Secure Data Fusion Analysis on Certificateless Short Signature Scheme Based on Integrated Neural Networks and Elliptic Curve Cryptography

Lina Zou¹, Xueying Wang¹ and Lifeng Deng²

¹. Software College, Shenyang Normal University, Shenyang, 110034 China
². Liaoning Rongke Zhiwei Cloud Technology Co., LTD, Shenyang, 110034 China

Abstract

In the traditional public key cryptosystem based on certificates, the issuance and management of user certificates are realized through the authoritative certificate center, but a lot of time is spent in the transmission and verification of user public key certificates. After a malicious user obtaining legitimate users’ private keys, he can select a secret value and signature process to generate the final private key, public key and signature. And he will announce that he is the legal user, while others are unable to distinguish this process. This is the defect of traditional digital signature scheme without certificate. Therefore, this paper proposes a certificateless short signature scheme based on integrated neural networks and elliptic curve cryptography for secure data fusion analysis. The security of the solution is based on Inv-CDH problem. The complete security proof is given under the stochastic predictor model. It is proved that the new model can resist existence forgery in adaptive selective message attack with new adversary. Experiment results show that the calculation amount of our proposed certificateless short signature scheme is small and the efficiency is high compared with other state-of-the-art schemes.

1. Introduction

Messages transmitted through wireless sensor network nodes or users must be verified to become the useful information. The signature algorithm based on Public Key Infrastructure (PKI) provides a guarantee tool for information security [1-3]. The signature scheme needs to provide a certificate issued by an authoritative authority to prove that the public key is corresponding to the user and has not been tampered with or replaced by a third party. Certificate retraction, storage, distribution and verification are managed by an authoritative certification authority. The calculation, communication latency and storage space caused by these operations are unacceptable in wireless sensor networks. Therefore, people use Identity Public Key Cryptography in wireless sensor network to realize message authentication. In identity-based certificate-free cryptography, there is a trusted Key Generation Center (KGC) [4-7], which generates the system master key and generates part of the user’s private Key by using the user’s identity and master Key. It solves the problem of certificate management and key escrow at the same time.

Digital signature plays an important role in public key cryptography. Since the identity-based public key cryptosystem was proposed in reference [8], the PKI certificate management problem in public key cryptosystem had been simplified. In reference [9], a bilinear pair was used to construct a short-signature scheme in the army conference, which was the shortest short-signature scheme in the classical password. The certificateless public key cryptosystem solved the key escrow problem of identity-based public
key cryptosystem. Certificateless short signature draw
the advantages of certificateless signature and short
signature and widely used in the field of electronic
payment and e-commerce. In recent years, scholars
have studied more certificateless signature schemes.
Huang [10] proposed an efficient certificate-free
signature scheme, which did not need pairing operation
to improve the operational efficiency of the scheme.
Dong [11] proposed an improved certificate-free
signature scheme, which used the secret value selected
by the user as the signature private key to improve
the security of the scheme. He [12] proposed a
certificateless short signature scheme that could prove
security, which did not use Hash function mapping.
Pang [13] presented a certificateless short signature
scheme under the standard model, which only needed
one bilinear pair operation. Zuo [14] proposed a
strongly provable secure certificate-free short signature
scheme, which could resist public-key substitution
signature scheme based on bilinear pairings, which
could resist public key replacement adversary attacks.
Islam [16] proposed an efficient short signature scheme
based on certificate, which improved the operation
efficiency by reducing double-line pair operation. Wang
[17] proposed an efficient certificateless short signature
scheme based on bilinear pairings, and gave the security
proof of the scheme. Liu [18] presented an efficient and
provably secure certificateless short signature scheme,
which could resist two types of super attacks and existential
forgery attacks. Dan [19] proposed an irrevocable short
signature scheme without certificate, which had strong
unforgery against adaptive selective message attack.
Sahu [20] proposed a certificate-safe and efficient
certificate-free signature scheme, which proved its
unfalsifiability based on the difficulty of discrete
logarithm. Liu [21] proposed a certificateless group
signature scheme based on bilinear pairings, which had
the advantages of certificateless cryptography and met
the requirements of group signature scheme.
However, in the certificate-free public key cryptog-
raphy system, the public key is not bound to the
user’s identity, so there is no authentication relationship
between the public key and the holder. In this paper,
our motivation is that we modify the definition of
certificateless signature, and propose a certificateless
short signature scheme.
This paper is organized as follows. After some
preliminary works, Section 3 detailed introduces the
integrated neural networks for feature extraction of
certificateless signature. Section 4 presents our new
certificateless signature scheme. Section 5 and section
6 analyze our proposed scheme from performance and
security points of view. In Section 7, the paper ends with
some concluding remarks.

