IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 3, 2021, accepted May 16, 2021, date of publication May 25, 2021, date of current version June 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3083529

Identity-Based Partially Blind Signature Scheme: Cryptanalysis and Construction

YUHONG JIANG, LUNZHI DENG, AND BINGQIN NING
School of Mathematical Sciences, Guizhou Normal University, Guiyang 550001, China

Corresponding author: Lunzhi Deng (denglunzhi@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61962011, in part by the Guizhou Provincial Science and Technology Foundation under Grant [2019]1434, and in part by the 2019 Academic New Seedling Cultivation And Innovative Exploration Special Project of Guizhou Normal University.

ABSTRACT Blind signature is a special type of digital signature, the signer cannot see the specific content signed. However, blindness may cause users to abuse their rights. Partial blind signature allows the signer to embed pre-negotiated public information in the blind signature without losing blindness, which can prevent users from abusing their rights. Islam *et al.* presented an identity-based partial blind signature scheme and claimed that it is provable secure. However, in this paper we proved that the scheme is vulnerable against the tampering attacks with public information. We then proposed a new identity-based partial blind signature scheme, showed the security proofs under the assumption that the elliptic curve discrete logarithm problem (ECDLP) is difficult. The new scheme does not use pairing operations and enjoys less computation cost.

INDEX TERMS Partially blind signature, identity-based cryptography, elliptic curve, random oracle, tampering.

I. INTRODUCTION

Chaum [4] introduced the blind signature. It is a special kind of digital signature, which means the signer generates a signature without knowing the specific information. Blind signature scheme is an interactive protocol between a user and a signer. The user obtains a signature generated by the signer on a message. Although the signer generates the signature in person, he does not know the specific content of the signed message. Due to the blindness, blind signature can effectively protect the specific content of signed messages, so it is widely used in e-commerce and electronic election. On the one hand, such complete blindness can easily lead to the illegal use of signatures. On the other hand, In order to resolve issues such as proof of payment, commercial money laundering, black market transactions, and tax evasion, government agencies and tax authorities can require audits of electronic payments, which contradicts the untraceable nature of blind signatures.

In order to solve the above problems, Abe *et al.* [1] first proposed the concept of partial blind signature (PBS) in 1996. Partial blind signatures have the characteristics of blind signatures, and signers can add an agreement negotiated in advance with the user when signing. Since the agreed-upon agreement cannot be tampered with, it can prevent users from providing illegal information and misusing their rights. This not only protects the user's privacy, but also effectively restricts the user's scope of authority.

Traditional public key infrastructure requires a lot of computing, communication, and storage costs. To solve the problem, Shamir [16] introduced identity-based cryptography. In the setting, the user does not need to exchange the public key, and his/her public key is his/her identity.

A. RELATED WORKS

In 1994, Camenisch *et al.* [3] designed two blind signature schemes, however, they did not provide the proofs on unforgeability. In 2009, Tahat *et al.* [17] presented a blind signature scheme and showed the security proofs based on factoring problem and discrete logarithm problem. In 2002, Zhang *et al.* [21] proposed a blind signature scheme, that requires pairing operations [2]. In 2003, Zhang *et al.* [22] constructed a blind signature scheme, that requires only two pairing operations. In 2011, He *et al.* [8] put forward a new blind signature scheme, that does not require pairing operation and enjoys lower computing cost. In 2018, Tsaur *et al.* [19] presented an efficient PBS scheme and analyzed the security of the scheme based on the assumption of ECDLP. In 2019, Cui *et al.* [7] proposed a restrictive PBS scheme that does not

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Yan.

use bilinear pairings and improves computing efficiency, but they did not show the security proofs.

In 2005, Chow *et al.* [6] combined identity-based public key cryptography and partially blind signature, for the first time put forward an identity-based partially blind signature (IB-PBS) scheme, and proved that the scheme is secure under the random oracle machine. In 2007, Chen *et al.* [5] proposed an IB-PBS scheme based on the assumption of CDHP, what they provide is only a security analysis rather than a security proof. Hu and Huang [9] came up with an efficient and secure IB-PBS scheme. However, Tseng *et al.* [20] pointed out that this scheme [9] is vulnerable against forgery attacks. In 2009, Tian *et al.* [18] proposed a security enforcement IB-PBS scheme with security analysis, but this scheme was broken by Tseng *et al.* [20]. In 2013, Li *et al.* [14] presented an IB-PBS scheme and applied it in electronic cash, but the use of bilinear pairing limited the efficiency. In 2016, Kumar *et al.* [12] constructed an efficient IB-PBS scheme, what they provide is only a security analysis rather than a security proof. In 2017, Kumar *et al.* [13] proposed a new IB-PBS scheme, they provided the security proofs based on the difficult assumption of ECDLP.

B. OUR MOTIVATIONS AND CONTRIBUTIONS

PBS can not only protect the user's personal privacy, but also prevent the user from abusing the signer-generated signature, which is suitable for online transactions. Islam *et al.* [10] presented an IB-PBS scheme and claimed that it is secure. However, we found that the scheme [10] is not secure. So it is quite significant to design an efficient and secure IB-PBS scheme. The contribution of this paper can be summarized as follows:

1) We have studied the scheme [10]. In the scheme, signatures obtained after tampering with public information can pass verification, which indicates that the scheme cannot resist tampering with public information attacks. Namely, the scheme does not capture partial blindness.
2) We proposed a new scheme to prevent tampering with public information attacks. At the same time, we proved that the new scheme is secure in the random oracle model under the assumption that elliptic curve discrete logarithm problem (ECDLP) is intractable.
3) We gave a comparison of the efficiency of the new scheme and previous schemes, The new scheme does not require pairing operations, the computation cost is lower than that of other schemes.

C. ARRANGEMENT OF ARTICLES

The rest of this paper is structured as follows. In section 2, we introduce ECDLP and elliptic curve cryptosystem. In section 3, we present the formal definition and introduce the security attributes of identity-based partial blind signature(IB-PBS). In section 4, we review and analyze the scheme [10]. In section 5, we propose a new IB-PBS scheme. In section 6, we show the security proofs of new scheme. In section 7, we compare the efficiency of the new scheme with several other schemes. In section 8, we give a summary of this paper.

