Proposing a solution to keep the order of the generating element secret to enhance the security of the digital signature scheme

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

[1]University of Information and Communication Technology, Thai Nguyen University, Z115 Street, Quyet Thang, Thai Nguyen, Vietnam;
[2]Academy of Cryptography Technique, 141 Chien Thang, Thanh Liet, Hanoi, Vietnam.
[*]Corresponding author: truongnd-it@actvn.edu.vn
Received 20 Dec. 2025; Revised 4 Feb. 2026; Accepted 10 Apr. 2026; Published 25 Apr. 2026.
DOI: https://doi.org/10.54939/1859-1043.j.mst.110.2026.150-158

ABSTRACT

Publicising the order of the generating element can sometimes cause a security risk for a digital signature scheme. Our proposed solution is a new digital signature scheme, in which the order of the generating element is kept secret. So, the proposed new digital signature scheme is more secure than the variants of the same type of digital signature scheme. Furthermore, the speed of computing of our scheme is faster than that of some similar schemes. For this reason, it can be applied in practice.

Keywords: Hash function; Discrete logarithmic problem; New digital signature scheme.

1. INTRODUCTION

Digital signatures play an increasingly important role in social life. Thanks to digital signatures, a series of digital services have been born, such as e-government, online tax payment service, and online money transfer service. Currently, many digital signature schemes have been invented and developed by scientists [2-5, 7, 9] such as the ElGamal Signature Scheme [19]; the DSA(Digital Signature Algorithm) of US [2, 6, 10]; the GOST digital signature scheme of Russia[9]; C. P. Schnorr [5, 10, 18]; or Okamoto's key distribute system based on the identification information [16]. Surveys show that most digital signature schemes reveal the order of the generating element. This leads to the risk of being attacked by algorithms that solve the discrete logarithm problem based on the order of the generating element, such as the Pohlig-Hellman algorithm and the Index calculus algorithm. Furthermore, the order of the generator element is public, which caused them to be attacked in some situations: The session keys are hijacked or used repeatedly. In this paper, we propose a new digital signature scheme in which the order of the generator is kept secret. Because the subgroup's order is kept secret that can avoid attacks by Pohlig-Hellman algorithm, Index calculus algorithms, etc. Moreover, the signing speed of the proposed signature scheme is faster than the signing speed of the digital signature schemes with the same variant. Therefore, the proposed signature scheme can be applied in practice.

2. NOTATION AND TERMINOLOGY

In this section, we present the Schnorr signature scheme[5, 10, 18] and the ECDSA(Elliptic Curve Digital Signature Algorithm) because our proposed digital signature scheme will be compared with this scheme. Furthermore, we are going to define some functions which are used in the following sections.

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) returns an integer value in$[a, b]$

Definition 2.3. The $L(m)$ function returns the size of m in bits

Definition 2.4. Cho s $\in \{0,1\}^{\mathrm{H}}$, supose $\mathrm{s} = \mathrm{s}_0 \ldots \mathrm{s}_{\mathrm{H}-2}\mathrm{s}_{\mathrm{H}-1}$. Denote $\bar{\mathrm{s}} \in \mathbb{N}$ is defined by the equation as follows: $\bar{\mathrm{s}} = \mathrm{s}_0 2^{\mathrm{H}-1} + \cdots + \mathrm{s}_{\mathrm{H}-2} 2 + \mathrm{s}_{\mathrm{H}-1}$.

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

Definition 2.6. x || y: concatenating string x with string y.

2.2. The digital scheme algorithm (DSA)

The parameter domain

g is a primitive element of the subgroup $< g >$ on Z_{p}, $0 < g < p$

p is a prime, $L(p) = L$.

q is a prime divisor of $p - 1$, $L(q) = N$.

x is the private key that must be kept secret; x is randomly in $[1, q - 1]$.

y is the public key, $y = g^{\mathrm{x}} \bmod p$.

A set of (p, q, g, x) is also called a private key and a set of (p, q, g, y) is a public key of the signer.

Algorithm 2.1. Signature generation algorithm

Input: $(p, q, g, x), k, M$.

Output: (r, s).