2. Preliminaries

**Definition 1.** Assuming $G_1$ is the $q$-order additive cyclic
group. $G_2$ is the $q$-factorial cyclic group. $Z_q^*$ is the
non-zero modular. Bilinear pair [22] is defined as the
following mapping:

$$
e: G_1 \times G_2 \rightarrow G_2.
$$

This mapping satisfies the following three properties:
- **Bilinear.** There is $P, Q \in G_1$ and $a, b \in Z_q^*$;
- **Non-degeneracy.** There is $P, Q \in G_1$, and
  $e(P, Q) \neq 1$;
- **Computability.** For all $P, Q \in G_1$, there is an
effective algorithm to calculate $e(P, Q)$. 

**Definition 2.** Elliptic Curve Discrete Logarithm
Problem (ECDLP) [23]. Given two elements $P, Q \in G_1$,
and the integer $a \in Z_q^*$, so that $Q = aP$ is established.

**Definition 3.** Inv-CDH problem. Given $b \in Z_q^*$, $P,
aP \in G_1$, ($a \in Z_q^*$ is an unknown random number), to
calculate $(a + b)^{-1}P \in G_1$.

3. Integrated Neural Networks (INN)

3.1. Architecture of INN

The multi-classification integrated neural network
system constructed in this paper is an organic whole,
each sub-network is independent of each other, but also
cooperates with each other as shown in figure 1.

Here, the realization of neural network subject is
divided into two parts, one is how to train the
network, the other is how to perform classification. The
specific realization process is as follows. The training
process of neural network is the learning process. The training set consists of two parts [24,25]. The authentication training set includes the same type of real signatures and forged short signatures to enhance the sensitivity of the neural network to the same type of real signatures. The recognition training set consists of the real signatures of this category and the real signatures of other categories randomly selected in a certain proportion. Neural networks mainly learn about the differences between different categories. For ease of use and administration, it creates a file for each subnetwork. The archive consists of two parts. One part records the structural characteristics of the network and the meaning of the input and output units. The other part contains the weights and accuracy learned for the two training set networks. Since an independent classifier is built for each person's signature sample, and a new classification sub-network is added to the recognition network body and trained without retraining the whole integrated network.

Sub-networks can begin to perform classification when their archives and knowledge are sound. The feature vectors transmitted by the feature assignment network and the weights learned by the neural network are calculated to score the signature categories independently. The scoring results of the three neural networks are sent to the decision fusion sub-network. For decision fusion with D-S evidence theory fusion, and the confidence degree of the corresponding categories is obtained. The fusion rules are as follows;

**Theorem 1.** $\Theta$ is an identification framework. For $n$ evidences $E_1, E_2, \ldots, E_n \subset \Theta$, the corresponding basic probability allocation is $M_1, M_2, \ldots, M_n$, then the obtained combined evidence after the combination of the $n$ evidences is:

1. $M(\phi) = 0$.  
2. $M(A) = K \cdot M_1 \oplus M_2 \oplus \cdots \oplus M_n = \sum_A M_1(A_1) \cdot M_n(A_{in})$.

$K$ reflects the degree of conflict between evidences, which is called conflict probability. The coefficient $1/(1-K)$ is called the normalization factor.

3.2. Implementation of decision fusion subnetwork

For the decision fusion sub-network $i$, let the score of neural network $NN_{ij}$ be $score_{ij}$, and the accuracy is $r_j$. Recognition framework $D = sort_1, \ldots, sort_m$ belongs to category $i$. $\neg sort_i$ does not belong to category $i$.

Probability distribution function is $M_j : 2^{D_i} \rightarrow [0,1]$, and satisfies the $M_j(\neg sort_i, D_j, \phi) = (m_j, n_j, 1-m_j-n_j, 0)$. Here, $m_j = score_j \cdot r_j, n_j = (1-score_j) \cdot r_j, i \in \{1,\ldots,m\}, j \in \{1,2,3\}$.

So the problem of finding the confidence of class $i$ is transformed into finding $M_i = M_1 \oplus M_2 \oplus M_3$. In signature authentication, the confidence of class $i$ is the possibility that the signature is a real signature, if $M_i > 0.5$, it is a real signature; otherwise, it is a forged signature.

3.3. Implementation of decision fusion recognition network

Let the classification vector formed by the recognition network be $sort = sort_1, \ldots, sort_m$. The confidence vector is $T = T_1, \ldots, T_m$. The confidence weight vector is $R = R_1, \ldots, R_m$. For decision fusion identification network, identification framework $D = sort_1, \ldots, sort_m, \neg sort_1, \ldots, \neg sort_m$. The probability assignment function is $M_i : 2^D \rightarrow [0,1]$ and satisfies: $M_i(sort_1, sort_2, \ldots, sort_m, \neg sort_1, \neg sort_2, \ldots, \neg sort_m, D, \phi) = (0, 0, \ldots, m_i, 0, 0, \ldots, n_i, 1 - m_i - n_i, 0)$.

The output of the converged network is:

$$M = M_1 \oplus M_2 \oplus \cdots \oplus M_m.$$ (2)

Where, the probability that the signature sample belongs to the i-th category is $M(sort_i)$.

