Telecommunication Systems (2018) 69:217–222
https://doi.org/10.1007/s11235-018-0458-9

# ID-based proxy re-signature without pairing

**Zhiwei Wang**[1] · **Aidong Xia**[1] · **Mingjun He**[1]

Published online: 18 April 2018


**Abstract**
Proxy re-signature is a powerful cryptographic primitive, in which a proxy acts as a translator converts Alice's signature into Bob's signature by using the re-signature key. Proxy re-signature is a very useful tool for the interoperable DRM architecture and the passed path proof in cloud computing. However, the number of cloud users is very huge, so it is unsuitable to construct PKI in cloud computing. Moreover, the cloud users are usually mobile devices, which are constrained with processing and power limitations, and pairing is a very costly operation to them. Thus, ID-based proxy re-signature without pairing is an attractive issue for the applications in cloud computing. In this paper, based on Chai et al's ID-based signature from quadratic residues, we propose the first unidirectional and single-use ID-based proxy re-signature, which is existential unforgeable in the random oracle model based on the factoring assumption.

**Keywords** Proxy re-signature · ID-based signature · Resource-limited user · Factoring · Random oracle

## 1 Introduction

Proxy re-signature is a novel cryptographic primitive, which allows a proxy transform Alice's(delegatee) signature to Bob's(delegator) signature on the same message by using the re-signature key. With the development of cloud computing, many secure problems have been proposed, such as constructing the interoperable DRM architecture in cloud computing, and proving the passed path that has been taken, since the cloud server is not so creditable to be given the user's private key. Proxy re-signature is a good solution to these problems, in which a semi-trusted proxy (cloud server) act as a translator between Alice and Bob. Proxy re-signature was introduced by Blaze et al.(BBS) [1] in 1998, and Ateniese and Hohenberger [2] formalized it in 2005. After then, some proxy re-signature schemes have been proposed [3–5]. ID-based cryptography, proposed by Shamir [6], eliminates the necessity for the public key certificates. Identity-based cryptography could particularly be suitable for cloud computing. Since the number of cloud users is very huge, the absence of certificate can greatly eliminate the costly certificate verification process. In an ID-based proxy re-signature scheme, the signature can be verified by delegatee's or delegator's identity information. To our knowledge, Shao et al. [7] firstly proposed an ID-based proxy re-signature scheme in 2011. In general, there are eight properties for proxy re-signature [7].

*Unidirectional* We call that a proxy re-signature scheme is an unidirectional scheme, on the condition that the re-signature key allows proxy to transform A's signature to B's, but B's signature cannot be transformed to A's.
*Multi-use* If the signature can be re-signed for multi-times, then we call that the proxy re-signature scheme is a multi-use scheme.
*Private proxy* If the re-signature key should be kept secretly by an honest proxy, then we call that the proxy re-signature scheme is a private proxy scheme.
*Transparent* If a user cannot know whether a proxy exists in a scheme, then the proxy re-signature scheme is a transparent scheme. In a transparent scheme, the re-signature cannot be distinguished whether it is transformed by a proxy or generated by a signer.
*Key-optimal* If a user only needs to keep a small number of secret keys regardless of how many re-signature processes he attends, then we call that the proxy re-signature scheme is a key-optimal scheme.
*Non-interactive* If the delegatee's secret key is not used to compute the re-signature key, then the scheme is a non-interactive scheme.
*ID-based* If the user's private key is generated from user's identity information, and the signature should be verified

✉ Zhiwei Wang
zhwwang@njupt.edu.cn

[1] School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, Jiangsu, China

**Table 1** We compare the properties of several ID-based proxy re-signature schemes discussed in this work

| Property | Shao's scheme | $S_{ia}$ | $S_{Non-ia}$ |
|---|---|---|---|
| Unidirectional | Yes | Yes | Yes |
| Multi-use | Yes | No | No |
| Private proxy | Yes | Yes | Yes |
| Transparent | Yes | Yes | Yes |
| Key-optimal | Yes | Yes | Yes |
| Non-interactive | Yes | No | Yes |
| ID-based | Yes | Yes | Yes |
| Pairing-free | No | Yes | Yes |

by the user's identity, then the proxy re-signature scheme is an ID-based scheme.

*Pairing-free* If pairing is not used in the construction of the proxy re-signature scheme, then we call that the scheme is pairing free.

We compare our ID-based scheme and Shao et al.'s ID-based scheme in terms of the satisfied properties Table 1. We denote our scheme (interactive version) and our scheme (Non-interactive version) as $S_{ia}$ and $S_{Non-ia}$ respectively.

