Proposing a method to design secure digital signature scheme on the ring structure Z_n

Nguyen Dao Truong, Le Van Tuan

***Abstract*—In this paper, we propose a method to design signature scheme on ring structure with residual classes modulo composite. At the same time, we develop several new digital signature schemes that are more secure, with faster signature generation than ElGamal digital signature scheme and its variants. Furthermore, our proposed signature scheme has overcome some weaknesses of some published signature scheme of the same type, which are built on ring structure.**

***Tóm tắt*—Trong bài báo này, chúng tôi đề xuất một phương pháp thiết kế lược đồ chữ ký trên cấu trúc vành các lớp thặng dư theo modulo hợp số, đồng thời phát triển một số lược đồ chữ ký số mới an toàn hơn, tốc độ sinh chữ ký nhanh hơn so với lược đồ chữ ký số ElGamal cùng với những biến thể của nó. Hơn nữa, lược đồ chữ ký do chúng tôi đề xuất cũng khắc phục được một số nhược điểm của một số lược đồ chữ ký cùng loại, được xây dựng trên cấu trúc vành.**

Keywords—Digital signature scheme; Hash; Discrete logarithm problem on rings.

Từ khóa—Lược đồ chữ ký số; hàm băm; bài toán logarit rời rạc trên vành.

I. INTRODUCTION

The ElGamal digital signature scheme was first published in 1985, mentioned in [1-4]. After that, a series of research to improve and develop variant schemes of the ElGamal scheme, typically the GOST R34.10-94[5]; Schnorr[6-7]; DSA[8-10]. In general, the order of the multiplicative group $\mathbb{Z}_p$ of the ElGamal scheme and its variants can't be kept secret. That leeds to be insecure when the session key is revealed or be coincided [3], [11]. Furthermore, these signature schemes publicize the order of multiplicative group $\mathbb{Z}_p$, that led to be insecurity when adversaries attacked them with the Pollard's Rho algorithm, the Pohlig Hellman algorithm, and the Index algorithm calculate[12]. On the other hand, the computational cost of generating signatures of the ElGamal schema and its variants on the field structure $\mathbb{Z}_p$ is usually higher than the computational cost of schemes of the same type on the ring structure $\mathbb{Z}_n$. Because on the ring $\mathbb{Z}_n$, if you know how to factorize $n = p.q$, by using the Chinese remainder theorem to calculate the exponentiation, the inverse modulo, the computational cost of these operations will be much reduction[12] (more details in Lemma 3). In addition to the signature schemes in the field structure $\mathbb{Z}_p$, there are a number of signature schemes on the ring structure $\mathbb{Z}_n$ published in [11] and [13-19]. However, these schemes have overcome the insecurity in situations where the session key is revealed or be coincided. But the signature generation speed of these schemes is not faster than that of signature schemes of the same type in the field structure $\mathbb{Z}_p$. So, we have developed a new digital signature scheme that overcomes the disadvantages pointed out in previously published schemes of the same type.

II. DEFINITIONS AND SYMBOLS

This section will review some notations, terminologies, definitions and lemmas relating to the following parts of the article.

Definition 1. Convention on some symbols:

Element k is randomly chosen from the set X: $k \in_R X$

The concatenation operation the string x with the string y: $x \,||\, y$

This manuscript is received on October 28, 2021. It is commented on November 04, 2021 and accepted on November 12, 2021 by the first reviewer. It is commented on November 04, 2021 and accepted on November 21, 2021 by the second reviewer.

The bit length of the binary representation of a: L_a.

The order of g in ring $\mathbb{Z}_n$: $\mathrm{Ord}_n(g)$

The greatest common divisor of two integers a and b: $GCD(a, b)$

Discrete Logarithm Problem on ring $\mathbb{Z}_n$, denoted DLP_n If $n = p$ (p is prime) it is denoted DLP_p, it is called the discrete logarithmic problem in the field $\mathbb{Z}_p$.

Hash: $\{0,1\}^{\infty} \to \{0,1\}^H$ where $H \in \mathbb{N}$.

