A NEW TYPE OF POST-QUANTUM DIGITAL SIGNATURE SCHEME

Kim Tuan Nguyen[1], *Minh Duc Tong*[2,*], *Tuan Hao Hoang*[2],
Hong Dung Luu[2]

Abstract

In this paper, the authors propose a new type of post–quantum digital signature scheme based on hard problems, which belongs to the group of unsolvable problems that currently have no solution. Therefore, this type of digital signature scheme can be resistant to quantum attacks based on the quantum algorithm proposed by P. Shor. In addition to quantum resistance, the type of signature scheme proposed here can also be used as pre–quantum digital signature schemes that are widely used in current practical applications.

Index terms

Digital signature scheme; post–quantum digital signature scheme; quantum–resistant digital signature scheme; discrete logarithm problem; new hard problem.

1. Introduction

Quantum computing is expected to bring huge benefits to scientific and social fields once it becomes a reality. However, it will also become a huge risk, threatening the survival of the field of cryptography and in fact it has been a factor that directly impacts cryptography since P.Shor proposed efficient quantum algorithms to solve the prime factorization problem and the discrete logarithm problem in 1994 and later [1]–[3]. Since then, quantum computing is expected to make all public–key cryptography algorithms and digital signatures based on the difficulty of these problems insecure. But on the other hand, it also opens up a new approach to the problem of building quantum-resistant digital signature algorithms, that is, if we use difficult problems with no solutions as the basis for building post–quantum digital signature algorithms, then these types of algorithms will be able to resist quantum attacks according to P.Shor's method, because then Shor's quantum algorithm will be disabled. In this paper, the authors propose a new type of post–quantum digital signature scheme based on new hard problems. The schemes proposed here, in addition to being resistant to quantum attacks, can also be used completely as the current digital signature schemes in use (RSA, DSA, etc.), and suitable for many applications requiring high secure in practice.

[1] School of Information Technology, Phenikaa University

[2] Institute of Information and Communication Technology, Le Quy Don Technical University

[*] Corresponding author, email: ductm@lqdtu.edu.vn

DOI: 10.56651/lqdtu.jst.v14.n01.1042.ict

2. The new hard problems

2.1. The discrete logarithm problems

The DLP or discrete logarithm problem on the finite field F_p is described as follows: Given a prime *p*, a generator *g* of Z_p^*, find the integer *x*, $0 \leq x \leq p-2$, such that:

$$y = g^x \bmod p$$

The ECDLP or discrete logarithm problem on elliptic curve is described as follows: Suppose *G* is a point on an elliptic curve *E*, generating the cyclic group <*G*>. Let a point *P* ∈ <*G*>. Find the integer *k* such that:

$$P = k \times G$$

2.2. The new hard problems on the finite filed

From the discrete logarithm problem on the finite field F_p, we see that if the parameter *g* is also kept secret, then the discrete logarithm problem on F_p will become an unsolvable problem. In the simplest case, it is possible to choose the secret key *x* itself for the role of the parameter *g*, the new hard problems is stated in the forms as follows:

Form 1: *Given p is a prime number, for each positive integer y in* F_p*, find the number x that satisfies the following equation:*

$$y = x^x \bmod p$$

Form 2: *Given p is a prime number and (a,b) are positive integers in* F_p*, find the number x that satisfies the following equation:*

$$a^x \equiv x^b \bmod p$$

It is easy to see that those existing algorithms for the discrete logarithm problem on F_p [4]–[9] cannot be used to solve this problem. At present, there is no solution to this problem other than the “brute force attack” method.

