Hash Algorithm with Adaptive Hash Value Length Based on Weighted Probability Model

WANG Jielin [1,2], GAO Jinding [1,3]

1 School of Information, Mechanical and Electrical Engineering, Hunan International Economics University, Changsha, 410205, China

2 Hunan YESINE Communication Technology Co., Ltd. Changsha, 410600, China

3 School of physics and electronics, Central South University, Changsha, 410205, China

***Abstract*—In view of the deficiency that the hash length of standard hash algorithm cannot be adjusted adaptively according to the attack strength, this paper defines the weighted probability quality function and weighted distribution function using the interval coding method of distribution function, and proposes a hash algorithm based on weighted generality model. The theoretical limit of algorithm collision is analyzed. Combining segmentation, exclusive OR operation and nonlinear round function of weighting coefficients, the algorithm implementation steps are given and the experimental test is carried out. The results show that the hash length of the proposed hash algorithm can be customized and adjusted adaptively according to the attack strength, which can reach the theoretical limit of hash collision. The algorithm can be widely used in digital signature, file verification and data transmission verification.**

***Keywords*—One-way hash function; HASH functions; Weighted probability model; Adaptive Hash Value Length**

I: INTRODUCTION

Hash algorithm can compress messages of any length into fixed-length message digests, and has been widely used in the fields of digital signatures, file verification, and information encryption. The current standard Hash algorithms mainly include several series of Message Digest Algorithm (MD) [1] [2] [3], Secure Hash Algorithm (SHA) [4] [5], Lattice-based hashing algorithm, and Message Authentication Code (MAC) [6]. Among them, MD message digest algorithms mainly include MD2, MD4, and MD5 series. The mainstream SHA secure hash algorithms mainly include SHA-1 and SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512) series, and SHA-3 (KECCAK algorithm [9]). The MAC algorithm is mainly a hash function algorithm of HMAC containing a key. It is based on the original MD and SHA algorithms, and the key is added. There are mainly HmacMD series (HmacMD2, HmacMD4, HmacMD5) and HmacSHA series (HmacSHA1, HmacSHA224, HmacSHA256, HmacSHA38, HmacSHA512). The length of the generated message digest is consistent with the original MD series and SHA series. The digest length is different, and the internal calculation structure of the algorithm is different, so that the system needs to integrate a large number of hash algorithms with different digest lengths to meet the security requirements, resulting in a waste of system resources and costs. With the advent of quantum computing, computing performance has greatly improved, and security systems need to adopt a

hash algorithm with a longer message digest or a hash algorithm with a more reliable computing structure to ensure security. However, the longer message digest also brings a huge burden to the network transmission, check operation and storage, and the hash algorithm with a complex operation structure has a computational burden.

This paper defines the weighted probability distribution function and weighted probability model based on the probability distribution function [10], defines the length of the coding result of the Hash algorithm through the weighted probability distribution function, and gives a brand new Hash algorithm. The algorithm has the following characteristics:

1. Linearly calculate the probability interval corresponding to each symbol in the message, and construct a non-linear round function of the weighting coefficient, so that the length of the message, the probability of the symbol and the sequence of the symbols are different, then the encoding result will be different. The differences in the message will cause the "Avalanche Effect".
2. The length of the encoding result can be customized, and the security system does not need to integrate a large number of hash algorithms with different digest lengths, saving system resources and reducing costs.
3. The bit-based linear coding process is suitable for bit streams, and the coding operation can be terminated or started at any time.

Compared with the current mainstream MD, SHA, and MAC hash algorithm with fixed hash length, the length of the algorithm proposed in this article can be customized, and it can be adjusted adaptively according to the information security level, which can not only ensure information security, but also effectively reduce the pressure of message network transmission, verification calculation and storage.

II: HASH ALGORITHM BASED ON WEIGHTED PROBABILITY MODEL

Let source sequence $X = (X_1, X_2, \dots, X_i, \dots, X_n)$ be a discrete sequence $X_i \in A = \{0,1,2, \dots, k\}$ that takes on a finite or countable number of possible values. Then for all numerical values in A we have a probability space, that is

$$\begin{bmatrix} X_i \\ P \end{bmatrix} = \begin{bmatrix} 0 & 1 & \dots & k \\ p(0) & p(1) & \dots & p(k) \end{bmatrix}$$

Since the source sequence must be transferred into some symbol, at any time we have

$$\sum_{X_i=0}^{k} p(X_i) = 1, \qquad 0 \le p(X_i) \le 1$$

Therefore, the distribution function of any symbol X_i is

$$F(X_i) = \sum_{s \le X_i} p(s) \tag{2-1}$$

$p(0) \le F(x) \le 1,\ s \in A.$

Definition 2.1 We let discrete random variable be X, where $X \in A = \{0,1, \dots, k\}$. Furthermore, $\mathrm{P}\{X = a\} =$

$p(a)(a \in A)$, and the weighted probability mass function is $\varphi(a) = r\mathrm{P}\{X = a\} = rp(a)$. $p(a)$ is the probability mass function, $0 \le p(a) \le 1$, and r is the weight coefficient, and

$$F(a) = \sum_{i \le a} p(i) \tag{2-2}$$

If $F(a,r)$ satisfies $F(a,r) = rF(a)$, then $F(a,r)$ is called a weighted cumulative distribution function, or a weighted distribution function for short. Then the sum of the weighted probabilities of all the symbols is $\sum_{a=0}^{k} \varphi(a) = r$.