1. $k \leftarrow \boldsymbol{Random}(1, q)$.
2. $r \leftarrow (g^k \bmod p)$.
3. $\overline{z} \leftarrow \boldsymbol{Num}(N, \boldsymbol{Hash}(M||r))$.
4. $s \leftarrow (\overline{z}.\ x + k) \bmod q$.
5. if $(\overline{z} = 0)$ or $(s = 0)$, then goto 1.
6. return $(\overline{z}, s)$.

Algorithm 2.2. Algorithm for signature verification

Input: $(p, q, g, y), (\overline{z}, s), M$.

Output: "*accept*" or "*reject*".

1. $u = ((g^s . y^{\overline{-z}}) \bmod p)$.
2. $\overline{v} = \boldsymbol{Num}(N, \boldsymbol{Hash}(M||u))$.
6. if $(\overline{v} = \overline{z})$ then return "accept", else return "reject".

2.3. The security of the Schnorr digital signature scheme

From FIPS_186-4 standard [8, 14] proposed the size of bit of the *L* and *N* parameter that would allow: the Schnorr and some of its variations, such as: the *DSA* or the GOST- 1994, GOST R34.10 – 2012 etc.. can apply, such as: $(L, N) = (2048, 224)$, $(L, N) = (2048, 256)$ and $(L, N) = (3072, 256)$. However, when the order of the primitive element of the Schnorr scheme and some of its variations are public, that leads to the Schnorr is insecurity in some of the situations as follows:

The first: The session key k is exposed. An attacker can easily compute the secret key by using the following formula: $s \leftarrow (\overline{z}.\ x + k) \bmod q$ and $x = (s - k)(\overline{z}.)^{-1} \bmod q$.

The second: The session key is reused:

$$s \leftarrow (\overline{z}.\ x + k) \bmod q \quad (1)$$

$$s' \leftarrow (\overline{z'}.\ x + k) \bmod q \quad (2)$$

$$s - s' = \overline{z}.\ x + k - \overline{z'}.\ x - k \bmod q$$

$$s - s' = (\overline{z} - \overline{z'})x \bmod q$$

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

The third: The Schnorr digital signature scheme and some of its variants built on the field structure Z_p can be attacked by algorithms that solve the discrete logarithm problem based on the degree of the subgroup, such as the baby step- giant step algorithm, the Pohlig-Hellman algorithm.

Example 2.1: Suppose that the Schnorr digital signature scheme is built on the field structure Z_p, $p = 8101$; $y = 7531$ is the public key, $g = 6$ is generation element where $y = g^x \ mod\ p$. In order to attack the DSA scheme, we have to find x. To find solution x by the–Pohlig-Hellman algorithm is as follows**:**

Since Order of g is $8101 - 1 = 8100$. Factoring $8100 = 2^2.3^4.5^2$. Apply the Pohlig-Hellman algorithm, we have the system of equations as follows:

$$\begin{cases} x = 1\ modulo\ 4 \\ x = 47\ modulo\ 81 \\ x = 14\ modulo\ 25 \end{cases}$$

Apply the CRT theorem, we have a secret key be $x = 6639$ modulo 8100. We can write $x = 8100.k + 6639, k \in Z$. We substitute x into the original DLP ($7531 = 6^x\ mod\ 8101$), we have: $7531 = 6^{8100.k + 6639}\ mod\ 8101$. Since $6^{8100.k}\ mod\ 8100 = 1$ (Little Fermat theorem), so $7531 = 6^{6639}\ mod\ 8101$

2.4. The ECDSA domain parameters

In cryptography, the Elliptic Curve Digital Signature Algorithm ($ECDSA$) offers a variant of the DSA which uses elliptic curve cryptography. The set of domain parameters includes:

- p is a prime number specifying the underlying finite field F_p.

- $E(F_p)$ is an elliptic curve defined on the finite field F_p by equation:

$y^2 = x^3 + ax + b$ with: $a, b \in F_p$ and satisfied: $4.a^3 + 27.b^2 \neq 0\ mod\ q$

The domain parameters here can be generated as specified in ISO/IEC 15946, ANSI X9.62, FIPS 186 – 4 or GOST R34.10 – 2012.

