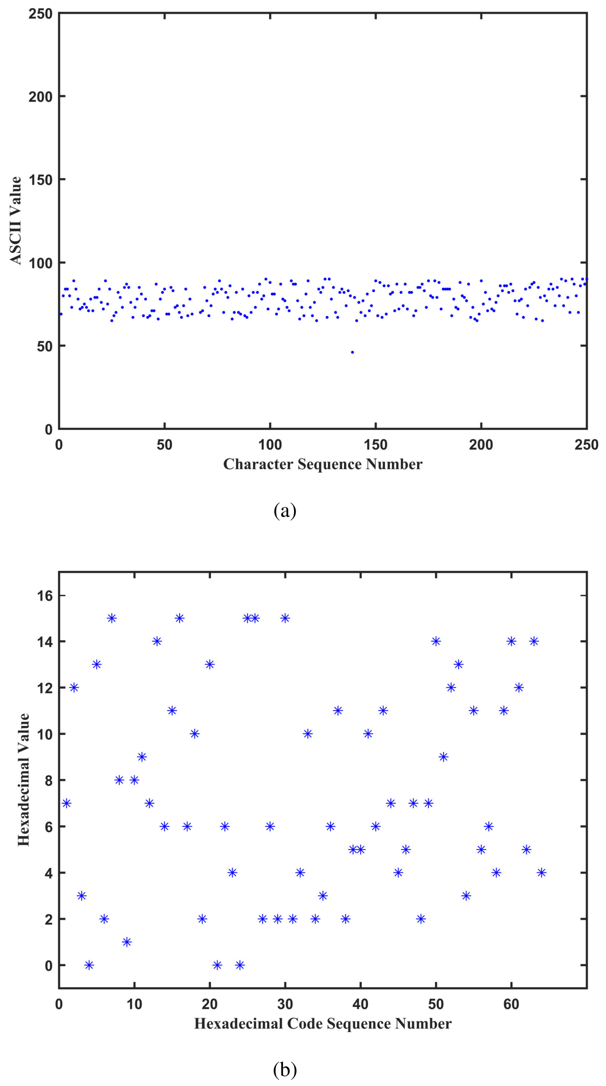


FIGURE 3. **Random characters and their hash value distribution.**

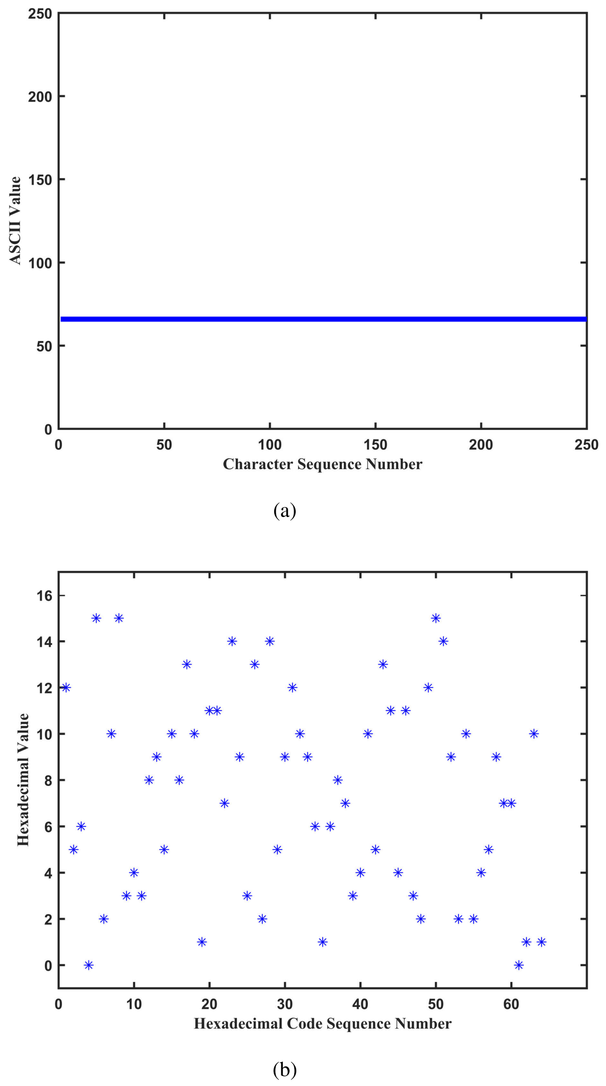


FIGURE 4. **Extreme characters and their hash value distribution.**

A. ANALYSIS OF HASH VALUE DISTRIBUTION

The Hash Algorithm with excellent performance can make full use of the plaintext space to make the hash value randomly and evenly distributed in the ciphertext space, thus hiding the relationship between the plaintext message and the hash value. To analyze the security of the Hash Algorithm in this paper, the random and extreme plaintext messages are selected to test the hash value distribution characteristics. As shown in Figure 3 (a), select a group of random characters as input, whose ASCII code values are mainly distributed in the range of [60, 95]. As shown in Figure 4 (a), select a group of the same characters as input, whose ASCII code values remain unchanged. As shown in Figure 3 (b) and Figure 4 (b), the two groups of plaintext messages are basically distributed in the finite value region, and the hash values output by the two groups of messages are randomly and uniformly distributed. Experimental results show that applying the compression function based on chaotic S-box to the SHA-256 Hash Algorithm can fully hide the statistical information of the plaintext message, and the plaintext message cannot be speculated by the hash value.

B. ANALYSIS OF SENSITIVITY

The unidirectionality of the Hash Algorithm means that it is computationally infeasible to find the plaintext message for a given hash value, which requires that the Hash Algorithm should have high message sensitivity. In this paper, we use the method proposed in the literature [25] to select a piece of plaintext message " University of Science and Technology is located in the beautiful ice city of Harbin, a university characterized by mining engineering.", and then performs 9 different tests to analyze the sensitivity of the Hash Algorithm, respectively

C1: The original plaintext message;

C2: Change the letter "a" in the word "Heilongjiang" to"A";

