A New Digital Signature Algorithm Similar to ELGamal Type

Haipeng Chen
College of Computer Science and Technology, Jilin University, Changchun, China
E-mail: chenhp@jlu.edu.cn

Xuanjing Shen and Yingda Lv
College of Computer Science and Technology, Jilin University, Changchun, China
E-mail: {xjshen@jlu.edu.cn, lvyingda1983@163.com}

***Abstract*—Application of digital signature technology becomes more extensive, but many exposed digital signature algorithms have increasingly revealed some shortcomings and deficiencies. Aiming directly at the frequently used digital signature technologies, which are weak to Substitution Attack and Homeostasis Attack, the authors perform the hash transformation on messages before signature. Then, a hash round function is constructed, which simultaneously satisfies the characters of balance, high nonlinearity, strict avalanche criterion and realization of software. Moreover, making use of the hash round function, a new hash algorithm named HRFA (Hash Round Function Algorithm) is contrived. On this basis, aiming at the defect that the existing digital signature algorithms are weak to active attack and impersonation attack, using the hash algorithm named HRFA and the self-certified public key system, a new kind of digital signature algorithm, which is similar to ELGamal, named H-S DSA (Hash Round Function and Self-certified Public Key System Digital Signature Algorithm) is raised and realized. Finally, the authors analyze the H-S DSA from two aspects of security and time-complexity. And, the results show that the new designed digital signature algorithm named H-S DSA not only has better security strength, but also has lower time-complexity.**

***Index Terms*—digital signature, similar to ELGamal, HRFA, H-S DSA, algorithm analysis**

I. INTRODUCTION

Digital signature technology is presented for the first time in 1976 by Diffie and Hellman in their famous paper named "New Direction in Cryptography"[1]. Subsequently, it has aroused interest in many researches and was grown by leaps and bounds.

At present, there are many mature digital signature algorithms, among which, RSA digital signature algorithm, ELGamal digital signature algorithm, U.S. digital signature standard/algorithm (DSS/DSA) and LUC digital signature algorithm are the most representative ones. However, developing to the present, these commonly used digital signature algorithms more or less exist kinds of problems. RSA scheme holds the characteristic of homeostasis, and is consequently weak to active attack and impersonation attack [2]. Meanwhile, ELGamal digital signature algorithm has severe technological defect although no detailed cryptanalysis test is employed [3]. It is very fragile to substitution attack and forgery attack. In addition, DSS, one variation of ELGamal scheme, suffers the same attacks as ELGamal. Worse still, the public modulus and too short secret key leaves a further security risk to DSS [4] [5]. As for LUC solution, signature could be forged using the signed information[5]. Moreover, LUCELG and LUCDIF are very weak to the sub-index time attacking algorithm [6].

Next, we will make detailed analysis of ELGamal digital signature algorithm.

The scheme can be briefed as below: Set p is a large prime number, q is a large prime number's factor of P-1, g is a large prime number's factor with order q over GF (P), and gcd(g,p)=1. User A selects a random number x, here $x \in (1, p-1)$, calculating $y = g^x \bmod p$. The public keys are Y, g, p, the secret key is x. If users want to sign for message m, the following steps must be carried out:

1) Select integer k randomly, $k \in (1,p-1]$ and gcd (k,p-1)=1;
2) Calculate: $r = g^k \pmod p$;
3) Calculate s which satisfies $m=xr+ks \pmod{(p-1)}$, that is, $s=(m-x^r)k^{-1} \pmod{(p-1)}$.

Then A's signature on m is (r, s), after receiving the signature (r, s) of m from A, B verifies if (r, s) meets the equation:

$$gm=Yxr^s \pmod p.$$

If it meets the above equation, the signature will be accepted; otherwise, the signature will be rejected.

Drawbacks of ELGamal digital signature algorithm:

1) After introduction of RSA in 1978, great deal of

Scientific and technological development project of Jilin Province (20050305);

Corresponding author: Xuanjing Shen(1958-), male, PhD supervisor, research interests: Computer Network Security, Digital Image Processing and Pattern Recognition, Intelligent Measurement System.

doi:10.4304/jsw.5.3.320-327

energy has been spent to find its defects which can be deciphered. There is no risk when it is used in some scope of the protocol, however, ELGamal algorithm is not tested by a detailed password's analysis and decipher, there are still serious technical defects.

2) It is consequently weak to active attack and impersonation attack[7]. If the attacker replaces some legitimate users' public key in SA by public key corresponding to the randomly selected private key successfully, he will be able to fake any of those users' signatures.

3) Suffer substitution attack[8]. These attacks include using part of the signature s and only use the public key Y. The substitution attack carried out by part of signature s is the most important attack ELGamal signature scheme faces.

4) Suffer fake attack[3]. The forgery starts from the signature, making any changes to the message to form signature of another message m, and, this signature will meet the same equation as the original signature.

5) Random key k can not be used to sign for different messages repeatedly[3], otherwise, the attacker can easily obtain the signature's key x.