II. PREREQUISITE KNOWLEDGE

A. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic curve cryptography (Ecc) plays an increasingly important role in cryptography because of its better nature. $p > 3$ is a prime number, and the Weieratrass equation of elliptic curve E on prime field F_p can be set as

$$y^2 \bmod p = (x^3 + ax + b)$$

and its discriminant is $\triangle = (4a^3 + 27b^2) \bmod p \neq 0$, $a, b \in F_p$. Let $E_p(a, b)$ be a set of elliptic curve points over the prime field F_p, which is defined as (1) trivia. The additive elliptic curve group defined as $G = \{(x, y) : x, y \in F_p$ and $(x, y) \in E/F_p\} \cup \{O\}$, where O is known as "point at infinity" and the group G becomes an additive cyclic group of elliptic curve points. The operation rules of E on F_p are as follows:

If $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are the two points on the curve E and O are the points at infinity, then

1) $O + P_1 = P_1 + O$;
2) $-P_1 = (x_1, -y_1)$;
3) If $P_3 = (x_3, y_3) = P_1 + P_2 \neq O$, then

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_2) - y_1 \end{cases}$$

where

$$\begin{cases} \lambda = \dfrac{y_1 - y_2}{x_1 - x_2}, & if \quad P_1 \neq P_2 \\ \lambda = \dfrac{3x_1^2 + a}{2y_1}, & if \quad P_1 = P_2 \end{cases}$$

When $P_1 = P_2$, the point addition operation in the case of elliptic curve cryptography is known as the point doubling operation. The scalar multiplication on the cyclic group G defined as $kP = P + P + \ldots + P$ (k times), where $k \in Z_p^*$ is a scalar, among them $P \in G$ is the generator of order n.

B. ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM ($ECDLP$)

Given a tuple (P, aP), among them $P \in G$, it is computationally hard for any Probabilistic Polynomial Time algorithm $\mathcal{ADV}$ to calculate the integer $a \in Z_p^*$. The probability that any polynomial-time bounded algorithm $\mathscr{A}$ can solve the ECDLP is defined as $\mathcal{ADV}_{\mathscr{A}}^{ECDLP} = Pr[\mathscr{A}(P, aP) = a : a \in Z_p^*]$

C. ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM ($ECDLP$) ASSUMPTION

If there is no polynomial time algorithm that can solve the elliptic curve discrete logarithm problem ($ECDLP$) with a non-negligible probability, then $ECDLP$ is hard to resolve.

III. FORMAL MODEL OF IB-PBS

A. GENERIC SCHEME

An IB-PBS scheme consists of four algorithms: System setup, Key extraction, Signature agreement and Verify. The signature protocol is an interaction protocol between signer and requester, which includes Commitment, Blind, Sign and Unblind.

- System setup: Input security parameter k, output system public parameter Ω and system master key msk, msk confidential.
- Key extraction: Enter the public parameter Ω, the master key msk and the identity ID_i of signer i, and output the public key P_i and private key d_i of the user.
- Signature agreement: Assuming that the message provided by the requester is m and the public information negotiated by the signer and the requester is c, the signer and the requester interact as follows:
  1) Commitment: For the random number r, the signer makes a commitment R and sends it to the user.
  2) Blind: The requester selects blind factor α, processes message m with α, outputs blind message h, and sends h to the signer.
  3) Sign: The signer uses its private key d_i to sign h, outputs blind signature δ', and sends δ' to the requester.
  4) Unblind: The requester uses the original selected blind factor α to process δ' and outputs signature δ.
- Verify: The input of this algorithm is public parameter Ω, signer's identity ID_i, message m, public information c and signature δ. If signature δ is valid, print *True*, otherwise print *False*.

B. SECURITY PROPERTIES OF AN IB-PBS SCHEME

An IB-PBS scheme should meet three security requirements: integrity, partially blindness and unforgeability. The following is a brief introduction of these security features.

- Integrity: The signature correctly generated by the signature algorithm must pass the verification algorithm.
- Partial blindness: Partial blindness means that it is impossible for the signer to associate the signature obtained by the requester with its signature process when the information is the same. Partial blindness of identity-based partially blind signatures can be defined through a game between adversary $\mathscr{A}$ and challenger $\mathscr{C}$. For specific definitions, refer to literature [10].
- Unforgeability: Unforgeability means that only the signer can produce a valid signature, and no one else can produce a valid signature. The unforgeability of identity-based partial blind signatures under adaptive selection messages and identity attacks can be defined by a game between adversary $\mathscr{A}$ and challenger $\mathscr{C}$. For specific definitions, refer to literature [10].

IV. ANALYZE ISLAM'S SCHEME

In this section, we will make a retrospective analysis of Islam et al's scheme [10] and point out the unsafe parts of its scheme.

A. SCHEME REVIEW

The IB-PBS scheme given by Islam *et al.* [10] includes four parts: system setup, key extraction, signature agreement, and verify, which are completed by the signer, requester, and verifier.

1) SETUP

The input to the algorithm is the security parameter $k \in Z^+$, and the output is the system parameter and the master key of PKG. As this stage PKG does the following:

1) Choose a quad $\{F_p, E/F_p, G, P\}$. p is a k-bit prime number.F_p is the prime field of order q. E/F_p is a set of elliptic curve points. G is an additive cyclic group of elliptic curve points. P is the generator of G.
2) Select $x \in {}_R Z_p^*$ calculated as the system master key, $P_{Pub} = xP$ as the system master public key.
3) Select two hash functions for secure collisions: $H_0 : \{0,1\}^* \times G \longrightarrow Z_p^*, H_1 : \{0,1\}^* \times \{0,1\}^* \times G \longrightarrow Z_p^*$
4) Publish system parameters:
   $\Omega = \{F_p, E/F_p, G, P, P_{Pub}, H_0, H_1\}$. keep x secret.

2) KEY EXTRACT

This algorithm takes (Ω, x, ID_B) as input and outputs the private key d_B based on the identity of signer $\mathcal{B}$. First, $\mathcal{B}$ passes ID_B to PKG through a secure channel, PKG works as follows:

1) Select $r_B \in_R Z_p^*$, calculate $R_B = r_B P$, $h_B = H_0(ID_B, R_B)$
2) Calculate $d_B = r_B + h_B x$

And then PKG sends (d_B, R_B) to $\mathcal{B}$, the public key of $\mathcal{B}$ is $P_B = R_B + h_B P_{Pub}$. $\mathcal{B}$ confirms whether to accept the private-public key pair (d_B, R_B) by judging equation $P_B = d_B P = R_B + h_B P_{Pub}$. Accept $P_B = R_B + h_B P_{Pub}$ if the equation is true, otherwise do not accept $P_B = R_B + h_B P_{Pub}$.

