IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 18, 2018, accepted August 15, 2018, date of publication September 6, 2018, date of current version October 17, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2869041

Blind Proxy Re-Signature Scheme Based on Isomorphisms of Polynomials

LI HUIXIAN[1,2], HAN ZHIPENG[1], WANG LIQIN[1], AND PANG LIAOJUN[2,3], (Member, IEEE)

[1]School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072, China
[2]Department of Computer Science, Wayne State University, Detroit, MI 48202, USA
[3]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China

Corresponding authors: Li Huixian (lihuixian@nwpu.edu.cn) and Pang Liaojun (liaojun.pang@wayne.edu)

This work was supported in part by the National Key Technologies Research and Development Program of China under Grant 2018YFB1105303, in part by the National Natural Science Foundation of China under Grant 61103178 and Grant 61473214, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2018JM6064, and in part by the National Cryptography Development Fund under Grant MMJJ20170208.

ABSTRACT Most of the existing blind proxy re-signature schemes are designed based on the traditional public key cryptosystems, whose security relies on the hardness of big integer factoring, discrete logarithm, elliptic curve discrete logarithm, and so on. However, these problems will be unsecure when facing the attack of quantum computers. Motivated by these concerns, we proposed a blind proxy re-signature scheme based on the problem of isomorphisms of polynomials, which can resist quantum attack, and gave its security proof under the random oracle model. In the proposed scheme, the message can be kept blind by using the hash function, and the delegatee's identity can be kept anonymous by using the problem of isomorphisms of polynomials. Compared with the existing schemes, the new scheme has more secure properties, such as quantum resistance, high efficiency, message blindness, and delegatee anonymity. It is concluded that the proposed scheme has a good application prospect in the future quantum computing environment with low-power hardware.

INDEX TERMS Digital signatures, public key, blindness, security.

I. INTRODUCTION

The digital signature has become an indispensable part in the field of the network security, because it has the function of ensuring the authenticity, integrity and non-repudiation of the sender's information. According to the different application scenarios, various types of digital signature schemes were put forward, such as proxy signature [1], group signature [2], multi-signature [3], and transitive signature [4]. Since Blaze *et al.* [5] proposed the proxy re-signature in Eurocrypt in 1998, the proxy re-signature [5] has become a new branch in the field of the digital signature technology. In the proxy re-signature scheme, the semi-trusted proxy signer transforms the delegatee Alice's signature on a message into the delegator Bob's signature on the same message. Different from proxy signatures, proxy re-signatures require that proxy signers be unable to sign on behalf of Alice or Bob on their own. Therefore, proxy re-signatures focus on securely implementing the delegation of signature without fully trusting the proxy signer, and they play an important role in the space-efficient proof, transparent certification, managing group signatures, and sharing and conversion of digital certificate.

The proxy re-signature scheme proposed by Blaze *et al.* [5] realized a semi-trusted proxy signer. In 2005, Ateniese and Hohenberger [6] proposed a new definition for properties and applications of the proxy re-signature. They pointed out the defect of the previous schemes and put forward a more secure proxy re-signature scheme by exploiting bilinear maps. The security of their scheme was based on CDH (Computational Daffier-Hellman) and 2-DL (2-Discrete Logarithm). At the same time, they formally defined the algorithm and security model of the proxy re-signature. Since then, many proxy re-signature schemes with different properties have been put forward. In 2007, Shao *et al.* [7] proposed a new proxy re-signature scheme based on Water's signature scheme [8]. In this scheme, the correctness of the re-signature transformation was ensured by the property of the bilinear maps and the security of Shao *et al.*'s scheme relied on the CDH assumption. In 2008, Libert and Vergnaud [9] proposed a proxy re-signature scheme with the construction of two-level signature, which can improve the security of proxy signer's re-signature key. In this scheme, the proxy signer used the first level signature to verify and translate signatures and used the second level signature as the translated signature after the re-sign. This scheme was more secure than Shao's scheme [7] because of Flexible Diffie-Hellman assumption, but it caused a problem of large computation and storage costs. In 2012, Vivik *et al.* [10] proposed a more secure proxy re-signature scheme with two-level signature based on the CDH assumption. In the process of signing and re-signing of this scheme, they used the combination of exponentiation and hash function to make the scheme more secure, but it also caused the large computation and storage costs. In 2015, Wang [11] proposed a new proxy re-signature scheme supporting conditional delegation based on Water's signature scheme [8]. In this scheme, the delegatee can decide whether to authorize the proxy signer to translate the signature by altering the random number in the process of the generation of re-signature key.

In order to satisfy more application scenarios, the study of the proxy re-signature is not only restricted to the proxy re-signature algorithm, but also combines proxy re-signature algorithm and other algorithms according to the requirements of the different properties over the past decade. For example, the certificateless proxy re-signature scheme [12] is the combination of the certificateless cryptography and the proxy re-signature, and the threshold proxy re-signature scheme [13] is the combination of the threshold signature and the proxy re-signature. The blind proxy re-signature technology combined proxy re-signature with blind signature technology, which has become a new field of proxy re-signature research because of its good application prospect. The blind signature was proposed by Chaum [14] in 1982, which is a digital signature technology with the property of information hiding. In 2010, Deng *et al.* [15] put forward the definition of the blind proxy re-signature through combining the message blindness and the proxy re-signature. In the blind proxy re-signature scheme, the proxy signer cannot get the correct message in the process of re-signing. Therefore it is suitable for the occasions that need to hide the signed message. The blind re-signature scheme uses the bilinear mapping computation in the process of signature blinding and unblinding, which ensures the message blindness and the correctness of this scheme.

So far, almost all blind proxy re-signature schemes are designed based on the traditional public key cryptography, whose security relies on the problems of integer factorization, discrete logarithm and the elliptic curve discrete logarithm problems, etc..Nevertheless, Shor's theory [16] points out that the quantum computer could easily solve the hard problems of prime factorization, discrete logarithm and the elliptic curve discrete logarithm, etc.. Hence, the security of current blind proxy re-signature schemes is threatened by the quantum computer. In addition, the identity protection of delegates is not considered in the existing blind proxy re-signature schemes, that is, the existing schemes do not take into account the delegatee's anonymity, so it is necessary to study the new quantum resistant blind proxy re-signature schemes. In 2012, Tang and Xu [17] proposed a simple and efficient signature scheme based on IP(Isomorphism of Polynomials)assumption, which can resist the quantum computing attack. For the sake of convenience, we call it IP signature scheme in this paper. This scheme gives a good idea for designing quantum resistant signature, and enlightens us to construct quantum resistant blind proxy re-signature schemes.

OUR CONTRIBUTION

In this paper, we propose a new blind proxy re-signature scheme based on IP signature [17]. By using the hash function and Isomorphisms of Polynomials problem, our scheme is with message blindness and delegatee anonymity. In the process of the re-signature translation of our scheme, the proxy signer cannot get the correct message and the identity of delegatee, which is conducive to protecting the user privacy in the re-signing process. Therefore, our scheme is with the properties of quantum resistance, high efficiency, message blindness and delegatee's anonymity.

The rest of the paper is organized as follows. In section II, we introduce the preliminaries of blind proxy re-signature scheme. In section III, we describe the concrete algorithms of the proposed scheme. In section IV, we analyze the correctness of the proposed scheme. In section V, we give the security proof of our scheme. We analyze the properties and efficiency of the proposed scheme in section VI. In section VII, we summarize our work.

II. PRELIMINARIES

First of all, we briefly introduce the definition of the affine transformation, the Isomorphism of Polynomials and the general structure model of blind proxy re-signature scheme.

A. AFFINE TRANSFORMATION

The reversible affine transformation is a linear transformation. In a multivariable public key cryptosystem, the reversible affine transformation can be used to hide the central mapping quadratic polynomial in the multivariable public key cryptosystem by transforming the input and output of the central mapping.

Definition 1: Reversible affine transformation. Let S_i $(i = 1, 2, \ldots, n)$ be n polynomials over a finite field F_q, where n is a positive integer and q is a prime, and we have

$$S_i(x_1, x_2, \ldots, x_n) = \alpha_{i,1}x_1 + \alpha_{i,2}x_2 + \ldots + \alpha_{i,n}x_n + \beta_i\alpha_{i,n},$$
$$\beta_i \in F_q, \ 1 \le i \le n,$$

Let $S(x) = (S_1(x), S_2(x), \ldots, S_n(x))$, and $x = (x_1, x_2, \ldots, x_n)$ is a vector over F_q. S is a reversible affine transformation.

Given a fixed value

$$x = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}),$$

and

$$S_1(x) = z_1 \quad S_2(x) = z_2, \ldots, S_n(x) = z_n,$$

so we have $S(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}) = (z_1, z_2, \ldots, z_n)$.