Although Shao et al.'s scheme [7] is ID-based, it requires three pairings in the verify algorithm. To our knowledge, pairings are very costly operation when compared to the other operations in the base group [8].

### 1.1 Our contribution

In this paper, we propose an unidirectional and single-use ID-based proxy re-signature scheme from quadratic residues with two versions, the one is interactive, and the other is Non-interactive. Our construction is based on Chai et al.'s ID-based signature scheme from quadratic residues [9]. We revise the security model of ID-based proxy re-signature, and give the security proof of our scheme in the random oracle model based on the factoring assumption.

### 1.2 Organization

This paper is organized as follows. In Sect. 2, we describe some preliminaries. In Sect. 3, we give the definition and security model of unidirectional and single-use ID-based proxy re-signature. In Sect. 4, we present two versions of ID-based proxy re-signature scheme from quadratic residues. In Sect. 5, we prove that our scheme is secure under random oracles. Finally, we draw the conclusion in Sect. 6.

## 2 Preliminaries

### 2.1 Complexity assumptions

**Factoring problem.** *Takes as input a composite modulus $N$, which is a multiple of two large primes $p$ and $q$, and outputs $p$ or $q$.*

**Factoring Assumption.** *$k$ is a security parameter. An integer $N$ is a multiple of two $k$-bit, large odd primes $p, q$. Takes as input $N$, we call that the $(t_R, \epsilon_R)$-factoring assumption holds on the condition that all $t_R$-time adversaries $\mathcal{A}$*

$$Pr[(p, q) \leftarrow \mathcal{A}(N), pq = N] \leq \epsilon_R,$$

*where the probability is over the random coins of $\mathcal{A}$.*

### 2.2 Some concepts in number theory

In this section, we would review some concepts in number theory which are adopted from [9]. Let $\mathbb{Q}_N$ denote the subgroup of squares in $\mathbb{Z}_N^*$. Then, $\mathbb{Q}_N$ is a cyclic group with the order $\phi(N)/4 = (p-1)(q-1)/4$ [10].

**Theorem 2.1** *Let $a \in \mathbb{Q}_N$, $N = p \times q$, where $p, q$ are large primes. Then $a^{2d} \equiv a \pmod N$, where $d = (N - p - q + 5)/8$.*

***Proof*** The proof is omit.

Indeed, Theorem 2.1 gives us a way to compute a square root of a quadratic residue $a \in \mathbb{Q}_N$.

If $s_1$ and $s_2$ are two square roots satisfying $s_1 \neq \pm s_2 \pmod N$, then $N$ could be factored by computing $GCD(s_1 + s_2, N)$ or $GCD(s_1 - s_2, N)$. However, if $s_1 = \pm s_2 \pmod N$, it is useless to the factorization of $N$. Thus, if takes as input two distinct square roots, then we can factor the composite modulus $N$, and the probability is $1/2$. □

## 3 ID-based proxy re-signature

### 3.1 Definition

An unidirectional and single-use ID-based Proxy Re-Signature scheme consists of six algorithms, namely, *Setup*, *Extract*, *Rekey*, *Sign*, *ReSign* and *Verify*. We define these algorithms as follows:

– The tuple of (*Setup*,*Extract*,*Sign*, *Verify*) is a standard ID-based signature scheme.

– Takes as input $(ID_A), sk_A^*, ID_B, sk_B)$, the *Rekey* algorithm outputs a key $rk_{A\to B}$ for the proxy. Here, $rk_{A\to B}$ allows to transform A's signature to B's signature. A is the delegatee, and B is the delegator. $sk_A^*$ is optional in the input. If $sk_A^*$ is included in the input, then the scheme is an interactive scheme. Otherwise, it is a Non-interactive scheme.
– On input $rk_{A\to B}$, A's identity information $ID_A$, a signature $\sigma_A$ and a message $m$, the *ReSign* algorithm, outputs $\sigma_B$ on the same message $m$, if

$$Verify(ID_A, m, \sigma_A) = 1.$$

*Correctness* For any message $m$ and two key pairs $(ID_A, sk_A)$ and $(ID_B, sk_B)$, let $rk_{A\to B} \leftarrow Rekey(ID_A), sk_A^*, ID_B, sk_B)$, an ID-based proxy re-signature scheme should satisfy the following two properties:

(1) $Verify(\sigma_A, m, ID_A) = 1$;
(2) $Verify(ReSign(\sigma_A, rk_{A\to B}, ID_A), m, ID_B) = 1$.