The secret key of the signature entity is d. The corresponding public key Q is: $Q = d.G$

Algorithm 2.3. The signature generation algorithm

Input: Message m, Private d
Output: signature (r, s)
Step 1: Choose random $k \in [1, n-1]$
Step 2: Compute $R = k.P = (x_1, x_2)$
Step 3: Compute $r = x_1 (mod\ n)$
Step 4: If $r = 0$, goto Step 1
Step 5: Compute $e = H(m)$ //
Step 6: Compute s=$k^{-1}\ (e + d.r) (mod\ n)$
Step 7: If $s = 0$, goto 1.
Step 8: Return (r, s) is the signature of message m

Algorithm 2.3.Verifying signature

Input: Pair of (r, s) is a signature of message m
Output: {"$Accept$" or "$Reject$".}
Step 1: Compute s^{-1} is the inverse of s and compute $e' = H(m)$
Step 2: Compute P as:
$P = s^{-1}.e'.m.G + s^{-1}.R.Q$
Step 3: if $Px = r$ then return "$Accept$", if $Px \neq r$ then return "$Reject$".

The security of ECDSA

The ECDSA's order of the primitive element (value n) is published, that lead to the ECDSA being insecure in some of the situations as follows:

The first: if the session key (value k) is revealed, the secret key x is calculated by the following equation:

Compute $s = k^{-1}\ (e + d.r)(mod\ n)$

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

$$d = \ ((s.k\ -e).r^{-1})\ mod\ n$$

The second: Since the order of the primitive element (value n) is published, the ECDSA can be attacked by some algorithms that solve the discrete logarithm problem based on the order of the generator, such as the Pohlig-Hellman algorithm. The attack of the ECDSA is illustrated by the following example:

Example 2.2: Suppose the ECDSA with *E* is the elliptic curve $y^2 = x^3$ + 130.x + 565 defined over $GF[719]$. The number of points is 699, and below the report will list some points of the E curve.

(0,224) (0,495) (1,290) (1,429) (3,118) (3,601) (10,154) (10,565) (11,173) (11,546) (14,152) (14,567) (16,350) (16,369) (17,44) (17,675) (19,52) (19,667) (21,236) (21,483) (22,177) (22,542) (24,82) (24,637) (25,226) (25,493) (27,104) (27,615) (28,331) (28,388) (30,317) (30,402) (32,172) (32,547) (38,134) (38,585) (39,49) (39,670) (41,54) (41,665) (44,294) (44,425) (47,297) (47,422) (48,355) (48,364) (50,58) (50,661) (51,162) (51,557) (53,196) (53,523) (56,159) (56,560) (60,161) (60,558) (61,130) (61,589) (62,347) (62,372) (63,276) (63,443) (64,163) (64,556) (65,217) (65,502) (66,230) (66,489) (67,55) (67,664) (73,226) (73,493) (74,113) (74,606) (78,325) (78,394) (80,351) (80,368) (85,354) (85,365) (87,350) (87,369) (88,19) (88,700) (89,121) (89,598) (95,274) (95,445) (98,304) (98,415) (101,71) (101,648) (102,284) (102,435) (106,143) (106,576) (107,276) **(107,443)=P** (108,62) (108,657) (109,146) (109,573) .....**(608,427)=Q** (609,116) (609,603) (611,40) (611,679) (614,209) (614,510) (616,350) (616,369) (617,115) (617,604) (618,287) (618,432) (621,226) (621,493) (624,334) (624,385) (626,168) (626,551) (627,111) (627,608) (628,359) (628,360) (629,160) (629,559) (630,190) (630,529) (631,182) (631,537) (635,358) (635,361) (636,198) (636,521) (638,1) (638,718) (642,100) ..... (705,255) (705,464) (706,331) (706,388) (709,292) (709,427) (710,129) (710,590) (711,106) (711,613) (712,104) (712,615) (713,293) (713,426) (715,134) (715,585) (716,322) (716,397) (718,282) (718,437)

Suppose that $P = (107, 443)$ and that $Q = (608, 427)$. Now we want to determine the unique integer *x* such that $Q = x.P$. It should be noted that it can be shown that P has order 699. Since the order of $\langle P \rangle$ known, so it is factored as: 699 = 3.233. Using the Pohlig-Hellman attack, we need to compute *x* modulo 3 and x modulo 233, we will then obtain a unique solution by using the CRT.

- With x modulo 3: We start by computing our small list: $T = \{j.([\frac{699}{3}].P) \mid 0 \leq j \leq 2\}$ = {O↔ j = 0, (24, 82) ↔ j = 1, (24, 637) ↔ j = 2} and we compute $\frac{699}{3}$Q = $\frac{699}{3}$(608, 427) = (24, 82). We now appeal to T to determine a match and we find that $x_1 \equiv 1$ modulo 3.

