Constructing digital signature scheme based on the new hard problem on the elliptic curve

Nguyen Kim Tuan[1], Nguyen Thi Thu Thuy[2], Luu Xuan Van[3], Luu Hong Dung[4*]

[1]Duy Tan University;
[2]Quang Nam College;
[3]People's Security Academy;
[4]Military Technical Academy.
[*]Corresponding author: luuhongdung@mta.edu.vn
Received 11 Sep. 2023; Revised 10 Dec. 2023; Accepted 12 Dec. 2023; Published 30 Dec. 2023
DOI: https://doi.org/10.54939/1859-1043.j.mst.CSCE7.2023.90-**!Invalid Character Setting**

ABSTRACT

In this paper, the authors propose a solution to improve the security of the digital signature scheme, this solution is implemented on two levels of digital signature scheme construction. At the first level, the authors propose a new hard problem, different from the hard problems used before, and importantly, this hard problem belongs to the class of hard problems for which there is currently no solution (except by the "brute force attack" method). At the second level, the authors propose a method to construct new digital signature algorithms based on this hard problem.

Keywords: Digital signature; Digital signature scheme; Discrete logarithm problem; Elliptic curve discrete logarithm problem; Elliptic curve cryptography.

1. INTRODUCTION

In [1], the paper proposed a method for constructing digital signature schemes based on a new hard problem on the prime finite field F_p. In this paper, we continue to propose a new form of that problem on elliptic curves and a method for constructing digital signature schemes based on this new hard problem. With this method, it is possible to generate a family of highly secure digital signature schemes suitable for different choices in practical applications.

2. THE NEW HARD PROBLEM ON THE ELLIPTIC CURVE

2.1. The Elliptic curve discrete logarithm problem (ECDLP)

ECDLP or Discrete Logarithm Problem on Elliptic Curves is described as follows: Suppose G is a point on an elliptic curve E, generating the cyclic group $<G>$. Let point $P \in <G>$. Find the integer k such that:

$$P = k \times G$$

2.2. The new hard problem on the elliptic curve

From ECDLP, we see that if 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 problem on the elliptic curve is stated in the first form as follows:

Form 1: Let $E(F_p)$ 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 a point P in <G>, find point G that satisfies the following equation:

$$P = x_G \times G$$

On the other hand, this hard problem can also be stated in the second form as follows:

Form 2: Let $E(F_p)$ 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 existing algorithms for the ECDLP cannot be used to solve this problem. At present, there is no other solution to this problem other than the "*brute force attack*" method.

In the proposed digital signature scheme construction method, 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 signatures in the Signing algorithm, while the second form of this hard problem is used as the basis for the construction of the Verification algorithm.

3. CONSTRUCTING DIGITAL SIGNATURE SCHEME BASED ON THE NEW HARD PROBLEM ON THE ELLIPTIC CURVE

In this section, the method for constructing a digital signature scheme is presented through the design of a specific signature scheme, including the key generation algorithm, the Signing algorithm, and the Verification algorithm. This scheme is constructed as follows:

3.1. The proposed scheme

3.1.1. 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 the Elliptic curve defined on the finite field F_p by equation:

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

The domain parameters here can be generated as specified in ISO/IEC 15946 [2], ANSI X9.62 [3], FIPS 186 – 4 [4] , or GOST R34.10 – 2012 [5].

The 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:

$$P = x_G \times G \quad (1)$$

The Key generating algorithm (**Algorithm 1.1**) is described as follows:

Algorithm 1.1:

Input: $E(F_p)$.
Output: G, P.

[1]. **Select** $G \in E(F_p)$

[2]. $P \leftarrow x_G \times G$

[3]. **return** (G, P)

Note:

– p, a, b: System parameter/domain parameters;

– G, P: Private and public key of the signer.

3.1.2. The Signing algorithm

Assuming (R,s,Z) is the signature on the message to be signed M, here: R,Z are points on the elliptic curve $E(F_p)$, and s is a value in the range (1,q). Also assumed that the condition that (R,s,Z) needs to be satisfied to make it difficult to forge a signature is:

$$s \times R = e \times Z + \pi\left(Z + s \times R\right) \times P \tag{2}$$

In (2), $\pi()$ is a function that converts a point on the elliptic curve to an integer. With E defined on F_p and Q as a point on E, then $\pi(Q) = x_Q$.