If $M(D) < \max M(sort_1, M(sort_2), \ldots, M(sort_m)$, then the signature sample belongs to the category with the largest probability.

If $M(D) \geq \max M(sort_1, M(sort_2), \ldots, M(sort_m)$, then the signature sample is rejected (the signature sample is not in the known category of neural network learning).

4. Proposed Certificateless Signature Scheme

1. **Setup.** Key Generation Center (KGC) sets parameter $k$ to generate system public parameter $params$ and system master key $s$. Build a certificateless system. KGC secretly stores $s$ and publishes $params$.

2. **ssv.** Set a secret value. Given the user's identity $ID$, the Private Key Generator (PKG). It uses the system parameter $params$ and ID to generate the user's secret value $x_{ID}$ and calculate the generated user part public $y_{ID} = x_{ID}P$.

3. **Extract Partial Private Key.** Send the system parameters $params$, the system master key and the user ID to KGC, which generates part of the private key $d_{ID}$. Then it sends part private keys to the corresponding user through the secure channel.

4. **SPK.** Set Private Key Creation. The user uses the system parameter $params$, the user's partial private key $d_{ID}$ and the secret value $x_{ID}$ to generate private key $sk_{ID}$. 

EAI Endorsed Transactions  
Scalable Information Systems  
10 2021 - 01 2022 | Volume 9 | Issue 34 | e3
5. Set Public Key. The user generates the user’s public key \( pk_{ID} \) through \( params \) and the user’s secret value \( x_{ID} \) and exposes the public key. The public key space is defined by the system public parameter \( params \) and the user’s identity \( ID \).

6. Sign. Signature. Given the system parameter \( params \), signature information \( m \), user \( ID \), public key \( pk_{ID} \) and private key \( sk_{ID} \). The signature algorithm is executed to generate signature \( S \).

7. \( sv \). Signature verify. Given the system parameter \( params \), the signer’s identity \( ID \), public key \( pk_{ID} \), message \( m \) and signature \( S \), verify the signature \( S \). If it returns 1, then it indicates that the signature is valid. If it returns 0, then it indicates that the signature is invalid.

4.1. Attacker model

The traditional certificateless cryptography mainly discusses two adversary types. Type 1: adversary is dishonest user. Type 2 adversary is malicious but passive KGC. Their specific capabilities are as follows:

1. Type 1. Adversary \( A_1 \) does not know the master key and the user’s partial private key. It can replace the user’s public key.

2. Type 2. Adversary \( A_2 \) knows the system master key and the user’s partial private key. It cannot replace the user’s public key.

In the scheme, part user’s private keys are bound to part users’ public keys and users’ \( ID \), respectively. There is an authentication relationship between the public key and the holder, so that the user’s public key cannot be replaced by the type 1 adversary. That is, there is no type 1 adversary [26,27]. However, the above reasons cannot completely exclude the type 2 adversary. In the actual situation, it is considered that KGC is not necessarily malicious, that is, the master key of the system will not be disclosed. However, it may leak users’ private keys during key management or key transmission. The users may also have the possibility of disclosure when using part of the private keys. However, the attack mode of type 2 adversary is malicious KGC leaking the system master key. Therefore, this paper no longer considers type 2 adversary and proposes two new adversaries.

1. Type 3. Adversary \( A_3 \) does not hold the system master key but knows part of the private key. It cannot replace the user’s public key.

2. Type 4. Adversary \( A_4 \) holds the system master key but does not know part of the private key. It can replace the user’s public key.

4.2. Certificateless Short Signature Scheme

The scheme contains seven steps as follows:

1. System establishment. Set security parameter \( k \), q-order addition cyclic group \( G_1 \) and q-order multiplication cyclic group \( G_2 \). \( q \) is prime and \( q > 2^k \). Given a bilinear pair \( e : G_1 \times G_1 \rightarrow G_2 \), \( P \) is the generator of \( G_1 \). Let \( g = e(P,P) \). KGC selects two different security hash functions: \( H_1 : 0,1^* \rightarrow Z_q^* \) and \( H_2 : 0,1^* \times G_1 \rightarrow Z_q^* \). Randomly choose a number \( s = Z_q^* \) as the system master key, system public key \( y_{pub} = sP \in G_1 \). KGC secretly saves \( s \) and publishes system parameters \( k, G_1, G_2, e, q, P, g, y_{pub}, H_1, H_2 \).

2. Secret value establishment. The user \( ID \) randomly selects \( x_{ID} = Z_q^* \) as its secret value and calculates part of the user’s public key \( y_{ID} = x_{ID}P \in G_1 \).

3. Partial private key extraction. Given \( ID \in \{0,1\}^* \), KGC calculates \( Q_{ID} = H_1(ID, y_{ID}) \), \( k = H_1(ID, timestamp) \) and then calculates partial private key \( d_{ID} = \frac{k}{sQ_{ID}}P \). Let \( K = kP \), \( k \) is as the authorization identification code of partial private key application. Where \( timestamp \) is the time of partial private key application. \( (k, timestamp) \) is used to distinguish partial private keys applied at different times, which is saved by KGC. It can be used to broadcast to revoke part of the leaked private key. Finally, KGC is sent to the user through the secure channel (\( d_{ID}, K \)).