3) SIGNATURE AGREEMENT

The identity of signer $\mathcal{B}$ is ID_B, the identity of signer $\mathcal{C}$ is ID_C. The message to be signed is $m \in \{0,1\}^*$, suppose that the public information $\mathcal{B}$ and $\mathcal{C}$ have agreed to is Δ. In order to get a partial blind signature on m and Δ, $\mathcal{B}$ and $\mathcal{C}$ interact as follows:

1) Commitment: $\mathcal{B}$ chooses a number $r \in {}_R Z_p^*$ and computes $R = rP_B$, $\mathcal{B}$ sends (R, R_B) to the requester $\mathcal{C}$.
2) Blind: When $\mathcal{C}$ receives (R, R_B) it chooses $a, b \in {}_R Z_p^*$ and calculates $R' = aR + abP + ab[R_B + H_0(ID_B, R_B)P_{Pub}] = aR + abP + abP_B$, $h = a^{-1}H_1(m, R', \Delta) + b$, C sends h to $\mathcal{B}$.
3) Sign: After $\mathcal{B}$ receives h, it calculates $S = (r + h)d_B$ and sends S to $\mathcal{C}$.
4) Unblind: After $\mathcal{C}$ receives S, it calculates $S' = a(S + b)$.The final partial blind signature is (m, Δ, R_B, R', S').

4) VERIFY

In order to verify the partial blind signature (m, Δ, R_B, R', S') for message m and public information Δ, the verifier performs the following steps:

1) Calculates $H_1(m, R', \Delta)$.
2) Verify whether equation $S'P = R' + H_1(m, R', \Delta)[R_B + H_0(ID_B, R_B)P_{Pub}]$ is valid, namely verify whether equation $S'P = R' + H_1(m, R', \Delta)P_B$ is valid.

If the equation is valid, accept partial blind signature (m, Δ, R_B, R', S'); otherwise, do not accept.

B. ATTACK ON THE SCHEME

Assume that the dishonest user $\mathcal{T}$ wants to illegally tamper the public information. $\mathcal{T}$ replaces the public information Δ with $\hat{\Delta}$. Where Δ is the information agreed by the signer $\mathcal{B}$ and the requestor $\mathcal{C}$ in advance, $\hat{\Delta}$ is the information tampered with by $\mathcal{T}$. $\mathcal{B}$ still signs with the original public information Δ, while the verifier identifies the public information as $\hat{\Delta}$ when verifying.

Only the signature part and the verification part are changed in the whole scheme, so signer $\mathcal{B}$ interacts with dishonest user $\mathcal{T}$ as follows:

1) SIGNATURE AGREEMENT

1) Commitment: $\mathcal{B}$ chooses a number $r \in {}_R Z_p^*$ and computes $R = rP_B$,$\mathcal{B}$ sends (R, R_B) to the requester $\mathcal{T}$.
2) Blind: When $\mathcal{T}$ receives (R, R_B) it chooses $a, b \in {}_R Z_p^*$ and calculates $R' = aR + abP + ab[R_B + H_0(ID_B, R_B)P_{Pub}] = aR + abP + abP_B$, $\hat{h} = a^{-1}H_1(m, R', \hat{\Delta}) + b$, $\mathcal{T}$ sends $\hat{h}$ to $\mathcal{B}$.
3) Sign: After $\mathcal{B}$ receives $\hat{h}$, it calculates $\hat{S} = (r + \hat{h})d_B$ and sends $\hat{S}$ to $\mathcal{T}$.
4) Unblind: After $\mathcal{T}$ receives $\hat{S}$, it calculates $\hat{S}' = a(\hat{S}+b)$. The final partial blind signature is $(m, \hat{\Delta}, R_B, R', \hat{S}')$.

2) VERIFY

In order to verify the partial blind signature $(m, \hat{\Delta}, R_B, R', \hat{S}')$ for message m and public information $\hat{\Delta}$, the verifier performs the following steps:

1) Calculates $H_1(m, R', \hat{\Delta})$.
2) Verify whether equation $\hat{S}'P = R' + H_1(m, R', \hat{\Delta})[R_B + H_0(ID_B, R_B)P_{Pub}]$ is valid, namely verify whether equation $\hat{S}'P = R' + H_1(m, R', \hat{\Delta})P_B$ is valid. If the equation is valid, accept partial blind signature $(m, \hat{\Delta}, R_B, R', \hat{S}')$; otherwise, do not accept.

The following shows that if the dishonest user $\mathcal{T}$ has changed the signature of the public information to pass the verification equation, it means that this scheme cannot resist the tampering with public information attacks, which is not safe.

$$\begin{aligned}
\hat{S}'P &= a(\hat{S} + b)P \\
&= a[(r + \hat{h})d_B + b]P \\
&= ard_BP + a\hat{h}d_BP + abP \\
&= ard_BP + a[a^{-1}H_1(m, R', \hat{\Delta}) + b]d_BP + abP \\
&= arP_B + a[a^{-1}H_1(m, R', \hat{\Delta}) + b]P_B + abP \\
&= arP_B + H_1(m, R', \hat{\Delta})P_B + abP_B + abP \\
&= aR + abP_B + abP + H_1(m, R', \hat{\Delta})P_B \\
&= R' + H_1(m, R', \hat{\Delta})P_B
\end{aligned}$$

That is to say, equation $S'P = R' + H_1(m, R', \hat{\Delta})P_B$ is established, so this scheme cannot resist tampering with public information attacks and is not safe.

V. PROPOSED IB-PBS SCHEME

The proposed IB-PBS scheme consists of four parts: system setup, key extract, signature agreement, and verify. In the entire scheme, the signer is set to $\mathcal{A}$ and the requester is $\mathcal{B}$.

A. SETUP

The input to the algorithm is the security parameter $k \in Z^+$, and the output is the system parameter and the master key of PKG. As this stage PKG does the following:

1) Choose a quad $\{F_p, E/F_p, G, P\}$. p is a k-bit prime number.F_p is the prime field of order q. E/F_p is a set of elliptic curve points. G is an additive cyclic group of elliptic curve points. P is the generator of G.
2) Select $x \in {}_R Z_p^*$ calculated as the system master key, $P_{Pub} = xP$ as the system master public key.
3) Select three hash functions for secure collisions: $H_0 : \{0, 1\}^* \times G \longrightarrow Z_p^*, H_1 : \{0, 1\}^* \times \{0, 1\}^* \times G \longrightarrow Z_p^*, H_2 : \{0, 1\}^* \longrightarrow Z_p^*$.
4) Publish system parameters:
$\Omega = \{F_p, E/F_p, G, P, P_{Pub}, H_0, H_1, H_2\}$. keep x secret.