In addition, the parameter $\boldsymbol{e}$ in (2) is generated as follows: $e = H(x_R, M)$, here: $H()$ is the hash function.

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

$$R = k \times G \tag{3}$$

Here, k is a randomly chosen value in the range (1,q).

Also, assume that the component Z is generated from a value u according to the formula:

$$Z = u \times G \tag{4}$$

Here, the u is also randomly chosen in the range (1,q).

The generation of the s component of the signature is done as follows:

From (4), we have:

$$Z + s \times R = u \times G + s \times k \times G = \left(u + k \times s\right) \times G \tag{5}$$

Set:

$$v = (u + k \times s) \bmod q \tag{6}$$

Then (5) will become:

$$Z + s \times R = v \times G = Q \tag{7}$$

From (1), (3), (4) and (7) we have:

$$s \times k \times G = e \times u \times G + \pi\left(Q\right) \times x_G \times G \tag{8}$$

From (8) deduce:

$$s \times k \equiv \left(e \times u + x_Q \times x_G\right) \bmod q \tag{9}$$

On the other hand, from (6) we have:

$$u = \left(v - k \times s\right) \bmod q \tag{10}$$

Substituting (10) into (9), we get:

$$s \times k \equiv \left(e \times \left(v - k \times s\right) + x_Q \times x_G\right) \bmod q \tag{11}$$

From (11) deduce:

$$s = \left(e \times v + x_Q \times x_G\right) \times \left(k \times \left(e + 1\right)\right)^{-1} \bmod q \quad (12)$$

From (10) and (12), the Z is calculated according to:

$$Z = \left(v - k \times s\right) \times G$$

The Signing algorithm (**Algorithm 1.2**) is described as follows:

Algorithm 1.2:

Input: E(F_p),G,M.
Output: R,s,Z).

[1]. **select** k,v: 1 < k, v < q

[2]. $R \leftarrow k \times G$

[3]. $Q \leftarrow v \times G$

[4]. $e \leftarrow H(\, x_R \,||\, M\,)$

[5]. $s \leftarrow \left(e \times v + x_Q \times x_G\right) \times \left(k \times \left(e + 1\right)\right)^{-1} \bmod q$

[6]. $Z \leftarrow \left(v - k \times s\right) \times G$

[7]. **return** (R,s,Z)

Note:

– M: The message to sign, with: $M \in \{0,1\}^{\infty}$;

– (R, s, Z): the signature on M;

– H(): the hash function, with $H : \{0,1\}^{*} \mapsto Z_h$, q<h<p. (eg: SHA-1,SHA-256,... [6]);

– "||": the operator that concatenates of bit strings.

3.1.3. The Verification algorithm

The verification algorithm of the schema is constructed on the assumption:

$$s \times R = e \times Z + \pi\left(Z + s \times R\right) \times P \quad (13)$$

That is, if M and the signature (R,s,Z) 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. Therefore, if the left-hand side of the verification equality is computed as:

$$A = s \times R$$

The right-hand side of the verification equality is:

$$B = e \times Z + \pi\left(Z + s \times R\right) \times P$$

Then the condition for a valid signature is: A = B

The Verification algorithm (**Algorithm 1.3**) of the schema is described as follows:

Algorithm 1.3:

input: E(F_p),P,M,(R,s,Z).
output: TRUE/FALSE.

[1]. $e \leftarrow H(x_R \| M)$

[2]. $A \leftarrow s \times R$ (14)

[3]. $Q \leftarrow Z + s \times R$

[4]. $B \leftarrow e \times Z + x_Q \times P$ (15)

[5]. **if** (A = B) **then return** (TRUE)}
else return (FALSE)

Note:

– M, (R, s, Z): message and signatures 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. The correctness of the proposed signature schema

What needs to be proved here is: if $A = s \times R$ and $B = e \times Z + x_Q \times P$ with $Q = (Z + s \times R)$ then: A = B.