B. PROBLEM OF ISOMOPHISM OF POLYNOMIALS

Patarin illustrated the IP (Isomorphism of Polynomials) problem in [18], namely Polynomial Isomorphism problem. All operations are on the finite field F_q. Let u and n be positive integers, and A be a set of u quadratic equations in formula (1) with n variables $x_1, x_2, \ldots, x_n$.

$$y_k = \sum_{i=1}^{n} \sum_{j=i}^{n} \gamma_{ijk} x_i x_j + \sum_{i=1}^{n} \mu_{ik} x_i + \delta_k, \quad k = 1, 2, \ldots, u \tag{1}$$

B is a set of u quadratic equations in formula (2) with n variables $x'_1, \ldots, x'_n$:

$$y'_k = \sum_{i=1}^{n} \sum_{j=i}^{n} \gamma'_{ijk} x'_i x'_j + \sum_{i=1}^{n} \mu'_{ik} x'_i + \delta'_k, \quad k = 1, 2, \ldots, u \tag{2}$$

S is a bijective affine transformation with u variables $y_1, y_2, \ldots, y_u$, which can be denoted as follows:

$$S(y_1, y_2, \ldots, y_u) = (y'_1, y'_2, \ldots, y'_u) \tag{3}$$

T is a bijective affine transformation with n variables $x'_1, x'_2, \ldots, x'_n$, which can be denoted as follows:

$$T(x'_1, x'_2, \ldots, x'_n) = (x_1, x_2, \ldots, x_n) \tag{4}$$

If there exists the transformation pair (S, T) that satisfies the equation $B = S \circ A \circ T$, where the symbol "$\circ$" is the synthesis of operations, then A and B are isomorphic. And the isomorphism from A to B is the bijective affine transformation pair (S, T).

The Problem of IP is to find the isomorphism (S, T) according to two isomorphic u quadratic equations A and B. Because the IP problem is NP hard, it usually is used to hide (S, T) which often acts as the private key.

C. MODEL OF BLIND PROXY RE-SIGNATURE

The general model of the blind proxy re-signature consists of the following eight algorithms: Global-Setup, KeyGen, ReKey, Sign, SignBlind, ReSign, ReSignUnblind and Verify.

Global-Setup: This algorithm is run by a trusted party to generate the global system parameters.

KeyGen: Taking as input the global system parameters, Key generation algorithm (KeyGen) generates a pair of signer's public key and private key (pk, sk).

ReKey: With the delegatee Alice's key (pk_a, sk_a) and delegator Bob's key (pk_b, sk_b) as inputs, the re-signature key generation algorithm (ReKey) outputs the re-signature key $rk_{a\to b}$. The proxy signer can transform the delegatee's signature into the delegator's by using the re-signature key.

Sign: With the message m and the private key sk as inputs, the signature algorithm (Sign) outputs the signature V on m.

Verify: Taking as input the public key pk, a message m and the signature V on m, the Verify algorithm judges whether Verify$(m, V, pk) = 1$ holds. If signatures generated by the algorithms Sign and ReSign make Verify$(m, V, pk) = 1$ true, the blind proxy re-signature scheme is correct.

SignBlind: With the message m, Alice's public key pk_a and Alice's signature V_a on m as inputs, if Verify$(m, V_a, pk_a) = 1$ is true, the blind signature algorithm (SignBlind) blinds the message m and signature V_a, and outputs the blinded signature V'_a, the blinded message ε and the blinded public keys pk'_a of the delegatee.

ReSign: Taking as input the re-signature key by re-signature key generation algorithm, a blinded message ε, the blinded public key pk'_a of the delegatee and the blinded signature V'_a on ε, if Verify$(\varepsilon, V'_a, pk'_a) = 1$ is true, the re-signature generation algorithm (ReSign) will be done and outputs the signature V'_b which is the blinded delegator Bob's signature. Otherwise, outputs $\perp$.

ReSignUnblind: With the blinded re-signature V'_b and message m as input, the unblind re-signature algorithm (ReSignUnBlind) outputs the unblended re-signature V_b and judges whether Verify$(m, V_b, pk_b) = 1$ holds. If the equation is true, the algorithm outputs the unblended re-signature V_b. Otherwise, outputs $\perp$.

D. BLIND PROXY RE-SINGNATURE SECURITY MODLE ON THE RANDOM ORACLES

The blind proxy re-signature scheme based on Isomorphism of Polynomials is unforgeable under the chosen message attack if there is no polynomial bounded adversary A to win the game with a non-negligible advantage in the following game.

The proxy re-signature security game is played by the attacker A and the challenger C as follows:

1) ORACLE QUERIES

The attacker A's queries will be tackled with as follows.

O_{KeyGen}: Key Generation Oracle. In the query, A inputs a public key pk generated by key generation algorithm (KeyGen), and the oracle O_{KeyGen} outputs the signer's private key sk to A.

O_{ReKey}: Re-Signature Key Oracle. In the query, A inputs public keys pk_a and pk_b generated by key generation algorithm (KeyGen), and the oracle O_{ReKey} outputs the re-signature key $rk_{a\to b}$ to A.

O_{Sign}: Signature Oracle. In the query, A inputs a public key pk generated by key generation algorithm (KeyGen) and arbitrary message m in the message space, and the oracle O_{Sign} outputs the signature V to A, which can be verified by public key pk.

O_{ReSign}: Blind Proxy Re-Signature Oracle. In the query, A inputs $(pk'_a, pk_b, \sigma, V'_a)$, where pk'_a and pk_b are public key generated by key generation algorithm (KeyGen), and V'_a is a signature on the blinded message σ which can be verified by public key pk'_a, and the oracle O_{ReSign} outputs the signature V_b to A.

2) FORGERY

The attacker A outputs (pk^*, m^*, V^*) and obtains a forged original signature successfully if the following conditions hold:

V^* is an valid signature on message m which can be verified by public key pk^*,

pk^* is not from the O_{KeyGen},

(pk^*, m^*) is not from the O_{Sign}.

The attacker A outputs (pk^*, m^*, V^*) and obtains a forged unblinded re-signature successfully if the following conditions hold:

V^* is an valid re-signature on message m which can be verified by public key pk^*,

pk^* is not from the O_{KeyGen},

(pk^*, m^*) is not from the O_{Sign}.

(Δ, pk^*) is not from the O_{ReKey}, where Δ is the public key of arbitrary signer,

$(\Delta, pk^*, m*, \square)$ is not from the O_{ReSign}, where $\square$ is arbitrary signature.

If the final outputs of A satisfy the above conditions, we say A wins the game and define the advantage of A in the above game to be Adv_A.

III. BLIND PROXY RE-SIGNAURE SCHEME BASED ON ISOMORPHISMS OF POLYNOMIALS

Our scheme consists of the following eight algorithms: Global-Setup, KeyGen, ReKey, Sign, SignBlind, ReSign, ReSignUnblind and Verify. There are five entities in this scheme, that is, the delegatee Alice, the delegator Bob, the proxy signer, the blinding proxy signer and a system administrator.

A. GLOBAL-SETUP ALGORITHM

The administrator runs this algorithm to generate the global system parameters $(n, u, q_1, q_2, q, K, Q, H_1(x), H_2(x))$. The specific steps are as follows:

1) Let K be a finite field of order 2^p, where p is a positive integer selected by the system. Let n, u, q_1, q_2 and q be positive integers, and q_1 and q_2 satisfy the equation $q = q_1 + q_2$.
2) Choose two collision-resistant hash functions $H_1(\bullet) : \{0, 1\}^* \rightarrow \{0, 1\}^{q_1}$ and $H_2(\bullet) : \{0, 1\}^* \rightarrow \{0, 1\}^{q_2}$.
3) Let Q be a set of u quadratic polynomial equations in the equation (5) with n variables:

$$y_k = \sum_{i=1}^{n}\sum_{j=i}^{n} \gamma_{ijk} x_i x_j + \sum_{i=1}^{n} \mu_{ik} x_i + \delta_k, \quad k = 1, 2, \ldots, u \tag{5}$$

where x_i and x_j are defined as the variables of polynomial equations, γ_{ijk} is defined as the coefficient of the quadratic terms of polynomial equations, μ_{ik} is defined as the first-order coefficient of polynomial equations, and δ_k is defined as the constant term.
4) The administrator publishes the global system parameters $(n, u, q_1, q_2, q, K, Q, H_1(x), H_2(x))$.

B. KEYGEN ALGORITHM

The user V runs this algorithm to generate keys by selecting his (or her) own parameters, and the details are as follows:

1) V randomly selects a pair of reversible affine transformations (M_v, N_v), which are in the following form:

$$M_v(\overline{y_1}, \overline{y_2}, \ldots, \overline{y_u}) = (y_1, y_2, \ldots, y_u),$$
$$N_v(x_1, x_2, \ldots, x_n) = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}),$$

where M_v is a reversible affine transformation with u variables, and N_v is a reversible affine transformation with n variables.
By using (M_v, N_v), V can compute her part public key A_v as follows:

$$A_v = M_v \circ Q \circ N_v.$$

2) V randomly selects reversible affine transformations $sk_v = (S_v, T_v)$ as her/his private key, which is in the following form:

$$S_v: \; S_v(y_1, y_2, \ldots, y_u) = (y'_1, y'_2, \ldots, y'_u),$$
$$T_v: \; T_v(x'_1, x'_2, \ldots, x'_n) = (x_1, x_2, \ldots, x_n),$$

where S_v is defined as a reversible affine transformation with u variables, and T_v is defined as a reversible affine transformation with n variables.
By using (S_v, T_v), V can compute her/his another part public key B_v as follows:

$$B_v = S_v \circ A_v \circ T_v.$$

3) Through the above steps 1) and 2), V gets her/his public key $pk_v = (A_v, B_v)$.

By using the above steps, Alice chooses randomly her private key $sk_a = (S_a, T_a)$, and compute her public key $pk_a = (A_a, B_a)$. In the same way, Bob randomly selects reversible affine transformations $sk_b = (S_b, T_b)$ as his private key, and compute his public key $pk_b = (A_b, B_b)$.

C. REKEY ALGORITHM

The proxy signer runs this algorithm to generate the re-signature key $rk_{a \rightarrow b} = (rk_1, rk_2, rk_3, rk_4)$.