Let $s \in \{0,1\}^H$, $s = s_0 \dots s_{H-2} s_{H-1}$ is a binary string. So $\bar{s} \in \mathbb{N}$, that is computed by the following formula:

$$\bar{s} = s_0 2^{H-1} + \cdots + s_{H-2} 2 + s_{H-1}. \quad (1)$$

Let $n \in \mathbb{N}$, $n = n_0 2^{k-1} + \cdots + n_{k-2} 2 + n_{k-1}$ where $n_j \in \{0,1\}$ ($j = 0..(k-1)$) and $n_0 \neq 0$. Let $H \in \mathbb{N}$ denote n[H] is the string $H-$bit which calculated by:

$$n[H] = n_{k-H} n_{k-H-1} \dots n_{k-1} \text{ if } H \leq k \quad (2)$$

$$n[H] = \underbrace{0 \dots 0}_{H-k} n_0 n_1 \dots n_{k-1} \text{ if } H > k \quad (3)$$

Definition 2. (Public key digital signature scheme) The public key digital signature scheme includes the following sets (M, S, k_s, k_b, K) where:

M is a finite set of messages.

S is a finite set of signatures.

Ks is a finite set of secret keys.

Kb is a finite set of public keys.

K is a finite set of session keys.

$KeyGen$: $k_s \to k_b$ is the key generation function.

Sig: $M \times k_s \times K \to S$ is the signature generation function.

Ver: $M \times S \times k_b \to$ {"accept", "reject"} is the signature verification function.

Each member of the system randomly selects a secret key $k_s \in Ks$, calculate the public key using the following formula:

$$k_b = \mathrm{KeyGen}(k_s) \quad (4)$$

The member who owns the secret key k_s signs the message m, $m \in M$ by the following formula:

$$s = Sig(m, k_s, k \in_R K). \quad (5)$$

The signature s of the message m is sent together to the receiver. The receiver who owns the public key k_b will verify the signature via the function $Ver(m, s, k_b)$, if the return value of the function $Ver(m, s, k_b)$ is "accept", the signature s of the message m is called valid, otherwise the function $Ver(m, s, k_b)$ returns "reject", the signature s is invalid for the message m.

Lemma 1. Let $n = u.v$ where $\mathrm{GCD}(u, v) = 1$ and $A = v.(v^{-1} \bmod u)$. Then for all $x \in \mathbb{Z}_n$, denoted $x_u = x \bmod u$ and $x_v = x \bmod v$ then we have:

$$x = (A.(x_u - x_v) + x_v) \bmod n \quad (6)$$

III. ELGAMAL SIGNATURE SCHEME AND ITS WEAKNESSES

A. ElGamal Scheme

ElGamal digital signature scheme is designed on field structure $\mathbb{Z}_P$, P is a prime number, a generator g in the field $\mathbb{Z}_P$, with order of t, $t = \mathrm{Ord}_p g = p - 1$. Sets (M, S, Ks, Kb, K) in the scheme as:

$M = \mathbb{Z}_p$, $S = \mathbb{Z}_P^* \times \mathbb{Z}_{P-1}^*$, $Ks = K = \mathbb{Z}_{P-1}^*$, $Kb = \mathbb{Z}_P^*$.

The value x, $x \in \mathbb{Z}_{P-1}^*$ is the secret key, $y = \mathrm{KeyGen}(x) = g^x \bmod P$ is the public key.

Algorithm 1. Generate signature in ElGamal

Input: (x, p), $m \in \mathbb{Z}_p$.

Output: $(r, s) \in S$ is the signature

1. $k \in_R K = [2, P-2]$.
2. $\mathrm{GCD}(k, p-1) \neq 1$ goto 1.
3. $r = g^k \bmod P$.
4. if $(r = 0)$ then goto 1.
5. $z = (m - r.x) \bmod P - 1$
7. $s = k^{-1}.z \bmod P - 1$
8. if (s = 0) then goto 1.
9. return (r, s)

Algorithm 2. Verify the signature in ElGamal

Input: (p, y); the signature (r, s) of the message m $\in \mathbb{Z}_p$.

Output: "accept" or "reject".

1. If (r = 0) or (s = 0) then return "reject"
2. $u = r^s \bmod P$
3. $v = y^r \bmod P$
4. If $r^s . y^r = g^m (\bmod P)$ then return "accept" else return "reject"

B. Some weaknesses

The computational cost of the ElGamal scheme and its variants is concentrated on the exponentiation and inverse calculations. These calculations performed in the field structure $\mathbb{Z}_p$ have a higher computational cost than doing them on the ring structure $\mathbb{Z}_n$. The reason is: on ring structure, if you know how to factorize *n=p.q*, by using the Chinese remainder theorem to calculate the exponentiation, the inverse modulo, the computational cost of these operations will be much reduction (see Lemma 3).

