Developing a digital signature scheme avoids attacks based on the order of the generator element

Doan Thi Bich Ngoc[1], Nguyen Dao Truong[2*]

[1]University of Information and Communication Technology, Thai Nguyen University, Z115 Road, Thai Nguyen City, Thai Nguyen, Vietnam;
[2]Academy of Cryptography Technique, 141 Chien Thang, Thanh Tri, Hanoi, Vietnam.
[*]Corresponding author: truongnd-it@actvn.edu.vn
Received 23 Oct. 2024; Revised 23 Dec. 2024; Accepted 5 Feb. 2025; Published 25 Feb. 2025.
DOI: https://doi.org/10.54939/1859-1043.j.mst.101.2025.131-139

ABSTRACT

In this paper, we propose a new digital signature scheme based on composite discrete logarithms, which is a variant of the DSA signature scheme. Our new proposed scheme is more secure than the DSA signature scheme. Furthermore, the speed of computing of our scheme is faster than to some similar schemes. For this season, it can be applied in practice.

Keywords: Digital signature scheme; Discrete logarithm problem; Hash function.

1. INTRODUCTION

The advent of the Internet and the rapid development of science and technology has caused information insecurity for users, even businesses and state agencies. In that context, some digital signature schemes are proposed by scientists over the world [2-5, 7-9] such as ElGamal Signature Scheme [19]; the DSA digital signature scheme of US from [2, 6, 10]; the GOST digital signature scheme of Russia[9]; C. P. Schnorr [5, 18]; Okamoto's key distribute system based on the identification information [16]. When studying some digital signature schemes on the field structure, All these schemes, the subgroup's order are public. If they are designed according to the world's security standards and used according to the designers' regulations, these schemes have high security. However, sometimes, signature scheme designers or signature users still violate some safety principles. Especially during use, users still violate the designer's rules, such as revealing the session key, duplicating the session key, etc. These digital signature schemes can be attacked based on the group's order [2, 10]. Therefore, a digital signature scheme is proposed for the ring structure Z_n, ($n = p.q$, with p, q is strong prime), in which the subgroup's order is private. Therefore, my proposed signature scheme can be avoided attacking, that based on the group's order, such as: Pohlig-Hellman algorithm; Index culculate algorithm, rho Pollard algorithm. A famous digital signature scheme on the ring structure Z_n is the RSA scheme [1] and its security of the RSA scheme is based on the number factorization problem; The schnorr scheme [5, 10, 18]; The scheme from [4] of Chik How Tan and its security is based on the hard of solving the discrete logarithm problem on the ring Z_n, etc. In the digital environment, the digital signature attacks [10, 12, 15] are becoming more and more common. Therefore, research and development of more secure signature schemes plays an important role [4, 17]. In this article, we will propose a new digital signature scheme, its security is based on the hard of solving the discrete logarithm problem on the ring Z_n. My proposed signature scheme overcomes the disadvantages of the digital signature scheme in the field Z_p. Furthermore, the computation speed of my digital signature scheme is faster than similar schemes on field structures by applying the CRT theorem, so it can be suitable for applications that require high security when applied in practice.

2. NOTATION AND TERMINOLOGY

In this section, we are going to define some functions that are used in the following sections. Furthermore, we present the DSA signature scheme [6, 8, 10, 20] because our proposed scheme will be compared with this scheme.

2.1. Some definition

Definition 2.1. The Number function converts a binary string into an integer that does not exceed T bits, notated **Num**: $\mathbb{N}\times\{0,1\}^{\mathrm{H}} \to \mathbb{Z}$. $(T, b_0 b_1 \dots b_{H-1}) \in \mathbb{N}\times\{0,1\}^{\mathrm{H}}$ converts to a, and $a = b_0 + 2b_1 + \dots + 2^{\min(\mathrm{T},\mathrm{H})-1} b_{\min(\mathrm{T},\mathrm{H})-1}$.

Definition 2.2. **Random** (a, b) return a integer value in $[a, b]$.

Definition 2.3. The L(m) function returns the size of m by bit.