### 3.2 Security model

In this section, we define the existential unforgeability of unidirectional and single-use ID-based proxy re-signature (USIPRS) through a security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:

*Setup* The challenger runs the *Setup* algorithm, and obtains the master public key $mpk$ and master secret key $msk$. It gives $mpk$ to the adversary, and keeps $msk$ secretly.
*Queries* The adversary $\mathcal{A}$ can make some different queries for polynomial times as follows:

– Extract queries: When the adversary $\mathcal{A}$ makes a query on an identity $ID$, the challenger $\mathcal{C}$ runs the *Extract* algorithm by using $msk$, and returns the private key $sk$ to $\mathcal{A}$.
– Re-signature key queries: While the adversary $\mathcal{A}$ makes a query on $(ID_A, ID_B)$, the challenger $\mathcal{C}$ returns the re-signature key $rk_{A\to B} = Rekey(ID_A, ID_B, Extract(msk, ID_B))$.
– Re-signature queries: When the adversary $\mathcal{A}$ makes a query on $(\sigma_A, m, ID_A, ID_B)$, the challenger $\mathcal{C}$ returns the re-signature

$$\sigma_B = ReSign(\sigma_A, m, ID_A, \times Rekey(ID_A, ID_B, Extract(msk, ID_B))).$$

– Signature queries: When the adversary $\mathcal{A}$ makes a query on $(ID, m)$, the challenger $\mathcal{C}$ returns the signature $\sigma = Sign(Extract(ID, msk), ID, m)$.

*Forgery* Finally, the adversary $\mathcal{A}$ outputs a message $m^*$, an identity $ID^*$, and a signature $\sigma^*$. We say $\mathcal{A}$ wins the game if all the following conditions are all satisfied:

1. $Verify(\sigma^*, m^*, ID^*) = 1$
2. $ID^*$ has not been queried to the *Extract* oracle.
3. $(ID^*, m^*)$ has not been queried to the *Sign* oracle.
4. The adversary $\mathcal{A}$ has not made a re-signature query on $(\sigma_i, m^*, ID_i, ID^*)$ for any identity $ID_i$.

We define $ADV^{ES}_{USIPRS}(1^k)$ to be the probability that $\mathcal{A}$ wins the above game. We say that an USIPRS scheme is existentially unforgeability with respect to adaptive chosen message and identity attacks if for all ppt adversaries $\mathcal{A}$, $ADV^{ES}_{USIPRS}(1^k)$ is negligible.

## 4 ID-based proxy re-signature scheme without pairing

In this section, We present the first unidirectional and single-use ID-based proxy re-signature scheme without pairing from Chai et al.'s ID-based signature scheme [9].

### 4.1 Interactive version

We first present an interactive version which means that the the re-signature key $rk_{A\to B}$ is computed from Bob(delegator)'s secret key $sk_B$ and Alice(delegatee)'s secret key $sk_A$, i.e., the delegatee should participate in the delegation process.

– $Setup(1^k)$: This algorithm selects two large safe $k/2$-bit primes $p$ and $q$, and computes $N = pq$. After then, it computes $d = (N - p - q + 5)/8$. Following that, it selects a hash function $H() : \{0,1\}^* \to \mathbb{Q}_N$, where $\mathbb{Q}_N$ is a subgroup of squares in $\mathbb{Z}_N^*$. It also chooses another common hash function $h() : \{0,1\}^* \to \{0,1\}^l$, where $l$ is security parameter for common hash function. At last, set $mpk = (N, H(), h()), msk = (p, q, d)$.
– $Extract(ID, msk, mpk)$: The algorithm takes as input the user's identity information $ID$, the master secret key $msk$, and the master public key $mpk$, and outputs the user's secret key as $sk = H(ID)^d \mod N$. (Note: From **Theorem** 2.1, we can deduce that $sk$ is a square root of $H(ID)$.)
– $Rekey(sk_A, sk_B)$: The algorithm takes as input Alice (delegatee) and Bob(delegator)'s secret keys $sk_A, sk_B$, and outputs the re-signature key as $rk_{A\to B} = \frac{sk_B}{sk_A} \mod N$.
– $Sign(m, sk, mpk)$ The algorithm takes as input a message $m$, the user's secret key $sk$, and the master public key $mpk$. Then, it selects a random number $r \in \mathbb{Z}_N^*$, and

computes $R = r^2 \mod N$. Following that, it computes $\sigma = r \cdot sk^{h(m)} \mod N$. Finally, it outputs the signature as $sig = (\sigma, R, ID)$.