We let the discrete source sequence be $X = (X_1, X_2, \dots, X_n)$, where $X_i \in A$, and $F(X_i - 1) = F(X_i) - p(X_i)$. The weighted distribution function of sequence X is denoted as $F(X,r)$. When $n = 1$,

$$F(X,r) = rF(X_1 - 1) + rp(X_1)$$

When $n = 2$,

$$F(X,r) = rF(X_1 - 1) + r^2F(X_2 - 1)p(X_1) + r^2p(X_1)p(X_2)$$

When $n = 3$,

$$F(X,r) = rF(X_1 - 1) + r^2F(X_2 - 1)p(X_1) + r^3F(X_3 - 1)p(X_1)p(X_2) + r^3p(X_1)p(X_2)p(X_3)$$

Let $\prod_{j=1}^{0} p(X_j) = 1$, by analogy we can obtain

$$F(X,r) = \sum_{i=1}^{n} r^i F(X_i - 1) \prod_{j=1}^{i-1} p(X_j) + r^n \prod_{i=1}^{n} p(X_i) \tag{2-3}$$

The set of weighted distribution functions satisfying Equation (2-3) is defined as a weighted probability model, which is henceforth referred to as a weighted model, and is denoted as $\{F(X,r)\}$. If $X_i \in A = \{0,1\}$, then $\{F(X,r)\}$ is called a binary weighted model. We let

$$H_n = F(X,r) \tag{2-4}$$

$$R_n = rp(X_1)rp(X_2) \dots rp(X_n) = r^n \prod_{i=1}^{n} p(X_i) \tag{2-5}$$

$$L_n = H_n - R_n \tag{2-6}$$

where $X_i \in A, n = 1,2, \dots$. When $r = 1$,

$$F(X,1) = \sum_{i=1}^{n} F(X_i - 1) \prod_{j=1}^{i-1} p(X_j) + \prod_{i=1}^{n} p(X_i) \tag{2-7}$$

From Equation (2-4), (2-5) and (2-6), $H_n = F(X,1)$, that is, the range coding (arithmetic coding) [11][12] is a lossless coding method based on the weighted distribution function of $r = 1$.

Since X_i must take a value in A, $p(X_i) > 0$. Equations (2-4), (2-5) and (2-6) are interval columns, and $[L_i, H_i)$ is the corresponding interval superscript and subscript of variable X_i of source sequence X at time $i(i = 0,1,2, \dots, n)$. $R_i = H_i - L_i$ is the length of the interval. According to Equation (2-4), (2-5) and (2-6), let $R_0 = H_0 = 1$ and $L_0 = 0$ when $i = 1,2, \dots, n$, its expressions are:

$$\begin{aligned} R_i &= R_{i-1}\varphi(X_i) \\ L_i &= L_{i-1} + R_{i-1}F(X_i - 1, r) \\ H_i &= L_i + R_i \end{aligned} \tag{2-8}$$

Through (2-8), the weighted probability model coding operation is performed on the source sequence X. L_n, a real number, is the coding result of the weighted probability model. L_n obtains the binary sequence through the decimal-binary conversion.

III: INFORMATION ENTROPY OF WEIGHTED PROBABILITY MODEL AND COLLISION LIMIT ANALYSIS

3.1 Information entropy of weighted probability model

We let the discrete memoryless source sequence X be $X = (X_1, X_2, \dots, X_n)(X_i \in A, A = \{0,1,2, \dots, k\})$. When $r = 1$, $\varphi(X_i) = p(X_i)$. According to the definition of Shannon information entropy [13] [14] [15], the entropy of X is

$$H(X) = -\sum_{X_i=0}^{k} p(X_i)\log_{k+1} p(X_i) \tag{3-1}$$

When $r \neq 1$, we define the self-information of the random variable X_i with probability $\varphi(X_i)$ as

$$I(X_i) = -\log_{k+1} p(X_i) \tag{3-2}$$

We let the number of a in set $\{X_i = a_j\}(j = 0,1, \dots, k; i = 1,2, \dots, n)$ be c_a. When the value of r is determined, the total amount of information of source sequence X is

$$-\sum_{a=0}^{k} c_a \log_{k+1} p(a)$$

Thus, the average amount of information per symbol is

$$-\frac{1}{n}\sum_{a=0}^{k} c_a \log_{k+1} p(a) = -\sum_{a=0}^{k} p(a)\log_{k+1} p(a)$$

Definition 3.1 Let $H(X,r)$ be

$$\begin{aligned} H(X,r) &= -\sum_{a=0}^{k} p(a)\log_{k+1}\varphi(a) \\ &= -\log r - \sum_{a=0}^{k} p(a)\log_{k+1} p(a) \\ &= -\log r + H(X) \end{aligned} \tag{3-3}$$