4. Private key establishment. Given the user’s partial private key \( d_{ID} \), secret value \( x_{ID} \) and public parameter \( params \). User’s private key is \( (d_{ID}, x_{ID}) \).

5. Public key establishment. Known user’s secret value \( x_{ID} \), parameter \( params \), generate partial private key \( pk_{ID} = x_{ID}y_{pub} + Q_{ID}y_{ID} \). Where the user’s public key is \( (y_{ID}, pk_{ID}, K) \) and user exposes the user’s public key.

6. Signature. Known message \( m \in \{0,1\}^* \). The user signs the message \( m \). The steps to get the signature are as follows:

   (a) Computing \( h_{1ID} = H_2(ID, m, pk_{ID}) \).
   (b) Computing \( S = \frac{1}{x_{ID}+h_{1ID}}d_{1ID} \).

7. Signature verification. Known \( m, S \). The verification steps are as follows:

   - Computing \( Q_{ID} = H_1(ID, y_{ID}) \).
   - Computing \( h_{1ID} = H_2(ID, m, pk_{ID}) \).
   - If \( e(S, pk_{ID} + h_{1ID}y_{pub} + Q_{ID}P) = e(K, P) \) is correct, then signature verification is successful. Otherwise, signature verification is failed.
The correctness of the scheme is proved as follows:

\[
\text{Proof} = e(S, pk_{ID} + h_{ID}(y_{pub} + Q_{ID}P)) \\
= e(S, x_{ID}y_{pub} + Q_{ID}y_{ID} + h_{ID}(s + Q_{ID}P)) \\
= e(S, x_{ID}(s + Q_{ID}P) + h_{ID}(s + Q_{ID}P)) \\
= e(S, (x_{ID} + h_{ID})(s + Q_{ID}P)) \\
= e((x_{ID} + h_{ID})(s + Q_{ID}P), (x_{ID} + h_{ID})(s + Q_{ID}P)) \\
= e(K, P)
\]

(3)

5. Security Analysis

Many references had proved the type 1 and type 2. This paper only gives the proof for type 1, type 2, type 3 adversary in the random predictor model. The proof for type 4 adversary is basically similar to type 3. It will not give the detailed proof in this paper.

Theorem. Let \( A_1 \) be the type 1 attacker. Given \( C \) an instance \((g, g^2, g^3, \ldots, g^{2^{q-1}})\). \( C \) can obtain a new \((c, g^\pi)\). Obviously, for any polynomial \( h(t) = \sum_{i=0}^{n} b_i t^i (n \leq q_s + 1) \), \( C \) can calculate \( g^{h(a)} = g^{\sum_{i=0}^{n} b_i a^i} = \prod_{i=0}^{n} (g^{b_i})^{a^i} \).

Proof. \( C \) randomly selects \((c_1, c_2, \ldots, c_q) \in Z_q^*\). Let \( f(t) = \prod_{i=1}^{q_s} (t + c_i) \), then \( f(t) = \sum_{i=0}^{q_s} b_i t^i \). Calculating \( u(t) = f(t) \sum_{i=0}^{q_s} b_i t^{i+1} \), so \( f(t) \) and \( u(t) \) are polynomials.

- Calculating \( h = g^{f(a)} \) and \( h^a = g^{u(a)} \), so \( h \) is the generator of \( G_1 \) with q-order.
- If \( h = 1 \), then \( c_i = -a \) can solve Inv-CDH problem, this probability is negligible. Therefore, \( c_i \neq -a \).
- For \( i = 1, 2, \ldots, q_s \), calculating \( f_i(t) = \frac{f(t)}{r_{ci}} = \prod_{j=1, j \neq i}^{q_s} (t + c_i) = \sum_{i=0}^{q_s} d_i t^i \). So \( h^{\frac{1}{r_{ci}}} = (g^{f_i(a)})^{\frac{1}{r_{ci}}} = g^{f_i(a)} \).

Then \( h^{\frac{1}{r_{ci}}} = (g^{f_i(a)})^{\frac{1}{r_{ci}}} = g^{f_i(a)} \). Thereby, it outputs \((c_i, h^{\frac{1}{r_{ci}}})\).

C will execute \text{setup} \text{ algorithm} and generate system parameter \( params = G_1, G_2, g, h, y, e, H \). Here, the main public key is \( y = h^x \). \( C \) returns \( params \) to \( A_1 \) and executes the following simulation algorithm.