- *ReSign*$(m, sig_A, ID_A, mpk, rk_{A\to B})$: The algorithm takes as input A's signature $sig_A$ on a message $m$, A's identity $ID_A$, the master public key $mpk$ and the re-signature key $rk_{A\to B}$. If $\sigma_A^2 = R \cdot H(ID_A)^{h(m)} \mod N$ holds, then it outputs the re-signature as $sig_B = (\sigma_A \cdot rk_{A\to B}^{h(m)}, R, ID_B)$.
- *Verify*$(m, sig)$: The algorithm takes as input a signature $sig = (\sigma, R, ID)$ on a message $m$. If $\sigma^2 = R \cdot H(ID)^{h(m)} \mod N$ holds, then the algorithm outputs "Valid"; otherwise, outputs "Invalid".

*Correctness* The interactive version can be proved correctly by the following equations:

(1)

$$\begin{aligned}\sigma^2 &= (r \cdot sk^{h(m)})^2 \mod N \\ &= R \cdot H(ID)^{h(m)} \mod N\end{aligned}$$

(2)

$$\begin{aligned}\sigma_A \cdot rk_{A\to B}^{h(m)} &= \sigma_A \cdot \left(\frac{sk_B}{sk_A}\right)^{h(m)} \mod N \\ &= r \cdot sk_A^{h(m)} \cdot \left(\frac{sk_B}{sk_A}\right)^{h(m)} \mod N \\ &= r \cdot sk_B^{h(m)} \mod N \\ &= \sigma_B\end{aligned}$$

### 4.2 Non-interactive version

In this section, we present a Non-interactive version which means that the the re-signature key $rk_{A\to B}$ is computed from Bob(delegator)'s secret key $sk_B$ and Alice(delegatee)'s identity $ID_A$, i.e., the delegatee does not require to participate in the delegation process.

- *Setup*$(1^k)$: This algorithm is the same as the interactive version.
- *Extract*$(ID, msk, mpk)$: This algorithm is also the same as the interactive version.
- *Rekey*$(ID_A, sk_A, ID_B, sk_B)$: The algorithm takes as input Alice(delegatee)'s identity $ID_A$ and Bob(delegator)'s secret keys $sk_B$, and outputs the re-signature key as $rk_{A\to B} = \frac{sk_B}{H(ID_A)} \mod N$.
- *Sign*$(m, sk, mpk)$ This algorithm is also the same as the interactive version.
- *ReSign*$(m, sig_A, ID_A, mpk, rk_{A\to B})$: The algorithm takes as input A's signature $sig_A$ on a message $m$, A's identity $ID_A$, the master public key $mpk$ and the re-signature key $rk_{A\to B}$. If $\sigma_A^2 = R \cdot H(ID_A)^{h(m)} \mod N$ holds, then it selects a random number $r' \in \mathbb{Z}_N^*$, and computes $R' = (r'R)^2 \mod N$. Finally, it outputs the re-signature as $sig_B = (r' \cdot \sigma_A^2 \cdot rk_{A\to B}^{h(m)}, R', ID_B)$.
- *Verify*$(m, sig)$: The algorithm takes as input a signature $sig = (\sigma, R, ID)$ on a message $m$. If $\sigma^2 = R \cdot H(ID)^{h(m)} \mod N$ holds, then the algorithm outputs "Valid"; otherwise, outputs "Invalid".

*Correctness* The Non-interactive version can be proved correctly by the following equation:

$$\begin{aligned}r' \cdot \sigma_A^2 \cdot rk_{A\to B}^{h(m)} &= r' \cdot R \cdot H(ID_A)^{h(m)} \cdot \left(\frac{sk_B}{H(ID_A)}\right)^{h(m)} \mod N \\ &= r' \cdot R \cdot sk_B^{h(m)} \mod N \\ &= \sigma_B \mod N\end{aligned}$$

## 5 Security proof

In this section, we provide the security proof to the interactive version of our scheme. The proof of Non-interactive version is similar, so it is omit.

**Theorem 5.1** *If the factoring assumption holds in $\mathbb{Z}_N^*$, then the interactive version of our scheme is secure under the random oracle model.*

***Proof*** We assume that there exists an attacker $\mathcal{A}$ can break our scheme with non-negligible probability. Then, we can construct an algorithm $\mathcal{B}$ to solve the factoring problem. The input of $\mathcal{B}$ is $N = pq$ ($\mathcal{B}$ doesn't know $p$ or $q$.), and $\mathcal{B}$'s challenge is to output $p$ or $q$ with non-negligible probability. $\mathcal{B}$ sends $N$ to $\mathcal{A}$ as a public parameter.