- With x modulo 233: We have a much larger list to compute this time. $T = \{j.(\frac{699}{233}.P) \mid 0 \leq j \leq 222\}$ = {j(3.P) | $0 \leq j \leq 222$}. List T includes:

Table 1. *List computing result of a part of* T.

j	3.P	j	3.P	j	3.P	j	3.P	j	3.P	j	3.P
0	θ	1	460,25	2	631,182	3	325,326	4	213,106	5	425,144
6	(392,319)	7	(670,460)	8	(404,91)	9	(635,361)	10	(242,221)	11	(422,363)
12	(663,494)	13	(617,604)	14	(284,505)	15	(541,392)	16	(168,508)	17	(591,204)
18	(80,368)	19	(290,673)	20	(421,410)	21	(567,681)	22	(548,262)	23	(704,331)
24	(436,453)	25	(161,275)	26	(133,221)	27	(306,52)	28	(475,57)	29	(41,54)
...	...	...	...	...	...	...	...	...	...	...	...
222	(422,356)										

Calculating $\frac{699}{233}.Q = 3.Q = (306, 52)$, we find that this matches with entry 27 in our list T. This then yields $x \equiv 27$ modulo 233. Using the CRT on the system of congruences:

$$\begin{cases} x \equiv 1 \, modulo \, 3 \\ x \equiv 27 \, modulo \, 233 \end{cases}$$

We find that we obtain the unique solution x(secret key) $x \equiv 493$ modulo 699 as required, $x.\mathrm{P} = \mathrm{Q} \leftrightarrow 493.(107, 443) = (608, 427)$.

3. THE PROPOSED SOLUTION

In this section, we propose a solution to keep the order of the generating element secret to enhance the security of the digital signature scheme. Our solution are construction a new digital signature scheme based on the composite discrete logarithm.

3.1. Parameters domain

Choose two sufficiently strong prime numbers p and q.$n = p.q$ Chose 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 compute $y = g^x \bmod n$.

x is private key and y is public key.

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

3.2. Algorithm 3.1. Generation signature

Input: $(n, m, g, x, \boldsymbol{mbit})$, M.//$mbit = L(m)$

Output: (r, s).

1. $\overline{z} \leftarrow 0$; $tg \leftarrow 0$;
2. While $(\overline{z} = 0) or \left((tg, m) <> 1\right)$
2.1. $k = Random(1, m-1)$
2.2. $r = g^k \bmod n$.
2.3. $\overline{z} = \boldsymbol{num}(\boldsymbol{mbit}, H(M||r))$
2.4. $tg \equiv (k.r - \overline{z}) \bmod m$.
End while
3. $s = x^{-1}.tg \bmod m$
4. return (r, s).

3.3. Algorithm 3.2. Verifying signature

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

Output: "accept" or "reject".

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

3.4. Correctness of the algorithm

It's easy to see that:

$$g^{\bar{z}}.y^{s} \ mod \ n = g^{\bar{z}}.g^{x.s} \ mod \ n = g^{\bar{z}+x.s} \ mod \ n = g^{\bar{z}+x.(k.r-\bar{z})x^{-1}} \ mod \ n$$
$$= g^{\bar{z}-\bar{z}+k.r} \ mod \ n = g^{k.r} \ mod \ n = r^{r} mod \ n \ \blacksquare$$

3.5. The security of the new digital signature scheme

The security of the new digital signature scheme is as follows: 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, from the following formula: $\mathrm{s} = \mathrm{x}^{-1}.(k.r - \bar{z}) \ mod \ m$, 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:

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

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, Pollard's rho algorithm to calculate the discrete logarithm. This is obvious, since the Pohlig-Hellman, Index calculate and Pollard's rho algorithm for calculating discrete logarithms, which require the input to have the order of the generator) and the new signature scheme keeps secret the order of the generator g, so they do not work.■

3.6. Speed of computing

The speed of computing of the proposed scheme is faster than that of 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, this is proven by the 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). 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.