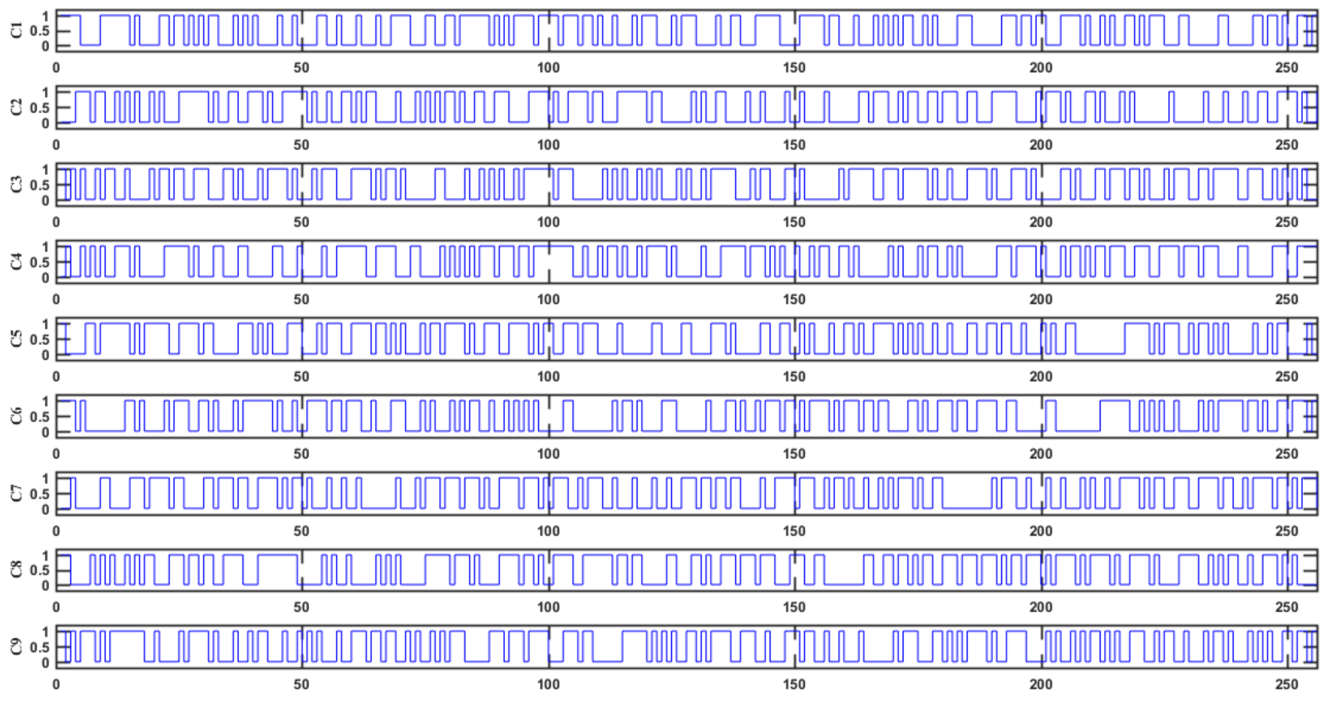


FIGURE 5. **Hash values under different conditions.**

C3: Change the "." at the end of the sentence to ",";
C4: Change the word "Heilongjiang" to "Heilong-jiang";
C5: Add "3" before the word "city";
C6: Add a space at the end of the sentence;
C7: Change the letter "f" in the word "beautiful" to "h";
C8: Add "3" before the word "Science";
C9: Delete the letter "i" in the word "Harbin".

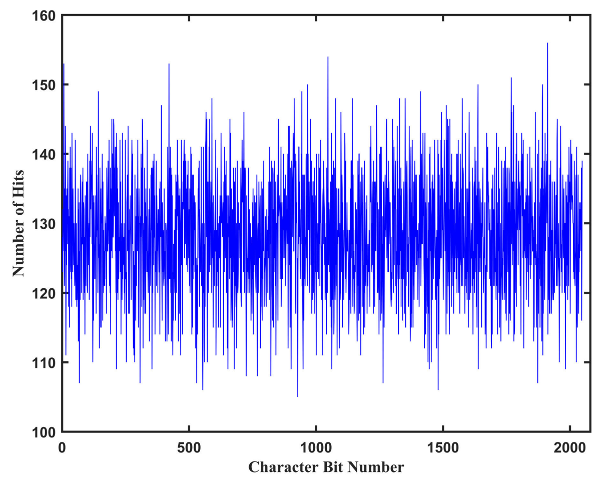
C1 represents the original plaintext message, C2-C9 represent the minimum change of the plaintext message under different conditions, the corresponding 256-bit hexadecimal hash values of C1-C9 are obtained by simulation experiments as follows. It can be seen that a small change in any position of the plaintext message will cause a great change in the output hash value.

C1:f0fd0dab15090c889e337eaf7bb48b8b84c7c3ec4559 4381db1e94d71818369b
C2:1d9528fd33a7d26b08ab4ce1e9ec7ec0a6121210372 cdcc7c2e8d140408899e9
C3:a9642d9c68bd171eb40c254bf702a5452f8b1a02f9d f48f4741a3768b9be72a8
C4:4a9d07d18c1884fcf1855776ff25a5e823eea4b340a2 ca03bd16b739eee1839f
C5:86fd7cf60ea38b9ed45bd39eb3b840c30e833289a7a a48cd914c00fae5a82584
C6:1d00d0b9a2fd27b7238a2b25506016870234bb5ebb 61a767981803f2a42bed375
C7:20c3bd86ecf5a24808bbba7bae5c8893b6c27b9692b a6005c89149eb39ecbdae
C8:c2a563b678ff0520a83ee87b4f3f6c18d3bb8e6019b5 9ddd6ebd7a7a1e564ea0
C9:aebf90bd12c4a89db295c1dde3b03ea9af391d92205c 6af4f0a56a72c8b454db