Definition 2.4. Cho s $\in \{0,1\}^{\mathrm{H}}$, supose $\mathrm{s} = \mathrm{s}_0 \dots \mathrm{s}_{\mathrm{H}-2}\mathrm{s}_{\mathrm{H}-1}$. Denote $\bar{\mathrm{s}} \in \mathbb{N}$ define by equation as follow: $\bar{\mathrm{s}} = \mathrm{s}_0 2^{\mathrm{H}-1} + \dots + \mathrm{s}_{\mathrm{H}-2}2 + \mathrm{s}_{\mathrm{H}-1}$.

Definition 2.5. **Hash**: $\{0,1\}^{\infty} \to \{0,1\}^{\mathrm{H}}$.

2.2. The Digital Scheme Algorithm (DSA)

DSA parameters

p is a prime, its length is bit, $bitlength(p) = L$. q is a divisor prime of $p-1$, $bitlength(q) = N$. g is a primitive element of subgroup q on Z_{p}, $0 < g < p$. x is the private key that must be kept in secret; x is randomly chosen or pseudorandomly in $[1, q-1]$. y is the public key, where $y = g^{\mathrm{x}} \, mod \, p$. k is a secret number for each message (another name is session key); k is randomly chosen or pseudorandomly in $[1, q-1]$. Set of (p, q, g, x) is also called private key and set of (p, q, g, y) is a public key of signer.

Algorithm 2.1. Signature generation algorithm

Input: $(p, q, g, x), k, M$.
Output: (r, s).
1. $z \leftarrow \boldsymbol{Num}(N, \boldsymbol{Hash}(M))$.
2. $k \leftarrow \boldsymbol{Random}(1, q)$.
3. $r \leftarrow (g^k \, mod \, p) \, mod \, q$.
4. $w \leftarrow (\bar{z} + x.r) \, mod \, q$.
5. if $(r = 0)$ or $(w = 0)$, then goto 2.
6. s $\leftarrow (k^{-1}.(\bar{z} + x.r)) \, mod \, q$.
7. return (r, s).

Algorithm 2.2. Algorithm of signature verify

Input: $(p, q, g, y), (r, s), M$.
Output: "*accept*" or "*reject*".
1. $w \leftarrow s^{-1} \, mod \, q$.
2. $z \leftarrow \boldsymbol{Num}(N, \boldsymbol{Hash}(M))$.
3. $u_1 \leftarrow (\bar{z}.w) \, mod \, q$.
4. $u_2 \leftarrow (r.w) \, mod \, q$.
5. $v \leftarrow ((g^{u_1}.y^{u_2}) \, mod \, p) \, mod \, q$.
6. if $(v = r)$ then return "accept" Else return "reject".

2.3. The security of DSA digital signature scheme

The security of DSA digital signature scheme based on the discrete logarithm problem on field Z_{p} (the hard problem) and FIPS_186-4 standard [8, 14] proposed the size of bit of the L and N parameter that would allow the DSA to be safe until 2030 as follows: $(L, N) = (2048, 224)$, $(L, N) = (2048, 256)$ and $(L, N) = (3072, 256)$. However, when the order of the primitive element is published, that lead to the DSA is insecurity in some of the situations as following:

The first: if the session key is revealed, the secret key x is calculated by the following formula:

$$s = (k^{-1}(\bar{z} + r.x)) \, mod \, q$$

The secret key x is computed easily using the following formula:

$$x = ((s.k - \overline{z}).r^{-1}) \ mod \ q$$

The second: If the session key of the message M, M' are k and two signatures corresponding to M, M' are respectively (r, s) and (r, s'). The secret key x will be found by the attacker as follows: The first z and z' are calculated from the formula $\overline{z} = Num\,(N, Hash\,(M))$ và $\overline{z}' = \boldsymbol{Num}\,(N, \boldsymbol{Hash}\,(M'))$, then s and s' are calculated as follows:

$$s = \left(k^{-1}(\overline{z} + r.x)\right) mod \ q \leftrightarrow k = s^{-1}(\overline{z} + r.x) mod \ q \quad (1)$$

$$s' = \left(k^{-1}\left(\overline{z}' + r.x\right)\right) mod \ q \leftrightarrow k = s'^{-1}(\overline{z}' + r.x) mod \ q \quad (2)$$