6) He and Keisler pointed out that, the signer can be faked to sign any messages[9]. If three random keys k_i (i = 1,2,3) satisfy $k_3 = k_1 + k_2$, then r_i (i = 1,2,3) will satisfy $r_3 = r_1 r_2$. Thus, the attacker can find key x. This is similar to homomorphism attacks RSA signature faces. What is different is, the homomorphism of RSA signature is only to fake the signature by attackers, and it can be overcome by the use of hash function. But, for the homomorphism attack ELGamal digital signature scheme faces, there is still no effective solution.

7) The issue of Subliminal Channel[3]. For closing Subliminal Channel on ELGamal digital signature scheme, so far there are no results of any research.

Concerning about these problems, we propose the H-S DSA (hash round function and self-certified public key system digital signature algorithm) based on hash round function and self-certified public key system after relevant study. We have also analyzed this scheme with respect to security and time-complexity. The results demonstrate that this newly designed algorithm H-S DSA possesses adequate security and relatively low time-complexity.

II. Design for Hash Round Function Algorithm

Hash function is a kind of function named h, which compresses numeric string denoted by M with arbitrary length to an output numeric string denoted by H with fixed length, and, $H=h(M)$ is called as Hash Value of M, it also can be called as Digital Finger Print of M or Message Digest of M.

For a hash function h, if it is easy to calculate H=h(M) using M, but the calculation is not feasible, which yields a M' to make $h(M')=H$, that is to say h is one-way function, then h is called as one-way hash function. It is introduced mainly based on the consideration of digital signature or message authentication.

Applying hash function to the digital signature can bring the following benefits: [10]

1) Can undermine some kind of mathematical structure of digital signature scheme, such as homomorphism structure.

2) Can increase the speed of digital signature. When the signer would like to sign a message x, he firstly constructs a message digest z=h(x) (h is a hash function), and then calculates signature y=Sigk (z).

3) Can leak a signature without disclosure of the message corresponding to the signature. For example, y=Sigk (z) is a signature for message x, where z=h(x), it can make (z, y) known to public, but keep x a secret.

4) Can distinguish signature transformation and encryption transformation, allow using private key cryptography to achieve confidentiality, and using public key cryptography to achieve digital signature. In the Open System Interconnection Reference Model(OSIRM) of ISO, one of the merits of this separation is to provide integrity and confidentiality between different layers.

Boolean Algebra, which was discovered and constructed by X.M.Zhang and J.Seberry et al, holds an excellent property of cryptography, which is widely believed witch needs to be satisfied as much as possible when designing cryptographic algorithm like one-way hash round function[11]. Three properties are considered to be essential:

1. Satisfying 0-1 balance;
2. High nonlinear;
3. Satisfying strict avalanche criterion;

Furthermore, as a function set, it has the following characters:

4. Mutual linear nonequivalent;
5. Mutual output irrelevant.

A. Construction of Boolean algebra on V_{2k+1}

Assume k≥1, f be Bent function on V_{2k} [12], and h for non-constant affine function on V_{2k}, so consequently $f(x) \oplus h(x)$ will be also Bent function. Yet generally, the values of f(x) and $f(x) \oplus h(x)$ can be considered as adding different times of 1. (We can replace h(x) by $h(x) \oplus 1$ and $f(x) \oplus h(x)$ by $f(x) \oplus h(x) \oplus 1$).

On V_{2k+1}, define function g as:

$$\begin{aligned} & g(y, x_1, \ldots, x_{2k}) \\ & = (1 \oplus y) f(x_1, \ldots, x_{2k}) \oplus y(f(x_1, \ldots, x_{2k}) \oplus h(x_1, \ldots, x_{2k})) \\ & = f(x_1, \ldots, x_{2k}) \oplus y(h(x_1, \ldots, x_{2k})) \end{aligned} \quad (1)$$

The above function g is Bent function that is balanced, high non-linear and satisfies strict avalanche criterion on V_{2k+1} [13].

Lemma 1 function g defined by (1) is a balanced function.

Lemma 2 non-linearity of function defined by (1), satisfies $N_g \geq 2^{2k} - 2^k$.

Lemma 3 function g defined by (1) satisfies strict avalanche criterion.

Some theorems can be summarized from lemma 1, 2 and 3:

Theorem 1 if $k \geq 1$, function g that is defined by (1) is a balanced function on V_{2k+1}, with its non-linearity $N_g \geq 2^{2k} - 2^k$ and satisfies strict avalanche criterion.

According to theorem 1, to build a Bent function on V_{2k+1} that is balanced, high non-linear and satisfies strict avalanche criterion, some steps can be followed:

1) Choose an appropriate Bent function f on V_{2k};

2) Choose the optimum affine function h on V_{2k+1} (with respect to the non-linearity of the finally built Bent function) and figure out the needed result g according to (1);

3) Do essential linear transformation to g (this transformation won't change the balance or non-linearity), so as to meet the final appliance need.

