Identity Based Proxy Signature from RSA without Pairings

Lunzhi Deng, Huawei Huang, and Yunyun Qu
(Corresponding author: Lunzhi Deng)
School of Mathematics Science, Guizhou Normal University
Guiyang 550001, China
(Email: denglunzhi@163.com)
(Received Sept. 16, 2015; revised and accepted Jan. 23 & Mar. 29, 2016)

Abstract

RSA is a key cryptography technique and provides various interfaces for the applied software in real-life scenarios. Although some good results were achieved in speeding up the computation of pairing function in recent years, the computation cost of the pairings is much higher than that of the exponentiation in a RSA group. So it is still interesting to design efficient cryptosystems based on RSA primitive. A proxy signature scheme allows a proxy signer to sign messages on behalf of an original signer within a given context. Most identity based proxy signature schemes currently known employ bilinear pairings. In this paper, an identity based proxy ring signature (IBPS) scheme from RSA without pairings is constructed, and the security is proved under the random oracle model.
Keywords: Identity based cryptography, proxy signature, RSA

1 Introduction

Public key cryptography is an important technique to realize network and information security. In traditional public key infrastructure, each user generates his own secret and public keys [13]. A certification authority must sign a digital certificate which links the identity of the user and his public key. The validity of this certificate must be checked before using the public key of the user, when encrypting a message or verifying a signature. Obviously, the management of digital certificates decreases the efficiency of practical implementations of public key cryptosystem. To solve the problem, Shamir [21] defined a new public key paradigm called identity-based public key cryptography. In this system, each user needs to register at a trusted private key generator (PKG) with identity of himself before joining the network. Once a user is accepted, the PKG will generate a private key for the user and the user's identity (e.g., user's name or email address) becomes the corresponding public key. In order to verify a digital signature or send an encrypted message, a user needs to only know the identity of communication partner and the public key of the PKG [20].

Shamir [21] proposed an identity-based signature scheme from the RSA primitive. Guillou and Quisquater [6] proposed a similar RSA identity-based signature scheme, which is constructed from a zero-knowledge identification protocol. Herranz [8] proposed an identity-based ring signatures from RSA whose security is based on the hardness of the RSA problem. After initial schemes, the following breakthrough result in the area of identity-based cryptography came in 2001, when Boneh and Franklin [2] designed an efficient identity-based public key encryption scheme. In the design, they used as a tool bilinear pairings, a kind of maps which can be constructed on some elliptic curves. Using bilinear pairings, a lot of identity-based schemes have been proposed for encryption, signature, key agreement, etc. However, it is still desirable to find schemes for identity-based scenarios which do not need to employ bilinear pairings.

The concept of proxy signatures was first introduced by Mambo et al. [16]. Based on the delegation type, proxy signature schemes are classified into three types: full delegation, partial delegation and delegation by warrant. In a full delegation scheme, the original signer's private key is given to the proxy signer. Hence the proxy signer has the same signing right as the original signer. Obviously, such schemes are impractical and insecure for most of real-world settings. In a partial delegation scheme, a proxy signer has a new key, called proxy private key, which is different from the original signer's private key. Although proxy signatures generated by using proxy private key are different from the original signer's standard signatures, the proxy signer is not limited on the range of messages he can sign. This weakness is eliminated in delegation by warrant schemes. One of the main advantages of the use of warrants is that it is possible to include any type of security policy (that specifies what kinds of messages are delegated, and may contain other information, such as the identities of the original signer and the proxy signer, the

delegation period, etc.) in the warrant to describe the restrictions under which the delegation is valid. Therefore, proxy signature schemes which use the method of this approach attract a great interest, and it is often expected that new proxy signature schemes will implement the functionality of warrants.

In order to adapt different situations, many proxy signature variants are produced, such as one-time proxy signature, proxy blind signature, multi-proxy signature, and so on. Using bilinear pairings, people proposed many new ID-based proxy signature (IBPS) scheme [1, 4, 5, 9, 10, 11, 14, 15, 17, 18, 19, 24, 25, 26]. All the above IBPS schemes are very practical, but they are based on bilinear pairings and the pairing is regarded as the most expensive cryptography primitive. Recently, He et al. [7] proposed an ID-based proxy signature schemes without bilinear pairings. Tiwari and Padhye [22] proposed a provable secure multi-proxy signature scheme without bilinear maps. Kim et al. [12] constructed a provably secure ID-based proxy signature scheme based on the lattice problems. The computation cost of the pairings is much higher than that of the exponentiation in a RSA group. Therefore, IBPS schemes based on RSA primitive would still be appealing.