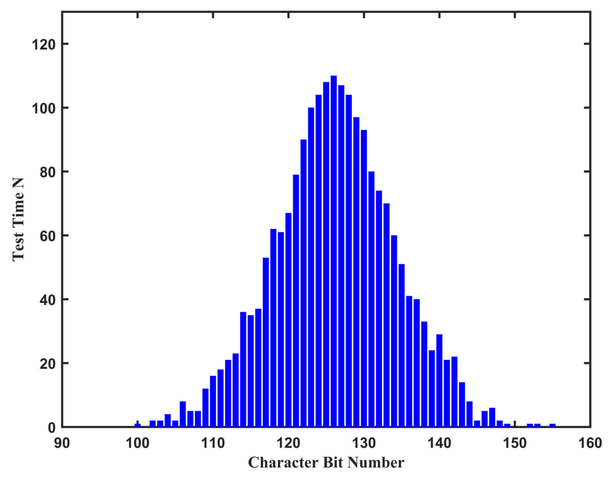
It is the most ideal that the new hash value generated by randomly changing the 1 bit value of the plaintext message has a 50% probability change in the corresponding position compared with the original hash value. The binary representations of hash values of C1-C9 under different conditions are shown in Figure 5. It can be intuitively seen that approximately half of the values change at the corresponding positions of the hash values, which indicate that the compression function constructed based on chaotic S-box will make the SHA-256 Hash Algorithm has strong message sensitivity, to better meet the performance requirement of unidirectional.

C. STATISTICAL ANALYSIS OF CONFUSION AND DIFFUSION

Confusion and diffusion are two basic principles for designing a Hash Algorithm. Confusion makes the relationship between plaintext messages and hash values complex and unpredictable, and diffusion strengthens the dependence between plaintext messages and hash values. To ensure security, the hash value generated by the designed algorithm should have unified distribution and good statistical characteristics, otherwise, the attacker will easily find the collision when the computational complexity is less than the exhaustive attack. For the hash value represented by binary, a slight change in the plaintext message causes the hash value to change with a probability of 50% per bit and the number of



(a)



(b)

FIGURE 6. Statistical results and histograms of the average number of changed bits.

bits of the final hash value change should be $1/2$ of the length of the original algorithm, which indicates that the distribution effect of the algorithm is the most ideal.

The purpose of the confusion and diffusion test is to statistics the change of Hash value when the plaintext value changes slightly through a large number of experiments. Randomly select a group of plaintext messages and calculate the corresponding hash value. Arbitrarily choose a bit in the plaintext message to change and then generate a new hash value. Comparing the results of two hash values to get variable bits B_i. The number of tests $N = 2048$ times and the statistical results and histograms of the average change bits B_i are shown in Figure 6. The change bits are mainly concentrated in about 128 bits, and their values vary between 100 and 158 bits. The average change value is 128.12, which is very close to the ideal value of 128 bits. The experimental results show that the compression function based on chaotic S-box will make the confusion and diffusion of the SHA-256 Hash

TABLE 2. Statistics of confusion and diffusion test results.

	N=256	N=512	N=1024	N=2048
B_{min}	107	107	105	104
B_{max}	153	153	153	156
$\bar{B}$	128.28	128.19	128.02	127.99
$P\%$	50.10	50.07	50.06	40.98
ΔB	7.89	7.345	7.19	7.26
$\Delta P\%$	3.09	3.13	3.10	3.05

Algorithm closer to the ideal statistical value, indicating that the algorithm has strong confusion and diffusion.

To test the stability of confusion and diffusion, the tests are repeated 256, 512, 1024, 2048 times, and the following formulas are used for statistics.

minimum change bit number

$$B_{\min} = \min(B_i)_1^N \tag{18}$$

maximum change bit number

$$B_{\max} = \max(B_i)_1^N \tag{19}$$

average change bit number

$$\bar{B} = \frac{1}{N}\sum_{i=1}^{N} B_i \tag{20}$$

average change probability

$$P = (\bar{B}/n) \times 100\% \tag{21}$$

mean square deviation of B

$$\Delta B = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i - \bar{B})^2} \tag{22}$$

mean square deviation of P

$$\Delta P = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\frac{B_i}{n} - P)^2} \times 100\% \tag{23}$$

In formula (23), n is the length of the hash value, N is the number of tests, and B_i is the number of bits that the hash value changes in the i test. The statistical results of the Hash Algorithm in this paper are shown in Table 2. The average change bit number $\bar{B}$ and the average change probability $P\%$ are very close to the ideal situation of 128 and 50 %, indicating that the ciphertext space has been fully used so that the attacker cannot use some known pairs of plaintext ciphertext to statistics the plaintext information. As the number of experiments increased, the test results did not change significantly, indicating that the Hash Algorithm has strong stability. When $N = 2048$, the comparison with other literatures is shown in Table 3. The average change bit number $\bar{B}$ is the closest to the ideal value of 128, and the average change probability $P\%$ is also very close to the ideal value of 50.