B. Constructing a new hash round function

Firstly, choose four Bent functions on V_4:

$$\begin{cases} f_0(x_1,x_2,x_3,x_4) = x_1x_3 \oplus x_3x_4 \oplus x_1x_2x_4 \oplus x_1x_3x_4 \\ f_1(x_1,x_2,x_3,x_4) = x_1x_2 \oplus x_3x_4 \\ f_2(x_1,x_2,x_3,x_4) = x_1x_3 \oplus x_1x_4 \oplus x_2x_4 \\ f_3(x_1,x_2,x_3,x_4) = x_1x_4 \oplus x_3x_4 \oplus x_1x_3x_4 \end{cases}$$

Then choose non-constant affine functions l_0, l_1, l_2 and l_3 on V_4.

$$\begin{cases} l_0(x_1,x_2,x_3,x_4) = 1 \oplus x_1 \\ l_1(x_1,x_2,x_3,x_4) = 1 \oplus x_2 \\ l_2(x_1,x_2,x_3,x_4) = 1 \oplus x_3 \\ l_3(x_1,x_2,x_3,x_4) = 1 \oplus x_4 \end{cases}$$

Use (1) to calculate:

$$g_i(x_1,x_2,x_3,x_4) = f(x_1,x_2,x_3,x_4) \oplus x_3 l_i(x_1,x_2,x_3,x_4), i = 0,1,2,3$$

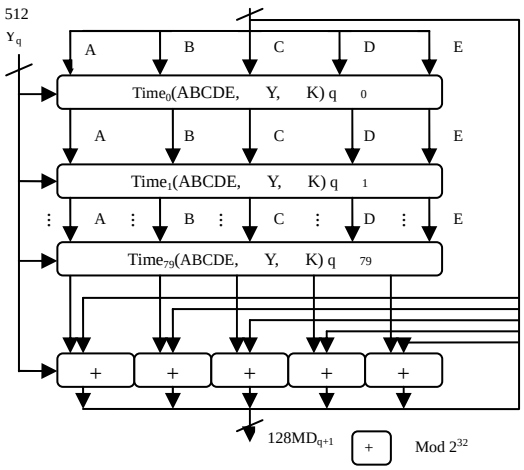

Figure 1. Dealing with every 512bit group of HRFA

Four Bent functions are figured out that satisfy balance, high non-linearity and strict avalanche criterion simultaneously.

$$\begin{cases} g_0(x_1,x_2,x_3,x_4,x_5) = x_1x_3 \oplus x_3x_4 \oplus x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_1x_5 \oplus x_5 \\ g_1(x_1,x_2,x_3,x_4,x_5) = x_1x_3 \oplus x_3x_4 \oplus x_2x_5 \oplus x_5 \\ g_2(x_1,x_2,x_3,x_4,x_5) = x_1x_4 \oplus x_3x_4 \oplus x_1x_3x_4 \oplus x_3x_5 \oplus x_5 \\ g_3(x_1,x_2,x_3,x_4,x_5) = x_1x_3 \oplus x_1x_4 \oplus x_2x_4 \oplus x_4x_5 \oplus x_5 \end{cases}$$

Do linear transformation to these four Bent functions afterwards:

$$\begin{cases} \sigma_0(x_1,x_2,x_3,x_4,x_5) = (x_1,x_5,x_2,x_3,x_4) \\ \sigma_1(x_1,x_2,x_3,x_4,x_5) = (x_1,x_5,x_2,x_3,x_4) \\ \sigma_2(x_1,x_2,x_3,x_4,x_5) = (x_1,x_5,x_2,x_3,x_4) \\ \sigma_3(x_1,x_2,x_3,x_4,x_5) = (x_1,x_5,x_2,x_3,x_4) \end{cases}.$$

Then, we get the final results:

$$\begin{cases} h_0(x_1,x_2,x_3,x_4,x_5) = x_1x_2x_3 \oplus x_1x_3x_5 \oplus x_1x_2 \oplus x_2x_3 \oplus x_1x_4 \oplus x_4 \\ h_1(x_1,x_2,x_3,x_4,x_5) = x_1x_2 \oplus x_3x_4 \oplus x_2x_5 \oplus x_5 \\ h_2(x_1,x_2,x_3,x_4,x_5) = x_1x_2x_4 \oplus x_1x_4 \oplus x_2x_4 \oplus x_2x_3 \oplus x_3 \\ h_3(x_1,x_2,x_3,x_4,x_5) = x_2x_4 \oplus x_2x_3 \oplus x_1x_4 \oplus x_4x_5 \oplus x_5 \end{cases}.$$

Here, h_0, h_1, h_2 and h_3 are the built hash round functions we want, all of which are balanced functions on V_5 that satisfy the strict avalanche criterion. The non-linearity is all $2^4 - 2^2 = 12$, actually the maximum linearity possible for balanced function on V_5. As a function set they are mutually linear nonequivalent [14].

Use (1) directly on Bent function, we have received four functions g_i（i= 0,1,2,3）, and the four functions have already been balanceable, high nonlinear and satisfied strict avalanche criterion, however, these functions as a function set to be output are related to each other[12]. The role of linear transformation is to make the output isolated by transforming the coordinates of the input. This transformation is just one of the transformations which meet the requirements.

By now, we have built up the new hash round functions. These functions fully satisfy cryptography Bent function characters 1~3; and as a function set, they also satisfy both character 4 and 5, which means a largely reinforced security for hash algorithm.