Substituting (3) into (14), we have:

$$A = s \times R = k \times s \times G$$

Similarly, by substituting (1), (3), (4), (7) and (10) into (15), we get:

$$B = e \times Z + x_Q \times P = e \times u \times G + x_Q \times x_G \times G$$

Now, what to prove would be:

$$k \times s \equiv \left(e \times \left(v - k \times s\right) + x_G \times x_Q\right) mod\ q$$

It is equivalent to:

$$k \times s \times \left(e + 1\right) \equiv \left(v \times e + x_G \times x_Q\right) mod\ q$$

Therefore, it can be re-stated what needs to be proved as follows:

If

$$E = k \times s \times \left(e + 1\right) mod\ q \quad (16)$$

and

$$F = \left(v \times e + x_G \times x_Q\right) mod\ q \quad (17)$$

then E = F.

Indeed, by substituting (12) into (16), we get:

$$\begin{aligned} E &= k \times s \times (e+1) \bmod q \\ &= k \times (e+1) \times (e \times v + x_Q \times x_G) \times (k \times (e+1))^{-1} \bmod q \\ &= (e \times v + x_Q \times x_G) \bmod q \end{aligned} \tag{18}$$

From (17) and (18) deduce: E = F.

Thus, the correctness of the schema has been proved.

3.3. Some assessments of the security of the proposed signature scheme

The security of the proposed signature scheme can be evaluated through its ability to resist some types of attacks, such as:

3.3.1. Secret key attack

In the proposed scheme, a secret key attack can be made on the Key generation algorithm (Algorithm 1.1) and steps 2, 3, 5 and 6 of the Signing algorithm (Algorithm 1.2). In step 2 and 3 of the Signing algorithm, since *k* or *v* are also secret parameters, finding *G* from steps 2 and 3 of the Signing algorithm is as difficult as finding *G* from the Key generation algorithm, as it is known this is a type of hard problem that currently there is no solution 7 - 16. In steps 5 and 6 of the Signature algorithm, in addition to x_G being the secret parameter to be found, *k* and *v* are also secret parameters, then finding x_G from steps 5 and 6 of the Signing algorithm is also impossible.

3.3.2. Signature forgery attack

From the Verification algorithm (Algorithm 1.3) of the scheme, a set of 3 components (R,s,Z) will be recognized as a valid signature with the message to be verified M if the condition is satisfied:

$$s \times R = e \times Z + \pi(Z + s \times R) \times P \tag{19}$$

It can be seen that, even if 2 of the 3 components (R,s,Z) of the signature are pre-selected, calculating the remaining component that satisfies condition (19) is essentially the second form of the hard problem mentioned in *section 2*. As we know, this is a type of hard problem for which, in mathematics, there is currently no other solution than the "brute force attack" method [7-16].

3.4. Performance of the proposed signature scheme

The performance of the signature scheme constructed according to the proposed method is basically evaluated by comparing the computational cost of this scheme with the computational cost of the ECDSA digital signature scheme in the US DSS standard [4] and GOST R34.10-2012 of the Russian Federation [5].

The computational cost (or cost) is the number of operations to be performed, where the symbols are defined as follows:

- N_{mp} : The number of multiplications on $E(F_p)$;
- N_{mul}: The number of modulo multiplications;
- N_{inv}: The number of modulo division (inversion);
- N_h : The number of hash operations.

Note:

The algorithm for generating parameters and keys only needs to be done once for every schema. Therefore, the computational cost for the key and parameter generation algorithms can be ignored when comparing the costs of the schemas.

The cost for the Signing algorithms and the Verification algorithms of the ECDSA and GOST R34.10-2012 compared with the proposed scheme is shown in table 1 and table 2 as follows:

Table 1. *The computational cost of the Signing algorithms.*

	N_{mp}	N_{mul}	N_{inv}	N_h
EC DSA	1	2	1	1
GOST R34.10 – 2012	1	2	0	1
The proposed scheme	3	5	1	1

Table 2. *The computational cost of the Verification algorithms.*

	N_{mp}	N_{mul}	N_{inv}	N_h
EC DSA	2	2	1	1
GOST R34.10 – 2012	2	2	1	1
The proposed scheme	4	0	0	1

Comment: Comparing the cost of the proposed scheme with the ECDSA and GOST R34.10-2012, as shown in table 1 and table 2, it shows that the performance of the proposed scheme is lower than that of ECDSA and GOST R34.10-2012. It can be seen that this is the cost of improving the security of the signature schemes constructed according to the proposed method.