1. Generating user request. \( C \) randomly selects \( i \in 1, 2, \ldots, q_{CU} \). \( ID' = ID \). For the \( j \)-th request of \( A_1 \), if \( j \neq i \), then \( C \) randomly selects \( z_j, s_j \in Z_q^* \) and computes \( u_j = h^{z_j}, w_j = h^{s_j}, d_j = s_j + xH(ID_j, u_j, w_j) \). \((ID_j, z_j, u_j, s_j, w_j, d_j) \) will be added into the table \( E \). If \( j = i \), then \( C \) randomly selects \( z^* \in Z_q^* \) and calculates \( u^* = h^{z^*} \). Let \( w^* = h^{t} \). \((ID', z^*, u^*, w^*) \) will be added into table \( E \).

2. Private key extraction query. \( A_1 \) queries the part of the private key corresponding to \( ID_j \). If \( ID_j = ID' \), then \( C \) outputs "stop", then the simulation is failed. Otherwise, \( C \) queries table \( E \) and returns the private key \( d_i \) corresponding to identity \( ID_j \).

3. Secret value query. \( A_1 \) queries the secret value corresponding to \( ID_j \). If \( ID_j = ID' \), then \( C \) outputs "stop", then the simulation is failed. Otherwise, \( C \) queries table \( E \) and returns the secret value \( z_i \) corresponding to identity \( ID_j \).

4. Public key query. \( A_1 \) queries the public key corresponding to \( ID_j \). \( C \) queries table \( E \) and returns the public key \((u_i, w_i) \) corresponding to identity \( ID_j \).

5. Public key replacement request. For public key replacement request \( ID_j, u'_i, w'_i \) of \( A_1 \), \( C \) uses \((u'_i, w'_i) \) to replace \((u_i, w_i) \).

6. Signature query. Assuming that \( A_1 \) makes signature query \((ID_j, m, \sigma) \), if \( ID_j = ID' \), the algorithm is failed. Otherwise, \( C \) computes

\[
c' = m' + xH(ID', u', w') + z' r'.
\]

(4)

It can be seen that \((c', \sigma') \) satisfies \( \sigma' = h^{\frac{1}{r_{ci}}} \). Obviously, the probability that \( A_1 \) does not query the private key corresponding to \( ID_j \) is at least \((1 - (1/q_{CU}))^{q_{p}} \). The probability that \( A_1 \) does not query the secret value is at least \((1 - (1/q_{CU}))(\frac{1}{q_{p}}) \). The probability that \( ID_j = ID' \) in a forged signature \((ID_j, m', r', \sigma') \) is at least \( 1/q_{CU} \). In the signature query, the probability that the corresponding public key has not been replaced is \( (1 - q_{p}/q_{CU})^{q_{p}} \). If \( A_1 \) can successfully forge a valid signature with probability \( \epsilon \), then \( C \) solves the Inv-CDH problem with probability \( \epsilon' = \epsilon - (1 - 1/q_{CU})^{q_{p}+q_{p}}(1 - q_{p}/q_{CU})^{q_{p}} \).
\[ \frac{q_{H}}{q_{X}} = \epsilon. \] According to the difficulty of Inv-CDH problem, \( \epsilon \) is negligible. Therefore, the scheme is unforgeable under type \( A_{1} \) attack.

**Theorem.** In the random predictor model under the Inv-CDH assumption problem, for the adaptive selective message attack of type 3 adversary, the proposed scheme can resist existential forgery.

**Lemma.** Assume that type 3 adversary \( A_{3} \), after finite inquiries, it breaks the scheme in polynomial time \( t \) with a non-negligible advantage \( \epsilon \). \( q_{X} \) and \( t_{X} \) are secret value query number and one query time, respectively. \( q_{Y} \) and \( t_{Y} \) are part of the public inquiry number and one query time, respectively. \( q_{E} \) is the number of times that adversary \( A_{3} \) first queries the predictor in the partial private key extraction stage. \( t_{E} \) is the one query time. \( q_{H_{1}} \) is the number of times that adversary \( A_{3} \) second queries the predictor in the partial private key extraction stage. \( t_{H_{1}} \) is the one query time. \( q_{H_{2}} \) is the number of times that adversary \( A_{3} \) queries the predictor. \( t_{H_{2}} \) is the one query time. \( q_{E} \) is number of private key parsing queries. \( t_{E} \) is one query time. \( q_{pk} \) is the number of public key queries. \( t_{pk} \) is the one query time. \( q_{s} \) is the number of signature queries. \( t_{s} \) is the one query time. So there is an algorithm \( C \), which can solve Inv-CDH problem with a non-negligible advantage \( \epsilon' \) in time \( t' \).

\[
\begin{align*}
    t' &< t + (q_{X}t_{X} + q_{Y}t_{Y} + q_{E}t_{E}) \\
    &+ q_{t}t_{s} + q_{pk}t_{pk} + 2q_{H_{1}}t_{H_{1}} \\
    &+ 2q_{H_{2}}t_{H_{2}} + 2q_{E}t_{E} \\
    \epsilon' &\geq (\epsilon - \frac{1}{2^{t_{s}}})(1 - \frac{1}{q_{X}})(1 - \frac{1}{q_{s}}).
\end{align*}
\]  

**Proof.** Suppose the Inv-CDH problem instance of challenge \( C \) is that given \( b \in Z_{q}^{*} \) and \( (P, aP) \in G_{1} \), where \( a \in Z_{q}^{*} \) is unknown to calculate \( \frac{1}{aP} \).