According to definition 3.1, when the value of r is determined, the binary length encoded by the weighted probability model is $nH(X,r)(bit)$. The simplest source sequence is a binary sequence. Let the bit length of the binary source sequence X be n, and the symbols 0 and 1 in X have the probabilities $p(0)$ and $p(1)$, and the source sequence X encoded by a weighted probability model obtains a sequence of length $L(bit)$. When $k = 1$, from Equation (3-3) we obtain

$$-n\log_2 r + nH(X) = L \tag{3-4}$$

where $H(X)$ is the information entropy of sequence X, that is, $H(X) = -p(0)\log_2 p(0) - p(1)\log_2 p(1)$, so simplify (3-4) and obtain

$$\begin{aligned} r &= 2^{\left(H(X)-\frac{L}{n}\right)} \\ &= 2^{\left(-p(0)\log_2 p(0)-p(1)\log_2 p(1)-\frac{L}{n}\right)} \end{aligned} \tag{3-5}$$

According to the lossless coding theorem, $H(X)$ is the lossless coding limit of the discrete memoryless source sequence X, so when $H(X,r) \geq H(X)$, the weighted model function $F(X,r)$ can restore source X without loss. When $H(X,r) < H(X)$, the weighted model function $F(X,r)$ cannot restore the source X, that is, when $L < nH(X)$, the encoding result L_n cannot restore the source X.

From (3-4) and (3-5), when $H(X) > L/n$, $r > 1$ and then $H(X,r) < H(X)$, so the weighted model functions $F(X,r)$ that satisfy Equation (3-5) and $r > 1$ are all one-way hash function (Hash function).

3.2 Collision limit analysis

Theorem 3.2 The probabilities of symbol "0" and symbol "1" in the hash value obtained by the weighted probability model hash algorithm for any binary sequence are equal.

Proof Let the bit length of the hash value obtained by the weighted probability model hash algorithm of the binary sequence be L. The binary sequence of the hash value is recorded as Y. Its information entropy is $H(Y) = -p(0)\log_2 p(0) - p(1)\log_2 p(1)$. According to Definition 3.1, $nH(X,r) = -n\log_2 r + nH(X)$ (n is the bit length of the binary sequence X), so $LH(Y) = -n\log_2 r + nH(X)$. If and only if $H(Y) = 1$, Equation (3-4) is tenable, that is, r satisfies Equation (3-5). Otherwise, r does not satisfy Equation (3-5). And if and only if $p(0) = p(1) = 0.5$, $H(Y) = 1$, so the probabilities of symbol "0" and symbol "1" in sequence Y are equal.

According to Theorem 3.2, the probabilities of symbols in the hash value obtained by the hash algorithm in this paper are equal. Let the bit length of the hash value be L, then the value range of the value space is $\{0,1,\ldots,2^L-1\}$. Let $d = 2^L$, according to the probability of hash collision (or "birthday attacj") [2][4][16][17], the hash collision probability of this algorithm can be obtained through N tests:

$$p(N,d) \approx 1 - e^{\frac{-N(N-1)}{2d}} \tag{3-6}$$

3.3 Non-linear piecewise iterative operations and round functions of weighted coefficients

Common Hash algorithm, AES [18][19] and DES [20] symmetric encryption algorithms all adopt a round function of nonlinear byte replacement to eliminate linear correlation, usually called S box. Based on the idea of round function, this section adopts section iterative and exclusive-or operation to eliminate linear correlation, and constructs a nonlinear round function with weighted coefficients based on Equation (3-5).

(1) Piecewise iteration and exclusive OR operation of Sequence X

The bit length of the sequence X is n, and the bit length of each segment is m^2, so the sequence X is linearly divided into $\lceil n/m^2 \rceil$ segments. Let $m = 16$, $j = 1,2,\dots,\lceil n/m^2 \rceil$, and v is the number of bits corresponding to each segment. Obviously when $j = \lceil n/m^2 \rceil$, $v = n - (\lceil n/m^2 \rceil - 1) * m^2 - 1$; when $j < \lceil n/m^2 \rceil$, $v = m^2$. Randomly generate m^2 bits and store them in a $16 * 16$ two-dimensional table, as shown in Table 1, where x and y are row and column subscripts.

Table 1 Randomly generate a two-dimensional table of m^2 bits

x \ y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	1	0	1	1	0	1	0	0	0	0	1	0	0
1	1	1	0	0	0	1	0	1	0	0	1	1	1	1	1	1
2	0	1	1	1	0	1	1	0	1	0	0	1	0	1	0	0
3	1	0	0	1	0	0	1	0	1	1	1	0	1	1	0	1
4	1	1	0	1	1	1	1	0	0	0	1	1	0	1	0	1
5	0	0	0	1	0	1	0	0	1	1	0	0	0	0	1	0
6	1	0	1	1	1	1	0	1	0	0	0	1	1	0	0	1
7	1	1	0	0	0	0	1	0	0	0	0	0	1	1	1	1
8	0	0	0	0	1	1	1	1	1	0	1	0	1	1	1	0
9	0	1	1	1	1	1	0	0	1	0	1	0	1	1	0	0
10	0	0	0	0	1	1	0	1	0	0	1	0	0	0	1	1
11	0	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0
12	0	1	1	0	0	0	1	1	0	0	1	1	0	0	0	0
13	1	1	1	0	0	0	0	0	1	0	1	1	0	0	0	1
14	0	0	1	1	1	0	1	1	0	1	1	1	0	1	0	1
15	0	1	0	1	0	1	1	0	1	0	0	1	0	1	0	1