B. KEY EXTRACT

This algorithm takes (Ω, x, ID_A) as input and outputs the private key d_A based on the identity of signer $\mathcal{A}$. First, $\mathcal{A}$ passes ID_A to PKG through a secure channel, PKG works as follows:

1) Select $r_A \in_R Z_p^*$, calculate $R_A = r_AP$, $h_A = H_0(ID_A, R_A)$.
2) Calculate $d_A = r_A + h_Ax$.

And then PKG sends (d_A, R_A) to $\mathcal{A}$, The public key of $\mathcal{A}$ is $P_A = R_A + h_AP_{Pub}$. $\mathcal{A}$ confirms whether to accept the private-public key pair (d_A, R_A) by judging equation $P_A = d_AP = R_A + h_AP_{Pub}$. Accept $P_A = R_A + h_AP_{Pub}$ if the equation is true, otherwise do not accept $P_A = R_A + h_AP_{Pub}$.

C. SIGNATURE AGREEMENT

The identity of signer $\mathcal{A}$ is ID_A, the identity of signer $\mathcal{B}$ is ID_B. The message to be signed is $m \in \{0, 1\}^*$, suppose that the public information $\mathcal{A}$ and $\mathcal{B}$ have agreed to is c. In order to get a partial blind signature on m and c, $\mathcal{A}$ and $\mathcal{B}$ interact as follows:

1) Commitment: $\mathcal{A}$ chooses a number $s \in {}_R Z_p^*$ and computes $S = H_2(c)[P_A + sP]$, $\mathcal{A}$ sends (S, R_A) to the requester $\mathcal{B}$.
2) Blind: When $\mathcal{B}$ receives (S, R_A) it chooses $\alpha, \beta, \gamma \in {}_R Z_p^*$ and calculates $E = \alpha S + \beta P + \gamma[R_A + H_0(ID_A, R_A)P_{Pub}] = \alpha S + \beta P + \gamma P_A$, $l = H_1(m, E, c)$, $g = a^{-1}(\gamma + l) + H_2(c)$, $\mathcal{B}$ sends g to $\mathcal{A}$.

3) Sign: After $\mathcal{A}$ receives g, it calculates $y = gd_A + sH_2(c)$ and sends y to $\mathcal{B}$.
4) Unblind: After $\mathcal{B}$ receives y, it calculates $f = \alpha y + \beta$. The final partial blind signature is $\delta = (m, c, R_A, E, f)$.

D. VERIFY

In order to verify the partial blind signature $\delta = (m, c, R_A, E, f)$ for message m and public information c, the verifier performs the following steps:

1) Calculates $l = H_1(m, E, c)$, $H_2(c)$.
2) Verify whether equation $fP = E + lP_A$ is valid. If the equation is valid, accept partial blind signature $\delta = (m, c, R_A, E, f)$; otherwise, do not accept.

VI. ANALYSIS OF THE PROPOSED IB-PBS SCHEME

In this section, we analyze the proposed partial blind signature scheme from the perspective of security and computation costs.

A. SECURITY ANALYSIS

Theorem 1 (Proof of Correctness): The IB-PBS scheme proposed by us satisfies the correctness.

Proof: The following equation shows the correctness of our proposed IB-PBS scheme.

$$\begin{aligned}
fP &= (\alpha y + \beta)P = \alpha yP + \beta P \\
&= \alpha[gd_A + sH_2(c)]P + \beta P \\
&= \alpha g d_A P + \alpha s H_2(c)P + \beta P \\
&= \alpha[\alpha^{-1}(\gamma + l) + H_2(c)]P_A + \alpha s H_2(c)P + \beta P \\
&= (\gamma + l)P_A + \alpha H_2(c)P_A + \alpha s H_2(c)P + \beta P \\
&= \gamma P_A + lP_A + \alpha H_2(c)(P_A + sP) + \beta P \\
&= \gamma P_A + lP_A + \alpha S + \beta P \\
&= \gamma P_A + \alpha S + \beta P + lP_A \\
&= E + lP_A
\end{aligned}$$

Theorem 2: (Resistance to tampering with public information attacks) Our IB-PBS scheme is resistant to tampering with public information attacks.

Proof: Assume that the dishonest user $\mathcal{T}$ wants to illegally tamper the public information. $\mathcal{T}$ replaces the public information c with $\hat{c}$, which agreed by signer $\mathcal{A}$ and requestor $\mathcal{B}$, but signer $\mathcal{A}$ is not aware of it. $\mathcal{A}$ still signs with the original public information c, while the verifier identifies the public information as $\hat{c}$ when verifying. Only the signature part and the verification part are changed in the whole scheme, so signer $\mathcal{A}$ interacts with dishonest user $\mathcal{T}$ as follows:

1) SIGNATURE AGREEMENT

1) Commitment: $\mathcal{A}$ chooses a number $s \in {}_R Z_p^*$ and computes $S = H_2(c)[P_A + sP]$, $\mathcal{A}$ sends (S, R_A) to the requester $\mathcal{T}$.
2) Blind: When $\mathcal{T}$ receives (S, R_A) it chooses $\alpha, \beta, \gamma \in {}_R Z_p^*$ and calculates $E = \alpha S + \beta P + \gamma[R_A + H_0(ID_A, R_A)P_{Pub}] = \alpha S + \beta P + \gamma P_A$, $\hat{l} = H_1(m, E, \hat{c})$, $\hat{g} = a^{-1}(\gamma + \hat{l}) + H_2(\hat{c})$, $\mathcal{T}$ sends $\hat{g}$ to $\mathcal{A}$.
3) Sign: After $\mathcal{A}$ receives $\hat{g}$, it calculates $\hat{y} = \hat{g}d_A + sH_2(c)$ and sends $\hat{y}$ to $\mathcal{T}$.
4) Unblind: After $\mathcal{T}$ receives $\hat{y}$, it calculates $\hat{f} = \alpha\hat{y} + \beta$. The final partial blind signature is $\delta = (m, \hat{c}, R_A, E, \hat{f})$.