2.3. The new hard problems on the elliptic curve

From ECDLP we see that if the point G is also kept secret, then the ECDLP will become an unsolvable problem. In the simplest case, the *x*–coordinate of *G* (*x G*) can be chosen as the secret parameter *k*, then the new hard problems on the elliptic curve is stated in the forms as follows:

Form 1: *Let E(F_p) be an elliptic curve defined on the finite field Fp and G be a point on E(F_p) generating the cyclic group <G>. Given a point P in <G>, find point G that satisfies the following equation:* $P = x_G \times G$

Form 2: *Let E(Fp) be an elliptic curve defined on the finite field* F_p *and G be a point on E(F_p) generating the cyclic group <G>. Given point P in <G> and an integer k in* F_p*, find point G that satisfies the following equation:* $x_G \times P = k \times G$

It is easy to see that those existing algorithms for the ECDLP cannot be used to solve this problem. At present, there is no solution to this problem other than the "brute force attack" method. In the type of proposed digital signature scheme, the first form of the hard problem is used to generate the public and private key pairs of the signer in the Key Generation algorithm, it is also used to generate signature in the signature generation algorithm, while the second form of this hard problem is used as the basis for construction the signature verification algorithm.

3. The post-quantum digital signature schemes base on the new hard problems

This section will present the construction of post–quantum digital signature schemes based on the new hard problems mentioned in Section 2.

3.1. The first scheme

The proposed scheme here includes the Key Generation algorithm (Algorithm 1), the signature generation algorithm (Algorithm 2) and the Signature Verification algorithm (Algorithm 3). These algorithms are presented in the following Sections 3.1.1, 3.1.2 and 3.1.3. The proof of the correctness and evaluation of the security of the algorithm are presented in Sections 3.1.4 and 3.1.5.

3.1.1. The Key Generation algorithm: The End–User's public/private key pair is generated by the Key Generation algorithm based on the domain parameter *p* which is a prime number. The domain parameter here can be generated as specified in ISO/IEC 14888-3 [10], FIPS 186– 4 [11] or GOST R34.10–94 [12].

To generate a private and public key pair, each signer first needs to choose a secret key x : 1 < x < p–1 and GCD(x, p–1) = 1.

The public key y is generated from x and p according to the formula:

$$y = x^{-(x)^{-1}} \bmod p \tag{1}$$

The Key Generation algorithm (Algorithm 1) of the proposed scheme is described as follows:

Algorithm 1

```
Input: $L_p$
Output: $p, x, y$
[1]. Generate $p$: len$(p) = L_p$
[2]. Select $x$: $1 < x < p$ and $\gcd(x, p-1) = 1$
[3]. $y \leftarrow x^{-(x^{-1})} \bmod p$ if $(y = 1)$ then go to [2]
[4]. Return $(p, x, y)$
```

Note: len(.) is function to calculate length (in bits) of an integer; L_p is length (in bits) of prime number p; *p* is system parameter/domain parameter; *x*,*y* are private and public key of the signer.

3.1.2. The signature generation algorithm: Assuming (*r*,*s*) is the signature on the message to be signed M. The first component of the signature *r* is calculated according to the following formula:

$$r = (x)^{\left(x^{-1} \times h + k\right) \times x^{-k}} \bmod p \qquad (2)$$

here, the *k* is a randomly value in the range (1, *p*–1); the *h* is the representative value (hash value) of the message to be signed M: *h* = H(M), here: H() is the hash function (eg: SHA-1, SHA-256,... [13]).

The second component s of the signature is calculated according to the following formula:

$$s = r \times (x)^k \bmod \ n \qquad (3)$$

here, $n = p \times (p - 1)$.

The signature generation algorithm (Algorithm 2) of the proposed scheme is described as follows:

```
Algorithm 2
Input: $p$, $x$, $M$.
Output: $(r, s)$.
[1]. select $k$: $1 < k < p - 1$
[2]. $h \leftarrow H(M)$
[3]. $r \leftarrow (x)^{(x^{-1} \times h + k) \times x^{-k}} \bmod p$
[4]. $n \leftarrow p \times (p - 1)$
[5]. $s \leftarrow r \times (x)^k \bmod n$
[6]. return $(r, s)$
```

Note: M is message to be signed, with: $M \in \{0, 1\}^\infty$; H(.) is hash function, with $H : \{0, 1\}^* \mapsto Z_h$, with: h < p.