Then, $\mathcal{B}$ responses for $\mathcal{A}$'s following queries to simulate the real world for $\mathcal{A}$.

- *H() queries*: To response the H() queries of $\mathcal{A}$, $\mathcal{B}$ should maintain a $T_H$ table, and each entry in the table is a tuple of $(ID, H, s)$. If $\mathcal{A}$ queries for an identity $ID$, then $\mathcal{B}$ searches $T_H$ for $(ID, H, s)$. If $(ID, H, s)$ has been existed in $T_H$, then $\mathcal{B}$ responses to $\mathcal{A}$ with $H$. Otherwise, $\mathcal{B}$ randomly chooses $s \in \mathbb{Z}_N^*$, and returns $H = s^2 \mod N$ as the answer. Then, $\mathcal{B}$ adds the new tuple $(ID, H, s)$ to $T_H$.
- *h() queries*: To response the h() queries of $\mathcal{A}$, $\mathcal{B}$ should maintain a $T_h$ table, and each entry in the table is a tuple of $(m, h)$. If $\mathcal{A}$ queries for a message $m$, then $\mathcal{B}$ searches $T_h$ for $(m, h)$. If $(m, h)$ has been existed in $T_h$, then $\mathcal{B}$ responses to $\mathcal{A}$ with $h$. Otherwise, $\mathcal{B}$ randomly chooses

$h \in \mathbb{Z}_N^*$, and returns $h$ as the answer. Then, $\mathcal{B}$ adds the new tuple $(m, h)$ to $T_h$.

- *Extract queries* If $\mathcal{A}$ makes an extract query for an identity $ID$, then $\mathcal{B}$ searches $T_H$ for $(ID, H, s)$ as in **H() queries**. If $(ID, H, s)$ has been existed in $T_H$, then $\mathcal{B}$ returns $s$ as the secret key to $\mathcal{A}$. If not exists, then $\mathcal{B}$ adds a new tuple $(ID, H, s)$ to $T_H$ as in **H() queries**, and returns $s$ as the answer.
- *Re-signature key queries* If $\mathcal{A}$ queries the re-signature key for $ID_A$(delegatee) and $ID_B$(delegator), then $\mathcal{B}$ searches $T_H$ as in **H() queries**. If $(ID_A, H_A, s_A)$ and $(ID_B, H_B, s_B)$ have been existed in $T_H$, then $\mathcal{B}$ returns $s_A/s_B \mod N$ as the re-signature key to $\mathcal{A}$. Otherwise, $\mathcal{B}$ adds new $(ID_A, H_A, s_A)$ or $(ID_B, H_B, s_B)$ to $T_H$ as in **H() queries**, and returns $s_A/s_B \mod N$ as the answer.
- *Signature queries* If $\mathcal{A}$ makes a signature query for an identity $ID$ and a message $m$, then $\mathcal{B}$ first searches $T_H$ and $T_h$ for $(ID, H, s)$ and $(m, h)$ respectively. After then, $\mathcal{B}$ randomly selects $r \in \mathbb{Z}_N^*$, and computes $\sigma = r \cdot s^h \mod N$. Finally, $\mathcal{B}$ returns $(\sigma, r^2 \mod N, ID)$ as the answer.
- *Re-signature queries* If $\mathcal{A}$ makes a re-signature query for the signature $sig_A = (\sigma_A, R, ID_A)$ on a message $m$, then $\mathcal{B}$ firstly answers the re-signature query on $(ID_A, ID_B)$ to obtain the re-signature key $s_A/s_B \mod N$. Following that, $\mathcal{B}$ returns $(\sigma_A \cdot s_A/s_B \mod N, R, ID_B)$ as the answer.