1) The proxy signer randomly selects reversible affine transformations C, D, E and F, and sends them to Alice. Alice computes

$$Z_1 = C \circ M_a$$
$$Z_2 = N_a \circ D$$
$$Z_3 = E \circ S_a \circ M_a$$
$$Z_4 = N_a \circ T_a \circ F$$

Where C and E are defined in the equation (3), D and F are defined in the equation (4).

After computing, Alice sends Z_1, Z_2, Z_3 and Z_4 to Bob.

2) After receiving Z_1, Z_2, Z_3 and Z_4, Bob computes

$$\begin{aligned} Z_1' &= Z_1 \circ M_b^{-1} = C \circ M_a \circ M_b^{-1}, \\ Z_2' &= N_b^{-1} \circ Z_2 = N_b^{-1} \circ N_a \circ D, \\ Z_3' &= Z_3 \circ M_b^{-1} \circ S_b^{-1} = E \circ S_a \circ M_a \circ M_b^{-1} \circ S_b^{-1}, \\ Z_4' &= T_b^{-1} \circ N_b^{-1} \circ Z_4 = T_b^{-1} \circ N_b^{-1} \circ N_a \circ T_a \circ F, \end{aligned}$$

where the "-1" is an inverse operation. After computing, Bob sends Z_1', Z_2', Z_3' and Z_4' to the proxy signer.

3) After receiving Z_1', Z_2', Z_3' and Z_4', the proxy signer computes

$$\begin{aligned} rk_1 &= C^{-1} \circ Z_1' = C^{-1} \circ C \circ M_a \circ M_b^{-1}, \\ rk_2 &= Z_2' \circ D^{-1} = N_b^{-1} \circ N_a \circ D \circ D^{-1}, \\ rk_3 &= E^{-1} \circ Z_3' = E^{-1} \circ E \circ S_a \circ M_a \circ M_b^{-1} \circ S_b^{-1}, \\ rk_4 &= Z_4' \circ F^{-1} = T_b^{-1} \circ N_b^{-1} \circ N_a \circ T_a \circ F \circ F^{-1}. \end{aligned}$$

4) The proxy signer obtains the re-signature key $(rk_1, rk_2, rk_3, rk_4) = (M_a \circ M_b^{-1}, N_b^{-1} \circ N_a, S_a \circ M_a \circ M_b^{-1} \circ S_b^{-1}, T_b^{-1} \circ N_b^{-1} \circ N_a \circ T_a)$.

D. SIGN ALGORITHM

The delegatee Alice runs Sign algorithm to generate the signature which can be verified by her public key. This signature generation algorithm (Sign) is based on the IP signature scheme.

1) Alice inputs the parameters m, (A_a, B_a), (S_a, T_a), where m is the message, in the implementation the message m plus redundancy will be represented as a vector $m = (m_1, m_2, \ldots, m_n)$ over K, which will be assigned to the variables of polynomial equations used in the Sign algorithm; (A_a, B_a) is the pair of public key, and (S_a, T_a) is the pair of private key.

2) Alice randomly selects the following q bijective affine transformation pairs $((S_1', T_1'), (S_2', T_2'), \ldots, (S_q', T_q'))$, which can be used only once.

$$\begin{aligned} S_1'(y_1, y_2, \ldots, y_u) &= (y_1^{(1)}, y_2^{(1)}, \ldots, y_u^{(1)}), \\ S_2'(y_1, y_2, \ldots, y_u) &= (y_1^{(2)}, y_2^{(2)}, \ldots, y_u^{(2)}), \\ &\ldots, \\ S_q'(y_1, y_2, \ldots, y_u) &= (y_1^{(q)}, y_2^{(q)}, \ldots, y_u^{(q)}), \\ T_1'(x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}) &= (x_1, x_2, \ldots, x_n), \\ T_2'(x_1^{(2)}, x_2^{(2)}, \ldots, x_n^{(2)}) &= (x_1, x_2, \ldots, x_n), \\ &\ldots, \\ T_q'(x_1^{(q)}, x_2^{(q)}, \ldots, x_n^{(q)}) &= (x_1, x_2, \ldots, x_n). \end{aligned}$$

3) By using the above results in step 2, Alice computes

$$\begin{aligned} C_1 &= S_1' \circ A_a \circ T_1', \\ C_2 &= S_2' \circ A_a \circ T_2', \\ &\ldots, \\ C_q &= S_q' \circ A_a \circ T_q'. \end{aligned}$$

4) By using $C_1, C_2, \ldots, C_q$, Alice computes

$$H = H_1(m) \| H_2(C_1 \| C_2 \| \ldots \| C_q),$$

where the "$\|$" means the concatenation operation.

5) Alice computes the value of (S_i, T_i)

$$(S_i, T_i) = \begin{cases} (S_i', T_i'), & H[i] = 0 \\ (S_i' \circ S_a^{-1}, T_a^{-1} \circ T_i'), & H[i] = 1 \end{cases}, \quad i = 1, 2, \ldots, q$$

where $H[i]$ is the binary value of the i-th bit of H and the order of the binary string takes from the low-order to the high-order.

6) Alice computes the signature V_a on the message m as follows:

$$V_a = (H, (S_1, T_1), (S_2, T_2), \ldots, (S_q, T_q)). \tag{6}$$

Finally, Alice sends m, V_a and $pk_a = (A_a, B_a)$ to the blinding proxy signer.

E. SIGNBLIND ALGORITHM

Taking as input m, V_a and $pk_a = (A_a, B_a)$, the blinding proxy signer runs this algorithm to generate the blinded signature as follows.

1) First, the blinding proxy signer computes

$$C_i' = \begin{cases} S_i \circ A_a \circ T_i, & H[i] = 0 \\ S_i \circ B_a \circ T_i, & H[i] = 1 \end{cases}, \quad i = 1, 2, \ldots, q$$

2) Then, the blinding proxy signer computes $H' = H_1(m) \| H_2(C_1' \| C_2' \| \ldots \| C_q')$, and check whether $H' = H$ holds.

If $H' = H$ holds, the signature is true. And the blinding proxy signer executes the next step. Otherwise, the signature verification is false. And the blinding proxy signer returns $\perp$ and aborts the algorithm.

3) The blinding proxy signer selects M^* and N^* as follows:

$$\begin{aligned} M^*:\ & M^*(y_1, y_2, \ldots, y_u) = (y_1', y_2', \ldots, y_u') \\ N^*:\ & N^*(x_1', x_2', \ldots, x_n') = (x_1, x_2, \ldots, x_n). \end{aligned}$$

4) By using M^* and N^*, the blinding proxy signer computes the following values:

$$(E_i^*, F_i^*) = (S_i, T_i), \quad if\ H[i] = 1,\ i = 1, 2, \ldots, q,$$

$$(S_i^*, T_i^*) = \begin{cases} (S_i, T_i), & H[i] = 0 \\ (S_i \circ M^*, N^* \circ T_i), & H[i] = 1 \end{cases}, \quad i = 1, 2, \ldots, q$$

5) Then, the blinding proxy signer computes

$$\begin{aligned} A^* &= A_a, \\ B^* &= M^{*-1} \circ B_a \circ N^{*-1}. \end{aligned}$$

6) By using the above results, the blinding proxy signer computes and outputs the blinded signature $V^* = (H, (S_1^*, T_1^*), (S_2^*, T_2^*), \ldots, (S_q^*, T_q^*))$ of Alice, the blinded message $\varepsilon = H_1(m)$ and Alice's blinded public key $pk_a^* = (A^*, B^*)$.

F. RESIGN ALGORITHM

Take as input V^*, $\varepsilon = H_1(m)$, $pk_a^* = (A^*, B^*)$ and the re-signature key $rk_{a\to b} = (rk_1, rk_2, rk_3, rk_4)$, the proxy signer runs this algorithm to generate the blinded re-signature as follows.

1) The proxy signer computes

$$C_i' = \begin{cases} S_i^* \circ A^* \circ T_i^*, & H[i]=0 \\ S_i^* \circ B^* \circ T_i^*, & H[i]=1 \end{cases}, \quad i = 1, 2, \ldots, q$$

$$H' = \varepsilon \| H_2(C_1' \| C_2' \| \ldots \| C_q'), \quad \text{where } \varepsilon = H_1(m).$$

2) The proxy signer checks whether $H' = H$ holds.
If $H' = H$ holds, the signature is true. And the proxy signer executes the next step. Otherwise, the signature verification is false. And the proxy signer returns $\perp$ and aborts the algorithm.

3) The proxy signer generates the re-signature by computing

$$(S_{ib}^*, T_{ib}^*) = \begin{cases} (S_i^* \circ rk_1, rk_2 \circ T_i^*), & H[i]=0 \\ (S_i^* \circ rk_3, rk_4 \circ T_i^*), & H[i]=1 \end{cases}, \quad i = 1, 2, \ldots, q$$

4) The proxy signer outputs the blinded re-signature $V_b^* = (H, (S_{1b}^*, T_{1b}^*), (S_{2b}^*, T_{2b}^*), \ldots, (S_{qb}^*, T_{qb}^*))$.

G. RESINGUNBLIND ALGORITHM

Take as input the blinded re-signature V_b^*, M^*, N^*, and (E_i^*, F_i^*), the blinding proxy signer runs this algorithm to generate the re-signature.