3.1.3. The Signature Verification algorithm: The Signature Verification algorithm of the scheme is construction on the assumption:

$$(y)^{r \times h} \times (r)^{s + r} \bmod p = (s)^r \bmod p \qquad (4)$$

That is, if M and the signature (*r*,*s*) satisfy the equality (4), then the signature is considered valid, and the message is verified for origin and integrity. Otherwise, the signature is considered forged, and the message to be verified is denied in terms of origin and integrity.

The Signature Verification algorithm (Algorithm 3) of the proposed scheme is described as follows:

Algorithm 3

```
Input: $p$, $y$, $M$, $(r, s)$.
Output: TRUE/FALSE.
[1]. $h \leftarrow H(M)$
[2]. $a \leftarrow (y)^{r\times h} \times (r)^{\mathrm{s+r}} \mod p$
[3]. $b \leftarrow (\mathrm{s})^{r} \mod p$
[4]. if $(a = b)$ then return (TRUE) else return (FALSE)
```

Note: *M*,(*r*,*s*) are message and signature to be verified. If the result is TRUE, then the integrity and origin of M are asserted. Otherwise, if the result is FALSE, then M is denied for origin and integrity.

3.1.4. The correctness of the proposed scheme: What needs to be proved here is: if

$$a = (y)^{r\times h} \times (r)^{s+r} \bmod p \tag{5}$$

and

$$b = (s)^{r} \bmod p \tag{6}$$

then: *a* = *b*.

Indeed, if the signature and message to be verified are not forged, from (1), (2), (3) and (5) we will have:

$$\begin{aligned} a &= (y)^{r\times h} \times (r)^{s+r} \bmod p = (y)^{r\times h} \times (r)^{s} \times (r)^{r} \bmod p \\ &= (x)^{-(x)^{-1}\times r\times h} \times \left((x)^{(x)^{-k}\times\left((x)^{-1}\times h+k\right)}\right)^{r\times(x)^{k}} \times (r)^{r} \bmod p \\ &= (x)^{-(x)^{-1}\times r\times h} \times \left((x)^{(x)^{-1}\times h\times(x)^{-k}} \times (x)^{k\times(x)^{-k}}\right)^{r\times(x)^{k}} \times (r)^{r} \bmod p \\ &= (x)^{-(x)^{-1}\times r\times h} \times (x)^{(x)^{-1}\times h\times(x)^{-k}\times r\times(x)^{k}} \times (x)^{k\times(x)^{-k}\times r\times(x)^{k}} \times (r)^{r} \bmod p \\ &= (x)^{-(x)^{-1}\times r\times h} \times (x)^{(x)^{-1}\times r\times h} \times (x)^{k\times r} \times (r)^{r} \bmod p = (x)^{k\times r} \times (r)^{r} \bmod p \end{aligned} \tag{7}$$

From (3) and (6) we get:

$$b = (s)^{r} \bmod p = \left(r \times (x)^{k}\right)^{r} \bmod p = (x)^{k\times r} \times (r)^{r} \bmod p \tag{8}$$

From (7) and (8) we have: $a = b$.

Thus, the correctness of the scheme has been proved.

3.1.5. The quantum resistance mechanism of the proposed scheme: As mentioned in the introduction section, the type of hard problem used to construct the digital signature scheme here belongs to the class of hard problems with no solution, so for this type of hard problem, Shor's quantum algorithm is ineffective, so quantum attack by Shor's method is not infeasible for the type of scheme proposed here, and that is the quantum resistance mechanism of the proposed type of scheme. The secure of the proposed signature scheme can be further evaluated through its resistance to some types of attacks that will be considered below.