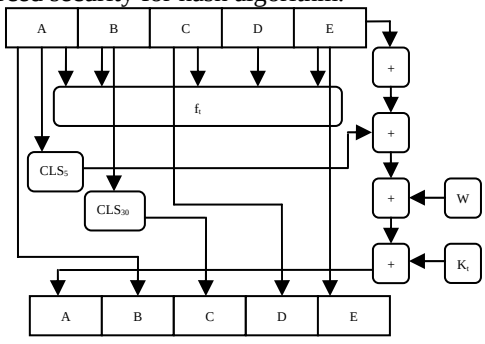

Figure 2. The basic calculation block diagram of t in HRFA

t	W_t
0≤t≤15	x_t
16≤t≤79	$(w_{t-3} \oplus w_{t-8} \oplus w_{t-14} \oplus w_{t-16})$ □ 1

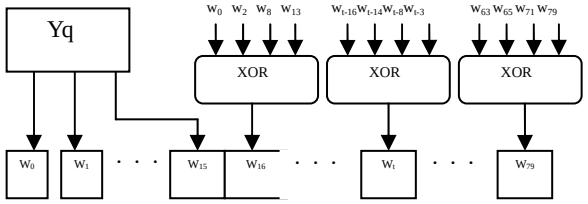

Figure 3. Eighty 32bit words produced by HRFA after treating an input group

C. Designing for Hash Round Function Algorithm

Input: the length of input messages should be less than 264bit;

Output: the output length is 160bit;

Verification: for receiver, they calculate the hash value of the received message, and verify it in compliance with the recovered hash value from decryption. It is unfeasible to calculate the hash value for a forgery message and make it the same as the given one, and it is also impossible to find out two different messages sharing the same hash value in calculation. Any change in message would lead to a different hash value with high probability, which would consequently result in the failure of signature verification.

Given a message x, the procedure of building a 160bit message abstract are as follows:

1. Pad the message x into an integral multiple of 512bit. Pad a 1 on the right side of x, and then cascade with enough 0 so as to extend the length of module to 448bit. At last, cascade with 64bit to represent the length of $|x|$ (use L to represent). Given that $M = x \| L \| 0d \| L$, the padding value is 1~512bit.

2. Use five 32bit variables (A, B, C, D, E) as the initial value (hexadecimal).

A = 67452301, B=EFCDAB89, C=98BADCFE, D=10325476, E=C3D22E1F0.

3. $M = Y_0 \| Y_1 \| \ldots \| Y_L$, here $Y_0, Y_1 \ldots Y_L$ are all 52bit groups, and Y_q in each group has sixteen 32bit words. Each time receiving 512bit, store values of A, B, C, D, E into another five 32bit variables AA, BB, CC, DD and EE, respectively (AA=A, BB=B, CC=C, DD=D, EE=E), then carry out four rounds of iteration with 20 completed in each round(Fig. 1). t denotes each basic operation (such as Fig. 2).

E=D, D=C, C=(B<<30), B=A, A=(A<<5)+f_t(A, B, C, D, E)+E+W_t+K_t. here , the values of $f_t(A,B,C,D,E)$ and constant K_t are shown in Tab. 1. Notice that $f_t(A,B,C,D,E)$ is hash round function built in Ⅱ.B. The values of W_t are shown in Fig. 3.

After dealing with all 512bit groups, that is, completing 80 basic calculations, A=A+AA, B=B+BB, C=C+CC, D=D+DD, E=E+EE. Then, cascade the values of A, B, C, D and E, and output them as hash value with a length of 160 bits.

Table 1. The Value Ranges of $f_t(A,B,C,D,E)$ and constant K_t

Calculate times t	f_t (A,B,C,D,E)	K_t
0≤t≤19	ABC ⊕ ACE ⊕ AB ⊕ BC ⊕ AD ⊕ D	5A827999
20≤t≤39	AB ⊕ CD ⊕ BE ⊕ E	6ED9EBA1
40≤t≤59	ABD ⊕ AD ⊕ BD ⊕ BCC	8F1BBCDC
60≤t≤79	BD ⊕ BC ⊕ AD ⊕ DE ⊕ E	CA62C1D6

4. Value of W_t: Extend the input 16bit words to 80bit words which are essential in treatment (Fig. 3).

The security of hash function is equivalent to that of hash round function, so hash round function plays a vital role in the design of hash function. Here, starting from four Bent functions on V_4, the authors have constructed four Boolean functions satisfying balance, highly nonlinear and strict avalanche criteria at the same time, and the four Boolean functions are as the hash round function of hash algorithm. Afterwards, using these hash round functions, HRFA has been designed, and in the process of the algorithm's realization, each step uses the results from the previous step, it has a good avalanche effect. Moreover, the hash value is 160 bits, the resistance to exhaustive search attack is stronger, and it has better security.