2) VERIFY

In order to verify the partial blind signature $\delta = (m, \hat{c}, R_A, E, \hat{f})$ for message m and public information $\hat{c}$, the verifier performs the following steps:

1) Calculates $l = H_1(m, E, \hat{c})$, $H_2(\hat{c})$.
2) Verify whether equation $\hat{f}P = E + \hat{l}P_A$ is valid. If the equation is valid, accept partial blind signature $\delta = (m, \hat{c}, R_A, E, \hat{f})$; otherwise, do not accept.

The following proves that the public information cannot be verified after passing the verification equation.

$$\begin{aligned}
\hat{f}P &= (\alpha\hat{y} + \beta)P = \alpha\hat{y}P + \beta P \\
&= \alpha[\hat{g}d_A + sH_2(c)]P + \beta P \\
&= \alpha\hat{g}d_A P + \alpha s H_2(c)P + \beta P \\
&= \alpha[\alpha^{-1}(\gamma + \hat{l}) + H_2(\hat{c})]d_A P + \alpha s H_2(c)P + \beta P \\
&= \alpha[\alpha^{-1}(\gamma + \hat{l}) + H_2(\hat{c})]P_A + \alpha s H_2(c)P + \beta P \\
&= (\gamma + \hat{l})P_A + \alpha H_2(\hat{c})P_A + \alpha s H_2(c)P + \beta P \\
&= \gamma P_A + \hat{l}P_A + \alpha[H_2(\hat{c})P_A + sH_2(c)P] + \beta P \\
&\neq \gamma P_A + \hat{l}P_A + \alpha S + \beta P \\
&\neq \gamma P_A + \alpha S + \beta P + \hat{l}P_A \\
&\neq E + \hat{l}P_A
\end{aligned}$$

Therefore, part of the blind signature $\delta = (m, \hat{c}, R_A, E, \hat{f})$ cannot pass the verification equation. In summary, the dishonest user $\mathcal{T}$ cannot break through some of our proposed IB-PBS schemes by tampering with the public.

B. PARTIAL BLINDNESS

Theorem 3: (Partial blindness) Our IB-PBS scheme satisfies partial blindness.

Proof: Given a valid partial blind signature $\delta = (m, c, R_A, E, f)$ and any set of signers to save the interactive intermediate variable (R, l, f) in the signature process,consider the following system of equations:

$$E = \alpha S + \alpha P + \gamma P_A \qquad (1)$$

$$g = \alpha^{-1}(\gamma + l) + H_2(c) \qquad (2)$$

$$f = \alpha y + \beta \qquad (3)$$

We can get $\gamma = [g - H_2(c)]\alpha^{-1} - l$ and $\beta = f - \alpha y$ from (3) and (4). If we replace γ and β with $[g - H_2(c)]\alpha^{-1} - l$ and $f - \alpha y$ in the (2) form, we can get the unique α, so we get the only γ and β. Next, we prove that the only $\alpha, \beta, \gamma \in Z_p^*$ can determine (2). Since the signature is $\delta = (m, c, , R_A, E, f)$ valid, the verification equation $fP = E + lP_A$ is satisfied, so there are:

$$\begin{aligned} E &= fP - lP_A \\ &= (\alpha y+\beta)P - lP_A \\ &= \alpha yP + \beta P - lP_A \\ &= \alpha[gd_A + sH_2(c))]P + \beta P - lP_A \\ &= \alpha[\alpha^{-1}(\gamma+l)+H_2(c)]d_AP + \alpha sH_2(c)P+\beta P - lP_A \\ &= \alpha[\alpha^{-1}(\gamma+l)+H_2(c)]P_A + \alpha sH_2(c)P + \beta P - lP_A \\ &= (\gamma+l)P_A + \alpha H_2(c)P_A + \alpha sH_2(c)P+\beta P - lP_A \\ &= \gamma P_A + lP_A + \alpha H_2(c)[P_A+sP]+\beta P - lP_A \\ &= \gamma P_A + lP_A+\alpha S+\beta P - lP_A \\ &= \alpha S+\beta P+\gamma P_A \end{aligned}$$

In summary, part of the blind signature $\delta = (m, c, R_A, E, f)$ and signature intermediate variables (R, l, f) have exactly the same relationship definition, and regardless of the value of $\delta = (m, c, R_A, E, f)$ and (R, l, f), such $\alpha, \beta, \gamma \in Z_p^*$ always exists.

Therefore, even an infinitely powerful $\mathscr{A}$ outputs a correct valueb', for example, the probability of $b = b'$ is $\frac{1}{2}$. So our IB-PBS scheme satisfies partially blind.

C. UNFORGEABILITY

Theorem 4 (Unforgeability): Under the assumption that ECDLP is difficulty, our IB-PBS scheme is unforgeable.

Proof: Suppose there is an attacker $\mathscr{A}$ who can successfully forge a valid partial blind signature with a non-negligible probability in polynomial time. Only the challenger $\mathscr{C}$ who proves the existence of a probabilistic polynomial time algorithm can solve the problem with $\frac{\varepsilon}{q_{H_0}-q_s}$ advantages. Let the challenger $\mathscr{C}$ receive an instance of ECDLP: Given a tuple (P, aP), find the integer $a \in Z_p^*$. In order to calculate $a \in Z_q^*$, challenger $\mathscr{C}$ interacts with attacker $\mathscr{A}$.

Challenger $\mathscr{C}$ sets the corresponding list: $L_{H_0}^{list}$, $L_{H_1}^{list}$, $L_{H_2}^{list}$,$L_{H_s}^{list}$,$L_{H_\delta}^{list}$ by answering the key query, hash query and signature query of attacker $\mathscr{A}$. The above list is initially empty, and the list value is simulated by a random oracle. $\mathscr{A}$ does at most q_{H_0} times H_0 queries, q_{H_1} times H_1 queries, q_{H_2} times H_2 queries, q_s times key queries,and q_δ signature queries.The specific inquiry process is as follows:

1) SYSTEM SETUP

The identity of the target user is denoted by ID^*, and the hash function H_i(i=0,1,2) is a random oracle. Challenger $\mathscr{C}$ generates and publishes the system public parameter $\Omega = \{F_p, E/F_p, G, P, P_{Pub} = xP, H_0, H_1, H_2\}$, where the system public key is set to $P_{Pub} = xP$, that is, the system master private key is set to x.