The ElGamal schema and its variants all have a level of security based on the difficulty of the DLP_p problem, so they are required to publicly the order of the generator element, leading to insecure in some of the following cases:

Case 1, session key is exposed. If the session key *k* is exposed during a signing on a certain message m, the following formula is used:

$$s = k^{-1}.(m - r.x) \bmod P - 1 \quad (7)$$

From formula (7), the secret key *x* can be calculated using the following formula (8):

$$x = (m - s.k).r^{-1} \bmod P - 1 \quad (8)$$

Case 2, session key is duplicated. If the session key *k* is duplicate, it means that the two messages share the same session key, then the attacker can use the following formulas to find the secret key calculation formula:

$$s = k^{-1}.(m - r.x) \bmod P - 1 \quad (9)$$

$$k = s^{-1}.(m - r.x) \bmod P - 1 \quad (10)$$

$$s' = \left(k^{-1}.(m' - r.x)\right) \bmod P - 1 \quad (11)$$

$$k = s'^{-1}.(m' - r.x) \bmod P - 1 \quad (12)$$

From formulas (10) and (12) deduce equality (13):

$$s^{-1}.(m - r.x) = s'^{-1}.(m' - r.x) \bmod P - 1 \quad (13)$$

From equality (13), it is easy to calculate the secret key x using formula (14):

$$x = (s'^{-1}.m' - s^{-1}.m).(s'^{-1}.r - s^{-1}.r)^{-1} \bmod P - 1 \quad (14)$$

Case 3, it faces the risk of being attacked by some algorithms that solve discrete logarithmic problems based on the order of the generator, typically Pohlig Hellman's Algorithm, Pollard's Rho algorithm and Index calculate algorithm, which has been demonstrated in [12].

IV. Proposing new digital signature scheme on the ring $\mathbb{Z}_N$

A. Basic signature scheme - BSS

The content of this section presents a method to design a secure digital signature scheme based on the discrete logarithm problem on ring of the residual classes modulo composite.

1. Definition of basis sets

The sets (M, S, k_s, k_b, K) is defined as follows:

The finite set of messages $M = \{0,1\}^\infty$.

The finite set of signatures $S = \mathbb{Z}_{2^H}^* \times \mathbb{Z}_t^*$.

The finite set of secret keys k_s, $k_s = \mathbb{Z}_t^*$.

The finite set of public keys k_b, $k_b = \mathbb{Z}_n^*$.

The finite set of session keys K, $K = \mathbb{Z}_t^*$.

Euler's function φ(n).

2. General parameters

Numerical size modulo *n* and the order of the generator g, are denoted respectively L_n, L_t, Hash function: $\{0,1\}^\infty \to \{0,1\}^H$, $L_t = H + 2$.

3. General Formula

Calculate keys: x is the secret key, $y = KeyGen(x) = g^x \bmod n$, y is the public key.

$r = (g^k \bmod n) \bmod 2^H$ với $k \in_R [1, t-1]$

$s = k.(f_1(m,r).x + f_2(m,r))^{-1} \ mod \ t.$

Algorithm 3. Generate parameters in BSS

Input: $L_p = \ L_q = \frac{L_n}{2}$;

Output: $\left((p,q), c_n, g, (g_p, g_q), t, (p_1, q_1), c_t, x, y\right)$

1. Generate two prime numbers p, q, corresponding size is L_p, L_q −bit. Calculate n, $n = p.q$ and $c_n = q.(q^{-1} mod \ p)$.

2. Generate two prime numbers p_1, q_1 satisfy:

$p_1 \neq q_1, p_1|(p-1), q_1|(q-1)$

$q_1 \nmid (p-1)$ *và* $p_1 \nmid (q-1)$, $L_{p_1} = L_{q_1} = \frac{H}{2} + 1$

3. Calculate $t = p_1.q_1$, $c_t = q_1.({q_1}^{-1} mod \ p_1)$,

4. The generator g in the group $\mathbb{Z}_t^*$, which has the order of *n*, denoted by $\mathrm{Ord}_n(g) = \ t$, it is calculated by the following formula:

5. $g = \alpha^{\frac{\varphi(n)}{p1.q1}}$ mod n với $\forall \alpha, (\alpha, n) = 1$

6. Calculate $g_p = g \ mod \ p$ and $g_q = g \ mod \ q$.

7. Generate the secret key *x*, it is randomly selected in $\mathbb{Z}_t^*$ and calculates the public key y based on the secret key x, $y = KeyGen(x) = g^x \ mod \ n$.

The tupble of sets $\left((p,q), c_n, (g_p, g_q), t, (p_1, q_1), c_t, x\right)$ for the sender to sign and tuple of parameters (n, g, y) for the reciever to verify.