- **Secret key attack:** In proposed scheme, attacking the secret key can be performed on the Key Generation algorithm (Algorithm 1) and the signature generation algorithm (Algorithm 2), however the attacker will encounter the first form of the hard problems mentioned in Section 2.2. Therefore, to find the signer's private key from the proposed scheme's key generation and signature generation algorithms, the attacker has no other choice but to solve the above hard problem by the "brute force attack" method.

- **Signature forgery attack:** From the Signature Verification algorithm (Algorithm 3) of the proposed scheme, a set of 2 values (*r*,*s*) will be confirmed as a valid signature with the message to be verified M it satisfies the condition (9):

$$(y)^{r\times h} \times (r)^{s+r} \bmod p = (s)^{r} \bmod p \tag{9}$$

It can be seen that condition (4) here is the second form of the hard problem mentioned in Section 2.2, which is known to be a hard problem (in mathematics) that currently has no other solution than the "brute force attack" method.

3.2. The second scheme

The proposed second scheme here includes the Key Generation algorithm (Algorithm 4), the signature generation algorithm (Algorithm 5) and the Signature Verification algorithm (Algorithm 6). These algorithms are presented in the following Sections 3.2.1, 3.2.2 and 3.2.3. The proof of the correctness and evaluation of the security of the algorithm are presented in Sections 3.2.4 and 3.2.5.

3.2.1. The Key Generation algorithm: In this scheme, the End–User's public/private key pair is generated by the Key Generation algorithm based on the set of domain parameters, including a pair of prime numbers p, q satisfy: $q|(p-1)$. The domain parameters here can be generated as specified in ISO/IEC 14888–3 [10], FIPS 186–4 [11] or GOST R34.10–94 [12]. Similar to the DSA signature scheme [11], the use of the Z_q subgroup here is intended to reduce the magnitude of the exponent in the power operations, thereby allowing to increase the efficiency of the algorithm. In addition, it also allows to reduce the size of the signature generated by this scheme.

To generate a private and public key pair, each signer first needs to choose a value $\alpha \in Z_p^*$, then compute the secret key *x* according to the formula: $x = \alpha^{\frac{p-1}{q}} \bmod p$.

The public key *y* is generated from *x* and *p*, *q* according to the formula:

$$y = x^{-(x)^{-1}} \bmod p \tag{10}$$

The Key Generation algorithm (Algorithm 4) of the proposed scheme is described as follows:

Algorithm 4

```
Input: $L_p \cdot L_q$
Output: $p, q, x, y$
[1]. generate $p, g$: len$(p) = L_p$, len$(q) = L_q$, $q|(p-1)$
[2]. select $\alpha$: $1 < \alpha < p$
[3]. $x \leftarrow \alpha^{\frac{p-1}{q}} \mod p$ if $(x = 1)$ then goto [2]
[4]. $y \leftarrow x^{-(x)^{-1}} \mod p$ if $(y = 1)$ then goto [2]
[5]. return $(p, q, x, y)$
```

Note: $len(.)$ is function to calculate length (in bits) of an integer; L_p,L_q are length (in bits) of prime numbers *p* and *q*; *p*, *q* are system parameter/domain parameters; *x*, *y* are private and public key of the signer.

3.2.2. The signature generation algorithm: Assuming (*r*,*s*) is the signature on the message to be signed M. The first component of the signature *r* is calculated according to the following formula:

$$r = (x)^{\left(x^{-1} \times h + k\right) \times x^{-k}} \mod p \tag{11}$$

here the *k* is a randomly value in the range (1, *q*) and *h* is the representative value (hash value) of the message to be signed M: *h* = H(M), here: H() is the hash function (eg: SHA-1, SHA-256,... [13].