2) HASH QUERIES TO H_0

$\mathscr{C}$ sets list $L_{H_0}^{list}$, which includes the tuple like (ID_i, R_i, h_i). $\mathscr{C}$ receives $\mathscr{A}$'s H_0 inquiry about identity ID_i. $\mathscr{C}$ checks list $L_{H_0}^{list}$, if there is (ID_i, R_i, h_i) in the list, let $h_i = H_0(ID_i, R_i)$ and return h_i directly to $\mathscr{A}$; Otherwise, $\mathscr{C}$ randomly selects $h_i \in_R Z_p^*$, makes $h_i = H_1(ID_i, R_i)$, returns h_i to $\mathscr{A}$, and adds (ID_i, R_i, h_i) to list $L_{H_0}^{list}$.

3) HASH QUERIES TO H_1

$\mathscr{C}$ sets list $L_{H_1}^{list}$, which includes the tuple like (m_i, c_i, E_i, l_i). $\mathscr{C}$ receives $\mathscr{A}$'s H_1 inquiry about identity m_i, the corresponding public information is c_i, and the signature public parameter is E_i. $\mathscr{C}$ checks list $L_{H_1}^{list}$, if there is (m_i, c_i, E_i, l_i) in the list, let $l_i = H_1(m_i, c_i, E_i)$ and return l_i directly to $\mathscr{A}$; Otherwise, $\mathscr{C}$ randomly selects $l_i \in_R Z_p^*$, makes $l_i = H_1(m_i, c_i, E_i)$, returns l_i to $\mathscr{A}$, and adds (m_i, c_i, E_i, l_i) to list $L_{H_1}^{list}$.

4) HASH QUERIES TO H_2

$\mathscr{C}$ sets list $L_{H_2}^{list}$, which includes the tuple like (c_i, h_2^i). $\mathscr{C}$ receives $\mathscr{A}$'s inquiry about public information c_i. $\mathscr{C}$ checks list $L_{H_2}^{list}$, if there is (c_i, h_2^i) in the list, let $h_2^i = H_2(c_i)$ and return h_2^i directly to $\mathscr{A}$; Otherwise, $\mathscr{C}$ randomly selects $h_2^i \in_R Z_p^*$, returns h_2^i to $\mathscr{A}$, and adds (c_i, h_2^i) to list $L_{H_2}^{list}$.

5) KEY INQUIRY

$\mathscr{C}$ sets list L_s^{list}, which includes the tuple like (ID_i, d_i, S_i, h_i). $\mathscr{C}$ receives $\mathscr{A}$'s key inquiry about identity ID_i. $\mathscr{C}$ queries list L_s^{list} and does the following:

1) When $ID_i \neq ID^*$, $\mathscr{C}$ chooses $r_i \in_R Z_q^*$, calculates $R_i = r_iP$. $\mathscr{C}$ checks list L_s^{list}, if there is (ID_i, R_i, h_i) in the list, let $h_i = H_0(ID_i, R_i)$; Otherwise, $\mathscr{C}$ chooses $h_i \in_R Z_p^*$ and makes $h_i = H_0(ID_i, R_i)$, store (ID_i, d_i, S_i, h_i) to list $L_{H_0}^{list}$. Then, $\mathscr{C}$ calculates $d_i = r_i + h_ix$ and returns d_i as the key of ID_i to $\mathscr{A}$.
2) When asked for the j time, set $ID_j = ID^*$,$R_* = aP$. $\mathscr{C}$ refuses to answer, and the inquiry fails.

6) SIGNATURE INQUIRY

$\mathscr{C}$ sets list L_δ^{list}, which includes the tuple like $(m_i, c_i, R_i, E_i, f_i)$. $\mathscr{C}$ receives $\mathscr{A}$'s signature inquiry about message m_i, and $\mathscr{C}$ performs the following operations:

1) select f_i, $l_i \in_R Z_p^*$;
2) Calculate $E_i = f_iP + l_iP_i$;
3) Let $l_i = H_1(m_i, c_i, E_i)$ and add (m_i, c_i, E_i, l_i) to list $L_{H_1}^{list}$. If H_1 collides, the first two operations are performed again;
4) Output signature $(m_i, c_i, R_i, E_i, f_i)$.

7) FORGERY

The attacker $\mathscr{A}$ is trained by the inquiry to the challenger $\mathscr{C}$, and outputs a valid signature of the target user whose identity is ID^*. By replaying the hash function, $\mathscr{C}$ can generate two valid signatures, $(m^*, c^*, R_A^*, E_1^*, f_1^*)$ and $(m^*, c^*, R_A^*, E_2^*, f_2^*)$, where $E_1^* = E_2^*$,$f_1^* \neq f_2^*$. Both of these signatures are valid, so they all satisfy the signature equation $f = \alpha y + \beta$, and the operation satisfies $f = ld_A + [\gamma + H_2(c)]d_A + \alpha sH_2(c) + \beta$. In fact:

$$\begin{aligned} f_1^* &= l_1^* d^* + (\gamma + h_2^*)d_A + \alpha sH_2(c) + \beta \\ f_1^* &= l_1^*(a + h_* x) + (\gamma + h_2^*)d_A + \alpha sH_2(c) + \beta \end{aligned} \tag{4}$$

$$\begin{aligned} f_2^* &= l_2^* d^* + (\gamma + h_2^*)d_A + \alpha sH_2(c) + \beta \\ f_2^* &= l_2^*(a + h_* x) + (\gamma + h_2^*)d_A + \alpha sH_2(c) + \beta \end{aligned} \tag{5}$$

TABLE 1. Operational efficiency.

Abbreviation	Representative operation	Operational efficiency
T_{ML}	Time needed to execute the modular multiplication operation	-
T_{EM}	Time needed to execute the elliptic curve point multiplication (Scalar multiplication in G_1)	$T_{EM} = 29T_{ML}$
T_{BP}	Time needed to execute the bilinear pairing operation	$T_{BP} = 87T_{ML}$
T_{IN}	Time needed to execute modular inversion operation in Z_p^*	$T_{IN} = 11.6T_{ML}$
T_{MTP}	Time needed to execute a map-to-point (hash function)	$T_{MTP} = 29T_{ML}$
T_{PA}	Time needed to execute addition of 2 elliptic curve points (point addition in G_1)	$T_{PA} = 0.12T_{ML}$

TABLE 2. Comparison of calculation efficiency.