Algorithm 4. Generate the signature in BSS

Input: m, $\left((p,q), c_n, (g_p, g_q), t, (p_1, q_1), c_t, x\right)$

Output: (r, s) is the signature of the message m.

1. $k \in_R [1, t-1]$; $k_p = k \ mod \ p_1$;

2. $k_q = k \ mod \ q_1$;

3. if $(\mathrm{k_p} = 0)$ or $(\mathrm{k_q} = 0)$ then goto 1;

4. $\mathrm{r_p} = \mathrm{g_p^{k_p}} \mathrm{mod\ p}$; $r_q = g_q^{k_q} mod \ q$;

5. $r = \left(\left(c_n(r_p - r_q) + r_q\right) mod \ n\right) \ mod \ 2^H$;

6. $f_1 = f_1(m,r)$;

7. if (f_1 = 0) then goto 1.

8. $f_2 = f_2(m,r)$; $w = (f_1.x + f_2)$;

9. $w_p = w \ mod \ p_1$; $w_q = w \ mod \ q_1$;

10. if $(w_p = 0)$ *or* $(w_q = 0)$ then goto 1;

11. $z_p = {w_p}^{-1} \ mod \ p_1$; $z_q = {w_q}^{-1} \ mod \ q_1$;

12. $z = \left(c_t.(z_p - z_q) + z_q\right) mod \ t$;

13. $s = k.z \ mod \ t$;

14. return (r, s);

Algorithm 5. Verify the signature in BSS

Input: m, with the signature (r, s), (n, g, y).

Output: Return "accept" or "reject".

1. if $(s \ = \ 0)$ theo return “reject”;

2. $f_1 = f_1(m,r)$;

3. if (f_1 = 0) theo return “reject”;

4. $a = y^{f_1} \ \mathrm{mod}\ n$; $f_2 = f_2(m,r)$;

5. $b = g^{f_2} \ \mathrm{mod}\ n$; $c = a.b \ \mathrm{mod}\ n$;

6. $w = (c^s \ (mod \ n)) \ \mathrm{mod}\ 2^H$;

7. if $(w \ = \ r)$ then return “accept” else “reject”;

4. Correctness

Whether Algorithm 4 can be terminated depends on three iterations, at steps 3, 7 and 10. Applying Lemma 4, the probability that these iterations occur once is approximately 1. The other steps of the algorithm will obviously be executed and algorithm 4 will terminate, resulting in the signature (r, s). The input to algorithm 5 is the signature (r, s) of the message m generated by the algorithm 4 and the public key y. Obviously, if (r, s) is a valid signature then $s \neq 0$ and the value w calculated in algorithm 5 will be equal to the value r calculated in algorithm 4, namely:

$$s = k.w^{-1} \ mod \ t \quad (15)$$

$$= \mathrm{k.}\left(\mathrm{f_1(m,r).x + f_2(m,r)}\right)^{-1} \mathrm{mod\ t} \quad (16)$$

We have:

$$k = s.(f_1(m,r).x + f_2(m,r)) \ mod \ t \quad (17)$$

Thus

$$g^k mod \ n = g^{s.f_1(m,r).x}.g^{s.f_2(m,r)} \ mod \ n \quad (18)$$

We have:

$$r = (g^k mod \ n) \ mod \ 2^H =$$

$= g^{s.f_1(m,r).x}.g^{s.f_2(m,r)} \ mod\ n)\ mod\ 2^H$ (19)

The value w calculated in algorithm 5 as follows:

$$w = \left(\left(y^{f_1(m,r)}.g^{f_2(m,r)}\right)^s \ mod\ n\right) mod\ 2^H \quad (20)$$

From (18) and (20) we have $w = r$, that is, algorithm 5 returns "accept", which proves that the signature scheme BSS is correct.

B. Proposing Signature scheme SS01

By replacing the general functions $f_1(m,r)$ and $f_2(m,r)$ in the BSS signature scheme with specific functions, we have the following SS01 signature scheme:

- With the general function $f_1(m,r) = 1$

- $f_2(m,r) = \overline{Hash(m||r[H])}$.

The signature scheme SS01, including signing algorithm, signature verification algorithm. Particularly, the parameter generation algorithm in SS01 is inherited from BSS.

Algorithm 6. Generate signature in SS01

Input: $m, \left((p,q), c_n, (g_p, g_q), t, (p_1, q_1), c_t, x\right)$

Output: (r,s) is the signature of the message m.