1) The blinding proxy signer computes

$$(S_{ib}, T_{ib}) = \begin{cases} (S_{ib}^*, T_{ib}^*), & H[i]=0 \\ (E_i^* \circ M^{*-1} \circ E_i^{*-1} \circ S_{ib}^*, & \\ \quad T_{ib}^* \circ F_i^{*-1} \circ N^{*-1} \circ F_i^*), & H[i]=1 \end{cases}, \quad i = 1, 2, \ldots, q$$

to obtain the unblinded re-signature $V_b = (H, (S_{1b}, T_{1b}), (S_{2b}, T_{2b}), \ldots, (S_{qb}, T_{qb}))$.

2) The blinding proxy signer computes

$$C_i' = \begin{cases} S_{ib} \circ A_b \circ T_{ib}, & H[i]=0 \\ S_{ib} \circ B_b \circ T_{ib}, & H[i]=1 \end{cases}, \quad i = 1, 2, \ldots, q$$

where $H[i]$ is the binary value of the i-th bit of H.

3) The blinding proxy signer computes $H_b' = H_1(m) \| H_2(C_1' \| C_2' \| \ldots \| C_q')$ to check whether $H_b' = H$ holds. If $H_b' = H$, the signature is true. And the blinding proxy signer executes the next step. Otherwise, the signature verification is false. And the blinding proxy signer returns $\perp$ and ends the algorithm.

4) The blinding proxy signer outputs the unblinded signature $V_b = (H, (S_{1b}, T_{1b}), (S_{2b}, T_{2b}), \ldots, (S_{qb}, T_{qb}))$.

H. VERIFY ALGORITHM

Take as input m (or $\varepsilon = H_1(m)$), Bob's public key (A_b, B_b), and the unblinded signature V_b, the verifier runs this algorithm to verify the signature.

The verifier computes

$$C_{ib}' = \begin{cases} S_{ib} \circ A_b \circ T_{ib}, & H[i]=0 \\ S_{ib} \circ B_b \circ T_{ib}, & H[i]=1 \end{cases}, \quad i = 1, 2, \ldots, q.$$

Then, he computes $H_b' = H_1(m) \| H_2(C_{1b}' \| C_{2b}' \| \ldots \| C_{qb}')$ to check whether $H_b' = H$ holds, where H is in the V_b.

If $H_b' = H$, the signature is true, and the algorithm runs successfully. Otherwise, the signature verification is false, and return $\perp$.

IV. CORRECTNESS ANALYSIS

In this section, we prove our scheme is correct from the following three parts: the correctness of SignBlind, the correctness of ReSignUnblind and the correctness of ReSign.

A. CORRECTNESS ANALYSIS OF SIGNBLIND

Theorem 1: The SignBlind algorithm of the blind proxy re-signature scheme based on Isomorphisms of Polynomials is correct. If the SignBlind algorithm is correct, the scheme must satisfy the conditions as follows. For arbitrary message m, the blinded message $\varepsilon = H_1(m)$ and Alice's blinded public key $pk_a^* = (A^*, B^*)$, the signature V^* generated by SignBlind are satisfied with Verify$(\varepsilon, V^*, (A^*, B^*)) = 1$.

Proof: The blinding proxy signer runs SignBlind to generate the blinded signature

$$V^* = (H, (S_1^*, T_1^*), (S_2^*, T_2^*), \ldots, (S_q^*, T_q^*)).$$

SignBlind generates Alice's blinded public key as

$$A^* = A_a,$$
$$B^* = M^{*-1} \circ B_a \circ N^{*-1}.$$

The blinding proxy signer computes:

$$C_i^{*\prime} = \begin{cases} S_i^* \circ A^* \circ T_i^*, & H[i]=0 \\ S_i^* \circ B^* \circ T_i^*, & H[i]=1 \end{cases}, \quad i = 1, 2, \ldots, q$$

$$(S_i^*, T_i^*) = \begin{cases} (S_i, T_i), & H[i]=0 \\ (S_i \circ M^*, N^* \circ T_i), & H[i]=1 \end{cases}, \quad i = 1, 2, \ldots, q$$

From the above two equations, we have

$$C_i^{*\prime} = \begin{cases} S_i \circ A^* \circ T_i, & H[i]=0 \\ S_i^* \circ B^* \circ T_i^*, & H[i]=1 \end{cases}, \quad i = 1, 2, \ldots, q.$$

If $H[i] = 1$, due to $B^* = M^{*-1} \circ B_a \circ N^{*-1}$, so we get

$$S_i^* \circ B^* \circ T_i^* = S_i \circ M^* \circ M^{*-1} \circ B_a \circ N^{*-1} \circ N^* \circ T_i$$
$$= S_i \circ B_a \circ T_i$$

$$C_i^{*\prime} = \begin{cases} S_i \circ A_a \circ T_i, & H[i]=0 \\ S_i \circ B_a \circ T_i, & H[i]=1 \end{cases}, \quad i=1, 2, \ldots, q.$$

According to the Sign algorithm, we have the following equations:

$$\begin{aligned} C_1 &= S_1' \circ A_a \circ T_1', \\ C_2 &= S_2' \circ A_a \circ T_2', \\ &\dots, \\ C_q &= S_q' \circ A_a \circ T_q'. \\ (S_i, T_i) &= \begin{cases} (S_i', T_i'), & H[i]=0 \\ (S_i' \circ S_a^{-1}, T_a^{-1} \circ T_i'), & H[i]=1 \end{cases}, \\ & \qquad\qquad i = 1, 2, \dots, q, \\ H &= H_1(m) \| H_2(C_1 \| C_2 \| \dots \| C_q). \end{aligned}$$

So, we can get

$$H^{*'} = H_1(m) \| H_2(C_1^{*'} \| C_2^{*'} \| \dots \| C_q^{*'}).$$

If $H[i] = 0$,

$$C_i^{*'} = S_i' \circ A_a \circ T_i' = C_i.$$

If $H[i] = 1$,

$$\begin{aligned} C_i^{*'} &= S_i \circ B_a \circ T_i \\ &= S_i' \circ S_a^{-1} \circ B_a \circ T_a^{-1} \circ T_i \\ &= S_i' \circ S_a^{-1} \circ S_a \circ A_a \circ T_a \circ T_a^{-1} \circ T_i' \\ &= S_i' \circ A_a \circ T_i' \\ &= C_i. \end{aligned}$$

We can get $H^{*'} = H$. It satisfies with Verify $(m, V^*, (A^*, B^*)) = 1$.

Therefore, the SignBlind algorithm is correct.

B. CORRECTNESS ANALYSIS OF RESIGNUNBLIND

Theorem 2: The ReSignUnblind algorithm of the blind proxy re-signature scheme based on Isomorphisms of Polynomials is correct. If the ReSignUnblind algorithm is correct, the scheme must satisfy the conditions as follows. For arbitrary message m, and Bob's public keys $pk_b = (A_b, B_b)$ by KeyGen, the signature V_b generated by ReSignUnblind are satisfied with Verify$(m, V_b, (A_b, B_b)) = 1$.

Proof: The blinding proxy signer runs ReSignUnblind to generate the unblinded signature

$$V_b = (H, (S_{1b}, T_{1b}), (S_{2b}, T_{2b}), \dots, (S_{qb}, T_{qb})),$$

The blinding proxy signer computes:

$$(E_i^*, F_i^*) = (S_i, T_i), \quad H[i] = 1, \ i = 1, 2, \dots, q$$

$$(S_i, T_i) = \begin{cases} (S_i', T_i'), & H[i]=0 \\ (S_i' \circ S_a^{-1}, T_a^{-1} \circ T_i'), & H[i]=1 \end{cases}, \quad i = 1, 2, \dots, q$$

$$(S_i^*, T_i^*) = \begin{cases} (S_i, T_i), & H[i]=0 \\ (S_i \circ M^*, N^* \circ T_i), & H[i]=1 \end{cases}, \quad i = 1, 2, \dots, q,$$

From the above equations, we have

$$(E_i^{*-1}, F_i^{*-1}) = (S_i^{-1}, T_i^{-1}) = (S_a \circ S_i'^{-1}, T_i'^{-1} \circ T_a), \quad H[i] = 1, \quad i = 1, 2, \dots, q$$

According to the ReSignUnblind algorithm, we have

$$(S_{ib}, T_{ib}) = \begin{cases} (S_{ib}^*, T_{ib}^*), & H[i]=0 \\ (E_i^* \circ M^{*-1} \circ E_i^{*-1} \circ S_{ib}^*, & \\ \ T_{ib}^* \circ F_i^{*-1} \circ N^{*-1} \circ F_i^*), & H[i]=1 \end{cases}, \quad i = 1, 2, \dots, q$$

According to the ReSign algorithm, we have

$$(S_{ib}^*, T_{ib}^*) = \begin{cases} (S_i^* \circ rk_1, rk_2 \circ T_i^*), & H[i]=0 \\ (S_i^* \circ rk_3, rk_4 \circ T_i^*), & H[i]=1 \end{cases}, \quad i = 1, 2, \dots, q$$

According to the ReKey algorithm, we have

$$(rk_1, rk_2, rk_3, rk_4) = (M_a \circ M_b^{-1}, N_b^{-1} \circ N_a, S_a \circ M_a \circ M_b^{-1} \circ S_b^{-1}, T_b^{-1} \circ N_b^{-1} \circ N_a \circ T_a)$$

So, when $H[i] = 0$, $i = 1, 2, \dots, q$, we can get:

$$\begin{aligned} (S_{ib}, T_{ib}) &= (S_{ib}^*, T_{ib}^*) \\ &= (S_i^* \circ rk_1, rk_2 \circ T_i^*) \\ &= (S_i' \circ rk_1, rk_2 \circ T_i') \\ &= (S_i' \circ M_a \circ M_b^{-1}, N_b^{-1} \circ N_a \circ T_i') \end{aligned}$$