Scheme	Signing Cost	Verification Cost	Total Cost	Security
Proposed Scheme	$5T_{EM} + 3T_{PA} + T_{IN}$	$T_{EM} + T_{PA}$	$186.08T_{ML}$	Yes
Tsaur [19]	$8T_{EM} + 2T_{PA} + T_{MTP}$	$9T_{EM} + 3T_{PA} + 2T_{MTP}$	$580.60T_{ML}$	Yes
Kumar [13]	$10T_{EM} + T_{PA} + 2T_{BP} + 7T_{MTP}$	$T_{EM} + T_{PA}$	$696.24T_{ML}$	Yes
Tian [18]	$7T_{EM} + 2T_{PA} + 2T_{BP} + 2T_{IN}$	$3T_{EM} + T_{PA} + 2T_{BP}$	$661.56T_{ML}$	Yes
Hu [9]	$7T_{EM} + 2T_{PA}$	$T_{EM} + T_{PA} + 2T_{BP}$	$406.36T_{ML}$	No
Islam [10]	$4T_{EM} + 2T_{PA} + T_{IN}$	$T_{EM} + T_{PA}$	$156.96T_{ML}$	No

(5) − (6) can get:

$$\begin{aligned} f_1^* - f_2^* &= (l_1^* - l_2^*)(a + h_* x) \\ f_1^* - f_2^* &= (l_1^* - l_2^*)a + (l_1^* - l_2^*)h_* x \\ (l_1^* - l_2^*)a &= (f_1^* - f_2^*) - (l_1^* - l_2^*)h_* x \\ a &= \frac{(f_1^* - f_2^*)}{(l_1^* - l_2^*)} - h_* x \end{aligned}$$

Finally, $\mathscr{C}$ can solve $a = \frac{(f_1^* - f_2^*)}{(l_1^* - l_2^*)} - h_* x$ as the solution of ECDLP. This means $\mathscr{C}$ successfully solves ECDLP, which contradicts the difficulty assumption of ECDLP, so attacker $\mathscr{A}$ can't break this scheme. For a target user with identity ID^*, the probability that the private key query does not fail is $\frac{1}{q_{H_0} - q_s}$, so the probability that successfully resolves ECDLP is: $\frac{\varepsilon}{q_{H_0} - q_s}$,. In summary, our IB-PBS scheme is unforgeable for adaptive selection messages and identity attacks in random oracles under the assumption that the ECDLP is difficult to solve.

VII. COMPARISON OF THE EFFICIENCY OF THE PROGRAM

In this section, we analyzed the performance of several IB-PBS schemes. we compared the performance of our scheme with several other schemes. Several notations are defined as Table 1. Third-party data is used to analyze several PBS schemes. James *et al.* [11] obtained the time overhead on basic cryptographic operations (Table 1) by using MIRACL (Shamus software), a standard cryptographic library and implemented on a hardware platform PIV (Pentium-4) 3GHZ processor with 512-MB memory and a windows XP operating system.

A simple and intuitive method is adopted to estimate the computation costs. Tsaur *et al.*'s scheme [19] requires 17 scalar multiplication operation in G_1, 5 scalar point addition in G_1 and 3 hash-to-point operations. So the computation efficiency is $17 \times 29 + 5 \times 0.12 + 3 \times 29 = 580.60\ T_{ML}$. Kumar *et al.*'s scheme [13] requires 11 scalar multiplication operation in G_1, 2 scalar point addition in G_1, 2 pairing operations in G_2 and 7 hash-to-point operations. So the computation efficiency is $11 \times 29 + 2 \times 0.12 + 2 \times 87 + 7 \times 29 = 696.24\ T_{ML}$. Tian *et al.*'s scheme [18] requires 10 scalar multiplication operation in G_1, 3 scalar point addition in G_1, 4 pairing operations in G_2 and 2 modular inversion operation in Z_q^*. So the computation efficiency is $10 \times 29 + 3 \times 0.12 + 4 \times 87 + 2 \times 11.6 = 661.56\ T_{ML}$. Hu *et al.*'s scheme [9] requires 8 scalar multiplication operation in G_1, 3 scalar point addition in G_1, 2 pairing operations in G_2. So the computation efficiency is $8 \times 29 + 3 \times 0.12 + 2 \times 87 = 406.36\ T_{ML}$. Islam *et al.*'s scheme [10] requires 5 scalar multiplication operation in G_1, 3 scalar point addition in G_1, 1 modular inversion operation in Z_q^*. So the computation efficiency is $5 \times 29 + 3 \times 0.12 + 1 \times 11.6 = 156.96\ T_{ML}$. Our scheme requires 6 scalar multiplication operation in G_1, 4 scalar point addition in G_1, 1 modular inversion operation in Z_q^*. So the

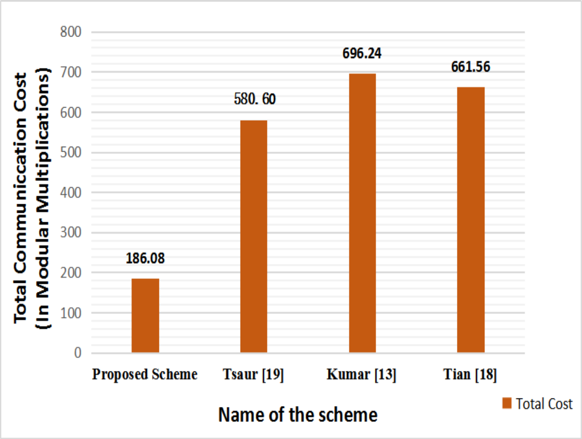

FIGURE 1. Graphical representation of total computation cost.

computation efficiency is $6\acute{A} \times 29 + 4 \times 0.12 + 1 \times 11.6 = 186.08T_{ML}$.

The total computational cost of our scheme is $186.08T_{ML}$, which is more efficient than the same type of scheme. Although our scheme is slightly more computationally expensive than Islam et al's scheme [10], our scheme is safe against tampering with public information attacks. It can be seen from Table 2 that the calculation cost of our scheme is $186.08T_{ML}$, which is 67.95% lower than the scheme in Tsaur *et al.* [19], 73.27% lower than the scheme in Kumar *et al.* [13], and 71.87% lower than the scheme in Tian *et al.* [18]. Therefore, our scheme is more efficient in calculation.