III. Design for H-S DSA

Digital Signature Algorithms based on Public key cryptosystem, such as Diffie-Hellman、RSA and ELGamal, are all algorithms which the signer uses the private key to generate the message's signature, and then the verifier uses the signer's public key to verify the signature. Generally speaking, public keys are all kept in the key directory maintained by the System Administrator (SA). When verifying the digital signature, the verifier will first obtain the public key from SA through public communication channel. There is a problem in this process, that is, a false public key is being substitute for a true public key. If the adversary replaces some legitimate users' public keys in key directory by public keys corresponding to the private keys he chose, or he replaces the public keys in its transmission process, he will be able to fake any of those users' signature, which are the so-called active attacks and fake attacks[15]. Most of the existing digital signature schemes have this problem. In order to overcome this shortcoming, it is necessary to verify the validity of public key firstly before using the public key to verify signatures. There are three methods to verify public key, however, since Self-certified public key method does not require an additional certificate, and relatively speaking, its storage and computation are greatly reduced, the authors present a new digital signature algorithm named H-S DSA(Hash Round Function and Self-Certified Public Key System Digital

Signature Algorithm), using HRFA algorithm and the above Self-certified public key method.

Signing for messages of this digital signature algorithm is composed of two parts, and this algorithm is similar to ELGamal signature algorithm in form, so the authors call it digital signature algorithm similar to ELGamal (H-S DSA). The security of this algorithm lies on the one-way hash round function, factorization (FAC) and discrete logarithm problem assumption (DL). Next, we first introduce the Self-certified public key system.

A. Self-certified public key system

Self-certified public key system (SCPKS) was proposed by Girault in 1991[16], which was commonly called RSA-based SCPKS, because the public/private key pair of this system are based on RSA cryptography. It consists of two steps including system initiation and user registration. Detailed descriptions are as follows:

1. System initiation

System Administration (SA) will choose two prime number p, q, calculate N=p.q, and get the integral number g (maximum exponent number in (Z/N^Z)). Then calculate the secret key according to RSA, with regards that (e, d)=1(mod φ(N)) be satisfied(φ is Euler's constant). And make public N, g, e whereas p, q, d would be kept confidential.

2. User registration

When user U_i with an identity ID_i wants to access the system, he should first choose a key x_i in $(Z/N^Z)^*$ and calculate:

$$v_i = g^{-x_i} \bmod N. \tag{2}$$

Then send $\{ID_i, v_i\}$ to SA to register. His public key would be then calculated by SA using (2).

$$y_i = (v_i - ID_i)^d \bmod N \tag{3}$$

Conclusion from (2) and (3): the public key of user U_i is actually the signature of his key and ID, which is produced by SA. Meanwhile, the private key of user is unknown to SA. User U_i can then verify the validity of public key y_i using (4):

$$y_i^e + ID_i = g^{-x_i} \bmod N. \tag{4}$$

The self-certification procedure of public key is:

If user U_i wants to verify his identity, these steps based on Beth's[17] or Schnorr's[18] authentication protocol need to be executed:

①User U_i sends $\{ID_i, v_i\}$ to the verifier, who would then calculate using (5):

$$v_i = (y_i^e + ID_i) \bmod N. \tag{5}$$

U_i chooses a random number r_i, and calculates t_i using (6) and send it to the verifier.

$$t_i = g^{r_i} \bmod N \tag{6}$$

Verifier would choose a random number k in (Z/N^Z), and send it to U_i.

② U_i calculates s_i using (7), and send it to the verifier.

$$s_i = r_i + x_i . k \tag{7}$$

③verifier verifies the (8):

$$g^{x_i} . v_i^k = t_i \bmod N. \tag{8}$$

Then, if (8) is tenable, verifier would consider the identity of U_i valid, otherwise invalid.

Based on the analysis above, no extra certifications are needed when verify the identity of U_i because the public key y_i is self-certified. Under FAC and DL, however, except for U_i, x_i cannot be derived from y_i or any other public information.

In the event that SA forges a U_i, saying he chooses a private key x_i, calculates the corresponding public key y_i using (3) and manages to pass the verifying equation of (4), the fact that one user U_i has two valid public key would however certificate the dishonesty of SA.

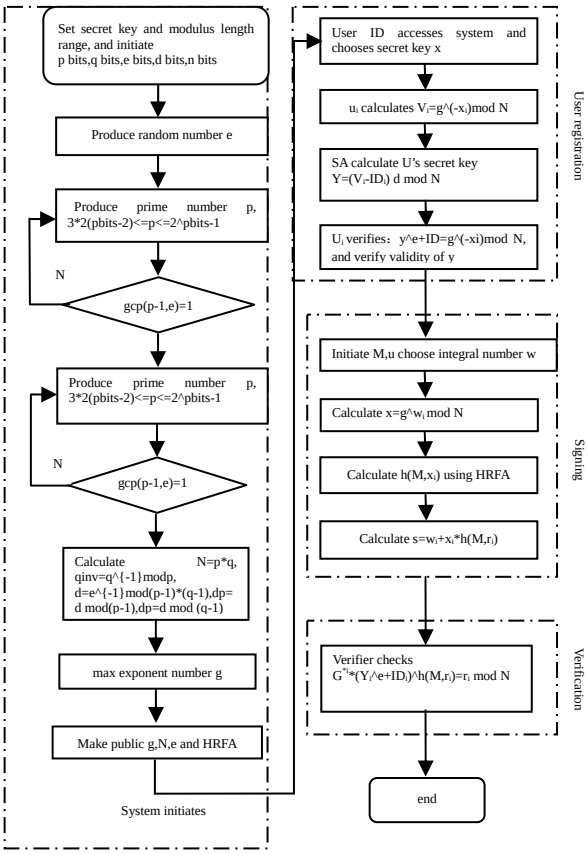

Figure 4. The flow-sheet diagram of H-S DSA

B. H-S DSA

Self-certified public key can effectively overcome active attacks and fake attacks, so on this basis, the digital signature algorithm similar to ELGamal (H-S DSA) is presented.