Set security parameter \( k \) and \( C \) for system initialization, select random number \( s \in Z_{q}^{*} \) as the system master key, \( y_{pub} = sP \). \( C \) selects identity \( ID' \) as the challenge identity, sends \( (k, G_{1}, G_{2}, P, y_{pub}, H_{1}, H_{2}) \) to \( A_{3} \). Assume that \( A_{3} \) cannot do the same query. The corresponding \( H_{1} \) and \( H_{2} \) predictions have been made before private key query, public key query, signature query and forged signature. All record lists are initialized empty.

- Secret value inquiry. \( C \) maintains a list \( L \) and records structure as an array \((ID, x, y, \cdot) \). When \( A_{3} \) submits a secret value query about \( ID \):
  1. When \( ID = ID' \), \( C \) terminates the simulation and prints "FALSE" to mark the event as \( E_{1} \).
  2. When \( ID \neq ID' \), query the list \( L \). If \( L \) has a record, then it returns the corresponding record \( x_{ID} \) to \( A_{3} \); Otherwise, it randomly selects \( x_{ID} \in Z_{q}^{*} \) to calculate \( y_{ID} = x_{ID}P \), return \( x_{ID} \) to \( A_{3} \), and add \((ID, x_{ID}, y_{ID})\) into \( L \).

- Partial public key query. When \( A_{3} \) submits a partial public key query about \( ID \):
  1. When \( ID = ID' \), \( C \) returns \( y_{ID} = aP \) to \( A_{3} \), and adds \((ID, \bot, aP)\) to list \( L \), where \( \bot \) means null.
  2. When \( ID \neq ID' \), \( C \) queries the list \( L \), and returns the \( y_{ID} \) of the corresponding record to \( A_{3} \), if \( L \) has records; Otherwise, it performs the secret value query first and returns the corresponding \( y_{ID} \) to \( A_{3} \).

- The first \( H_{1} \) query of partial private key extraction stage. \( C \) maintains a list \( LH_{1} \). This list is composed of \( ID, timestamp, k_{i} \). When \( A_{3} \) submits a \( H_{1} \) query about \((ID, timestamp, k)\), if \( (ID, timestamp \leq 1) \) is already in \( LH_{1} \), \( C \) returns \( k_{ID} \) to \( A_{3} \); Otherwise, it selects a random value \( k_{ID} \), returns \( k_{ID} \) to \( A_{3} \) and records \((ID, timestamp, k_{ID})\) to list \( LH_{1} \).

- The second \( H_{1} \) query of partial private key extraction stage. \( C \) maintains a list \( LH_{1} \). This list is composed of \( ID, timestamp, k_{i} \). When \( A_{3} \) submits a \( H_{1} \) query about \((ID, timestamp, k)\), if \( (ID, timestamp \leq 1) \) is already in \( LH_{1} \), \( C \) returns \( k_{ID} \) to \( A_{3} \); Otherwise, it selects a random value \( k_{ID} \), returns \( k_{ID} \) to \( A_{3} \) and records \((ID, timestamp, k_{ID})\) to list \( LH_{1} \).

- Partial private key query. When \( A_{3} \) submits a partial private key query about identity \( ID \), \( C \) first executes \( H_{1} \) predictor query to get array \((ID, y_{ID}, k_{ID})\). Then it executes \( H_{1} \) again and obtains array \((ID, timestamp, k_{ID})\), and returns \( d_{ID} \) to \( A_{3} \).

\[
d_{ID} = \frac{k_{ID}}{x_{ID} + QID}P. \tag{7}
\]

- Public key query. \( C \) maintains list \( L_{pk} \) and records structure as an array \((ID, y_{ID}, QID, pk_{ID}, x_{ID}) \). When \( A_{3} \) submits a public key query about identity \( ID \), \( C \) checks whether the query value already exists in the list, and returns the corresponding value \((y_{ID}, pk_{ID})\) to \( A_{3} \). Otherwise, the following operation is performed:
  1. When \( ID = ID' \), \( C \) finds \((ID, y_{ID}, QID)\) in \( LH_{1} \), returns \( pk_{ID} = aP \) to \( A_{3} \), and adds \((ID, y_{ID}, QID, pk_{ID}, x_{ID})\) to list \( L_{pk} \).

\[
pk_{ID} = x_{ID}y_{pub} + QIDy_{ID}. \tag{8}
\]
2. When $ID \neq ID^*$, C first queries the secret value to get the corresponding answer $(ID, x_{ID}, y_{ID})$, then executes $H_1$ query to get the array $(ID, y_{ID}, Q_{ID})$, returns $(y_{ID}, pk_{ID})$ to $A_3$. It records $(ID, y_{ID}, Q_{ID}, pk_{ID}, x_{ID})$ into list $L_{pk}$.