The proposed digital signature scheme's signing speed is faster than the signing speed of the same type of signature scheme on the field structure Z_{p} by applying the CRT theorem, which has been proven in theorem 3.1. Furthermore, the proposed signature scheme is more secure than some similar type signature schemes on Zp field structure**.**

Table 2. *Security comparison of the proposed scheme.*

Some situations and algorithms attacking	DSA	GOST - 1994	Schnorr scheme	ECDSA	Proposed scheme
The Pohlig-Hellman algorithm	Yes	Yes	Yes	Yes	No
The index calculus algorithm	Yes	Yes	Yes	Yes	No
Baby step- Giant step algorithm	Yes	Yes	Yes	Yes	Possible, but difficult
Session key revealed	Yes	Yes		Yes	No
Session key re-used	Yes	Yes		Yes	No

3.7. Contruct the parameter

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

$\boldsymbol{p}$ =6117232749284706947203239371920572680913581374344079905019539757091969779609195832178686393815797179231584450687350904654445900835503615065033361689021062568606447297148062202675498823110770692220088564171129697281476074871648513636925997756154333534212900701320193368505812731107304348554620095567369169517349827552049384353090543218947958563191 9 (1152 bit)

$\boldsymbol{q}$ =4282062924499294863042267560344400876639506962040855933513677829964378845726437082525080475671058025462109115481145633258112130584852530545523353182314743798024513108003643543717684825056512007739295238476671535742425595326736506180958215537261787414870785359381785486325992462495432632943017860515750501904145609000002639899443406995357214479261 9 (1152 bit)

$\boldsymbol{p_1}$=1884301757735223514072906698839139946883075054307835673864355039680854294971 (257 bit)

$\boldsymbol{q_1}$=1700315685696166233167452354083655615989777905867642850444655217034448211663871 (257 bit)

$\boldsymbol{n = p.q =}$

2619437555624493402657599841476825721458150176469303108543080910600343378135398115827547115100750310063690929630657824458975476451256150988243923984221390606660191893881674529559615419571209980570118940574554235662719196429284694287700518022626232838698176244785015686516187968708945834534024659098239615302213265487102202845427162354312066508027796986366886926846558492672712491586151688587523670810821178515326619693757480568940486591323422457539057131903045845448966481183736277834905684400474400867237279292315638403422043276999914652924339506174207572953515094391659745867231021679120145768360600822994095942038741144315074641660436539920846715170154964543607729936542920264715348322005861 (2304 bit)

$\boldsymbol{m = p_1.q_1 =}$

=3203907835262057874792321423919652197888290638022351505502953197384183726748933640648821105668144949183371862093265920367495331792618914443133459876833977 7 (514 bit)

$\boldsymbol{g}$ =1092805446860152044677914723154788527836625535134445151909784219238040855886588439498815235553434691689126389856238313865831260362789878354475270313674880233412657974474319506379230034067959426587310593157429607198102614300101708737287865610857093812320695028926450601583143760932226663551612219759337825698377254305009357324613055641201403266482876470647486415793328750182981109433228185818225957404600702347546997909277334754504508107869068075 97

461611752997979644611454037753692857918255971133378102415304891481821262675
249241295939127363375156634355900880103089891934611265386023693325783732849
871917235025057072343512347115232895225171689760593558859718451168266451823
9469020914514627367754 (2303 bit)

Private key, denote by $\boldsymbol{x}$:

$\boldsymbol{x} =$ 534896968734055943108224653962169224 2 (123 bit)

Public key $\boldsymbol{y}$:

y=165893389168891685004241592884367920336257389946261695304730687371993943 7
869926626486207953270697617601957205964987603877628157595005001347391707054
300937208164536989078121871334455477071975187961888487974282386196165652622
191356903887525155952914416100242781169377381501243720654495485160483485241
613036351962748404538692917041897274831296730110843730189058031798563378879
910709449809241361569962287467040972283560648176164279170907611462077535449
951914979417343997282090634955302751137360917948890285438531280164129718909
871012237477985461263542611838064582979985740694293203113286897592349473941
512844129898623481196443620937958214078949308354926285884751334892094941374
030980408663671670199 (2303 bit). It's easy to check if the key satisfies the following condition: $\boldsymbol{m = p_1.q_1}$ where $\boldsymbol{p_1, q_1}$ are two distinct primes and $\boldsymbol{p_1 \mid (p-1)}$, $\boldsymbol{q_1 \mid (q-1)}$, $\boldsymbol{p_1 \nmid (q-1)}$, $\boldsymbol{q_1 \nmid (p-1)}$;