H-S DSA consists of four steps: system initiation, user registration, signature creation and signature verification. The first two steps are the same as that of Girault's SCPKS. What is different is that SA needs to make public a one-way hash round function h during system initiation with the output length shorter than that of N, that is for any m, we have $|h(m)| \le |N|$. The main purpose of h is to condense the coming signature message into message abstract so as to avoid plaintext attack.

The signature creation and certification steps are described as:

1. Signature creation

M is a message that needs to be signed. The signer U_i chooses a random number w_i, and calculates the signature (r_i, s_i) of M, where we have:

$$r_i = g^{w_i} \bmod N. \quad (9)$$

$$s_i = w_i + x_i . h(M, r_i) \quad (10)$$

Afterwards, U_i sends M and the signature (r_i, s_i) to the verifier.

2. Signature verification

After receiving M and (r_i, s_i), verifier will verify (11):

$$g^{s_i} \times (y_i^e + ID_i)^{h(M, r_i)} = r_i \bmod N. \quad (11)$$

If (11) is tenable, the verifier accepts the signature validity of M, otherwise it will deny.

This implies that, public key y_i from secret key lists of SA needs also to be verified by signature certification equation. We are now proving that both the signature of M and public key of U_i can be verified, provided that they pass (11).

Theorem 2 if (11) is tenable, public key of U_i will be verified at the time the signature M is verified.

To prove: take on both side exponent with a base g, and we have:

$$g^{s_i} = g^{w_i} . g^{x_i \cdot h(M, r_i)} \bmod N. \quad (12)$$

According to (2), (4) and (9), transform (11) into (13):

$$g^{s_i} = r_i . \left(y_i^e + ID_i\right)^{-h(M, r_i)} \bmod N. \quad (13)$$

This equation in fact contains a hidden (11), which means (r_i, s_i) would be verified at the time public key y_i be verified.

In addition, from (9), (12) and (13), we have:

$$\left(y_i^e + ID_i\right)^{h(M,n)} = \left(g^{-x_i}\right)^{h(M,n)} \bmod N. \quad (14)$$

We can derive (4) from (14), which confirms that y_i is the signature of x_i and ID_i. In other words, once (r_i, s_i) be verified, y_i would be verified simultaneously.

In this scheme, the verification of signature and public key complete at the same time, so no extra time would be spent on verifying the public key. In addition, since the self-certified public key needs not to store extra certification, the storage and calculation both decrease a lot.

H-S DSA procedure is shown in Fig. 4.

IV. H-S DSA analysis

To ensure that an algorithm meets or exceeds the designed expectations, it is essential to analyze the performance of this algorithm to detect potential problems, this process is called as Algorithm Performance Analysis. Specific to the H-S DSA, its performance analysis including security analysis and time complexity analysis, is to check whether the algorithm can work effectively.

A. Security analysis

H-S DSA argorithm has used a one-way hash function, and its safety mainly lies in the hash round function used in each round. In addition to the one-way hash function, the safety of H-S DSA also depends on the following two well-known password assumptions: Facts Factorization Hypotheses (FAH) and Discrete Logarithm Problem (DLP)[19].

1、Facts Factorization Hypotheses. If N is the product of two large prime numbers, and two integers e and d satisfy: e·d=1(mod φ (N)), then, the three items as following will not be feasible in the calculation.

(1) Find the factors of N;

(2) Give integers M and C to find d which makes $C^d = M \pmod N$;

(3) Give integer C to find M which makes $M^e = C \pmod N$.

2、Discrete Logarithm Problem. Give a large prime number p, and g is the primitive element over GF(p). Integer $y \in (1, p-1)$ is not feasible in the calculation of finding out x to make $y = g^x \bmod p$. Next, on the basis of Facts Factorization Hypotheses and Discrete Logarithm Problem, we analyze the three possible attacks to H-S DSA. These attacks include exposing a secret parameter, forging of digital signature of a given information.

Attack 1: Adversary discloses the user's secret key x_i via U_i's public key y_i.

Security analysis: adversary could get

$$v_i = \left(y_i^e + ID_i\right) \bmod N$$

from $y_i = \left(v_i - ID_i\right)^d \bmod N$, which implies that he may calculate x_i directly by:

$$v_i = g^{-x_i} \bmod N$$

or

$$y_i^e + ID_i = g^{-x_i} \bmod N .$$

However, in such situation, FAC and DL assumption are inevitable for him to face.