And when $H[i] = 1$, $i = 1, 2, \dots, q$, we can get:

$$\begin{aligned} &(S_{ib}, T_{ib}) \\ &= (E_i^* \circ M^{*-1} \circ E_i^{*-1} \circ S_{ib}^*, T_{ib}^* \circ F_i^{*-1} \circ N^{*-1} \circ F_i^*) \\ &= (E_i^* \circ M^{*-1} \circ E_i^{*-1} \circ S_i^* \circ rk_3, rk_4 \circ T_i^* \circ F_i^{*-1} \\ &\quad \circ N^{*-1} \circ F_i^*) \\ &= (E_i^* \circ M^{*-1} \circ E_i^{*-1} \circ S_i' \circ S_a^{-1} \circ M^* \circ rk_3, \\ &\quad rk_4 \circ N^* \circ T_a^{-1} \circ T_i' \circ F_i^{*-1} \circ N^{*-1} \circ F_i^*) \\ &= (S_i' \circ S_a^{-1} \circ M^{*-1} \circ S_a \circ S_i'^{-1} \circ S_i' \circ S_a^{-1} \circ M^* \circ rk_3, \\ &\quad rk_4 \circ N^* \circ T_a^{-1} \circ T_i' \circ T_i'^{-1} \circ T_a \circ N^{*-1} \circ T_a^{-1} \circ T_i') \\ &= (S_i' \circ S_a^{-1} \circ rk_3, rk_4 \circ T_a^{-1} \circ T_i') \\ &= (S_i' \circ S_a^{-1} \circ S_a \circ M_a \circ M_b^{-1} \circ S_b^{-1}, \\ &\quad T_b^{-1} \circ N_b^{-1} \circ N_a \circ T_a \circ T_a^{-1} \circ T_i') \\ &= (S_i' \circ M_a \circ M_b^{-1} \circ S_b^{-1}, T_b^{-1} \circ N_b^{-1} \circ N_a \circ T_i') \end{aligned}$$

In the signature transformed by the blinding proxy signer, the q reversible affine transformation pairs can be regarded as the affine transformation pairs chosen by Bob. Those are $(S_1' \circ M_a \circ M_b^{-1}, N_b^{-1} \circ N_a \circ T_1')$, $(S_2' \circ M_a \circ M_b^{-1}, N_b^{-1} \circ N_a \circ T_2')$, $\dots$, $(S_q' \circ M_a \circ M_b^{-1}, N_b^{-1} \circ N_a \circ T_q')$.

According to the KeyGen algorithm, we have

$$A_b = M_b \circ Q \circ N_b,$$

So, we can get

$$C_{1b} = S_1' \circ M_a \circ M_b^{-1} \circ A_b \circ N_b^{-1} \circ N_a \circ T_1'$$

$$
\begin{aligned}
&= S_1' \circ M_a \circ M_b^{-1} \circ M_b \circ Q \circ N_b \circ N_b^{-1} \circ N_a \circ T_1' \\
&= S_1' \circ M_a \circ Q \circ N_a \circ T_1' \\
&= C_1 \\
C_{2b} &= S_2' \circ M_a \circ M_b^{-1} \circ A_b \circ N_b^{-1} \circ N_a \circ T_2' \\
&= S_2' \circ M_a \circ Q \circ N_a \circ T_2' \\
&= C_2 \\
&\dots, \\
C_{qb} &= S_q' \circ M_a \circ M_b^{-1} \circ A_b \circ N_b^{-1} \circ N_a \circ T_q' \\
&= S_q' \circ M_a \circ Q \circ N_a \circ T_q' \\
&= C_q.
\end{aligned}
$$

For the same message m, we can get

$$
\begin{aligned}
H_b' &= H_1(m)\|H_2(C_{1b}\|C_{2b}\|\dots\|C_{qb}) \\
&= H_1(m)\|H_2(C_1\|C_2\|\dots\|C_q) = H.
\end{aligned}
$$

The signature $V_b = (H, (S_{1b}, T_{1b}), (S_{2b}, T_{2b}), \dots, (S_{qb}, T_{qb}))$ generated by ReSignUnblind is satisfied with Verify $(m, V_b, (A_b, B_b)) = 1$.

Therefore, the ReSignUnblind algorithm is correct.

C. CORRECTNESS ANALYSIS OF RESIGN

Theorem 3: The ReSign algorithm of the blind proxy re-signature scheme based on Isomorphisms of Polynomials is correct. If the ReSign algorithm is correct, the scheme must satisfy the conditions as follows. For the arbitrary message m, the pair of public keys and private keys $(pk_i, sk_i) = ((A_i, B_i), (S_i, T_i))$ by KeyGen, the signature V_i generated by Sign are satisfied with Verify$(m, V_i, (A_i, B_i)) = 1$, and the signature generated by ReSign are satisfied with Verify$(m$, ReSignUnblind(ReSign$(rk_{a\to b}, m, V_a)), (A_b, B_b)) = 1$.

Proof: We first prove that the signature V_i generated by Sign is satisfied with Verify$(m, V_i, (A_i, B_i)) = 1$.

The signature of Alice is $V_a = (H, (S_1, T_1), (S_2, T_2), \dots, (S_q, T_q))$, where $H = H_1(m)\|H_2(C_1\|C_2\|\dots\|C_q)$. The "$\|$" means concatenation operator.

Alice randomly selects q'reversible affine transformation pairs $((S_1', T_1'), (S_2', T_2'), \dots, (S_q', T_q'))$, according to the Sign and KeyGen algorithms, we have

$$
\begin{aligned}
C_1 &= S_1' \circ A_a \circ T_1', \\
C_2 &= S_2' \circ A_a \circ T_2, \\
&\dots, \\
C_q &= S_q' \circ A_a \circ T_q', \\
A_a &= M_a \circ Q \circ N_a.
\end{aligned}
$$

From the above equations, we have

$$
\begin{aligned}
C_1 &= S_1' \circ M_a \circ Q \circ N_a \circ T_1', \\
C_2 &= S_2' \circ M_a \circ Q \circ N_a \circ T_2', \\
&\dots, \\
C_q &= S_q' \circ M_a \circ Q \circ N_a \circ T_q'.
\end{aligned}
$$

According to the Sign algorithm and KeyGen algorithms, we have

$$
(S_i, T_i) = \begin{cases} (S_i', T_i'), & H[i] = 0 \\ (S_i' \circ S_a^{-1}, T_a^{-1} \circ T_i'), & H[i] = 1 \end{cases}, \qquad i = 1, 2, \dots, q,
$$

where $H[i]$ is the value of the i-th bit of H.

$$
B_a = S_a \circ A_a \circ T_a.
$$

So, we can get

$$
C_i' = \begin{cases} S_i \circ A_a \circ T_i = S_i' \circ A_a \circ T_i' \\ \quad = C_i, \quad (H[i] = 0) \\ S_i \circ B_a \circ T_i = S_i' \circ S_a^{-1} \circ B_a \circ T_a^{-1} \circ T_i' \\ \quad = S_i' \circ S_a^{-1} \circ S_a \circ A_a \circ T_a \circ T_a^{-1} \circ T_i' \\ \quad = S_i' \circ A_a \circ T_i' \\ \quad = C_i, \quad (H[i] = 1) \end{cases}, \qquad i = 1, 2, \dots, q,
$$

$$
\begin{aligned}
H' &= H_1(m)\|H_2(C_1'\|C_2'\|\dots\|C_q') \\
&= H_1(m)\|H_2(C_1\|C_2\|\dots\|C_q) = H.
\end{aligned}
$$

Hence, the signature V_a generated by Sign is satisfied with Verify$(m, V_a, (A_a, B_a)) = 1$.

Then, we prove that the signature generated by ReSign is satisfied with Verify$(m$, ReSignUnblind(ReSign $(rk_{a\to b}, m, V_a)), (A_b, B_b)) = 1$.

By the theorem 1 and the theorem 2, we can get ReSignUnblind(ReSign$(rk_{a\to b}, m, V_a)) = V_b$.

By the theorem 2, we can get

ReSignUnblind(ReSign$(rk_{a\to b}, m, V_a)) = V_b = (H, (S_{1b}, T_{1b}), (S_{2b}, T_{2b}), \dots, (S_{qb}, T_{qb}))$.