Since the position of the symbol X_i in the j-th paragraph is i $(i = 1,2,\dots,m^2)$, the table look-up expression in Table 1 is:

$$y = x = (i + X_i) \bmod m, \quad y = \lfloor (i + X_i) / m \rfloor \tag{3-7}$$

The position (x, y) is random due to X_i, so the bit value $f(x, y)$ is unknown. Xor X_i and $f(x, y)$, and then the binary sequence after the XOR operation is denoted as Y:

$$Y_i = X_i \oplus f(x, y) \tag{3-8}$$

After the XOR operation, calculate the probability of symbol 0 and symbol 1 in sequence Y, and substitute it into Equation (3-5) to calculate the value r of the current segment: $f(x, y)$.

(2) Non-linear round function of weighting coefficient

After setting L, when each binary sequence is coded with a weighted probability model, if r does not change with i（$i = 1,2,\dots,m^2$）, then r is called a static weighting coefficient; if r changes with i, then we call r the dynamic weighting coefficient, denoted as $r(i)$. Randomly generate m^2 integers from 0 to 255 and store them in a $16 * 16$ two-dimensional table, as shown in Table 2, where x and y are row and column subscripts, and the calculation formulas for x and y are the same as Equation (3-7). After looking up Table 2 to get the value g(x,y), we can obtain that the nonlinear round function of $r(i)$ is:

$$r(i) = 2^{(H(X)-L/v)} - \frac{g(x,y)}{10^s} \tag{3-9}$$

In Equation (3-9), s can be an integer greater than 3. The actual value depends on the calculation accuracy of the computer. In this experiment, $s = 4$. When $r < 2^{(H(X)-L/n)}$, the encoding result will exceed L bits. Assuming that the encoding result is $L_z = L + t$ bits, there are t more bits, and the binary sequence of the hash value of the j-th segment is recorded as Z, and Z is the binary sequence converted from L_n through the decimal-binary conversion. Since $\frac{g(x,y)}{10^s}$ tends to 0, there is no situation where $t > L$. Let $l = 1,2,\dots,t$, then xor the last t bits and the first L bits:

$$Z_{l-1} = Z_{l-1} \oplus Z_{L+l-1} \tag{3-10}$$

In Equation (3-10), Z_{l-1} and Z_{L+l-1} are the $(l-1)$-th and $(L+l-1)$-th binary symbols of the sequence Z.

Table 2 Randomly generate a two-dimensional table of m^2 integers from 0 to 255

x \ y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	143	254	212	99	131	164	146	48	66	137	58	219	171	251	193	50
1	94	43	86	32	68	48	217	36	242	56	183	150	73	48	82	213
2	143	12	156	62	127	28	182	213	23	235	49	190	164	108	121	92
3	178	59	73	121	131	85	96	153	126	112	204	252	183	237	213	249
4	220	75	88	129	35	170	182	14	162	35	179	67	94	67	132	38
5	122	96	58	232	187	17	3	218	35	105	75	82	137	114	160	2
6	140	157	84	106	186	209	79	221	213	41	37	228	47	74	141	115
7	189	203	124	9	33	28	206	90	7	206	30	230	136	89	149	128
8	221	216	214	146	133	135	208	157	79	85	45	84	1	23	95	97
9	62	148	184	231	37	249	204	152	146	227	97	35	113	170	167	197
10	107	63	233	178	99	118	249	249	155	99	152	204	23	223	214	222
11	199	86	226	230	84	90	249	64	214	173	94	242	209	202	229	147
12	146	55	115	158	202	125	19	58	180	187	127	151	24	135	11	125
13	186	2	102	215	97	187	126	146	39	91	208	115	54	174	197	18
14	41	69	133	125	6	111	70	193	192	249	161	224	78	238	178	138
15	218	250	158	88	134	71	46	8	248	233	103	224	141	74	174	5

Based on Equation (3-9), the real number L_v corresponding to the j-th segment is obtained through the weighted probability model encoding, and then the hash value Z of the current segment can be obtained through Equation (3-10).

Ⅳ: IMPLEMENTATION OF HASH ALGORITHM IN BINARY WEIGHTED PROBABILITY MODEL

4.1 Algorithm flow of the customized hash value length

L is the bit length of the customized hash value, and the encoding steps of the binary source sequence X using the weighted probability model are as follows.

(1) Initialize parameters: $p = c = L_0 = 0$, $H_0 = R_0 = 1$, $i = t = L_z = 0$, $m = 16$, $j = l = 1$, $s = 4$, $v = m^2 - 1$, $T = x = y = \varphi(0) = \varphi(1) = 0$;

(2) Divide the sequence X into $\lceil n/m^2 \rceil$ segments linearly;

(3) When $j = \lceil n/m^2 \rceil$, $v = n - (\lceil n/m^2 \rceil - 1) * m^2 - 1$;

(4) Obtain the j-th binary sequence of sequence X, a total of v bits;

(5) Obtain the i-th symbol X_i of the j-th segment;

(6) Calculate x and y: $x = (i + X_i) \bmod m$ and $y = (i + X_i) / m$;

(7) Check table 1 to obtain $f(x,y)$;