2 Preliminaries

The notations of this paper are listed in the following:

- N: A large composite number, the product of two prime numbers p, q.
- b: A prime number satisfying $gcd(b, \varphi(N)) = 1$.
- p, q, a: The master key satisfying $N = pq$ and $a = b^{-1} \bmod \varphi(N)$.
- ID_i/D_i: The user's identity /private key.
- ID_o/D_o: The original signer's identity/private key.
- ID_p/D_p: The proxy signer's identity/private key.
- m_w: The warrant consisting of the identities of original signer and proxy signer, the delegation duration and so on.
- π: The proxy delegation.
- H_0, H_1, H_2: Three hash functions.

We define the RSA problem as follows.

Definition 1. *Let $N = pq$, where p and q are two k-bit prime numbers. Let b be a random prime number, greater than 2^l for some fixed parameter l, such that $\gcd(b, \varphi(N)) = 1$. Given $Y \in Z_N^*$, RSA problem is to find $X \in Z_N^*$ such that $X^b = Y \bmod N$.*

An identity based proxy signature scheme consists of the following six algorithms:

- Setup: This algorithm takes as input a security parameter k, then returns $params$ (system parameters) and a randomly chosen master secret key msk. After the algorithm is performed, the PKG publishes the system parameters $params$ and keeps the master key msk secret.
- Key extract: This algorithm takes as input $params, msk$, identity $ID_i \in \{0,1\}^*$ of an entity, and returns a private key D_i. The PKG carries out the algorithm to generate the private key D_i and send D_i to the corresponding owner ID_i via a secure channel.
- Delegate: This algorithm takes as input the $params$, original signer's private key D_o, a warrant m_w and outputs the delegation π.
- Delegation verify: This algorithm takes as input $params$, π and verifies whether π is a valid delegation from the original signer.
- Proxy sign: This algorithm takes as input the $params$, proxy signer's private key D_p, delegation π, a message m and outputs the proxy signature σ.
- Proxy signature verify: This algorithm takes as input the original signer's identity ID_o, the proxy signer's identity ID_p, a proxy signature σ, and outputs 1 if the proxy signature is valid or 0 otherwise.

Definition 2. *An identity based proxy signature scheme is unforgeable(UNF-IBPS) if no polynomially bounded adversary has a non-negligible advantage in the following game.*

Game. *Now we illustrate the game performed between a challenger $\mathscr{C}$ and a adversary $\mathscr{A}$ for an identity based proxy signature scheme.*

Initialization. *$\mathscr{C}$ runs the setup algorithm to generate a master secret key msk and the public system parameters $params$. $\mathscr{C}$ keeps msk secret and gives $params$ to $\mathscr{A}$.*

Query. *$\mathscr{A}$ performs a polynomially bounded number of queries. These queries may be made adaptively, i.e. each query may depend on the answers to the previous queries.*

- *Hash functions query: $\mathscr{A}$ can ask for the values of the hash functions for any input.*
- *Key query: When $\mathscr{A}$ requests the private key of the user ID_i, $\mathscr{C}$ responds with the private key D_i.*
- *Delegation query: When $\mathscr{A}$ submits original signer's identity ID_o and a warrant m_w to the challenger, $\mathscr{C}$ responds by running the delegate algorithm on the warrant m_w, the original signer's private key D_o.*

- *Proxy signature query: When $\mathscr{A}$ submits original signer's identity ID_o, proxy signer's identity ID_p, a warrant m_w and a message m to the challenger, $\mathscr{C}$ responds by running the proxy sign algorithm on the message m, the warrantm_w, the private keys D_o and D_p.*

Forge. *$\mathscr{A}$ outputs a tuple (π^*, ID_o) or $(m^*, m_w^*, \sigma^*, ID_o, ID_p)$. $\mathscr{A}$ wins the game, if one of the following cases is satisfied:*

Case 1: *The final output is (π^*, ID_o) and it satisfies:*

1) *π^* is a valid delegation.*
2) *$\mathscr{A}$ does not query the original signer ID_o's private key.*
3) *π^* is not generated from the delegation query.*

Case 2: *The final output is $(m^*, m_w^*, \sigma^*, ID_o, ID_p)$ and it satisfies:*