TABLE 3. Comparison with other algorithms when N = 2048.

Algorithm	B_{min}	B_{max}	$\bar{B}$	$P\%$	ΔB	$\Delta P\%$
Ref.[26]	104	153	129.0	50.38	8.08	3.15
Ref.[27]	100	155	128.1	50.05	7.94	3.10
Ref.[28]	100	157	127.9	49.95	7.94	3.10
Ref.[29]	101	168	128.1	50.03	8.12	3.21
Ref.[30]	N/A	N/A	128.1	50.07	10.16	4.00
SHA-256	104	154	128.01	50.00	7.94	3.10
This paper	1024	156	127.99	49.98	7.26	3.05

ΔB and $\Delta P\%$ mark the degree of stability of confusion and diffusion, the closer to zero, the more stable. In this paper, the Hash Algorithm ΔB and $\Delta P\%$ have the smallest values, indicating that their confusion and diffusion capabilities are more stable. The above results show that the test results of the proposed scheme have better statistical characteristics as a whole, which shows that the compression function based on the chaotic S-box construction will make the confusion and diffusion performance of the SHA-256 Hash Algorithm more stable.

D. ANALYSIS OF COLLISION RESISTANCE

Collision means that different plaintexts produce the same hash value. From a statistical point of view, plaintext space and hash value space are multiple-to-one mappings, and there is a possibility that different plaintexts share the same hash value. Collision resistance is a very important indicator to measure the security of the Hash Algorithm. The better the collision resistance, the higher the security.

In this paper, the method tested in the literature [31] is used for the collision resistance analysis. A piece of plaintext message is randomly selected to generate hash values and stored in ASCII format. Randomly changing a plaintext bit, generate new hash values, and store them in the same format. By comparing whether two ASCII characters in the same position between two hash values are equal, the smaller the maximum value of the same ASCII code, the lower the collision degree. If there is one is called one collision, the number of statistics is the number of collisions. Literature [31] also gives the number of hits in the Nth independent test, theoretically how many experiments should be hit, the calculation formula is as follows

$$\begin{cases} W_N(\omega) = N \times \dfrac{s!}{\omega!(s-\omega)!} \times (\dfrac{1}{2^8})^{\omega} \\ \qquad \times (1-\dfrac{1}{2^8})^{s-\omega} \\ \sum_{\omega=0}^{s} W_N(\omega) = W_N(0) + W_N(1) \\ \qquad + \cdots + W_N(\omega) = N \end{cases} \tag{24}$$

In formula (24), s = 256/8=32, 256 is the length of the hash value and 8 ASCII codes are required; w is the number of characters hit, and N is the number of tests.

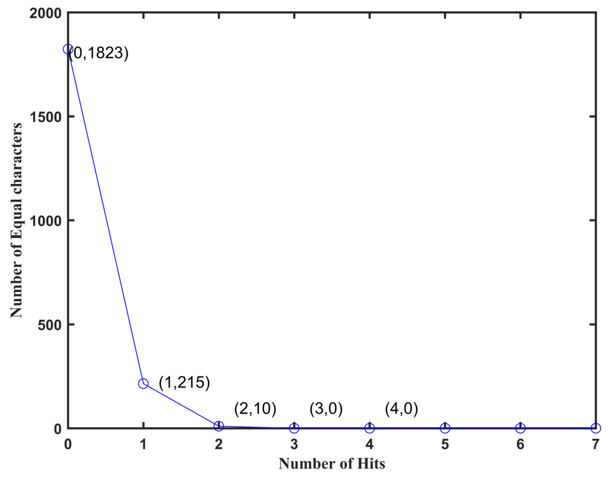


FIGURE 7. Distribution of collision.

TABLE 4. Comparison of absolute difference d between two hash values.