(8) $Y_i = X_i \oplus f(x,y)$;

(9) $i = i + 1$. If $i \le v$, repeat (5) to (9);

(10) $i = 0$. Count the number c of symbols 0 in the sequence Y to obtain $p = \frac{c}{v}$ and $H(Y) = -p\log_2 p - (1-p)\log_2(1-p)$;

(11) Calculate x and y: $x = (i + X_i) \bmod m$ and $y = (i + X_i) / m$;

(12) Check table 2 to obtain $g(x,y)$;

(13) Calculate $r(i)$, $\varphi(0)$ and $\varphi(1)$: $r(i) = 2^{(H(Y)-L/v)} - \frac{g(x,y)}{10^s}$, $\varphi(0) = r(i)p$ and $\varphi(1) = r(i)(1-p)$;

(14) Weighted model coding operation. If $Y_i = 0$, $R_i = R_{i-1}\varphi(0)$, or $L_i = L_{i-1} + R_{i-1}\varphi(0)$ and $R_i = R_{i-1}\varphi(1)$;

(15) $i = i + 1$. If $i \le v$, repeat (11) to (15);

(16) $L_v = L_v + T$, $T = L_v$;

(17) L_v is converted into a binary sequence Z by decimal-binary conversion;

(18) Count the bit length L_z of Z and calculate t, $t = L_z - L$;

(19) When $l \le t$, $Z_{l-1} = Z_{l-1} \oplus Z_{L+l-1}$;

(20) $l = l + 1$. Repeat (19) to (20);

(21) $i = 0$, $j = j + 1$;

(22) If $j \le \lceil n/m \rceil$, Repeat (3) to (22);

(23) End encoding and output Z, where Z is the hash value.

4.2 System process of adaptive attack intensity

The algorithm for customizing the length of the hash value in Section 4.1 is recorded as WPMHA (Weighted Probability Model Hash Algorithm). The following takes information transmission verification as an example to give the system process of adaptive attack intensity based on WPMHA.

The system process of the transmitter is shown in Figure 1, and the steps are as follows:

（1）Initialize parameters: Define the increase d (d can be a random number within a certain range, such as a random number belonging to $1,2 \ldots, m\}$, or a customized integer greater than 0), the number of verifications $i = 3$, the value of L (for example, $L = 512$, that is, L is the bit length of the hash value in the WPMHA algorithm), and $sign = 0$;

（2）The transmitter makes use of WPMHA to obtain the hash value A of the information (the information is marked as Data), and sends the hash value A;

（3）The transmitter sends information Data and waits for the receiver to return $sign$;

（4）When overtime or $sign = 1$, $i = i + 1$, $L = L + d$;

（5）Repeat (2) to (5) when $i > 0$, or end.

The system process of the receiver is shown in Figure 2, and the steps are as follows:

（1）Initialize parameters, and check times $i = 3$ and $sign = 0$;

（2）Receive the hash value A;

（3）Receive information Data;

（4）According to the length of the hash value A, calculate the hash value B of Data based on WPMHA;

（5）When A is equal to B, reply success identity $sign = 0$ and end, or discard data, and $j = j - 1$;

（6）When $j > 0$, reply failure identity $sign = 1$, or end.