1. k $\in_R [1, t-1]$; $k_p = k\ mod\ p_1$;

2. $k_q = k\ mod\ q_1$;

3. *if* $(k_p = 0)$ *or* $(k_q = 0)$ *then goto 1;*

4. $r_p = g_p^{k_p} mod\ p$; $r_q = g_q^{k_q} mod\ q$;

5. $r = \left(\left(c_n(r_p - r_q) + r_q\right)\ mod\ n\right)\ mod\ 2^H$;

6. $f_2 = \overline{Hash(m||r[H])}$;

7. $w = (x + f_2)$;

8. $w_p = w\ mod\ p_1$;

9. $w_q = w\ mod\ q_1$;

10. *if* $(w_p = 0)$ *or* $(w_q = 0)$ *then goto 1;*

11. $z_p = {w_p}^{-1}\ mod\ p_1$; $z_q = {w_q}^{-1}\ mod\ q_1$;

12. $z = \left(c_t(z_p - z_q) + z_q\right)\ mod\ t$;

13. $s = k.z\ mod\ t$;

14. return *(r, s)*

Algorithm 7. Verify the signature in SS01

Input: Parameters: the key (n, g, y); the signature (r, s) of message m

Output: Return "accept" or "reject"

1. if *(s = 0)* then return "reject";

2. $f_2 = \overline{Hash(m||r[H])}$;

3. $w = (y.g^{f_2})^s\ mod\ n)\ 2^H$;

4. if *(w = r)* then return "accept" Else "reject";

Correctness: Because the BSS scheme is correct, the SS01 scheme is obviously correct. ■

V. Analysis of proposing digital signature

To analyze the proposed signature scheme, suppose the number modulo n of the proposed scheme, $n = p.q$ where p and q are primes of equal size. ($L_p = L_q = \frac{L_n}{2} = L$). The number modulo P in the ElGamal signature scheme has $L_P = L_n = 2L$. Suppose $H \geq 512$ −bit, $t = p_1.q_1$ and $L_t \geq 514$ −bit. Some cost estimates of the algorithms of ElGamal schema and RSA schema in [12] are included to compare with the proposed scheme.

A. Computational cost

Suppose the notation M_l is the number of bit operations that perform the multiplication of two positive integers modulo of size $l-$ bit. The function that performs the calculation $a^e\ mod\ n$ is denoted by $Pow(a,e,n)$.

Lemma 2: Suppose a, e, n are positive integers of size L_a, L_e, L_n −bit. The cost of calculating the function $Pow(a,e,n) = a^e\ mod\ n$ based on the squaring and multiplying algorithm, denoted $C_{Pow_{(L_a,L_e,L_n)}}$ is as follows:

$$C_{Pow_{(L_a,L_e,L_n)}} \approx 1.5.L_e.M_{L_n} \quad (21)$$

Proof: Using the squaring and multiplying algorithm, to calculate the function $Pow(a,e,n)$, it is necessary to perform L_e squared and $W(e)$ reduced multiplication modulo L_n −bit. We have $C_{Pow_{(L_a,L_e,L_n)}} \approx L_e + W(e)$ calculation M_{L_n}. Given that an integer e has size L_e − bit the mean

of the number of 1 bits in the binary representation of *e*, denoted $W(e) = \frac{L_e}{2}$, thus:

$$C_{Pow_{(L_a,L_e,L_n)}} \approx (L_e + \frac{L_e}{2}).M_{L_n} = 1.5.L_e.M_{L_n} \blacksquare$$

Lemma 3. Let $n = p.q$ where p, q are two primes of the same size, and *e* has the same size as φ(n). Then if *p* and *q* are known, then:

(i) The cost of calculating $a^e \, mod \, n$ is only equal to $\frac{1}{4}$ of the cost of calculating the above value if p, q are not known.

(ii) The cost of calculating $a^{-1} \, mod \, n$ when factoring $n = p.q$ is known is only equal to $\frac{1}{2}$ the cost of calculating the above value if p, q are not known.

Proof:

(i). According to Lemma 3, to solve $x = a^e \, mod \, p$ and $x = a^e \, mod \, q$ where $L_p = L_q = \frac{L_n}{2} = L$ has cost of calculating as$1{,}5.2.L.M_{2L} = 12.L^3$ bit operations. We are going to prove that $n = p.q$ is known, the calculation $a^e \bmod n$ has a cost of calculating equal to $\frac{1}{4}$ the cost of calculating $a^e \bmod n$ in the usual way. Indeed, to compute $a^e \bmod n$[12], we have to perform$a^{e_p} \bmod p$ and $a^{e_q} \bmod q$ (where $e_p = e \bmod (p-1)$ and $e_q = e \bmod (q-1)$ has a computational cost of $2.(\frac{3}{2}.L^3)$ bit operations. We have $\frac{2.1{,}5.L^3}{12.L^3} = \frac{1}{4}$, which means that if we apply the Chinese remainder theorem to calculate exponentiation modulo, the computational cost will be reduced by 4 times.