The second component *s* of the signature is calculated according to the following formula:

$$s = r \times (x)^k \mod\ n \tag{12}$$

here: $n = p \times q$. The signature generation algorithm (Algorithm 5) of the proposed scheme is described as follows:

Algorithm 5

```
Input: $p$, $q$, $x$, $M$.
Output: $(r, s)$.
[1]. select $k$: $1 < k < p$
[2]. $h \leftarrow H(M)$
[3]. $r \leftarrow (x)^{(x^{-1} \times h + k) \times x^{k}} \mod p$
[4]. $n \leftarrow p \times q$
[5]. $s \leftarrow r \times (x)^k \mod n$
[6]. return $(r, s)$
```

Note: M is message to be signed, with: $M \in \{0, 1\}^{\infty}$; H(.) is hash function, with $H : \{0, 1\}^{*} \mapsto Z_h$, *q* < *h* < *p*.

3.2.3. The Signature Verification algorithm: The Signature Verification algorithm of the scheme is construction on the assumption:

$$(y)^{r\times h} \times (r)^{s+r} \bmod p = (s)^{r} \bmod p \tag{13}$$

That is, if M and the signature (*r*,*s*) satisfy the equality (13), then the signature is considered valid, and the message is verified for origin and integrity. Otherwise, the signature is considered forged, and the message to be verified is denied in terms of origin and integrity.

The Signature Verification algorithm (Algorithm 6) of the proposed scheme is described as follows:

Algorithm 6

```
Input: $p$, $q$, $y$, $M$, $(r, s)$.
Output: TRUE/FALSE.
[1]. $h \leftarrow H(M)$
[2]. $a \leftarrow (y)^{r\times h} \times (r)^{s+r} \bmod p$
[3]. $b \leftarrow (s)^{r} \bmod p$
[4]. if $(a = b)$ then return TRUE else return FALSE
```

Note: M,(*r*,*s*) are message and signature to be verified. If the result is TRUE, then the integrity and origin of M are asserted. Otherwise, if the result is FALSE, then M is denied for origin and integrity.

3.2.4. The correctness of the proposed scheme: What needs to be proved here is: if

$$a = (y)^{r\times h} \times (r)^{s+r} \bmod p \tag{14}$$

and

$$b = (s)^{r} \bmod p \tag{15}$$

then: *a* = *b*.

Indeed, if the signature and message to be verified are not forged, from (10), (11),(12) and (14) we will have:

$$\begin{aligned}
a &= (y)^{r\times h} \times (r)^{s+r} \bmod p = (y)^{r\times h} \times (r)^{(s)} \times (r)^{r} \bmod p \\
&= (x)^{-x^{-1}\times r\times h} \times \left((x)^{x^{-k}\times\left(x^{-1}\times h+k\right)}\right)^{r\times x^{k}} \times (r)^{r} \bmod p \\
&= (x)^{-x^{-1}\times r\times h} \times \left((x)^{x^{-1}\times h\times x^{-k}} \times (x)^{k\times x^{-k}}\right)^{r\times x^{k}} \times (r)^{r} \bmod p \\
&= (x)^{-x^{-1}\times r\times h} \times (x)^{x^{-1}\times h\times x^{-k}\times r\times x^{k}} \times (x)^{k\times x^{-k}\times r\times x^{k}} \times (r)^{r} \bmod p \\
&= (x)^{-x^{-1}\times r\times h} \times (x)^{x^{-1}\times r\times h} \times (x)^{k\times r} \times (r)^{r} \bmod p = (x)^{k\times r} \times (r)^{r} \bmod p
\end{aligned} \tag{16}$$

From (12) and (15) we get:

$$b = (s)^r \bmod p = \left(r \times (x)^k\right)^r \bmod p = (x)^{k \times r} \times (r)^r \bmod p \tag{17}$$

From (15) and (17) we have: $a = b$.

Thus, the correctness of the scheme has been proved.

3.2.5. The quantum resistance mechanism of the proposed scheme: The analysis and evaluation of the quantum resistance mechanism as well as the secure of this scheme can be performed similarly to the first scheme mentioned in Section 3.1.5.

3.3. The third scheme

This section will present the construction of quantum–resistant digital signature scheme based on the new hard problems on the elliptic curve mentioned in Section 2.3.