According to the theorem 2, we have

$$
\begin{aligned}
H &= H_1(m)\|H_2(C_{1b}\|C_{2b}\|\dots\|C_{qb}), \\
C_{1b} &= S_1' \circ M_a \circ M_b^{-1} \circ A_b \circ N_b^{-1} \circ N_a \circ T_1' \\
&= S_1' \circ M_a \circ M_b^{-1} \circ M_b \circ Q \circ N_b \circ N_b^{-1} \circ N_a \circ T_1' \\
&= S_1' \circ M_a \circ Q \circ N_a \circ T_1' \\
&= C_1, \\
C_{2b} &= S_2' \circ M_a \circ M_b^{-1} \circ A_b \circ N_b^{-1} \circ N_a \circ T_2' \\
&= S_2' \circ M_a \circ Q \circ N_a \circ T_2' \\
&= C_2, \\
&\dots, \\
C_{qb} &= S_q' \circ M_a \circ M_b^{-1} \circ A_b \circ N_b^{-1} \circ N_a \circ T_q' \\
&= S_q' \circ M_a \circ Q \circ N_a \circ T_q' \\
&= C_q.
\end{aligned}
$$

According to the ReSignUnblind and ReSign algorithms, when $H[i] = 0$, $i = 1, 2, \dots, q$, we have:

$$
\begin{aligned}
(S_{ib}, T_{ib}) &= (S_{ib}^*, T_{ib}^*) \\
&= (S_i^* \circ rk_1, rk_2 \circ T_i^*) \\
&= (S_i' \circ rk_1, rk_2 \circ T_i') \\
&= (S_i' \circ M_a \circ M_b^{-1}, N_b^{-1} \circ N_a \circ T_i'),
\end{aligned}
$$

and when $H[i] = 1, i = 1, 2, \dots, q$, we have:

$$
\begin{aligned}
&(S_{ib}, T_{ib})\\
&= (E_i^* \circ M^{*-1} \circ E_i^{*-1} \circ S_{ib}^*, T_{ib}^* \circ F_i^{*-1} \circ N^{*-1} \circ F_i^*)\\
&= (E_i^* \circ M^{*-1} \circ E_i^{*-1} \circ S_i^* \circ rk_3, rk_4 \circ T_i^* \circ F_i^{*-1}\\
&\qquad \circ N^{*-1} \circ F_i^*)\\
&= (E_i^* \circ M^{*-1} \circ E_i^{*-1} \circ S_i' \circ S_a^{-1} \circ M^* \circ rk_3,\\
&\qquad rk_4 \circ N^* \circ T_a^{-1} \circ T_i' \circ F_i^{*-1} \circ N^{*-1} \circ F_i^*)\\
&= (S_i' \circ S_a^{-1} \circ M^{*-1} \circ S_a \circ S_i^{'-1} \circ S_i' \circ S_a^{-1} \circ M^* \circ rk_3,\\
&\qquad rk_4 \circ N^* \circ T_a^{-1} \circ T_i' \circ T_i^{'-1} \circ T_a \circ N^{*-1} \circ T_a^{-1} \circ T_i')\\
&= (S_i' \circ S_a^{-1} \circ rk_3, rk_4 \circ T_a^{-1} \circ T_i')\\
&= (S_i' \circ S_a^{-1} \circ S_a \circ M_a \circ M_b^{-1} \circ S_b^{-1},\\
&\qquad T_b^{-1} \circ N_b^{-1} \circ N_a \circ T_a \circ T_a^{-1} \circ T_i')\\
&= (S_i' \circ M_a \circ M_b^{-1} \circ S_b^{-1}, T_b^{-1} \circ N_b^{-1} \circ N_a \circ T_i')
\end{aligned}
$$

In the signature transformed by the blinding proxy signer, the q reversible affine transformation pairs can be regarded as the affine transformation pairs chosen by Bob. Those are $(S_1' \circ M_a \circ M_b^{-1}, N_b^{-1} \circ N_a \circ T_1'), (S_2' \circ M_a \circ M_b^{-1}, N_b^{-1} \circ N_a \circ T_2'), \dots, (S_q' \circ M_a \circ M_b^{-1}, N_b^{-1} \circ N_a \circ T_q')$.

According to the KeyGen algorithm, we have

$$A_b = M_b \circ Q \circ N_b,$$

So, we can get

$$
\begin{aligned}
C_{1b} &= S_1' \circ M_a \circ M_b^{-1} \circ A_b \circ N_b^{-1} \circ N_a \circ T_1'\\
&= S_1' \circ M_a \circ M_b^{-1} \circ M_b \circ Q \circ N_b \circ N_b^{-1} \circ N_a \circ T_1'\\
&= S_1' \circ M_a \circ Q \circ N_a \circ T_1'\\
&= C_1,\\
C_{2b} &= S_2' \circ M_a \circ M_b^{-1} \circ A_b \circ N_b^{-1} \circ N_a \circ T_2'\\
&= S_2' \circ M_a \circ Q \circ N_a \circ T_2'\\
&= C_2,\\
&\quad \dots,\\
C_{qb} &= S_q' \circ M_a \circ M_b^{-1} \circ A_b \circ N_b^{-1} \circ N_a \circ T_q'\\
&= S_q' \circ M_a \circ Q \circ N_a \circ T_q'\\
&= C_q.
\end{aligned}
$$

For the same message m, we can get

$$
\begin{aligned}
H_b' &= H_1(m) \| H_2(C_{1b} \| C_{2b} \| \dots \| C_{qb})\\
&= H_1(m) \| H_2(C_1 \| C_2 \| \dots \| C_q)\\
&= H.
\end{aligned}
$$

Hence, The signature generated by ReSign is satisfied with Verify(m, ReSignUnblind(ReSign($rk_{a \to b}, m, V_a$)), (A_b, B_b)) = 1.

Therefore, the ReSign algorithm is correct.

By the analysis above, our scheme is proved.

V. SECURITY ANALYSIS

In this section, we prove our scheme is secure from the following two parts: consistency and unforgeability. In the proposed scheme, we assume that the proxy signer cannot collude with the delegatee Alice or the delegator Bob.

A. ANALYSIS OF CONSISTENCY

Theorem 4: The blind proxy re-signature scheme based on Isomorphisms of Polynomials is consistent. If the scheme is consistent, the blind proxy re-signature scheme must satisfy the condition that for an arbitrary message m, the public key $pk_i = (A_i, B_i)$ and the signature V_i, the results of respectively running the algorithm Verify($m, V_i, (A_i, B_i)$) twice are the same.

Proof: In our scheme, when the signature generated by Sign(pk_i, m) or Resign($rk_{i \to j}, m, V_i$) is verified, the same conditions are used. So we can get that H' is always equal to H for $V = (H, (S_1, T_1), (S_2, T_2), \dots, (S_q, T_q))$ and $H' = H_1(m) \| H_2(C_1' \| C_2' \| \dots \| C_q')$.

Therefore, the blind proxy re-signature scheme is consistent.

B. ANALYSIS OF UNFORGEABILITY

Theorem 5: Under the random oracle model, if the adversary A with a non-negligible probability ε wins the game at most Q_S times queries to the oracle O_{Sign}, Q_K times queries to the oracle O_{KeyGen}, Q_{RK} times queries to the oracle O_{ReKey}, Q_{RS} times queries to the oracle O_{ReSign}, Q_{H_1} times hash queries to the oracle H_1, and Q_{H_2} times hash queries to the oracle H_2, then there may exist an algorithm C which is able to solve the IP problem with a probability ε' in the polynomial time,

$$\varepsilon' \geq \frac{\varepsilon(1 - \frac{1}{2^{q_1}})}{Q_{H_1} \cdot Q_K \cdot (Q_{RS} + Q_S)}.$$

Proof: Let (S_t, T_t) be the target private keys of algorithm C, A is a subroutine of C, and C also plays the role of the challenger in the game.

1) INITIALIZATION PHASE

Set the public parameters (n, u, q_1, q_2, K, Q) of adversary A, where K is a finite field, n, u, q_1 and q_2 are positive integers, Q is a set of u quadratic equations with n variables. Let $H_1(\bullet) : \{0, 1\}^* \to \{0, 1\}^{q_1}$, $H_2(\bullet) : \{0, 1\}^* \to \{0, 1\}^{q_2}$ be random oracles.

2) ATTACK PHASE

O_{KeyGen}: Challenger C chooses (S_i, T_i) at random. If there is not the entry, output the corresponding value $(pk_i, sk_i) = (A_i, B_i)$, where $A_i = M_i \circ Q \circ N_i$, $B_i = S_i \circ A_i \circ T_i$, M_i and S_i is defined in the equation (3), N_i and T_i is defined in the equation (4). If there is the entry, output the corresponding public keys and private keys $(pk_i, sk_i) = ((A_i, B_i), (S_i, T_i))$.

O_{Sign}: Take as input (pk_i, m_j) after C receiving A's querying, challenger C chooses S_p', T_p' at random, $p = 1, 2, \dots, q$.

(a) If pk_i has been compromised, compute

$$\begin{aligned} C_1 &= S_1' \circ A_i \circ T_1', \\ C_2 &= S_2' \circ A_i \circ T_2', \\ &\ldots, \\ C_q &= S_q' \circ A_i \circ T_q'. \end{aligned}$$

The signer computes H and (S_p, T_p) as follows:

$$H = H(m_j \| C_1 \| C_2 \| \ldots \| C_q),$$

$$(S_p, T_p) = \begin{cases} (S_p', T_p'), (H[p] = 0) \\ (S_p' \circ S^{-1}, T^{-1} \circ T_p'), (H[p] = 1) \end{cases}, \quad p = 1, 2, \ldots, q$$

Output the signature $V = (H, (S_1, T_1), (S_2, T_2), \ldots, (S_q, T_q))$.

(b) If pk_i has not been compromised, select M defined in the equation (3) and N defined in the equation (4) and compute

$$\begin{aligned} A_i &= M \circ Q \circ N, \\ S_p^* &= S_p' \circ M^{-1}, \\ T_p^* &= N^{-1} \circ T_p', \\ C_1 &= S_1^* \circ A_i \circ T_1^* = S_1' \circ Q \circ T_1', \\ C_2 &= S_2^* \circ A_i \circ T_2^* = S_2' \circ Q \circ T_2', \\ &\ldots, \\ C_q &= S_q^* \circ A_i \circ T_q^* = S_q' \circ Q \circ T_q'. \end{aligned}$$

Then, the signer computes the hash value $H = H_1(m) \| H_2(C_1 \| C_2 \| \ldots \| C_q)$ and (S_p, T_p), and

$$(S_p, T_p) = \begin{cases} (S_p' \circ M^{-1}, N^{-1} \circ T_p'), H[p] = 0 \\ (S_p' \circ M^{-1} \circ S_i^{-1}, \\ \quad T_i^{-1} \circ N^{-1} \circ T_p'), H[p] = 1 \end{cases}, \quad p = 1, 2, \ldots, q$$

Output the signature $V = (H, (S_1, T_1), (S_2, T_2), \ldots, (S_q, T_q))$.