(ii) By the same argument, it is easy to prove that the cost of calculating of $a^{-1} \, mod \, n$ when we know that prime factorization $n = p.q$ focuses mainly on the two operations $x_p = a^{-1} \bmod p$ and $x_q = a^{-1} \, mod \, q$ has a computational cost of $2l^2$ bit operations. Also, the computational cost of the calculation $a^{-1} \bmod n$ is $(2L)^2 = 4L^2$. So we have $\frac{2L^2}{4L^2} = \frac{1}{2}$, that is, if we apply the Chinese remainder theorem to calculate exponentiation modulo, the computational cost will be reduced by 2 times.■

Lemma 4. Suppose the ring $\mathbb{Z}_n$ has a generator g where $t = Ord_n g$. Let t_p, t_q are distinct primes, each is about the same size and satisfies $t = t_p.t_q > 2^H$. The probability of the loop in the proposed scheme is executed only one time be approximately 1 when $H \geq 512$.

Proof:

Obviously, the event that each loop in the signature generation algorithm of the proposed digital signature schemes is approximately 1 is equivalent to the following events: $(k_p \neq 0)$ or $(k_q \neq 0)$ or $(w_p \neq 0)$ or $(w_q \neq 0)$ or $(f_1 \neq 0)$ occurs from the first loop.

It is easy to see that the event $(k_p = 0)$ occurs only when $k = p_1$ with probability $\frac{1}{t}$. Similarly, the event $(k_q = 0)$ occurs only when $k = q_1$ with probability $\frac{1}{t}$. The argument is similar with events $(w_p = 0)$ and $(w_q = 0)$ also occurring with probability $\frac{1}{t}$. Since $L_t = H + 2$, $\frac{1}{t} = \frac{1}{2^{H+2}} \approx 0$ when $H \geq 512$. Therefore, the probability of events $(k_p \neq 0)$, $(k_q \neq 0)$, $(w_p \neq 0)$ or $(w_q \neq 0)$ is approximately 1. Because the chosen Hash is a secure hash $(f_1 = \overline{Hash(m||r[H])})$, the event $(f_1 \neq 0)$ occurs with probability approximately 1.■

1. Computational costs

We only focus on analyzing the computational cost of the proposed scheme in the signature generation and verification algorithm. Obviously, the computational cost in these algorithms focuses mainly on exponentiation, inverse modulo and number of iterations in the algorithm. Lemma 4 has shown that the probability that the loop event in the signature generation algorithm occurs once is approximately 1, so we consider the algorithms to have no iterations. Based on Lemma 2 and 4, the results of calculation cost estimation for the proposed scheme are as follows: Algorithm 6: Assuming addition, subtraction and modulo reduction operations are ignored, the estimated cost of algorithm 6 is as follows: 2 exponents

modulo p, so the total costs is $2(1.5.\frac{L_t}{2}M_L)$; a multiplication modulo n costs of M_{2L}; two inversions modulo of size $\frac{L_t}{2}$, có costs of $2M_{\frac{L_t}{2}}$; two multiplications modulo t cost of $2M_t$. Therefore, the computational cost (number of bit operations) is: $3L_t.M_L + M_{2L} + 2M_{\frac{L_t}{2}} + 2M_t$.

Algorithm 7: Assuming that ignoring modulo reductions, the conditional test of s, algorithm 7 has two powers (Suppose if the output of the Hash function has $H = 512$, then $L_t = 514$ the size of the exponent is $L_s \approx 512$ and $L_{f_2s} \approx 1000$) and one multiplication modulo n. Thus, the estimated cost of the calculation is: $1.5.L_sM_{2L} + 1.5.L_{f_2s}.M_{2L} + M_{2L}$.