From (1) and (2) the equation is as following:

$$s^{-1}\,(\overline{z} + r.x) = s'^{-1}\,(\overline{z}' + r.x) mod \ q \leftrightarrow s^{-1}\overline{z} - s'^{-1}\overline{z}' = (s'^{-1} - s^{-1}).r.x \ mod \ q$$

from this equation it is easy to calculate the secret key x as follows:

$$x = r^{-1}.(s^{-1}.\overline{z} - s'^{-1}\overline{z}').(s'^{-1} - s^{-1})^{-1} \ mod \ q$$

The third: Designers violate the security regulations, these lead to the DSA digital signature scheme and some of its variants built on the field structure Z_p can be attacked by some algorithms that solve the discrete logarithm problem based on the degree of the subgroup, such as Baby step-Giant step, the Polig-Hellman algorithm etc.

2.4. The computational speed

The computational speed of a digital signature scheme depends mainly on modular exponentiation and the calculation of inverse elements in the modulus n. If the sizes of n and p are equal ($|n| = |p|$ and $n = p_1 x q_1$), since the number of modules in the DSA digital signature scheme is a prime number, the CRT theorem cannot be applied to improve the computational speed. Meanwhile, similar digital signature schemes built on the Z_n ring structure can apply the CRT theorem to improve the computational speed.

Example 2.1: In this example, the parameters p and q are used as prime numbers generated by Poclington's algorithm or Lucas's algorithm [17]. Suppose an attacker uses some method to obtain the session key k. In the DSA signature scheme, if the session key is revealed during the signing process, the attacker will be able to calculate the secret key x.

p =14383578043976381451913706752039738469623004360866638122248222455190546930
369244830019923669669820609781381265516767953019505092780747719197316895896033
923050473820476435340858346104130004346524661200587416016634368121704624962473
253316596972178043078846879826673904133207016786021648586046090536227620574780
001895492357772629937311853085255531975734634062961929205870117573324989204834
546860345924645157693651908360155105857390089538207293273379982298894353062730
411676760504167826293886102983997376006358039258141140427123074721182766269 9
q =234982940392365530955082920959567444538486197758354704698986589267930213399
968764530394763857765938297034912058582422856405123961617967469153675730044404
343838356816840351828617672006405567382046645828525254864385299832765872047853
924722353061093420297932291996148934649612666089909118791702805 3
g =107089596773351242962107610877873635558135263862336342149731837209565617444
326402901117534292466881925480294048768549765754224230334740160218134408096504
785514475800148249139896895049741897576599189302719118037222300929768761569516
550918198530649297610860787564264244737317519892626178505609377075922966866806
246931053588777198800843748478758623804122119501931771572496953975567844265322
333013070353838251859778946701118007693075578655295630292931307300988129942363
038622581708519110620410811512666535236115539550306378611060782827565332742 3

z =BF3EA03290BE6800952628DB78748174C0263E8521E8E30D43DF5E9A11F8278025110
5CA1C738EB0445B1060E5272C0601438296E3C117CF8E05873456FE2A90
r =110486040131365016443820369709729194629212082627086726579222623530301927780137828581108461272604019546618363238880371370488524552749280424208157286536686496825454244931756479548406558132432887845392058883363769309225627072704993521986487736964697929739350031439752783747700415603152245570720005
s =152446275901919590448012700971041790211280155644790844905463533933888425937348782424793995622081249525032382990691623895390827681704745033011480394417154345868826673830493699178640000164290677485678988315835827063480318852269490829844323937471964795494186489897308755266902038423861688938727631

Assuming that session key is exposed:

k=159951188481735615522781471166926283311697856634415698422596207441509513952121268972893755031379976854189992551408387024058029650233415357053787138056088054289726271099513066403628726035544644006203472185281656139795796171664784847393408164166506253229328649920448045396458451.

Then the secret key is calculated according to the formula following:

$$x = ((s.k - z).r^{-1}) \bmod q$$

and value x such as:

x =235382187544255435348708414084805240077197152593023889599595782161761074774243136016255020669970659908804806197964817857357609436873345027459690280103757031948222524687878238292128682628410877273898328944660245130207160 23

3. CONSTRUCTION A NEW SIGNATURE SCHEME

In this part, a new signature scheme that can be avoid attacking will be proposed, that based on the group's order, such as: Pohlig-Hellman algorithm; Index culculate algorithm, rho Pollard algorithm.