The proposed scheme here includes the Key Generation algorithm (Algorithm 7), the signature generation algorithm (Algorithm 8) and the Signature Verification algorithm (Algorithm 9). These algorithms are presented in the following Sections 3.3.1, 3.3.2 and 3.3.3. The proof of the correctness and evaluation of the secure of the algorithm are presented in Sections 3.3.4 and 3.3.5.

3.3.1. The Key Generation algorithm: The set of domain parameters includes:

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

- E(F_p) is elliptic curve defined on the finite field F_p by equation: $y^2 = x^3 + ax + b$ with: *a*,*b* ∈ F_p and satisfied: $4a^3 + 27b^2 \neq 0$ mod *q*.

The domain parameters here can be generated as specified in ISO/IEC 15946 [14], ANSI X9.62 [15], FIPS 186–4 [11] or GOST R34.10–2012 [16].

The private (secret) key of the signature entity is a point *G* of prime order *q* on the elliptic curve E(F_p). The corresponding public key *P* is:

$$P = (-x_G) \times G \tag{18}$$

The Key Generation algorithm (Algorithm 7) is described as follows:

Algorithm 7

```
Input: $E(\mathbb{F}_p)$.
Output: $G, P$.
[1]. select $G \in E(F_p)$
[2]. $P \leftarrow (-x_G) \times G$
[3]. return $(G, P)$
```

Note: *E*(F_p) is System parameter/domain parameters; *G*, *P* are private and public key of the signer.

3.3.2. The signature generation algorithm: Assuming (R, S) is the signature on the message to be signed M, here: R, S are points on the elliptic curve E(F_p). The first component of the signature R is calculated according to the following formula:

$$R = \left((x_G \times h + k) \times (x_G)^{-k} \bmod q\right).G \tag{19}$$

here the k is a randomly value in the range (1, q) and h is the representative value (hash value) of the message to be signed M: h = H(M).

The second component S of the signature is calculated according to the following formula:

$$S = R + k.G \tag{20}$$

The signature generation algorithm (Algorithm 8) of the proposed scheme is described as follows:

Algorithm 8

```
Input: $E(F_p)$, $G$, $M$.
Output: $(R, S)$.
[1]. select $k$: $1 < k < q$
[2]. $h \leftarrow H(M)$
[3]. $R \leftarrow \left((x_G \times h + k) \times (x_G)^{-\frac{1}{2}} \bmod q\right) G$
[4]. $S \leftarrow R + k \cdot G$
[5]. return $(R, S)$
```

Note: M is message to be signed, with:$M \in \{0, 1\}^{\infty}$; H(.) is hash function, with $H : \{0, 1\}^* \mapsto Z_h$, $q < h < p$.

3.3.3. The Signature Verification algorithm: The Signature Verification algorithm of the scheme is construction on the assumption:

$$(\pi_1(R) \times h).P + (\pi_2(S) + \pi_1(R)).R = \pi_1(R).S \tag{21}$$

That is, if M and the signature (*R*,*S*) satisfy the equality (21), then the signature is considered valid, and the message is verified for origin and integrity. Otherwise, the signature is considered forged, and the message to be verified is denied in terms of origin and integrity.

In (21), $\pi_1()$ and $\pi_2()$ are function that convert a point on the elliptic curve to an integer. With E(F_p) is an elliptic curve defined on F_p and Q , Q_1 , Q_2 are points on E(F_p), then $\pi_1(Q) = x_Q$ and $\pi_2(Q_1 + k.Q_2) = x_{Q_1} \times (x_{Q_2})^k$.

The Signature Verification algorithm (Algorithm 9) of the proposed scheme is described as follows:

Algorithm 9

```
Input: $E(F_p)$, $P$, $M$, $(R,S)$.
Output: TRUE/FALSE.
[1]. $h \leftarrow H(M)$
[2]. $A \leftarrow (\pi_1(R) \times h)P + (\pi_2(S) + \pi_1(R))R$
[3]. $B \leftarrow \pi_1(R) \cdot S$
[4]. if $(A = B)$ then return TRUE else return FALSE
```

Note: M,(*R*,*S*) are message and signature to be verified. If the result is TRUE, then the integrity and origin of M are asserted. Otherwise, if the result is FALSE, then M is denied for origin and integrity.