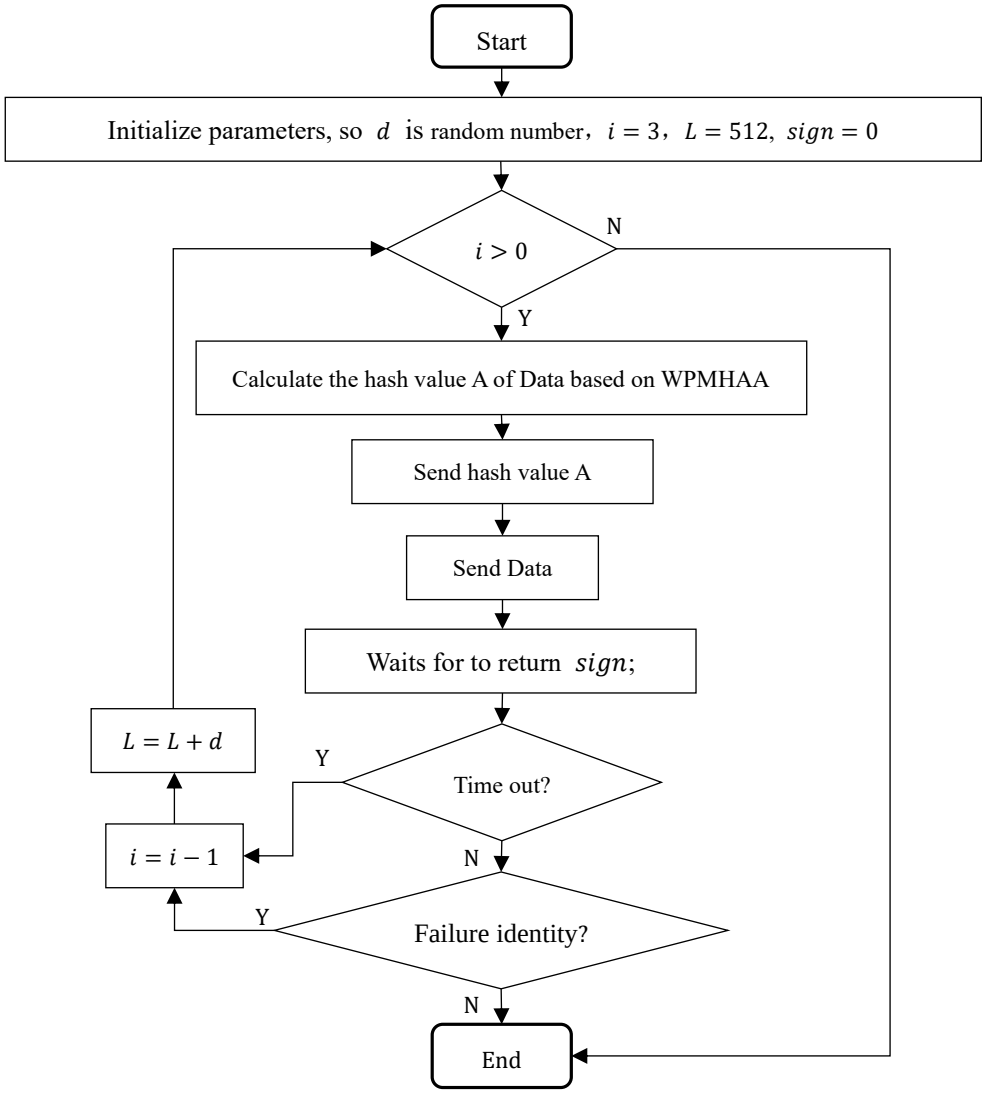

Figure 1 Schematic diagram of the transmitting end system

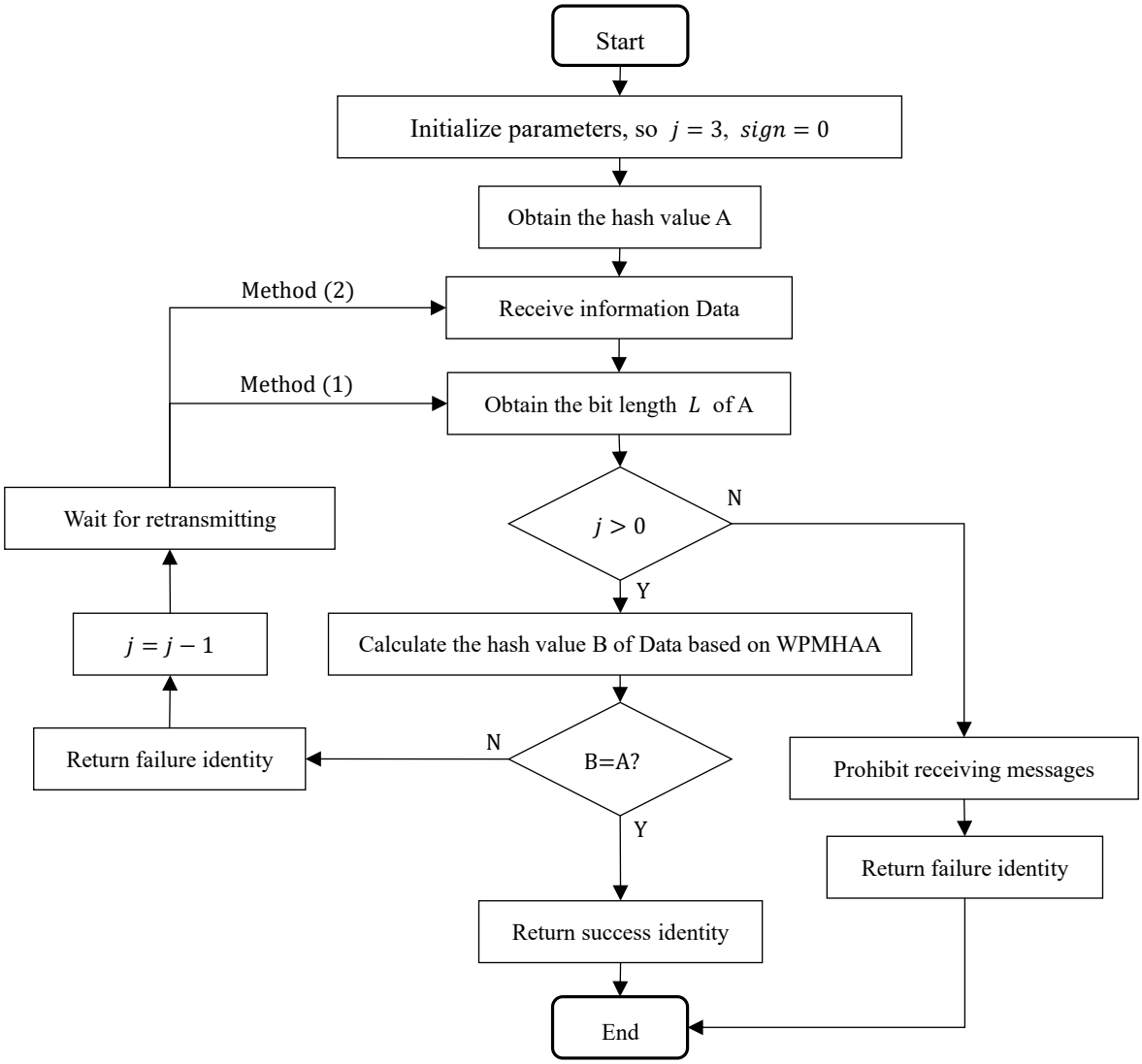

Figure 2 Schematic diagram of the receiving end system

The method (1) and method (2) proposed in Figure 2 are based on the system safety requirements. For example, when $j = 2$, Data does not need to be retransmitted for verification; when $j = 1$, Data must be retransmitted. The purpose is to reduce the burden of retransmission.