Algorithm	Hits	Maximum	Minimum	Mean	Mean/Char
Ref.[32]	2	2139	645	1388.6	86.79
Ref.[33]	2	3812	1693	2726.8	85.21
Ref.[34]	2	2017	695	1359.0	84.94
Ref.[35]	2	2096	646	1425.0	89.07
Ref.[36]	2	2230	731	1368.0	85.50
Ref.[37]	2	2399	537	1364.7	85.29
SHA-256	2	3819	1789	2707.1	84.59
This paper	2	4004	1590	2732.0	85.32

From formula (24), the theoretical value of the number of hits can be obtained: $W_N(0) = 1806.91$, $W_N(1) = 226.75$, $W_N(2) = 13.78$, $W_N(3) = 0.54, \ldots, W_N(32) = 1.75 \times 10^{-74}$.

After 2048 tests were performed, the distribution of the number of hash values of the same character is shown in Figure 7. There were 2 collisions in 10 tests, 1 collision in 215 tests, and no collision in 1823 tests. When the plaintext changes by only one bit, the maximum number of equal characters for hash values in the same position is only two. Compared with the theoretical value, the results show that the proposed scheme will make the SHA-256 Hash Algorithm have a low collision degree and excellent collision resistance performance.

To quantitative statistics the relationship between the two hash values, the absolute difference between the two hash values is defined as

$$d_{hash} = \sum_{i=1}^{N} (|t(a_i) - t(b_i)|) \tag{25}$$

Formula (25), d_{hash} represents the absolute difference between the two hash values, a_i and b_i represent the i-th ASCII characters of the original and new hash values, respectively. The function $t(x)$ is to convert the ASCII characters into the corresponding decimal number, and N is the number of ASCII characters corresponding to the hash value.

The comparison of the absolute difference is shown in Table 4. In 2048 tests, the maximum value of the absolute difference between the two hash values of the proposed algorithm is 4004, the minimum value is 1590, the average value is 2732.0, and the average character difference is 85.32, which is very close to the theoretical value of 85.33. Compared with hash values in other literatures, the compression function based on chaotic S-box will make the SHA-256 Hash Algorithm have an extremely high ability of collision resistance.

V. CONCLUSION

In this paper, nonlinear chaotic S-box and linear transformation P-box are introduced into the compression function of SHA-256 to realize information confusion and diffusion, and the iterative structure and operation mode are designed reasonably. The scheme of this paper is applied to the SHA-256 to test the performance of the distribution, sensitivity, confusion, diffusion and collision resistance, and compared with the original SHA-256 and the mainstream Hash Algorithms. The results show that compared with the original SHA-256 algorithm and other chaotic Hash Algorithms, the compression function constructed based on chaotic S-box only uses fewer cryptographic components and lower computational consumption, so that the Hash Algorithm has good distribution and sensitivity, stronger collision resistance attack ability, and more stable confusion and diffusion, thus achieving an effective balance between security intensity and computational efficiency. The research and experiment of this paper not only provide a reliable guarantee for the secure application of the SHA-256. It also provides a new idea for the improvement of the Hash Algorithm, especially the construction of compression function.

REFERENCES

[1] A. F. S. Ibrahim, "New secure solutions for privacy and access control in health information exchange," Ph.D. dissertation, Dept. Comput. Sci., Univ. Kentucky, Lexington, KY, USA, 2016.

[2] D. Ravilla and C. S. R. Putta, "Implementation of HMAC-SHA256 algorithm for hybrid routing protocols in MANETs," in *Proc. Int. Conf. Electron. Design, Comput. Netw. Automat. Verification (EDCAV)*, Jan. 2015, pp. 154–159.

[3] K. D. B. Utama, Q. M. R. Al-Ghazali, L. I. B. Mahendra, and G. F. Shidik, "Digital signature using MAC address based AES-128 and SHA-2 256-bit," in *Proc. Int. Seminar Appl. Technol. Inf. Commun. (iSemantic)*, Oct. 2017, pp. 72–78.

[4] R. Martino and A. Cilardo, "Designing a SHA-256 processor for blockchain-based IoT applications," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100254.

[5] Y. Yang, F. Chen, J. Chen, Y. Zhang, and K. L. Yung, "A secure hash function based on feedback iterative structure," *Enterprise Inf. Syst.*, vol. 13, pp. 281–302, Mar. 2019.

[6] F. Mendel, T. Nad, and M. Schläffer, "Improving local collisions: New attacks on reduced SHA-256," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2013, pp. 262–278.