The detailed comparison results of several different PBS schemes are illustrated in Table 2 (Fig.1).

VIII. CONCLUSION

Most PBS schemes currently known use bilinear pairings, the computation cost of the pairings is much higher than that of the scalar multiplication over the elliptic curve group. Therefore, it is quite significant to construct efficient PBS scheme without bilinear pairings. In this paper, we propose a new IB-PBS scheme and gave the proof of security in the random oracle model. Our scheme does not require pairing operation, the analysis on performance shows that it is more efficient than previous ones, so it can be effectively applied to practical applications such as electronic cash.

REFERENCES

[1] M. Abe and E. Fujisaki, "How to date blind signatures," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, in Lecture Notes in Computer Science, vol. 1163, Kyongju, South Korea, 1996, pp. 244–251.

[2] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Annu. Int. Cryptol. Conf.*, in Lecture Notes in Computer Science, vol. 2139, Santa Barbara, CA, USA, 2001, pp. 213–229.

[3] J. L. Camenisch, J.-M. Piveteau, and M. A. Stadler, "Blind signatures based on the discrete logarithm problem," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, in Lecture Notes in Computer Science, vol. 950, Perugia, Italy, 1994, pp. 428–432.

[4] D. Chaum, "Blind signatures for untraceable payments," in *Proc. Adv. Cryptol.*, Santa Barbara, CA, USA, 1982, pp. 199–203.

[5] X. Chen, F. Zhang, and S. Liu, "ID-based restrictive partially blind signatures and applications," *J. Syst. Softw.*, vol. 80, no. 2, pp. 164–171, Feb. 2007.

[6] S. Chow, L. Hui, S. Yiu, and K. Chow, "Two improved partially blind signature schemes from bilinear pairings," in *Proc. Australas. Conf. Inf. Secur. Privacy*, in Lecture Notes in Computer Science, vol. 3574, Brisbane, QLD, Australia, 2005, pp. 316–328.

[7] W. Cui and Q. Jia, "Provably secure pairing-free identity-based restrictive partially blind signature scheme," in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf.*, Mar. 2019, pp. 1038–1042.

[8] D. He, J. Chen, and R. Zhang, "An efficient identity-based blind signature scheme without bilinear pairings," *Comput. Electr. Eng.*, vol. 37, no. 4, pp. 444–450, Jul. 2011.

[9] X. Hu and S. Huang, "An efficient ID-based partially blind signature scheme," in *Proc. 8th ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw., Parallel/Distrib. Comput. (SNPD)*, Jul. 2007, pp. 291–296.

[10] S. H. Islam, R. Amin, G. P. Biswas, M. S. Obaidat, and M. K. Khan, "Provably secure pairing-free identity-based partially blind signature scheme and its application in online E-cash system," *Arabian J. for Sci. Eng.*, vol. 41, no. 8, pp. 3163–3176, Aug. 2016.

[11] S. James, N. B. Gayathri, and P. V. Reddy, "Pairing free identity-based blind signature scheme with message recovery," *Cryptography*, vol. 2, no. 4, Oct. 2018, Art. no. 29.

[12] M. Kumar and C. P. Katti, "An efficient ID-based partially blind signature scheme and application in electronic-cash payment system," *ACCENTS Trans. Inf. Secur.*, vol. 2, no. 6, pp. 36–42, Dec. 2016.

[13] M. Kumar, C. P. Katti, and P. C. Saxena, "A new blind signature scheme using identity-based technique," *Int. J. Control Theor*, vol. 10, no. 15, pp. 115–124, 2017.

[14] F. Li, M. Zhang, and T. Takagi, "Identity-based partially blind signature in the standard model for electronic cash," *Math. Comput. Model.*, vol. 58, nos. 1–2, pp. 196–203, Jul. 2013.

[15] R. A. Sahu and S. Padhye, "ID-based signature schemes from bilinear pairing: A survey," *Frontiers Electr. Electron. Eng. China*, vol. 6, no. 4, pp. 487–500, Dec. 2011.

[16] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, in Lecture Notes in Computer Science, vol. 196, Santa Barbara, CA, USA, 1984, pp. 47–53.

[17] N. Tahat, E. Ismail, and R. Ahmad, "A new blind signature scheme based on factoring and discrete logarithms," *Int. J. Pharmaceutical Res.*, vol. 1, no. 1, pp. 1–9, 2009.

[18] X.-X. Tian, H.-J. Li, J.-P. Xu, and Y. Wang, "A security enforcement ID-based partially blind signature scheme," in *Proc. Int. Conf. Web Inf. Syst. Mining*, Nov. 2009, pp. 488–492.

[19] W. Tsaur, J. Tsao, and Y. Tsao, "An efficient and secure ECC-based partially blind signature scheme with multiple banks issuing E-cash payment applications," in *Proc. Int. Conf. Enterprise Comput., E-Commerce E-Services*, 2018, pp. 94–100.

[20] Y. M. Tseng, T. Y. Wu, and J. D. Wu, "Forgery attacks on an ID-based partially blind signature scheme," *IAENG Int. J. Comput. Sci.*, vol. 35, no. 3, pp. 1–4, 2008.

[21] F. Zhang and K. Kim, "ID-based blind signature and ring signature from pairings," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, in Lecture Notes in Computer Science, vol. 2501, Queenstown, New Zealand, 2002, pp. 533–547.

[22] F. Zhang and K. Kim, "Efficient ID-based blind signature and proxy signature from bilinear pairings," in *Proc. Australas. Conf. Inf. Secur. Privacy*, in Lecture Notes in Computer Science, vol. 2727, 2003, pp. 312–323.

YUHONG JIANG received the B.S. degree from Yangtze Normal University, Chongqing, China, in 2017. She is currently pursuing the master's degree with Guizhou Normal University, China. Her recent research interests include cryptography protocol and information security.

LUNZHI DENG received the B.S. and M.S. degrees from Guizhou Normal University, Guiyang, China, in 2002 and 2008, respectively, and the Ph.D. degree from Xiamen University, Xiamen, China, in 2012. He is currently a Professor with the School of Mathematical Sciences, Guizhou Normal University. His recent research interests include cryptography and information security.

BINGQIN NING received the B.S. degree from Xiangnan University, Chenzhou, China, in 2017. She is currently pursuing the master's degree with Guizhou Normal University, China. Her recent research interests include cryptography protocol and information security.