3.1. Parameters domain

$n = p.q$ with p, q are two distinct primes and n-factorization is a hard problem; m is private, $m = p_1.q_1$ where p_1, q_1 are two distinct primes and $p_1 \mid (p-1)$, $q_1 \mid (q-1)$, $p_1 \nmid (q-1)$, $q_1 \nmid (p-1)$; $\boldsymbol{mbit} = L(m)$.

Value g is a primitive element, $0 < g < n$ its order, denoted by $ord_g = m$ and $< g >$ is cyclic subgroup on Z_n,

x is chosen randomly in $(1, m-1)$ and x is private.

y is public, where $y = g^{x^{-1}} \bmod n$.

The set of (m, x) are the secret key and set of four values $(n, g, y, mbit)$ are the public key.

3.2. Algorithm 3.1. Generation signature

Input: $(n, m, g, x, \boldsymbol{mbit})$, M.//$mbit = L(m)$
Output: (r, s).
1. $z \leftarrow 0$; $tg \leftarrow 0$;
2. While $(z = 0)$ or $((tg, m) <> 1)$
3. $k \leftarrow Random(1, m-1)$
4. $r = (g^k \bmod n)$.
5. $\overline{z} = \boldsymbol{num}(\boldsymbol{mbit}, H(M||r))$// The output is the value z, its size is not more than mbit
6. $tg = (k - \overline{z}) \bmod m$
End while // The loop stops when k-z exists the inverse value
7. $s = x.tg \bmod m$
8. return (r, s).

3.3. Algorithm 3.2. Verifying signature

Input: $M, (r, s), (n, g, y, \boldsymbol{mbit})$// **mbit=** L(m).
Output: "accept" or "reject".
1. $z = \boldsymbol{numr}(\boldsymbol{mbit}, H(M||r))$// value of z < **mbit**.
2. $u = ((g^z.y^s) \ mod \ n)$.
3. if $(u = r)$ return "*True*" else return "*False*".

3.4. Correctness of the algorithm

It's easy to see that:

$$u = ((g^{\overline{z}}.y^s) \ mod \ n) = g^{\overline{z}}.\left((g)^{x(k-\overline{z})}\right)^{x^{-1}} (mod \ n) = g^k (mod \ n) = r \blacksquare$$

4. IMPROVING PROPOSED SIGNATURE SCHEME

In this part, an algorithm is built for the proposed digital signature scheme in order to improve the computing speed. Therefore, the digital signature scheme proposed in part 3 needs to be designed appropriately as follows:

4.1. Parameters domain

$n = p.q$ with p, q are two distinct primes and n-factorization is a hard problem; m is private, $m = p_1.q_1$ where p_1, q_1 are two distinct primes and $p_1 \mid (p-1)$, $q_1 \mid (q-1)$, $p_1 \nmid (q-1)$, $q_1 \nmid (p-1)$; $\boldsymbol{mbit} = L(m)$.

g is a primitive element, $0 < g < n$ have order denoted by $ord_g = m$; $< g >$ is cyclic subgroup on Z_n,

x is chosen randomly in $(1, m-1)$ and x is private.

y is public, where $y = g^{x^{-1}} \ mod \ n$.

The signer parameter set (n, g, m, x) is replaced by the following parameter set: $\left((p, q), c_n, (g_p, g_q), m, (p_1, q_1), x\right)$, in which: $c_n = q.(q^{-1} \bmod p)$, $g_p = g \bmod p$ and $g_q = g \bmod q$. Value m is order of $< g >$.

- Generic parameter (n, g, y).

- The signer's parameter are set of $\left((p, q), c_n, (g_p, g_q), m, ((p_1, q_1), x)\right)$ be secret. At this time, the signature generation algorithm will be written as follows.