TABLE I. COMPUTATIONAL COSTS

Scheme	Signature generation	Signature verification	Note
SS01	$\approx 3L_t.M_L + M_{2L} + 2M_{\frac{L_t}{2}} + 2M_t$	$\approx 1.5.L_sM_{2L} + 1.5.L_{f_2s}.M_{2L} + M_{2L}$	M_i is bit operations

2. Storage space

Space to store the signatures of proposed schemes (r, s) equal $2L_t$, which is much smaller than the storage space cost of signatures in ElGamal and RSA schemes. However, considering the entire proposed digital signature scheme system for authentication, the cost of storage space will depend on the number of members in the system. Whenever two members exchange information and use SS01 signature schemes for authentication, requires the two members to use a separate parameter set, consisting of 11 elements whose total number of bits to store is $7L + 5L_t$ −bit. Suppose κ is the number of members in the system using the signature, then C_K^2 transactions are different, so the parameter storage space cost of the proposed digital signature schemes will be $C_K^2(7L + 5L_t)$ −bit. Therefore, the proposed digital signature scheme has a storage space cost many times greater than the cost of storage space compared to ElGamal signature scheme or its variant schemes in the field structure $\mathbb{Z}_p$. The results of the estimation of the computational cost and storage space of the proposed signature scheme compared with the ElGamal and the RSA signature schemes are shown in the following table:

TABLE II. COMPUTATIONAL COST AND STORAGE SPACE

Scheme	Signature generation	Signature verification	Storage space	Comp. cost
RSA	$\approx 3L.M_{2L}$	$\approx 128.M_{2L}$	$\approx 2L$	$\approx 6.L$
ElGamal	$\approx 3L.M_{2L} + 3.M_{2L}$	$\approx 9.L.M_{2L} + 2.M_{2L}$	$\approx 4.L$	$\approx 6.L$
SS01	$\approx 3L_t.M_L + M_{2L} + 2M_{\frac{L_t}{2}} + 2M_t$	$\approx 1.5.L_sM_{2L} + 1.5.L_{f_2s}.M_{2L} + M_{2L}$	$\approx 2.L_t$	$\approx C_K^2.(7L + 5L_t)$

B. Security

Proposition 1. The proposed signature scheme SS01 is secure in situations of coinciding or revealing of session key.

Proof:

The session key is generated in each transaction, so the possibility of it being revealed still occurs with a certain probability (side channel attack). Moreover, according to the "birthday paradox", the possibility of the same session key is quite high. However, our proposed schemes secure the order of the generator (the order of the generator, denoted *t*). When the session key is coincided or revealed, it is difficult for the attacker to calculate the secret key. In the following, we proceed to consider two cases of coinciding or revealing of session key that happens in the scheme SS01 but it is still secure.

(i) Assuming the session key k is revealed, because the parameter t is unknown, the attacker cannot determine z from the formula $s = k.z \bmod t$. Since he cannot determine z and t, $w = (f_1.x + f_2) \bmod t$ cannot be determined and therefore *x* cannot be determined.

(ii) The situation where the session key k is coincided, when signing the message m to get the signature is (r, s) and signing the message m' to

get the signature (r', s'). Based on the signature generation algorithm, the messages m, m' have the following formula for calculating s, s' respectively: $s = k.z \bmod t$ and $s' = k.z' \bmod t$. From these two formulas, we have: $s.z = s'^{-1}.z' \bmod t$. Since t is kept secret, it is difficult for an attacker to determine the z and w components, so it is difficult to determine the secret key x from the formula: $w = (f_1.x + f_2) \bmod t$. ■

Proposition 2. The proposed signature scheme SS01 is secure when the attacker uses algorithms to solve the discrete logarithmic problem based on the order of the generator, typically: Pohlig Hellman algorithm, Index calculate algorithm, rho Pollard algorithm to calculate the discrete logarithm [12].

Proof:

This is obvious, since the Pohlig Hellman, Index calculate and rho Pollard algorithms for calculating discrete logarithms, which require the input to have the order of the multiplication group (or the order of the generator g), denoted Ord_g. In addition, the public components of the proposed signature scheme hide the order of the multiplication group, so they do not work.■

CONCLUSION

In this paper, before proposing a new digital signature scheme, we have demonstrated that the ElGamal scheme is not secure in situations where the session key is coincided or revealed. Moreover, we have analyzed, evaluated and pointed out some weaknesses of the published signature scheme on ring structure $\mathbb{Z}_n$, which are: The Chinese remainder theorem has not been applied to calculate exponents and inverses in the signature generation algorithm, so the computational cost in the signature generation algorithms is still high. To overcome the weaknesses of the signature schemes that we have considered, we have proposed a method of designing a signature scheme based on that method. We have developed a new digital signature scheme that is secure in situations where the session key is coincided or revealed. Moreover, the signature generation speed of the proposed scheme is much faster than that of the published signature schemes of the same type on the ring $\mathbb{Z}_n$. However, we acknowledge that the proposed signature scheme uses much larger storage space than the published signature schemes of the same type on the field structure $\mathbb{Z}_p$.