\[ pk_{ID} = x_{ID}y_{pub} + Q_{ID}y_{ID}. \] (9)

- $H_2$ query. $C$ maintains a list $LH_2$, records structure as an array $ID, m, Q, k, pk, h$. When $A_3$ submits a $H_2$ query about $(ID, m_{ID}, pk_{ID})$, $C$ checks whether the query value already exists in the list, and returns the corresponding value $(h_{ID})$ to $A_3$. Otherwise, the following operation is performed:

1. When $ID = ID^*$, $C$ regards $b$ as the value of $H_2(ID, m_{ID}, pk_{ID})$ and returns $b$ to $A_3$. $(ID, m_{ID}, Q_{ID}, k_{ID}, pk_{ID}, b)$ is added to list $LH_2$.

2. When $ID \neq ID^*$, $C$ randomly selects $h_{ID}$, regards $h_{ID}$ as the value of $H_2(ID, m_{ID}, pk_{ID})$ and returns $h_{ID}$ to $A_3$. $(ID, m_{ID}, Q_{ID}, k_{ID}, pk_{ID}, b)$ is added to list $LH_2$.

- Signature query. When $A_3$ submits the signature query of $(ID, m_{ID})$, $C$ performs the following operations:

1. When $ID = ID^*$, it stops the query and returns "FALSE", records the event as $E_2$.

2. When $ID \neq ID^*$, $C$ obtains the record $(ID, x_{ID}, y_{ID})$ from $L$. Then it obtains the record $(ID, m_{ID}, Q_{ID}, k_{ID}, pk_{ID}, h_{ID})$ from $LH_2$, and obtains the signature $S_{ID}$ of $C$ to message $m_{ID}$ through calculation:

\[ S_{ID} = \frac{1}{x_{ID} + h_{ID}}d_{ID}k_{ID} \]

\[ = \frac{k_{ID}}{(x_{ID} + h_{ID})(s + Q_{ID})}. \] (10)

Finally, $A_3$ stops query and outputs a valid message signature pair $(m_{ID}, S_{ID})$ about $ID^*$. $C$ calls the array $(ID^*, y_{ID}, Q_{ID}, pk_{ID}, x_{ID})$ and $(ID^*, m^*, Q_{ID}, k_{ID}, pk_{ID}, h_{ID^*})$ respectively. Meanwhile, $h_{ID^*} = b$ and $y_{ID^*} = aP$. According to the verification equation:

\[ E = e(S_{ID^*}, pk_{ID^*} + h_{ID^*}(y_{pub} + Q_{ID^*}P)) \]

\[ = e(S_{ID^*}, x_{ID^*}y_{pub} + Q_{ID^*}y_{ID^*} + h_{ID^*}(s + Q_{ID^*}P)) \]

\[ = e(S_{ID^*}, (x_{ID^*}y_{pub} + Q_{ID^*}y_{ID^*} + h_{ID^*}(s + Q_{ID^*}P))) \]

\[ = e((a + b)(s + Q_{ID^*})P) \]

\[ = (a + b)(s + Q_{ID^*})P \]

$C$ can successfully calculates $\frac{1}{x_{ID^*}}P = k_{ID^*}^{-1}(s + Q_{ID^*})S_{ID^*}$. that is, $\pi$ it outputs $k_{ID^*}^{-1}(s + Q_{ID^*})S_{ID^*}$, as the answer for the Inv-CDH problem, so $C$ solves the Inv-CDH problem.

The following analysis shows the $C$’s time and advantages in successfully solving difficult problems:

- The answers for the query of $H_1$, $H_2$ are evenly and independently distributed in $Z_q^*$, and the answers are valid.

- Only when events $E_1$ and $E_2$ do not occur, the answers obtained by the private key query and the signature predictor query are valid.

- If $E_1$ and $E_2$ do not occur, $C$ can solve an instance of Inv-CDH problem, the probability of $E_1$ and $E_2$ neither occurring:

\[ Pr(\neg E_1 \land \neg E_2) = (1 - \frac{1}{q_X})(1 - \frac{1}{q_s}). \] (12)

When $A_3$ forges a valid signature without query $H_2$, there is a loophole in this simulation. The occurrence probability is $\frac{1}{2^t}$, so the advantage in this game is:

\[ \epsilon' = (\epsilon - \frac{1}{2^t})(1 - \frac{1}{q_X})(1 - \frac{1}{q_s}). \] (13)

Running time is:

\[ t' < t + (q_Xt_X + q_Yt_Y + q_{\text{el}}t_{\text{el}}) \]

\[ + q_{\text{el}}t_{\text{el}} + q_{\text{pk}}t_{\text{pk}} + 2q_{H_{11}}t_{H_{11}} \]

\[ + 2q_{H_{12}}t_{H_{12}} + 2q_{H_{2}}t_{H_{2}} \] (14)

6. Performance Analysis

we first give the unforgeability analysis. This paper proposes that the scheme cannot be forged under adaptive selection message attack. The security analysis of A-I and A-II forgery attacks is given below.