The verification process of signature correctness is as follows if pk_i is not been compromised.

First compute:

$$C_p' = \begin{cases} S_p \circ A_i \circ T_p = S_p' \circ M^{-1} \circ A_i \\ \quad \circ N^{-1} \circ T_p', & (H[p] = 0) \\ S_p \circ B_i \circ T_p = S_p' \circ M^{-1} \circ S_i^{-1} \\ \quad \circ B_i \circ T_i^{-1} \circ N^{-1} \circ T_p', & (H[p] = 1) \end{cases}, \quad p = 1, 2, \ldots, q$$

According to the KeyGen algorithm, we have

$$A_i = M \circ Q \circ N, \quad B_i = S_i \circ A_i \circ T_i$$

So, we can get

$$C_p' = \begin{cases} S_p \circ A_i \circ T_p \\ \quad = S_p' \circ Q \circ T_p', & (H[p] = 0) \\ S_p \circ B_i \circ T_p \\ \quad = S_p' \circ Q \circ T_p', & (H[p] = 1) \end{cases}, \quad p = 1, 2, \ldots, q$$

$$\begin{aligned} H_1(m_j) \| H_2(C_1' \| C_2' \| \ldots \| C_q') &= H_1(m_j) \\ &\| H_2(C_1 \| C_2 \| \ldots \| C_q) = H. \end{aligned}$$

O_{ReKey}: Take as input $(pk_i,\ pk_j)$ after C receiving A's querying, challenger C outputs $rk_{i \to j}(rk_1, rk_2, rk_3, rk_4) = (M_i \circ M_j^{-1}, N_j^{-1} \circ N_i, S_i \circ M_i \circ M_j^{-1} \circ S_j^{-1}, T_j^{-1} \circ N_j^{-1} \circ N_i \circ T_i)$ when pk_i and pk_j are both compromised or not. Otherwise, ends the program.

O_{ReSign}: Take as input $(pk_a', pk_b, \sigma, V_a')$ after C receiving A's querying, where $pk_a' \neq pk_a, V_a'$ is the blinded signature of the blinded message σ corresponding to the public key pk_a'. The challenger C verifies $\text{Verify}(pk_a', \sigma, V_a') = 1$, if $\text{Verify}(pk_a', \sigma, V_a') = 1$ and $pk_a' \neq pk_a, C$ outputs $O_{Sign}(pk_b, m_j)$ by running O_{Sign}. Otherwise, ends the program.

O_{H_1}: Take as input m_k after C receiving $A's$ querying, there is the entry in the list L_1, challenger C outputs the corresponding value; otherwise, C chooses $\omega \in Z_{q_1}$ at random and records it in the list L_1.

O_{H_2}: Take as input $(C_1, C_2, \ldots, C_q)$ after C receiving $A's$ querying, there is the entry in the list L_2, challenger C outputs the corresponding value; otherwise, C chooses $\mu \in Z_{q_2}$ at random and records it in the list L_2.

3) FORGERY PHASE

A outputs a purposed forgery $V'' = (H'', (S_1'', T_1''), (S_2'', T_2''), \ldots, (S_q'', T_q''))$ after the polynomial bounded time queries.

According to the analysis above, we know the simulation is equal to the actual attack environment. If the forgery of A succeeds, A must obtain (S_t, T_t). The probability of obtaining the correct (S_t, T_t) is $\frac{1}{Q_{RS}+Q_S}$ at least, and (S_t, T_t) is as the C's output for IP problem.

The probability that A will guess the target user correctly is $\frac{1}{Q_K}$. The probability that A will guess the correct message m^* is $\frac{1-\frac{1}{2^{q_1}}}{Q_{H_1}}$.

Thus, the probability that C succeeds is

$$\varepsilon' \geq \frac{\varepsilon(1 - \frac{1}{2^{q_1}})}{Q_{H_1} \cdot Q_K \cdot (Q_{RS} + Q_S)}.$$

VI. EFFICIENCY AND PERFORMANCE ANALYSIS

A. EFFICIENCY ANALYSIS

The proposed scheme has quite low time consumption throughout the algorithm. The time consumption mainly includes the stage of re-signature generation (including the algorithms of ReSign, SignBlind, and ReSignUnblind in the proposed scheme) and the stage of verification. The following table shows the efficiency of our scheme by analyzing the time consumption of the stage of re-signature generation and verification between existing proxy re-signature schemes and ours.

Exponentiation operations and pairing operations are inefficient [19]. According to Pang *et al.*'s data (see PlosOne, 2016, 11(11): e0166173), compared with the time consumption T_m of a modular multiplication operation, the time

TABLE 1. Efficiency comparison.

Scheme	Re-signature generation	Verification	Public key	Private key
Ateniese [6]	T_e+2T_p+T_h ≈246.5T_m	2T_p+T_h ≈203T_m	k	$\|q\|$
Shao[7]	2T_e+3T_p+T_h ≈377T_m	3T_p+T_h ≈290T_m	$\|p\|$	$\|q\|$
Libert[9]	3T_e+2T_p+T_h ≈333.5T_m	4T_p+T_h ≈377T_m	$\|p\|$	$\|p\|$
Vivik[10]	6T_e+3T_p+2T_h ≈580T_m	4T_e+5T_p+4T_h ≈725T_m	2$\|p\|$	2$\|p\|$
Wang[11]	6T_p+2T_h ≈580T_m	4T_p+2T_h ≈406T_m	2$\|p\|$	2$\|p\|$
Deng[15]	5T_e+6T_p+T_h ≈768.5T_m	3T_p+T_h ≈290T_m	$\|p\|$	$\|p\|$
Ours	5T_h≈145T_m	2T_h≈58T_m	8u(n+1) (n+2)	8u(u+1)+ 8n(n+1)

Note: k is a security parameter in the scheme, T_e donates the time required for executing an exponentiation operation, T_h donates the time required for executing a hash operation, T_P donates the time required for executing a pairing operation, T_m donates the time required for executing a modular multiplication operation, u and n are positive integers in our scheme, p and q are prime numbers in the schemes[6,7,9,10,11,15], $|p|$ and $|q|$ respectively donate the lengths of the binary strings over the finite fields defined by p and q.

consumptions of exponentiation operation, pairing operation and hash operation are $\mathrm{T}_p \approx 87\mathrm{T}_m$, $\mathrm{T}_e \approx 43.5\mathrm{T}_m$, $\mathrm{T}_h \approx 29\mathrm{T}_m$, respectively. From Table 1, we can see that there is no exponentiation operation and pairing operation in the stage of re-signature generation in our proxy re-signature scheme. So our scheme uses the hash operation with little computation to substitute the pairing operation and exponentiation operation of the scheme [15] in the algorithms of SignBlind, and ReSignUnblind. Therefore, our scheme is more efficient compared with other schemes [6], [7], [9]–[11], [15].

According to the analysis of the attack for IP signature scheme [17], the attack computational complexity is over 280 that required by the security level, where the parameters are $K = GF(28)$, $n = 18$, $u = 10$, and $q = 64$. In our scheme, the size of public key is $8u(n+1)(n+2) = 8*10*(18+1)*(18+2) = 30400$(bits), and the size of private key is $8u(u+1)+n(n+1) = 8*10*(10+1)+8*18*(18+1) = 3616$(bits). Because other schemes [6], [7], [9]–[11], [15] are based on the discrete logarithm problems and p, q are prime numbers, the size of public key and private key is 1024 bits. Therefore, compared with other schemes [6], [7], [9]–[11], [15], our scheme is large in the size of the public key and private key, which is the inherent defect of IP signature technology, and requires further study.

B. PERFORMANCE ANALYSIS

Although the size of public key and private key of our scheme is large, it has more secure properties. The following table compares the secure properties between the existing proxy re-signature schemes and ours.

According to table 2, we briefly analyze these properties.

1) MULTI-USE

The signature generated by ReSignUnblind or Sign of the blind proxy re-signature scheme can be used as input to the ReSign algorithm.

TABLE 2. Secure properties comparision.

Schemes	Multi-use	Transparent	Message blindness	Delegatee anonymity	Quantum resistance	Security model
Ateniese[6]	√	√	×	×	×	ROM
Shao[7]	√	√	×	×	×	SM
Libert[9]	√	×	×	×	×	SM
Vivik[10]	√	×	×	×	×	SM
Wang[11]	√	×	×	×	×	ROM
Deng[15]	√	√	√	×	×	SM
Ours	√	√	√	√	√	ROM

In our scheme, the type of ReSignUnblind algorithm input is $V_b^* = (H, (S_{1b}^*, T_{1b}^*), (S_{2b}^*, T_{2b}^*), \ldots, (S_{qb}^*, T_{qb}^*))$. And the type of the signature generated by ReSignUnblind is $V_b = (H, (S_{1b}, T_{1b}), (S_{2b}, T_{2b}), \ldots, (S_{qb}, T_{qb}))$. The type of the signature generated by Sign is $V_a = (H, (S_1, T_1), (S_2, T_2), \ldots, (S_q, T_q))$. These three signature types are exactly the same. Therefore, the signature can be retransformed again by the proxy signer. Therefore, our scheme is multi-use.