REFERENCES

[1] ElGamal, Taher. "A public key cryptosystem and a signature scheme based on discrete logarithms." IEEE transactions on information theory 31.4 (1985): 469-472.

[2] Li, Xiaofei, Xuanjing Shen, and Haipeng Chen. "ElGamal digital signature algorithm of adding a random number." Journal of Networks 6.5 (2011): 774.

[3] Liu, Jing-mei, Xiang-guo Cheng, and Xin-mei Wang. "Methods to forge ElGamal signatures and determine secret key." 20th International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06). Vol. 1. IEEE, 2006.

[4] Zhiming, C. H. E. N. "An improved encryption algorithm on ElGamal algorithm." Computer Applications and Software 22.2 (2005): 82-85.

[5] GOST, R. "R 34.11–94, Gosudarstvennyi Standard of Russian Federation, " Information technology. Cryptographic Data Security. Hashing function." Government Committee of the Russia for Standards (1994).

[6] Schnorr, Claus-Peter. "Efficient signature generation by smart cards." Journal of cryptology 4.3 (1991): 161-174.

[7] Ng, Tiong-Sik, Syh-Yuan Tan, and Ji-Jian Chin. "A variant of Schnorr signature scheme with tight security reduction." 2017 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2017.

[8] Morita, Hiraku, et al. "On the security of the schnorr signature scheme and DSA against related-key attacks." ICISC 2015. Springer, Cham, 2015.

[9] Lu, Chia-Yu, Wu-Chuan Yang, and Chi-Sung Laih. "Efficient Modular Exponentiation Resistant to Simple Power Analysis in DSA-Like Systems." 2010 International Conference

on Broadband, Wireless Computing, Communication and Applications. IEEE, 2010.

[10] Ping, Zhou, Kou Yingzhan, and Ji Keke. "Instruction-Cache Attack on DSA Adopting Square-Multiply Method." 2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control. IEEE, 2012.

[11] Lê Văn Tuấn, Phát triển và xây dựng tham số cho lược đồ chữ ký số trên bài toán logarit rời rạc theo modul hợp số, Học viện Kỹ thuật Quân sự, luận án tiến sĩ, Hà Nội, 2019.

[12] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, Handbook Applied Cryptography, Webster Professor of Electrical Engineering and Computer Science Massachusetts Institute of Technology June (1996).

[13] Hồ Ngọc Duy, Vũ Long Vân, Nguyễn Kim Tuấn, Nguyễn Thị Thu Thủy, "Giải pháp nâng cao độ an toàn cho lược đồ chữ kí số", SOIS Thành phố HCM, No 2, pp 13-16, 2017.

[14] Berezin, A. N., N. A. Moldovyan, and V. A. Shcherbacov. "Cryptoschemes based on difficulty of simultaneous solving two different difficult problems." Computer Science 21.2 (2013): 62.

[15] Binh V., Minh H. Nguyen, and Nikolay A. Moldovyan. "Digital Signature Schemes from Two Hard Problems." Multimedia and Ubiquitous Engineering. Springer, Dordrecht, 2013. 817-825.

[16] Lưu Hồng Dũng, Hoàng Thị Mai, Nguyễn Hữu Mộng "Một dạng lược đồ chữ ký số trên bài toán phân tích số", Hội nghị FAIR, pp 377-386, 2015.

[17] Meshram, Chandrashekhar. "Discrete Logarithm and Integer Factorization using ID-based Encryption." Bulletin of Electrical Engineering and Informatics 4.2 (2015): 160-168.

[18] Tan, Chik How, Xun Yi, and Chee Kheong Siew. "Signature scheme based on composite discrete logarithm." Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint. Vol. 3. IEEE, 2003.

[19] Tripathi, Shailendra Kumar, and Bhupendra Gupta. "An efficient digital signature scheme by using integer factorization and discrete logarithm problem." 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2017.

ABOUT THE AUTHORS

Nguyen Dao Truong

Workplace: Academy of Cryptography Techniques

Email:truongnd-it@actvn.edu.vn

Education: He received his undergraduate degree from Academy of Cryptography Techniques in 2001; Master's degree in Information Technology from Military technical Academy in 2010; PhD in Mathematics from Academy of Military Science and Technology in 2019.

Recent research direction: Information Security and Network Security.

Le Van Tuan

Workplace: Military Science Academy

Email:levantuan71@yahoo.com

Education: He received the BSc degree in mathematics from Thai Nguyen University in 1992; MSc degree from Military technical Academy in 2007; Doctorate degree in mathematics from Military technical Academy in 2020.

Recent research direction: public key cryptography, digital signatures, identification and authentication.