4.2. Algorithm 4.1. Generation signature

Input: Message denoted by M; $\left((p, q), c_n, (g_p, g_q), m, (p_1, q_1), x\right)$
Output: (r, s) are signature of message *M*
1. $k \in_R [1, m-1]$.
2. $k_p = k \bmod p_1$.
3. if $(k_p = 0)$ then goto 1.
4. $k_q = k \bmod q_1$.
5. if $(k_q = 0)$ then goto 1.
6. $r_p = g_p^{k_p} \bmod p$; $r_q = g_q^{k_q} \bmod q$.
7. $r = \left((c_n.(r_p - r_q) + r_q) \bmod n\right)$. //applying CRT theorem
8. $\overline{z} = \boldsymbol{num}(\boldsymbol{mbit}, H(M||r))$
9. if ($\overline{z} = 0$) then go to 1.
10. $w = k - \overline{z}$.
11. $s = x.w \bmod m$.
12. return (r, s).■

4.3. Algorithm 4.2. Verifying signature

Because the parameters p, q are secret to the signature verifier, the signature verification algorithm is exact the same as algorithm 3.2 above.

Input: $M, (r, s), (n, g, y, \boldsymbol{mbit})$// **mbit=** L(m).

Output: "*accept*" or "*Reject*".

1. $z = \boldsymbol{numr}(\boldsymbol{mbit}, H(M||r))$// value of z < **mbit**.
2. $u = ((g^{\overline{z}}.y^{s}) \ mod \ n)$.
3. if $(u = r)$ return "*accept*" else return "*Reject*".

4.4. The security of the new digital signature scheme

The security of the new digital signature scheme as follow: the order of g is not published, therefore it is secure in some situations:

(i) Assuming the session key k is revealed, since m is kept secret and $\mathrm{s} = \mathrm{x}.(\mathrm{k} - \overline{\mathrm{z}}) \ mod \ m,$ the attacker cannot determine x.

(ii) The situation where the session key k is coincided, the new digital signature scheme is still secure. This is proven as follows:

$r = (g^{k} \ mod \ n) \ \mathrm{mod\ m}.$
$r' = \mathrm{g}^{k} \ mod \ n' \ \mathrm{mod\ m}.$
$\overline{z} = \mathbf{num}\ (\mathbf{mbit}, \mathrm{H(M||r)})$
$\overline{z}' = \mathbf{num}\ (\mathbf{mbit}, \mathrm{H(M'||r')})$
$s = \mathrm{x}.(\mathrm{k} - \overline{\mathrm{z}}) \ \mathrm{mod\ m} \leftrightarrow k = \mathrm{x}^{-1}.(\overline{z} + \mathrm{s}) \ \mathrm{mod\ m}$
$s' = \mathrm{x}.(\mathrm{k} - \overline{\mathrm{z}}')\mathrm{mod\ m} \leftrightarrow k = \mathrm{x}^{-1}.(\mathrm{s}' + \overline{\mathrm{z}}') \ \mathrm{mod\ m}$
$\mathrm{x}^{-1}.(\overline{z} + \mathrm{s}) \ \mathrm{mod\ m} = \mathrm{x}^{-1}.(\mathrm{s}' + \overline{\mathrm{z}}') \ \mathrm{mod\ m}.$

Since m is kept secret, it is difficult for an attacker to determine the secret key *x* ■

(iii) The new digital signature scheme is still secure from Pohlig Hellman algorithm, Index calculate algorithm, rho Pollard algorithm to calculate the discrete logarithm. 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 generator g) and the new signature scheme keep secret the order of the generator g, so they do not work.■

Table 1. *Security comparison of proposed scheme*.

Some situation and algorithms attacking	**DSA**	**GOST R34.10-2012**	**ECDSA**	**Proposed scheme**
The Pohlig-Hellman algorithm	Yes	Yes	Yes	No
The index calculus algorithm	Yes	Yes	Yes	No
session key revealed	Yes	Yes	Yes	No
session key re-used	Yes	Yes	Yes	No

4.5. Speed of computing

The speed of computing of the proposed scheme is faster than signature schemes on the Z_{p} field structure, because the operations on the ring structure Z_{n} can apply the CRT theorem, the computational speed of the algorithms on the Z_{n} ring will be faster than the computational speed of the algorithms on the Z_{p} field structure, these is proven by theorem in[1].

Theorem 3.1: 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 $a^{e} \ mod \ n$ with p and q are known is only equal to $\frac{1}{4}$ of the cost of calculating $a^{e} \ mod \ n$ 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 $a^{-1}\ mod\ n$ if p, q are not known.