1) For A-I attackers. This type of attacker cannot obtain the system master key $s$, but it can replace the public key of a legitimate user. Assuming that the A-I attacker replaces the public key $PK_n = (X_n, R_n)$ of the valid user $ID_n$ with $PK_n^* = (X_n^*, R_n^*)$, $EAI$
and uses the replaced public key to successfully forge a signature \((R, v)\) for message \(M\), then based on the signature verification algorithm, it calculates \(h = H_2(ID_π, M, R_{π}', X_{π}')\), \(h_1 = H_1(ID_π, R_{π}', X_{π}')\), and the signature verification equation \(vP = R + h(R_{π}' + h_1P_{pub} + X_{π}')\) is correct. Because \(R, R_{π}'\) and \(X_{π}'\) participate in the computation of \(h = H_2(ID_π, M, R_{π}', X_{π}'),\) so there will be a \(vP = r + hr_{π}' + hh_1s + hx_{π}'\), where \(R = rP\), \(X_{π}' = x_{π}'P\), \(R_{π}' = r_{π}'P\), so it can launch \(s = (v - r - hr_{π}' - hx_{π}')/hh_1\). That is, the master key \(s\) can be calculated by \(P_{pub} = sP\) with A-I attacker, thus solving the ECDLP problem. However, ECDLP is a difficult problem that cannot be solved in the real world at present, so the counterfeiting attacks of A-I cannot be successful.

1) For A-II attackers. This type of attacker can obtain the system master key \(s\), but it cannot replace the public key of a legitimate user. Assuming that the A-I attacker replaces the public key \(PK_{π} = (X_{π}', R_{π}')\) of the valid user \(ID_π\) successfully to forge a signature \((R, v)\) for message \(M\). Then based on the signature verification algorithm, it calculates \(h' = H_2(ID_π, M, R_{π}, X_{π}), h_1 = H_1(ID_π, R_{π}, X_{π}),\) and the signature verification equation \(vP = R + h'R_{π} + h_1P_{pub} + X_{π}\) is correct. The true signature output value of user \(ID_π\) for message \(M\) is \((R, v)\). According to the improved signature verification algorithm, it can get \(h' = H_2(ID_π, M, R_{π}, X_{π}), h_1 = H_1(ID_π, R_{π}, X_{π}),\) \(vP = R + h'R_{π} + h_1P_{pub} + X_{π}\). The following will be obtained.

\[
R' = vP - h'(R_{π} + h_1P_{pub} + X_{π})
= vP - h'\left(\frac{v - r}{P}\right)
= (v' - h'\frac{v - r}{P})P
\]

According to \(R' = δP\), \(δ ∈ Z_{q'}^\ast\) it solves \(δ = v' - h'\frac{v - r}{P}\). That is, the A-II attackers are used as subroutines to solve the ECDLP problem successfully. However, the secure assumes that the ECDLP problem is a difficult problem that cannot be solved in the real world at present, so the forgery attack of the A-II attackers cannot be successful.

Table 1 gives the performance comparison between proposed scheme and other schemes including PFP[28], SRSA[29], PCPA[30] and IECS[31]. Where, \(mp\) represents the multiple point operation on group \(G_1\). \(bp\) represents the bilinear pair operation. \(eo\) denotes the exponential operation with relatively high computation cost.

In the signature stage, the scheme in this paper requires two multiple point operations. PFP requires two multiple point operations. SRSA requires two large exponential operations. IECS requires three multiple point operations. In signature verification phase, the proposed scheme needs three multiple point operations and three bilinear pairings computations. PFP requires three bilinear pairings computations. SRSA needs four larger index operations and two bilinear pairings computation. PCPA needs three multiple point operations and two bilinear pairings computations. IECS needs two multiple point operations and five bilinear pairings computations. In conclusion, this scheme has higher efficiency than other schemes. Moreover, the scheme in this paper has lower computational complexity and more advantages in computational efficiency.

### 7. Conclusions

This paper modifies the definition of certificateless signature and proposes a certificateless short signature scheme based on random predictor model. The scheme in this paper calculates the user’s partial public key while generating the secret value, and associates the user’s identity with the user’s partial public key when extracting the partial private key. Thus it establishes the authentication relationship between the user’s public key and the user. Compared with the classical signature schemes and the relevant certificateless signature schemes, the results show that the proposed scheme has better performance and lower computational complexity. It can meet the requirements of practical applications.

### Acknowledgement

This work was supported by Higher education Computer Research Association project, No. CERACU2020R08 and CERACU2020R02. National Computer basic Education Research Association project of higher education, No. 2020-AFCEC-309 and 2021-AFCEC-390.

### References