Attack 2: Adversary discloses user's secret key x_i from U_i's signature to M, i.e. (r_i, s_i).

Security analysis: suppose that adversary obtains w_i in advance, he could then calculate x_i from (10) even if only r_i is known to him. In other words, adversary can calculate w_i via (9), the same as that in attack 1, which means still that FAC and DL assumption would be still inevitable. Moreover, (3) could be another approach to x_i and w_i. However, the amount of unknown variables x_i and w_i is always larger than that of equations in system. This makes the attempt impossible.

Attack 3: With x_i unknown, adversary forges the signature (r_i, s_i) to the randomly chosen message M under the name U_i.

Security analysis: there exists two ways for adversary to forge valid signature to message M, both of which need (4):

1. Fixes r_i first, then calculates s_i;
2. Fixes s_i first, then calculates r_i from equation (4).

In the first method, adversary will have to face to breakdown hypothetic FAC and the discrete logarithm problem DLP assumption. And it will be more complex to use method (2) because with r_i being under the protection of one-way hash function, an extra obstacle would come to him.

From the analysis above, we can come to the conclusion that under the one-way hash function, FAC and DLP assumption, H-S DSA can endure those attacks and demonstrates a relatively high security.

From the above analysis and discussion, it can be considered that under the assumption of one-way hash function、factorization Hypotheses(FAH) and Discrete Logarithm Problem(DLP), H-S DSA has better security strength. It can resist a variety of password attacks effectively including linear analysis and differential attacks, with better security.

B. Time-complexity analysis

Because of attacks to ELGamal's signature scheme, ELGamal has to use hash function. Signature scheme Yen and Laib had made many efforts to find ELGamal's signature scheme without using hash function, but they failed finally[20]. Use hash function to make a hash round function transformation, it effectively increased the algorithm's security.

The time-complexity of H-S DSA signature scheme depends on one-way function、hash round function、Discrete Logarithm Problem and Self-certified public key system. The symbols below are used in performance analysis of H-S DSA:

$\lvert x \rvert$	Length of message x
$\lvert h \rvert$	Output length of hash round function h
T_h	Time in calculating one-way function
T_m	Multiplication time with no modulus N
T_{mm}	Multiplication time with modulus N
T_{me}	Exponent time with modulus N

In case of exhaustive searching attack, according to (3) that if chosen number w_i and $\lvert h \rvert$ be restricted to 220bit and 128bit respectively, the value of s_i would be restricted to $\lvert \varphi_{(N)} \rvert$, so the length of any signature would be $\lvert r_i \rvert + \lvert s_i \rvert$, being restricted into $2 \cdot \lvert N \rvert$ bit.

We use equation (2) and equation (3) to calculate the time-complexity of creating a digital signature. It costs $(T_m + T_{mm} + T_{me})$ to create it, where $(3 \cdot T_{me} + T_m + T_{mm})$ is needed to verify the signature.

ELGamal signature scheme is based on the difficulty of solving discrete logarithm, we can see from reference [21], if it is assumed that b is the bits of prime modulus p, then the computational complexity of calculating x from y, or calculating k from r, or calculating t from u is $O(\exp\sqrt{cblnb}$, which $c \in (0,1))$. Algorithm with computational complexity of $O(\exp\sqrt{cblnb})$ is called as sub-index time algorithm. Therefore, as long as the scale of b is expanded appropriately (For example, b achieves 1024 bits), it will become very difficult for forgers to decipher implicit signature scheme or ELGamal signature scheme simply by solving discrete logarithm. The complexity of mode index problem is $O(n)$, so signers can easily calculate y, u and r.

Due to the construction features of signature, the signing speed of H-S DSA is a bit slower compared with those of ELGamal signing schemes, which is the principal weakness of H-S DSA. However, since the calculation of u has nothing to do with message M, u calculation can be done to offset the weakness in speed. By the way, both signing and verifying speed in recently proposed reinforced ELGamal schemes are slower than that of the original schemes [22]. With security enhanced, signing and verifying speed are inevitable to be affected more or less. The time-complexity of H-S DSA can be regarded as relatively simple.

V. Conclusion

In this paper, we analyze and summarize some digital signature algorithms which are relatively mature and frequently used. On this basis, hash round function and self-certified public key system are being studied, upon which a new digital signature algorithm similar to ELGamal(H-S DSA) is designed and realized. And this signature algorithm is based on transformation of hash round function and self-certified public key system.

A performance evaluation for H-S DSA has been made in this article. From the security analysis for the algorithm, we can think that the new designed algorithm

has sufficient security strength, it can effectively resist all kinds of password attacks, including linear analysis and differential attacks. From the time-complexity analysis for the algorithm, we can think that the new designed algorithm has lower time-complexity. The security of H-S DSA is improved compared with that of digital signature algorithm of ELGamal, and the time-complexity of H-S DSA is not more than that of digital signature algorithm of ELGamal, so, it can be considered that H-S DSA is feasible.

With the more extensive application of digital signatures, new issues will emerge from a variety of digital signature algorithms, but people will try hard correspondingly to make new solutions.