4. CONCLUSIONS

In this paper, the authors propose a solution to improve the security of the digital signature scheme based on a new hard problem on the elliptic curve, which is developed from the elliptic curve discrete logarithm problem. Currently, this is a hard problem that belongs to the class of unsolvable problems, which is an important factor allowing to improve the security of the digital signature scheme constructed according to this new solution. From the proposed new solution, it is possible to generate a family of highly secure digital signature schemes suitable for different choices in practical applications. However, to apply the above results in practice, there needs to be a deeper assessment in terms of cryptanalysis, preventing spoofing attacks, improving implementation speed, etc.

REFERENCES

[1]. Nguyen Kim Tuan, Nguyen Vinh Thai, Luu Hong Dung, "*A new construction method of digital signature scheme based on the discrete logarithm combining find root problem on the finite field*", Journal of Military Science and Technology - ISSN 1859-1403. (2022). DOI: 10.54939/1859-1043.j.mst.FEE.2022.164-170

[2]. ISO/IEC 15946: Information technology – Security techniques – Cryptographic Techniques Based on Elliptic Curves, (1999).

[3]. ANSI X9.62. Public Key Cryptography for the Financial Services Industry: Elliptic Cuve Digital Signature Algorithm (ECDSA), (1999).

[4]. National Institute of Standards and Technology, NIST FIPS PUB 186-4. Digital Signature Standard, U.S. Department of Commerce, (2013).
[5]. GOST R34.10 – 2012, Russian Federation Standard Information Technology. Government Committee of the Russia for Standards, (2012) (in Russian).
[6]. Federal Information Processing Standards Publication 180-3 (FIPS PUB 180-3). Secure Hash Standard (SHS), (2008).
[7]. A. Menezes, P. van Oorschot, and S. Vanstone. *"Handbook of Applied Cryptography"*. CRC Press, (1996).
[8]. J. Katz, Y. Lindell. *"Introduction to Modern Cryptography"*. Chapman & Hall/CRC (2008).
[9]. Jeffrey Hoffstein, Jill Pipher and Joseph H. Silverman. *"An Introduction to Mathematical Cryptography"*. ISBN 978-0-387-77993-5. Springer - Verlag (2008).
[10]. L. C. Washington. *"Elliptic Curves. Number Theory and Cryptography"*. Chapman & Hall/CRC (2008).
[11]. D. R. Stinson. *"Cryptography. Theory and Practice"*. Chapman & Hall/CRC (2006).
[12]. J. Talbot and D. Welsh. *"Complexity and Cryptography: An Introduction"*. Cambridge University Press, (2006).
[13]. J. H. Silverman. *"Elliptic curves and cryptography"*. In Public-Key Cryptography, volume 62 of Proc. Sympos. Appl. Math., pages 91–112. Amer. Math. Soc., Providence, RI, (2005).
[14]. I. Shparlinski. *"Cryptographic Applications of Analytic Number Theory"*. Complexity Lower Bounds and Pseurandomness. Birkhäuser (2003).
[15]. I. F. Blake, G. Seroussi, and N. P. Smart. *"Elliptic Curves in Cryptography",* volume 265 of London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge, (2000).
[16]. I. Blake, G.Seroussi & N. Smart. *"Elliptic Curves in Cryptography"*. Cambridge University Press (2000).

TÓM TẮT

Xây dựng lược đồ chữ ký số dựa trên bài toán khó mới trên đường cong elliptic

Trong bài báo này, nhóm tác giả đề xuất giải pháp nâng cao tính an toàn của lược đồ chữ ký số, giải pháp này được triển khai trên cả hai cấp độ xây d ng lược đồ chữ ký số. Ở cấp độ đầu tiên, bài báo đề xuất một bài toán khó mới, khác với các bài toán khó đã sử dụng trước đây và quan trọng là bài toán khó này thuộc lớp các bài toán khó mà hiện tại còn chưa có cách giải. Ở cấp độ thứ hai, bài báo đề xuất một phương pháp xây d ng thuật toán chữ ký số mới d a trên bài toán khó mới này.

Từ khóa: Chữ ký số; Lược đồ chữ ký số; Bài toán logarit rời rạc; Bài toán logarit rời rạc đường cong elliptic; Mật mã đường cong elliptic.