3.3.4. The correctness of the proposed scheme: What needs to be proved here is: if

$$A = (\pi_1(R) \times h)\,.P + (\pi_2(S) + \pi_1(R))\,.R \tag{22}$$

and

$$B = \pi_1(R).S \tag{23}$$

then: *A* = *B*.

Indeed, if the signature and message to be verified are not forged, from (18),(19), (20) and (22) we will have:

$$\begin{aligned}A &= (\pi_1(R) \times h)\,.P + (\pi_2(S) + \pi_1(R))\,.R \\ &= (\pi_1(R) \times h)\,.P + \pi_2(S).R + \pi_1(R).R \\ &= (x_R \times h)\,.\,(-x_G.G) + \pi_2(R + k.G).R + x_R.R \\ &= -\,(x_R \times x_G \times h)\,.G + \pi_2\,(R + k.G)\,.\left(\left((x_G \times h + k) \times (x_G)^{-k}\right).G\right) + x_R.R \\ &= -\,(x_R \times x_G \times h)\,.G + \left(\left(x_R \times (x_G)^{k}\right) \times \left((x_G \times h + k) \times (x_G)^{-k}\right)\right).G + x_R.G \\ &= -\,(x_R \times x_G \times h)\,.G + (x_R \times x_G \times h)\,.G + (x_R \times k)\,.G + x_R.G \\ &= (x_R \times k)\,.G + x_R.G\end{aligned} \tag{24}$$

From (20) and (23) we get:

$$B = \pi(R).S = x_R.\,(R + k.G) = x_R.R + (x_R \times k)\,.G \tag{25}$$

From (24) and (25) we have: *A* = *B*. Thus, the correctness of the scheme has been proved.

3.3.5. The quantum resistance mechanism of the proposed scheme: The analysis and evaluation of the quantum resistance mechanism as well as the secure of this scheme can be done similarly to the two previous schemes mentioned in Sections 3.1.5 and 3.2.5 with one difference that the hard problems here are established based on elliptic curves instead of on finite fields.

4. Conclusions

In the paper, the authors propose a new type of post–quantum digital signature scheme based on new hard problems. Currently, these hard problems belong to the group of unsolvable problems, which is an important factor that creates the quantum–resistant mechanism of the proposed digital signature schemes. In addition to quantum resistance, the proposed signature schemes here can also be used as pre–quantum digital signature schemes that are widely used in current practical applications, and this is also an important advantage of the schemes proposed here compared with previously proposed post–quantum signature schemes.

References

[1] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS).* IEEE, 1994, pp. 124-134. DOI: 10.1109/SFCS.1994.365700

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," vol. 26, no. 5, 1997, pp. 1484–1509. DOI: 10.1137/S0097539795293172

[3] M. Eker, "Modifying Shor's algorithm to compute short discrete logarithms," iACR ePrint Archive, Report 2016/1128, 2016.

[4] L. C. Washington, *Elliptic curves: Number theory and cryptography*. Chapman & Hall/CRC, 2008.

[5] J. Hoffstein, J. Pipher, and J. H. Silverman, *An introduction to mathematical cryptography*. Springer, 2008. ISBN 978-0-387-77993-5

[6] D. R. Stinson, *Cryptography: Theory and practice*, 3rd ed. Chapman & Hall/CRC, 2006.

[7] J. Talbot and D. Welsh, *Complexity and cryptography: An introduction*. Cambridge University Press, 2006.

[8] I. Shparlinski, *Cryptographic applications of analytic number theory: Complexity lower bounds and pseudorandomness*. Birkhäuser, 2003.