[7] K. S. Yap, G. M. Wolrich, J. D. Guilford, V. Gopal, E. Ozturk, S. M. Gulley, W. K. Feghali, and M. G. Dixon, "Method and apparatus to process SHA-2 secure hashing algorithm," U.S. Patent 10 152 326, Dec. 11, 2018.

[8] Journal Applied Quantitative Methods, "SHA family functions," *IEEE Trans. Ind. Electron.*, vol. 10, no. 3, p. 48, Sep. 2015.

[9] G. Ye, K. Jiao, C. Pan, and X. Huang, "An effective framework for chaotic image encryption based on 3D logistic map," *Secur. Commun. Netw.*, vol. 2018, pp. 1–11, Oct. 2018.

[10] A. Gowthaman and M. Sumathi, "Performance study of enhanced SHA-256 algorithm," *Int. J. Appl. Eng. Res.*, vol. 10, no. 4, pp. 10921–10932, 2015.

[11] P. Zhang, X. Zhang, and J. Yu, "A parallel hash function with variable initial values," *Wireless Pers. Commun.*, vol. 96, no. 2, pp. 2289–2303, Sep. 2017.

[12] Y. Yang, J. Yu, Q. Zhang, and F. Meng, "Improved hash functions for cancelable fingerprint encryption schemes," *Wireless Pers. Commun.*, vol. 84, no. 1, pp. 643–669, 2015.

[13] Z. Lin, C. Sun, and Y. Li, "Design and implementation of the single-block hash FunctionBased on S-box," in *Proc. Int. Conf. Artif. Intell. Comput. Sci.*, Jul. 2019, pp. 62–69.

[14] Lu, Zhu, and Wang, "A novel S-Box design algorithm based on a new compound chaotic system," *Entropy*, vol. 21, no. 10, p. 1004, Oct. 2019.

[15] L. Yi, X. Tong, Z. Wang, M. Zhang, H. Zhu, and J. Liu, "A novel block encryption algorithm based on chaotic S-box for wireless sensor network," *IEEE Access*, vol. 7, pp. 53079–53090, 2019.

[16] F. Neugebauer, I. Polian, and J. P. Hayes, "S-box-based random number generation for stochastic computing," *Microprocessors Microsyst.*, vol. 61, pp. 316–326, Sep. 2018.

[17] Z.-Z. Wang and Y.-Z. Li, "Adaptive single-block hash function for short message," in *Proc. Int. Conf. Intell. Syst. Res. Mechatronics Eng.* Paris, France: Atlantis Press, 2015, pp. 1150–1155.

[18] H. Zhu, X. Tong, Z. Wang, and J. Ma, "A novel method of dynamic S-box design based on combined chaotic map and fitness function," *Multimedia Tools Appl.*, vol. 79, nos. 17–18, pp. 12329–12347, May 2020.

[19] R. Martino and A. Cilardo, "SHA-2 acceleration meeting the needs of emerging applications: A comparative survey," *IEEE Access*, vol. 8, p. 28415–28436, 2020.

[20] R. Martino and A. Cilardo, "A flexible framework for exploring, evaluating, and comparing SHA-2 designs," *IEEE Access*, vol. 7, pp. 72443–72456, 2019.

[21] Z. Hua, J. Li, Y. Chen, and S. Yi, "Design and application of an S-box using complete Latin square," *Nonlinear Dyn.*, vol. 12, pp. 1–19, Mar. 2021.

[22] Z. Hua, Y. Zhang, and Y. Zhou, "Two-dimensional modular chaotification system for improving chaos complexity," *IEEE Trans. Signal Process.*, vol. 68, pp. 1937–1949, 2020.

[23] Z. Hua, Z. Zhu, S. Yi, Z. Zhang, and H. Huang, "Cross-plane colour image encryption using a two-dimensional logistic tent modular map," *Inf. Sci.*, vol. 546, pp. 1063–1083, Feb. 2021.

[24] J. Wang, B. Pan, C. Tang, and Q. Ding, "Construction method and performance analysis of chaotic S-box based on fireworks algorithm," *Int. J. Bifurcation Chaos*, vol. 29, no. 12, Nov. 2019, Art. no. 1950158.