2) TRANSPARENT

In a transparent scheme, users can not judge the signature is generated by ReSignUnblind or Sign. Because the type of the signature generated by ReSignUnblind and Sign are the same, $(H, (S_1, T_1), (S_2, T_2), \ldots, (S_q, T_q))$. Therefore, our scheme is transparent.

3) MESSAGE BLINDNESS

If the proxy signer cannot get the message m in a blind proxy re-signature scheme, the scheme is message blindness.

In the SignBlind algorithm of our scheme, the blinding proxy signer runs hash operation on the message m. He sends the blinded message $\varepsilon = H_1(m)$ to the proxy signer. Because of the unidirectional of the hash operation, the proxy signer cannot get the message m from the blinded message ε. Therefore, our scheme is message blindness.

4) DELEGATEE ANONYMITY

If the proxy signer cannot get the public key of delegatee in a blind proxy re-signature scheme, the scheme is delegatee anonymity.

In the SignBlind algorithm of our scheme, the blinding proxy signer selects M^* and N^* as follows:

$$\begin{aligned} M^*:\ & M^*(y_1, y_2, \ldots, y_u) = (y_1', y_2', \ldots, y_u') \\ N^*:\ & N^*(x_1', x_2', \ldots, x_n') = (x_1, x_2, \ldots, x_n). \end{aligned}$$

The blinding proxy signer computes

$$\begin{aligned} A^* &= A_a \\ B^* &= M^{*-1} \circ B_a \circ N^{*-1}. \end{aligned}$$

The blinding proxy signer outputs the pair of blinded Alice's public keys $pk_a^* = (A^*, B^*)$.

If the proxy signer wants to get the public key of the delegatee $pk_a = (A_a, B_a)$, he must solve the equation as follows:

$$B^* = M^{*-1} \circ B_a \circ N^{*-1}.$$

According to the analysis, we have

$$B_a = S_a \circ A_a \circ T_a$$
$$A^* = A_a,$$

So, we can get

$$B^* = M^{*-1} \circ S_a \circ A^* \circ T_a \circ N^{*-1} = M^{*-1} \circ B_a \circ N^{*-1}.$$

If the proxy signer wants to get (S_a, T_a), he must first get $(M^{*-1} \circ S_a, T_a \circ N^{*-1})$. The difficulty of solving this problem is same with solving the IP problem.

Due to the difficulty of the IP problem, the proxy signer cannot get the public key of the delegatee. Therefore, the scheme is delegatee anonymity.

5) QUANTUM RESISTANCE

According to the analysis of the attack for the multivariate public key cryptosystem, the proposed scheme can resist quantum attack if the computational complexity of the attack to signature is over 2^{80}.

Our scheme is based on the multivariate public key cryptosystem. According to the analysis of the attack for IP signature scheme [17], the attack computational complexity is over 2^{80} that required by the security level, where the parameters are $K = GF(2^8)$, $n = 18$, $u = 10$, and $q = 64$. Therefore, the scheme can resist quantum attack.

6) SECURITY MODEL

We list the based security model of all schemes in the comparison. Schemes [7], [9], [10], [15] are secure in the standard model (SM). Our scheme and schemes [6], [11] are secure in the Random oracle model (ROM). In the standard model, the adversary is only limited by the amount of time and computational power available and there is no other assumption. Security proofs are notoriously difficult to achieve in the standard model, so in many practices, cryptographic primitives are replaced by the random oracle model. In addition, the scheme based on ROM is more efficient.

VII. CONCLUSION

In this paper, we proposed a blind proxy re-signature scheme based on Isomorphisms of Polynomials in the multivariate public key cryptosystems. We proved our scheme is correct, consistent and unforgeable. Meanwhile, our scheme is with the properties of multi-use, transparency, message blindness, delegatee anonymity, and quantum resistance. Therefore, it has more advantages of safety and efficiency in the low-power hardware and the environment under quantum computers attack. However, compared with the existing schemes, our scheme is large in the size of public key and private key, which is the inherent defect of the multivariate public key cryptosystems. Hence, we will focus on the blind proxy re-signature scheme based on the multivariate public key cryptosystems with smaller size of key in the future work.

REFERENCES

[1] N. Agarwal, A. Rana, and J. P. Pandey, "Proxy signatures for secured data sharing," in *Proc. 6th Int. Conf. Cloud Syst. Big Data Eng. (Confluence)*, Jan. 2016, pp. 255–258.

[2] B. Libert, F. Mouhartem, and K. Nguyen, "A lattice-based group signature scheme with message-dependent opening," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, 2016, pp. 137–155.

[3] M. Severens, J. Farquhar, J. Duysens, and P. Desain, "A multi-signature brain–computer interface: Use of transient and steady-state responses," *J. Neural Eng.*, vol. 10, no. 2, p. 026005, 2013.

[4] C. Lin, F. Zhu, W. Wu, K. Liang, and K.-K. R. Choo, "A new transitive signature scheme," in *Proc. Int. Conf. Netw. Syst. Secur.*, 2016, pp. 156–167.

[5] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1403. Berlin, Germany: Springer, 1998, pp. 127–144.

[6] G. Ateniese and S. Hohenberger, "Proxy re-signatures: New definitions, algorithms, and applications," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2005, pp. 310–319.

[7] J. Shao, Z. Cao, L. Wang, and X. Liang, "Proxy re-signature schemes without random oracles," in *Proc. Int. Conf. Cryptol. India*, Chennai, India, Dec. 2007, pp. 197–209.

[8] B. Waters, "Efficient identity-based encryption without random oracles," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 2005, pp. 114–127.

[9] B. Libert and D. Vergnaud, "Multi-use unidirectional proxy re-signatures," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2008, pp. 511–520.

[10] S. S. Vivek, S. D. Selvis, G. Balasubramanian, and C. P. Rangan, "Strongly unforgeable proxy re-signature schemes in the standard model," Cryptol. ePrint Arch., IACR, Las Vegas, NV, USA, 2012, vol. 2012, p. 80.

[11] X. A. Wang, "Proxy re-signature supporting conditional delegation," in *Proc. 10th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput. (3PGCIC)*, Nov. 2015, pp. 844–848.

[12] L. Chen, X. Chen, Y. Sun, and X. Du, "A new certificateless proxy re-signature scheme in the standard model," in *Proc. 7th Int. Symp. Comput. Intell. Design (ISCID)*, Dec. 2014, pp. 202–206.

[13] X. D. Yang, L. Zhang, and C. F. Wang, "A flexible threshold proxy re-signature scheme with provable security," *Comput. Eng. Sci.*, vol. 36, no. 7, pp. 1250–1254, 2014.

[14] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*. New York, NY, USA: Springer, 1983, pp. 199–203.

[15] Y. Deng, M. Du, Z. You, and X. Wang, "A blind proxy re-signatures scheme based on standard model," *J. Electron. Inf. Technol.*, vol. 32, no. 5, pp. 1119–1223, 2010.

[16] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, 1999.

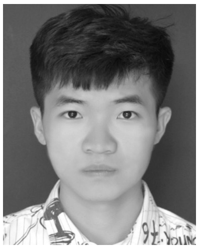
[17] S. Tang and L. Xu, "Proxy signature scheme based on isomorphisms of polynomials," in *Network and System Security* (Lecture Notes in Computer Science), vol. 7645. Heidelberg, Germany: Springer, 2012, pp. 113–125.

[18] J. Patarin, "Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms," in *Advances in Cryptology* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 1996, pp. 33–48.

[19] L. Chen, Z. Cheng, and N. P. Smart, "Identity-based key agreement protocols from pairings," *Int. J. Inf. Secur.*, vol. 6, no. 4, pp. 213–241, 2007.

LI HUIXIAN was born in Ulanhot, China, in 1977. She received the B.S. and M.S. degrees in computer science from Xidian University, China, in 2000 and 2003, respectively, and the Ph.D. degree in computer application technology from the Dalian University of Technology, China, in 2006.

From 2006 to 2008, she holds a post-doctoral position with the School of Computer Science and Engineering, Northwestern Polytechnical University, China. Since 2008, she joined the School of Computer Science and Engineering, Northwestern Polytechnical University, as an Assistant Professor. Her main research interests focus on information security, cryptography, secure protocols, and cloud computing.

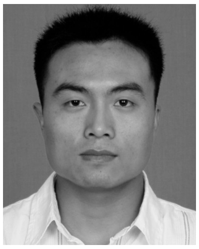


HAN ZHIPENG was born in Shenyang, Liaoning, China, in 1994. He received the B.S. degree in computer science and technology from Northwestern Polytechnical University, Xi'an, in 2016, where he is currently pursuing the master's degree in network and information security.

His research interests include cloud computing, homomorphic encryption, and privacy protection.

WANG LIQIN was born in Yulin, Shaanxi, China, in 1992. She received the B.S. degree in computer science and technology from the Xi'an University of Finance and Economics, Shaanxi, in 2016. She is currently pursuing the M.S. degree in software engineering with the School of Software and Microelectronics, Northwestern Polytechnical University.

Her research interests include information security and attribute-based access control for cloud computing.



PANG LIAOJUN (M'09) was born in Weinan, Shaanxi, China, in 1978. He received the bachelor's and master's degrees in computer science and technology and the Ph.D. degree in cryptography from the Xidian University of China, in 2000, 2003, and 2006, respectively.

He is currently a Full Professor with the State Key Laboratory of Integrated Services Networks, Xidian University, and at the same time he was a Visiting Scholar at the Department of Computer Science, Wayne State University, Detroit, MI, USA. His research interests include Internet security, cryptography, secure mobile agent system, and e-commerce security technology.