4.3 Experiment

(1) customize hash value length $bytelength$

Input: abcdefghijklmnopqrstuvwxyz123456789

The output $bytelength$ is the hash value of 4, 8, 15, 20, 32 (unit: byte).

Table 3 The customized hash values of different lengths

$bytelength$	Hash value
4byte	b21527d1
8byte	7a63428361901e1c
15byte	d99067a5ebd9dd1f5eefc63bddcfa
20byte	cd5c2c83aa1cf1468590303811895af11b

	f06943
32byte	233652b6ff6365c634bd1f7dbe58967ebd e64b849b5beed3a145f4b83efe365

(2) The hash value obtained by different experiments of the input data is different, and the probability $p(0)$ of the symbol 0 in the hash value approaches 0.5

Table 4 The value of $p(0)$ when the hash value length is 128 bits (16 bytes)

Input	Hash value	$p(0)$
0123456789	9083a6bf012fe055381bb0d1cee1b307	0.53125
abcdefghijklmnopqrstuvwxyz123456789	8aa3404995ca1e7c33a3785df04c9059	0.54687
0	14e5c8666b999536301be0347d0e625b	0.53125
AAAAAAA	4a96d54d2f5dea8f0cb955643eaaf1fe	0.44531

Table 5 The value of $p(0)$ when the hash value length is 512 bits (64 bytes)

Input	Hash value	$p(0)$
0123456789	bfb98538a61b26ecc102452b00b9cb63b 25be606c2707dcd0b7dbf6f411593cda1 025c280f78cc704f345d7264d4fa16c7a 5a3b9f518db58e6d56fddbcf2e107	0.49218
abcdefghijklmnopqrstuvwxyz123456789	a84a4fb59a0a1d7c7169a40c8e28c7f4b 601634e5e073eee47e20ad270fcf7be4bf de766b4d33450d69cafc26c4fad7becbc 3442de5a9a776dd50890a6a3db06	0.48046
0	b8ce085f673a3de4f67196de9b8edf786 d07606234baa19b5c2899040a11175e5 6c8f52e9287a87224a8dc111cddc6f1fae 3597c7bb20795235de3a014c223ab	0.50585
AAAAAAA	f1fbff90d76d3750840ae113c1c6e64803 5e5d123ae9c5854b64b98cab90c3c0b36 4c2dcaf26bd12bd6913fb38d8ec2b9227 576c0f3c9ed5a1897febbeaaf8f7	0.47851

It can be concluded from the experimental results that the hash length of the method in this paper can be customized, and the probabilities of symbol 0 and symbol 1 in the hash value tend to be equal. WPMHA is the core of the adaptive attack intensity security system. The simulation experiment is implemented based on the TCP/IP protocol Socket. Data and hash value A are processed for random bit errors during data transmission. In the experiment, d is set as a random number of 1-128, $i = j = 3$, and the initial value of L for each simulation is 512 bits. The conclusion is drawn from the not less than 10^5 number of simulations that the security system based on Figure 1 and Figure 2 can achieve adaptive attack intensity.

4.4 Safety analysis

Piecewise iteration, exclusive OR operation, and round function are commonly used methods to defend against

linear and differential attacks. According to the coding process in Section 4.1, firstly, since $X_i = 0$ and $X_i = 1$, the two-dimensional coordinates (x, y) are random when checking table 1 or table 2, so the values of $f(x, y)$ and $g(x, y)$ are random. Secondly, the probability of the symbol 0 in the sequence X is fixed, but the probability p of the symbol 0 in each binary sequence is different. The precision of p determines the values, so $H(Y)$ is unknowable. Since $2^{(H(Y)-L/v)}$ and $\frac{g(x,y)}{10^s}$ are both unknown, the binary length of L_v is unknown. $L_v = L_v + T$, that is, the coding result is piecewise-iteratively operated in the way of addition, so that the bit variation of L_v is random during operation. Therefore, it can be analyzed that the sufficient condition is: when the i-th symbol of sequence E (E is any binary sequence other than X) is being encoded, $f(x, y)$, $g(x, y)$, $r(i)$, R_i and L_i are consistent with encoding X_i, which means that you can generate the same hash value. There is uncertainty in the sufficient condition, and the probability of satisfying a certain sufficient condition in each segment of coding can be analyzed.

(1) When $X_i \in \{0,1\}$, $f(x, y)$ only acts on the symbol X_i. Each segment has v bits and the probability that each symbol correctly selects $f(x, y)$ is $p(x, y, f) = \frac{1}{2^v}$.