[25] H. Liu, A. Kadir, C. Ma, and C. Xu, "Constructing keyed hash algorithm using enhanced chaotic map with varying parameter," *Math. Problems Eng.*, vol. 2020, Jul. 2020, Art. no. 4071721.

[26] M. Todorova, B. Stoyanov, K. Szczypiorski, and K. Kordov, "SHAH: Hash function based on irregularly decimated chaotic map," 2018, *arXiv:1808.01956*. [Online]. Available: http://arxiv.org/abs/1808.01956

[27] H. Liu, A. Kadir, and J. Liu, "Keyed hash function using hyper chaotic system with time-varying parameters perturbation," *IEEE Access*, vol. 7, pp. 37211–37219, 2019.

[28] N. Abdoun, S. El Assad, O. Déforges, R. Assaf, and M. Khalil, "Design and security analysis of two robust keyed hash functions based on chaotic neural networks," *J. Ambient Intell. Hum. Comput.*, vol. 2, pp. 1–25, Feb. 2019.

[29] M. Asgari Chenaghlu, S. Jamali, and N. Nikzad Khasmakhi, "A novel keyed parallel hashing scheme based on a new chaotic system," *Chaos, Solitons Fractals*, vol. 87, pp. 216–225, Jun. 2016.

[30] A. Kumar, A. Fatima, and N. K. Nishchal, "An optical hash function construction based on equal modulus decomposition for authentication verification," *Opt. Commun.*, vol. 428, pp. 7–14, Dec. 2018.

[31] M. Alawida, J. S. Teh, D. P. Oyinloye, M. Ahmad, and R. S. Alkhawaldeh, "A new hash function based on chaotic maps and deterministic finite state automata," *IEEE Access*, vol. 8, pp. 113163–113174, 2020.

[32] Y. Li, "Collision analysis and improvement of a hash function based on chaotic tent map," *Optik*, vol. 127, no. 10, pp. 4484–4489, May 2016.

[33] A. Kanso and M. Ghebleh, "A structure-based chaotic hashing scheme," *Nonlinear Dyn.*, vol. 81, nos. 1–2, pp. 27–40, Jul. 2015.

[34] Y. Li, G. Ge, and D. Xia, "Chaotic hash function based on the dynamic S-box with variable parameters," *Nonlinear Dyn.*, vol. 84, no. 4, pp. 2387–2402, Jun. 2016.
[35] Y. Li, X. Li, and X. Liu, "A fast and efficient hash function based on generalized chaotic mapping with variable parameters," *Neural Comput. Appl.*, vol. 28, no. 6, pp. 1405–1415, Jun. 2017.
[36] A. Kanso and M. Ghebleh, "A fast and efficient chaos-based keyed hash function," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 1, pp. 109–123, Jan. 2013.
[37] M. Ahmad, S. Khurana, S. Singh, and H. D. AlSharari, "A simple secure hash function scheme using multiple chaotic maps," *3D Res.*, vol. 8, no. 2, p. 13, Jun. 2017.

JUAN WANG was born in 1981. She received the B.S. and M.S. degrees in electronic information engineering from the Heilongjiang University of Science and Technology, Harbin, China, in 2004 and 2010, respectively, and the Ph.D. degree in microelectronics and solid electronics from Heilongjiang University, in 2019. She is currently the Department Head, an Associate Professor, and a Master Tutor of electronic information engineering with the Heilongjiang University of Science and Technology. Her research interests include information security, and wireless and secure communication.

GE LIU received the Engineering degree from Taishan University, in 2019. She is currently pursuing the master's degree with the School of Electronics and Information Engineering, Heilongjiang University of Science and Technology, Harbin. Her research interests include hash algorithm and chaotic cryptography.

YONGQI CHEN received the B.Sc. degree from Xihua University, in 2019. He is currently pursuing the master's degree with the School of Electronics and Information Engineering, Heilongjiang University of Science and Technology, Harbin. His research interests include hash algorithm and blockchain.

SHU WANG received the master's degree in electrical and electrical appliances from the School of Electrical Engineering, Shenyang University of Technology, in 2012. She currently works with State Grid Anshan Electric Power Supply Company, mainly engaged in power system research and distribution network planning.