4.6. Construction parameters

From [14], NIST Special Publication 800-57 Part1, 2020 proposed size key apply for the RSA in 2024 to 2030 be 2048 bit. An example of the parameters domain is n = p. q with L(n) = 2048 bit for the proposed scheme. The parameters p and q are used as prime numbers generated by Poclington's algorithm or Lucas's algorithm [17]:

p=17976931348623159077293051907890247336179769789423065727343008115773267580550096313270847732240753602015229840324490311078953104933236990465599999437814003982586791729133133235488128219696644081404980207725901874790616294652072940269837941076230747357831686283196685711850638498904639383830108147262670117001910241 (1024 bit)

q=12583851944036211354105136335523173135325838852596146009140105681041287306385067419289593412568527521478407971590997535036115283817009163574499320144763188715634207825444087236018893223945722006825117438179814741269664209760942550315910521337339323605514481509873178703360301825825272618201997145064576424447124 (1024 bit)

p_1=22088637997671445121090678779188648898023676478637039381422140037568318257 9999 (257 bit)

q_1=13657205691658248661518726317620615574174716766490577395837357196376249807 5349 (257 bit)

n=p.q=22621904249917705110500413682068966372310871868800838822491241799267258597207523625238040988066039424460314105975182696006930547527592076883047563942152293530023442933779621686734652338324818349111859105025652798961668094610359244037241410509478225820268409566255222511242758096541776326215799410641261810612619069882355122896717294067035771929959015012407320774191729585551162902121036404203858380911834065730175822796874474790968732076552467686813895739558379771828356551005562937489953618957342329781174088697888570339469023798191441586737864571269220031524252533943592691169241166189932887932973706428411251714949 (2048 bit)

m=$p_1.q_1$=30166907258277712145269271006886798631606519267083565688797850512893158913627260512528298163018086055290797240573427177477590242699965603226273201722344651 (514 bit)

- Generation element g:

g=10939199530926224361030428276318519730680719598600717269809319194462611821526013876238875982851394919825280634965645163488958923172568697359383780722609206742950684460140968732986901626940835297963453311507747224326446877319544619879248701227654123887088711044515080641403464028730823127348639257868868984503668708353851474008244801244233669574978814173877109254551313331251626769150298601831822867274782545005638847703193759875285460050411535419814608762698757413790412214113394605679467546648069306553617990810766447179825132224235811902620484482502798861894944020883323333179152585129411293710382888121078772406921 (2047 bit)

- Secret key x:

x=61847778290232715991229668678563299495463704124304620256953944762334575758182633688689524914300456003457166341979510888343473570204828738006603335220900179 (511 bit)

- Public y:

y=167101973668695963894149937153650844029860760048414171568185459756385508402388071628806452556649216358069637707044505789845593546027777469989228994598716135420510864267461990400608701969031213339670021761920081946410952088441662180391652690797313626912368088154402998316149303821083314872243634157728650948587530039860246351986364522649202003707575659554061086298831541012305624102558185416674617707429243826223106718072464545565171752645784373990441630619062203017488926851221939517926487763331645365176648312457578021877967863211651875371830444980272308075408378509135488939024758324931732472418746189556802648155812048 bit)

5. CONCLUSIONS

A series of online services have been emerged due to the development of science and technology and the advent of the Internet, using online services, we face the risk of information insecurity. In that context, proposing a new digital signature scheme is more secure and necessary. This paper proposes a new signature scheme for the ring Z_n. Because the order of the primitive element g in our proposed signature scheme is kept secret, if a session key is revealed or coincides, adversaries cannot forge the corresponding signature. In terms of security, our scheme prevents from base attacks by: Pohlig Hellman algorithm, Index culculate algorithm, rho Pollard algorithm to calculate the discrete logarithm. Furthermore, the computation speed of my scheme is faster than the same type of schemes on the field structure Z_p thanks to the application of CRT theorem. However, if we want to apply the CRT theorem, the parameters and signature generation algorithm need to be designed appropriately. The research results are considered a solution to improve the security of the digital signature scheme. The next research is to design a suitable signing algorithm to be able to apply the CRT theorem to improve the calculation speed and build security parameters based on the world standards [8, 14] for the proposed signature scheme.