REFERENCES

[1]. Q. Wang, Z.F. Cao, "Formal model of proxy multi-signature and a construction", *Chinese Journal of Computers*. vol. 29, no. 9, pp. 1628-1635, 2006.

[2]. M. Michels, P. Horster, "On the risk of disruption in several multiparty signature schemes", *Advances in Cryptology-Aslacrypt'02. Pringer-Verlag. NewYork*, pp.334-338, 2002.

[3]. M. Qi, G. Z. Xiao, "Enhancing the Security of Generilized ElGamal Type Signature Schemes", *Acta Electronica Sinica*, vol.24, no.11, November 2003.

[4]. NIST, "A proposed federal information processing standard for digital signature standard", *Federal Register*. Vol. 56, no. 169, pp. 42980-42982, 1999.

[5]. P. Smith, "LUC public-key encryption", *Dr.Dobb's Journal*. pp. 44-49, 2003.

[6]. P. Smith and L. Skinner, "A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete Logarithms", *Proceedings of Asiacrypt'94*. Springer, pp. 357-364, 2004.

[7]. M. Girault, "Self-certified public keys", Advances in *Cryptology-Eurocrypt'91*,*Springer-Veriag.Berlin*,pp.491-497, 1991.

[8]. K. Nyberg and R.A. Rueppel, "New digital signature scheme based on discrete logarithm (comment)", *Electronic Letters*. vol. 30, no. 5, pp.481, 2004.

[9]. J. He and T. Keisler, "Enhancing the security of EIGamal's signature scheme", *IEEE proc. Comput. Digit.Tech*. vol. 141, no. 4, pp. 249-252, 2004.

[10]. W. F. Ji, X. X. Wu, S. Z. Jin, D. H. Yuan, "New On-Line Secret Sharing Scheme Using Hash Function", *Acta Electronica Sinica*. vol. 31, no. 1, pp. 45-47, 2003.

[11]. X. M. Zhang, J. Seberry, Y. Pieprzyk, "HAVAL-A One-Way Hashing Algorithm with Variable Length of Output", vol. 3, no. 13, November 1993, in press.

[12]. J.F. Dillon, "A Survey of Bent Functions", *NSA Mathematical Meeting*, 2002.

[13]. J. Seberry, X. M. Zhang and Y. Zheng, "Improving the Strict Avalanche criterion Characteristics of Cryptographic Functions ", *Information Processing Leters*. vol.50, 1996.

[14]. J. Seberry, X. M. Zhang and Y. Zheng, "Nonlinearity and Propagation Characteristics of Balanced Boolean Functions", *Information and Computation*, vol.119, no.1, 2003.

[15]. W. Alexi, B. Z. Chor, O. Goldreich, and C. P. Schorr, "RSA and rabin functions: certain parts are as hard as the whole", *SIAM Journal on Computing*, vol.17, no.2, pp.194-209, Apr.1998.

[16]. M. Girault, "Seif-certified public keys", *Advances in Cryptology-Eurocrypt'91*, *Springer-Verlag. Berlin*, pp. 491-497, 2001.

[17]. T. Beth, "A Fiat-Shamir-like authentication protocol for the ELGamal scheme", *Advances in Cryptology-Eurocrypt '88, Springer-Veriag. Berlin*, pp.77-86. 2001.

[18]. C. P. Schnorr, "Eficientid entification and signatures for smart cards", *Advances in Cryptology-Crypto'89, Springer- Verlag. Berlin*, pp.239-252, 1989.

[19]. X. F. Yuan, R. Y. Sun, J. Q. Sun, Y. H. Yang, "Signature scheme with message recovery based on discrete logarithm s and factoring", *Computer Applications*. vol. 27, no. 10, pp. 2460-2463, 2007.

[20]. S. M. Yen, and C. S. Laih, "New digital signature scheme based on discrete logarithm", *Electronic Letters*. vol. 29, no. 12, 1993.

[21]. S. Pohlig and M. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", *IEEE Transaction on Information Theory*. IT-24(1998).

[22]. L. Harn, and Y. Xu, "Design of generalized EIGamal type digitai signature scheme based on discrete logarithm", *Electronic Letters*. vol. 31, no. 6, 2005.

Haipeng A. Chen, male, was born in Cao County, Shandong, June, 1978. He received bachelor degree in 2003 and master degree in 2006 both from Jilin University.

Now he is a lecturer and a Ph.D candidate in the college of computer science and technology, Jilin University. His research interests are computer network security, digital image processing and pattern recognition.

Xuanjing B. Shen, male, was born in Helong County, Jilin Province, December, 1958. He received bachelor degree in 1982, master degree in 1984, and PhD degree in 1990 all from Harbin Institute of Technology respectively.

He is a professor and PhD supervisor currently in the college of computer science and technology, Jilin University. His research interests are multimedia technology, computer image processing, intelligent measurement system, optical- electronic hybrid system, and etc.

Yingda C. Lv, female, was born in Wenan County, Hebei Province, January, 1983. She received bachelor degree in 2007 from Jilin University.

Now she is a Master candidate in the college of computer science and technology, Jilin University. Her research interests are digital image processing and pattern recognition.