[9] S. S. Wagstaff, *Cryptanalysis of number theoretic ciphers*. Chapman & Hall/CRC, 2003.

[10] *ISO/IEC 14888-3: Information technology – Security techniques – Digital signatures with appendix*, 2nd ed., 2006-11-15, 2006.

[11] *FIPS PUB 186-4: Digital Signature Standard (DSS).* National Institute of Standards and Technology, U.S. Department of Commerce, 2013.

[12] *GOST R 34.10-94: Information Technology – Cryptographic data security – Procedures of electronic digital signature*, Russian Federation Standard, Government Committee of the Russia for Standards, 1994, (in Russian).

[13] *FIPS PUB 180-4: Secure Hash Standard (SHS).* National Institute of Standards and Technology, U.S. Department of Commerce, 2015.

[14] *ISO/IEC 15946: Information technology – Security techniques – Cryptographic techniques based on elliptic curves*, ISO/IEC Standard, 1999.

[15] *ANSI X9.62: Public key cryptography for the financial services industry: Elliptic curve digital signature algorithm (ECDSA)*, American National Standards Institute, 1999.

[16] *GOST R 34.10-2012: Information technology – Digital signature algorithm*, Russian Federation Standard, Government Committee of the Russia for Standards, 2012, (in Russian).

Manuscript received 19-03-2025; Accepted 15-06-2025.

■

Kim Tuan Nguyen received his Master's degree in Information Technology from Hanoi University of Science and Technology and his Ph.D. in Computer Science from Duy Tan University. He is currently a lecturer at the School of Information Technology, Phenikaa University. His research focuses on information security and post-quantum cryptography, with particular interest in their applications in cloud computing, IoT systems, software security, and e-commerce infrastructure, often leveraging machine learning techniques.
E-mail: tuannkim@gmail.com

Minh Duc Tong graduated from Le Quy Don Technical Unversity in 2000. He received PhD in Computer Science from University of Electrical Engineering, Russia, in 2007. Currently, he is a lecturer at Institute of Information and Communication Technology, Le Quy Don Technical Unversity. Research field: image processing, cryptography and information security. E-mail: ductm@lqdtu.edu.vn

Tuan Hao Hoang graduated from Le Quy Don Technical Unversity in 2001. He received PhD in Computer Science from University of New South Wales, 2009. Currently, he is the Senior Lecturer of Information Security Dept., Institute of Information and Communication Technology, LQDTU. His research interests are related to artificial intelligence, evolutionary computation and cyber security. E-mail: haoth@lqdtu.edu.vn

Hong Dung Luu graduated in Electronics and Communications in 1989 and PhD in 2013 at Le Quy Don Technical University. Currently, he is working in the Institute of Information and Communication Technology, Le Quy Don Technical University. Research field: cryptography and information security. E-mail: luuhongdung@lqdtu.edu.vn

MỘT DẠNG LƯỢC ĐỒ CHỮ KÝ HẬU LƯỢNG TỬ MỚI

Nguyễn Kim Tuấn, *Tống Minh Đức*, *Hoàng Tuấn Hảo*, *Lưu Hồng Dũng*

Tóm tắt

Trong bài báo này, các tác giả đề xuất một dạng lược đồ chữ ký hậu lượng tử mới dựa trên các bài toán khó, thuộc nhóm các bài toán không có cách giải ngoài phương pháp "vét cạn". Do đó, các lược đồ chữ ký dạng này có khả năng chống lại tấn công lượng tử dựa trên thuật toán do P. Shor đề xuất. Ngoài khả năng kháng lượng tử, các lược đồ chữ ký này còn có thể sử dụng như các lược đồ chữ ký đang được sử dụng rộng rãi trong các ứng dụng thực tế hiện nay.

Từ khóa

Lược đồ chữ ký số; lược đồ chữ ký số hậu lượng tử; lược đồ chữ ký số kháng lượng tử; bài toán logarithm rời rạc; bài toán khó mới.