REFERENCES

[1]. 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, (1996).

[2]. B. Yang, *"A DSA-Based and Efficient Scheme for Preventing IP Prefix Hijacking",* International Conference on Management of e-Commerce and e-Government, Shanghai, pp. 87 - 92, (2014).

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

[4]. Chik How Tan, 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, pp. 1702 – 1706, (2003).

[5]. C. P. Schnorr, *"Efficient signaturegeneration for smartcards"*, Journal of Cryptology Vol. 4, pp. 161-174, (1991).

[6]. C. Y. Lu, W. C. Yang and C. S. Laih, *"Efficient Modular Exponentiation Resistant to Simple Power Analysis in DSA-Like Systems",* International Conference on Broadband, Wireless Computing, communication and Applications, Fukuoka, pp. 401 - 406, (2010).

[7]. D.R Stinson, *"Cryptography: Theory and Practice"*, CRC Press, (2003).

[8]. FIPS PUB 186-4, *"Digital Sinature Standrd (DSS)",* (2013).

[9]. GOST R 34.10-94. *"Produce and check procedures of Electronic Digital Signature based on Asymmetric Cryptographic Algorithm, Russian Federation Standard"*. Information Technology. Cryptographic data Security, Government Committee of the Russia for Standards, (1994).

[10]. H. Morita, J.C. Schuldt, T. Matsuda, G. Hanaoka, T. Iwata. *"On the security of the schnorr signature scheme and DSA against related key attacks."* International Conference on Information Security and Cryptology '15, pp. 20–35, Springer, (2015).

[11]. H. Zhang, R. Li, L. Li and Y. Dong, *"Improved speed Digital Signature Algorithm based on modular inverse",* Proceedings of 2013 2nd International Conference on Measurement, Information and Control, Harbin, pp. 706 – 710, (2013).
[12]. L. Xiao-fei, S. Xuan-jing and C. Hai-peng, *"An Improved ElGamal Digital Signature Algorithm Based on Adding a Random Number",* 2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, Hubei, pp. 236 – 240, (2010).
[13]. M. Toorani and A. Beheshti Shirazi, *"SSMS - A secure SMS messaging protocol for the m-payment systems,"* IEEE Symposium on Computers and Communications, pp. 700–705, (2008).
[14]. National Institute of Standards and Technology (NIST), NIST Special Publication 800-57 Part1, (2020).
[15]. N. Saxena, N. S. Chaudhari, J. Thomas, *"Solution to An Attack on Digital Signature in SMS Security"* Department of Computer Science & Engineering Indian Institute of Technology Indore, India.
[16]. Okamoto E, *"Key distribution systems based on identification information"*. Proc. Of Crypto, (1987).
[17]. Richard Crandall, Carl Pomerance. *"Prime Numbers, A Computational Perspective",* Second Edition, Springer Science Business Media, Inc, (2005).
[18]. T. S. Ng, S. Y. Tan and J. J. Chin, *"A variant of Schnorr signature scheme with tight security reduction"* 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea (South), pp. 411-415, (2017).
[19]. W. C. Kuo, *"On ElGamal Signature Scheme,"* Future Generation Communication and Networking (FGCN 2007), Jeju, pp. 151-153, (2007).
[20]. https://en.wikipedia.org/wiki/Digital_Signature_ Algorithm.

TÓM TẮT

Phát triển một lược đồ chữ ký số chống lại các tấn công dựa trên bậc của phần tử sinh

Trong bài báo này, chúng tôi đề xuất một lược đồ chữ ký số mới dựa trên bài toán logarith rời rạc, đây là một biến thể của lược đồ chữ ký DSA. Lược đồ đề xuất của chúng tôi an toàn hơn lược đồ chữ ký DSA. Ngoài ra, tốc độ tính toán của lược đồ đề xuất của chúng tôi nhanh hơn một số lược đồ cùng loại. Chính vì vậy, lược đồ đề xuất có thể được áp dụng vào trong thực tế.

Từ khoá: Lược đồ chữ ký số; Bài toán logarit rời rạc; Hàm Hash.