4. CONCLUSIONS

In this paper, we propose a solution to keep the order of the generating element secret to enhance the security of the digital signature scheme on the ring structure $Z_{\mathrm{n}}(n = p.q,\ p,\ q$ are strong primitive). Because the order of the primitive element g is kept secret, our scheme is still secure when the session key is revealed or compromised. In terms of security, our scheme prevents base attacks by the Pohlig-Hellman algorithm, Index calculate algorithm, rho Pollard algorithm. 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. The next time, we will design a suitable signing algorithm how our scheme can apply the CRT theorem in order to improve the computing speed and build security parameters based on the standards of the world [8, 14].

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, 87–92, (2014).

[3]. Binh V, Minh H. Nguyen, 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, Chee Kheong Siew, *"Signature scheme based on composite discrete logarithm"*, Fourth International Conference on Information, Communications and Signal Processing and the Fourth Pacific Rim Conference on Multimedia, 1702–1706, (2003).

[5]. C. P. Schnorr, *"Efficient signature generation for smartcards"*, Journal of Cryptology, 4, 161–174, (1991).

[6]. C. Y. Lu, W. C. Yang, 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, 401–406, (2010).

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

[8]. NIST, *"Special Publication 800-57 Part 1"*, (2020).

[9]. GOST R34.10–2012, *"Russian Federation Standard Information Technology"*, Government Committee of Russia for Standards, (2012).

[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 — CRYPTOLOGY '15, Springer, 20–35, (2015).
[11]. Luu Hong Dung, Hoang Thi Mai, Nguyen Huu Mong, *"A form of digital signature scheme on the problem of number factoring"*, Conference FAIR, 377–386, (2015).
[12]. L. Xiao-fei, S. Xuan-jing, C. Hai-peng, *"An Improved ElGamal Digital Signature Algorithm Based on Adding a Random Number"*, International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, 236–240, (2010).
[13]. M. Toorani, A. Beheshti Shirazi, *"SSMS - A secure SMS messaging protocol for the m-payment systems"*, IEEE Symposium on Computers and Communications, 700–705, (2008).
[14]. NIST, *"Special Publication 800-57 Part 1 Revision 5 Recommendation for Key Management"*, (2020).
[15]. Neetesh Saxena, Narendra S. Chaudhari, Jaya Thomas, *"Solution to An Attack on Digital Signature in SMS Security"*, Department of Computer Science & Engineering Indian Institute of Technology Indore.
[16]. Okamoto E, *"Key distribution systems based on identification information"*, Proc. of Crypto, (1987).
[17]. R. Crandall, C. Pomerance, *"Prime Numbers, A Computational Perspective"*, Springer Science, (2005).
[18]. T. S. Ng, S. Y. Tan, J. J. Chin, *"A variant of Schnorr signature scheme with tight security reduction"*, International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, 411–415, (2017).
[19]. W. C. Kuo, *"On ElGamal Signature Scheme"*, Future Generation Communication and Networking (FGCN), Jeju, 151–153, (2007).
[20]. Tripathi Shailendra Kumar, Bhupendra Gupta, *"An efficient digital signature scheme by using integer factorization and discrete logarithm problem"*, International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, (2017).

TÓM TẮT

Đề xuất một giải pháp giữ bí mật bậc của phần tử sinh nhằm nâng cao mức độ an toàn của lược đồ chữ ký số

Việc công khai bậc của phần tử sinh trong một số trường hợp có thể làm phát sinh các rủi ro về an toàn đối với lược đồ chữ ký số. Trong bài báo này, chúng tôi đề xuất một lược đồ chữ ký số mới, trong đó bậc của phần tử sinh được giữ bí mật. Nhờ đó, lược đồ đề xuất đạt mức độ an toàn cao hơn so với các lược đồ chữ ký số cùng loại đã được công bố. Bên cạnh đó, kết quả phân tích cho thấy tốc độ tính toán của lược đồ đề xuất nhanh hơn so với một số lược đồ tương tự, do đó có tính khả thi và tiềm năng ứng dụng trong thực tiễn.

Từ khoá: Hàm hash; Bài toán lôgarit rời rạc; Lược đồ chữ ký số mới.