(2) $g(x, y)$ only acts on the weight coefficient $r(i)$. According to (1) the probability that each symbol correctly selects $g(x, y)$ is $p(x, y, g) = \frac{1}{2^v}$.

(3) Assuming that the precision of p is u-bit binary, then we obtain that the probability p of the symbol 0 in each binary sequence has 2^u possible values. Since $r = 2^{(H(Y)-L/v)}$, $p(r) = \frac{1}{2^u}$.

(4) The binary digits of T and L_v are L, and the iterative operation is $L_v = T + L_v$, then the probability that both L_v and T are correct is $p(T, L_v) = \frac{1}{2^L}\frac{1}{2^L} = \frac{1}{4^L}$.

The same coding result can be obtained when all the above sufficient conditions are met. $f(x, y)$ and $g(x, y)$ are S-boxes. Piecewise iteration and XOR operation based on S-box eliminate linear correlation to a certain extent. The probabilities of symbol 0 and symbol 1 of each sequence are different and the weighting coefficient used for encoding of each symbol changes due to the round function, so that the length of the weighted probability model after encoding is random. The bit length of the hash value in WPMHA is random, that is, L is a random value, and the collision probability is smaller when L is fixed. And the longer the hash value, the smaller the collision probability. Therefore, as the number of checking in the security system based on Figure 1 and Figure 2 increases, the randomness of L increases, making the probability of collision closer to zero.

V: CONCLUSION

On the basis of the distribution function interval coding technology, this paper proposes a weighted probability

model Hash algorithm. The algorithm has the advantages of self-defined hash value length and adaptive adjustment according to security level and attack intensity, which can not only ensure information security, but also effectively reduce the pressure of message network transmission, verification calculation and storage, and can be widely used in digital signatures, file verification, and data transmission verification. In order to facilitate inspection and application, the algorithm of this article has been published in GITHUB, and the access address is https://github.com/Jielin-Code/WjlHashAlgorithm.

REFERENCES

[1] Rivest R L . The MD5 Message-digest Algorithm. 1992.

[2] Wang X , D Feng, Lai X , et al. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. cryptology eprint archive report, 2004.

[3] Yu H , Wang G , Zhang G , et al. The Second-Preimage Attack on MD4[C] International Conference on Cryptology & Network Security. Springer Berlin Heidelberg, 2005.

[4] Wang X , Yin Y L , Yu H . Collision search attacks on SHA1. Springer Berlin Heidelberg, 2005.

[5] H Choi, Seo S C . Optimization of PBKDF2 using HMAC-SHA2 and HMAC-LSH Families in CPU Environment[J]. IEEE Access, 2021, PP(99):1-1.

[6]来齐齐, 杨波, 禹勇,等. 基于格的哈希证明系统的构造综述[J]. 密码学报, 2017(05):474-484.

[7] Nist F . The Keyed-Hash Message Authentication Code[J]. 2008.

[8] Michail H E , Kakarountas A P , Milidonis A , et al. Efficient implementation of the keyed-hash message authentication code (HMAC) using the SHA-1 hash function[C] IEEE International Conference on Electronics. IEEE, 2004.

[9] Vamsi T S , Kumar T S , Krishna M V . Impact Analysis of Black Hole, Flooding Attacks and Enhancements in MANET Using SHA-3 KeccakAlgorithm[M]. 2021.

[10]刘轩黄. 关于随机过程一阶概率分布函数的遍历性[J]. 大学数学, 1989.

[11] G. N. N. Martin, Range encoding: an algorithm for removing redundancy from a digitised message. Video & Data Recording Conference, held in Southampton July 24-27 1979.

[12] Ian H.Witten, Radford M.Neal,John G.Cleary. Arithmetic Coding for Data Compression.Communications of the ACM. 1987,30(6):520~539.

[13] C. E. Shannon. A mathematical theory of communication. Bell Syst. Tech. J., 27:379–423,623–656, 1948.

[14] T.M.Cover and J.A.Thomas,Elements of Information Theory.New YorK,Wiley 1991.

[15] Wang W , Feng A . Self-Information Loss Compensation Learning for Machine-Generated Text Detection[J]. Mathematical Problems in Engineering, 2021, 2021(1):1-7.

[16] Manral V . Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD[J]. Cryptology Eprint Archive Report, 2004, 2004.

[17] Weber B , Zhang X . Parallel hash collision search by Rho method with distinguished points[C] 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT). IEEE, 2018.

[18] Keller, Nathan, Shamir, et al. Improved Single-Key Attacks on 8-Round AES-192 and AES-256[J]. Journal of cryptology: the journal of the International Association for Cryptologic Research, 2015.

[19] Tromer E , Osvik D A , Shamir A . Efficient Cache Attacks on AES, and Countermeasures[J]. Journal of Cryptology, 2010, 23(1):37-71.

[20] Biham E , Biryukov A . An Improvement of Davies' Attack on DES[J]. Journal of Cryptology, 1997, 10(3):195-205.

About the author:

WANG Jielin, male, born in 1985 from Pingjiang, Hunan, a distinguished professor of Hunan International Economics University, and his main research direction is coding technology.

GAO Jinding, male, born in 1981 from Taojiang, Hunan, Ph.D (post), professor of electronics, and his main research direction is

digital signal processing FPGA real-time realization technology and information coding.