When the above game is over, $\mathcal{A}$ outputs a forgery $sig^* = (\sigma^*, R^*, ID^*)$ on message $m^*$, where $ID^*$ and $m^*$ have not been queried to the *Extract* oracle and *Sign* oracle, and the adversary $\mathcal{A}$ has not made a re-signature query on $(\sigma_i, m^*, ID_i, ID^*)$ for any identity $ID_i$. For factoring $N$, we should reset the game between $\mathcal{A}$ and $\mathcal{B}$ by using the rewind technology [11]. In the second game, we use the same random number, and $H()$ oracle gives the same answer. However, the $h()$ oracle provides the different answer. If $\mathcal{B}$ answers $h^*$ in the first game, then he returns $h^* + 1$ in the second game. Finally, $\mathcal{B}$ obtains two different signatures $(\sigma^*, R^*, ID^*)$ and $(\sigma', R^*, ID^*)$ on the same message $m^*$. Due to the same random number, these two signatures satisfy the following equation:

$$\sigma^{*2}/H(ID^*)^{h^*} = \sigma'^2/H(ID^*)^{h^*+1} \mod N$$
$$\Rightarrow H(ID^*) = \left(\frac{\sigma'}{\sigma^*}\right)^2 \mod N$$

Thus, from the above equation, $\frac{\sigma'}{\sigma^*}$ is a square root of $H(ID^*)$. $\mathcal{B}$ also can find another root of $H(ID^*)$ in the table $T_H$. So, if these two roots are different, then $\mathcal{B}$ can factor $N$ by computing $GCD(s_1 + s_2, N)$ or $GCD(s_1 - s_2, N)$. Otherwise, $\mathcal{B}$ outputs "failure". From the above game, we can see that $s$ is chosen by $\mathcal{B}$, which is independent from $H(ID^*)$. So, the probability that two square roots are different is $1/2$. □

## 6 Conclusions

ID-based proxy re-signature is very suitable for many applications in cloud computing. However, cloud users are usually resource constrained. So, pairing free scheme is a good solution. In this paper, we propose the first ID-based proxy re-signature scheme from quadratic residues. We proved that it is existential unforgeability under factoring assumption in the random oracle model. However, our scheme is unidirectional and single-use, which has some limitations in the applications. We plan to research bidirectional and multi-use scheme in the future.

**Acknowledgements** This research is supported by the National Natural Science Foundation of China under Grant No.61373006,61672016.

## References

1. Blaze, M., Bleumer, G., & Strauss, M. (1998). Divertible protocols and atomic proxy cryptography. In *EUROCRYPT 1998*. LNCS (Vol. 1403, pp. 127–144).
2. Ateniese, G., & Hohenberger, S. (2005). Proxy re-signatures: New definitions, algorithms, and applications. In *ACM CCS 2005* (pp. 310–319).
3. Libert, B., & Vergnaud, D. (2008). Multi-use unidirectional proxy re-signatures. In *ACM CCS 2008* (pp. 511–520).
4. Shao, J., Feng, M., Zhu, B., Cao, Z., & Liu, P. (2010). The security model of unidirectional proxy re-signature with private re-signature key. In *ACISP 2010*. LNCS (Vol. 6168, pp. 216–232).
5. Yang, P., Cao, Z., & Dong, X. (2011). Threshold proxy re-signature. *Journal of Systems Science and Complexity*, *24*, 816–824.
6. Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84, volume 196 of lecture notes in computer science* (pp 47–53). Springer.
7. Shao, J., Wei, G., Ling, Y., & Xie, M. (2011). Undirectional identity-based proxy re-signature. In *Proceeding of IEEE ICC 2011* (pp. 1–5).
8. Lauter, K., Montgomery, P. L., & Naehrig, M. (2010). An analysis of affine coordinates for pairing computation. In M. Joye, A. Miyaji, & A. Otsuka (Eds.), *Pairing 2010*. LNCS (Vol. 6487, pp. 1–20). Heidelberg: Springer.
9. Chai, Zhenchuan, Cao, Zhenfu, & Dong, Xiaolei. (2007). Identity-based signature scheme based on quadratic residues. *Science in China Series F: Information Sciences*, *50*(3), 373–380.
10. Shoup, V. (2005). *A computational introduction to number theory and algebra* (p. 534). Cambridge: Cambridge University Press.
11. Bellare, M., & Palacio, A. (2002). GQ and Schnoor identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *Crypto*. LNCS (Vol. 2442, pp. 162–177).

**Zhiwei Wang** received his Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications, Beijing in 2009. Currently, he is an associate professor in the department of information security at Nanjing University of Posts and Telecommunications. His research interests include digital signatures, provable security, cryptographic protocols, and network and cloud security. He has published over 40 papers at prestigious journals and conferences.

**Aidong Xia** is a master student of Nanjing University of Posts and Telecommunications. His research direction is cryptography and information security.

**Mingjun He** is a master student of Nanjing University of Posts and Telecommunications. Her research direction is cloud security.