1) *σ^* is a valid proxy signature.*
2) *$\mathscr{A}$ does not query the original signer ID_o's private key.*
3) *The tuple (ID_o, ID_p, m_w^*) is not appear in delegation query.*
4) *σ^* is not generated from the proxy signature query.*

Case 3: *The final output is $(m^*, m_w^*, \sigma^*, ID_o, ID_p)$ and it satisfies:*

1) *σ^* is a valid proxy signature.*
2) *$\mathscr{A}$ does not query the proxy signer ID_p's private key.*
3) *σ^* is not generated from the proxy signature query.*

The success probability of $\mathscr{A}$ is defined as: $Succ_{\mathscr{A}}^{UNF-IBPS} = Pr[\mathscr{A}\ win]$.

3 The Proposed Scheme

Setup: Given security parameters k, a trusted private key generator (PKG) generates two random k-bit prime numbers p and q. Then it computes $N = pq$. For some fixed parameter l (for example $l = 200$), chooses at random a prime number b satisfying $2^l < b < 2^{l+1}$ and $\gcd(b, \varphi(N)) = 1$, and computes $a = b^{-1} \bmod \varphi(N)$. Furthermore, PKG chooses cryptographic hash functions described as follows: $H_0 : \{0,1\}^* \to Z_N^*$, $H_1 : \{0,1\}^* \times Z_N^* \to Z_b^*$ and $H_2 : \{0,1\}^* \times Z_N^* \times Z_N^* \to Z_b^*$. The set of public parameters is: $params = \{N, b, H_0, H_1, H_2\}$, the secret information stored by PKG is master-key=(p, q, a).

Key extract: For an identity $ID_i \in \{0,1\}^*$, PKG computes $D_i = Q_i^a, Q_i = H_0(ID_i)$ and sends D_i to the user ID_i via a secure channel.

Delegate: m_w is the warrant consisting of the identities of original signer and proxy signer, the delegation duration and so on. On input the warrant m_w, the original signer ID_o, whose private key is D_o, performs the following steps:

1) Randomly selects $A \in Z_N^*$, computes $T = A^b \bmod N$, $h = H_1(m_w, T)$.
2) Computes $R = AD_o^h \bmod N$.
3) Outputs $\pi = (m_w, T, R)$ as the delegation.

Delegation verify: To verify a delegation $\pi = (m_w, T, R)$ for an identity ID_o, the verifier performs the following steps:

1) Computes $h = H_1(m_w, T)$.
2) Checking whether $R^b = TQ_o^h \bmod N$. If the equality holds, accepts the delegation. Otherwise, rejects.

Proxy sign: For a message m, the proxy signer (whose identity is ID_p) who owns the delegation $\pi = (m_w, T, R)$ does the following:

1) Randomly selects $B \in Z_N^*$, computes $S = B^b \bmod N$, $k = H_2(m, m_w, T, S)$.
2) Computes $Z = RBD_p^k \bmod N$.
3) Outputs the signature $\sigma = (m, m_w, T, S, Z)$.

Proxy signature verify: To verify the validity of a proxy signature (where the original singer's identity is ID_o, the proxy singer's identity is ID_p), a verifier first checks whether the original signer and proxy signer conform to m_w, then performs the following steps.

1) Computes $h = H_1(m_w, T)$ and $k = H_2(m, m_w, T, S)$.
2) Checking whether $Z^b = TSQ_o^h Q_p^k \bmod N$. If the equality holds, outputs 1. Otherwise, outputs 0.

On correctness, we have $Z^b = (RBD_p^k)^b = R^b B^b D_p^{bk} = TSQ_o^h Q_p^k$.

4 Security of The Proposed Scheme

Theorem 1. *The scheme is unforgeable against the adversary in randomly oracle model if the RSA problem is hard.*

Proof. Suppose the challenger $\mathscr{C}$ receives a random instance (N, b, Y) of the RSA problem and and has to find an element $X \in Z_N^*$ such that $X^b = Y$. $\mathscr{C}$ will run $\mathscr{A}$ as a subroutine and act as $\mathscr{A}$'s challenger in the UNF-IBPS game.

Initialization. At the beginning of the game, $\mathscr{C}$ runs the setup program with the parameter k, gives $\mathscr{A}$ the system parameters params=$\{N, b, H_0, H_1, H_2\}$.

Queries. Without loss of generality, it is assumed that all the queries are distinct and $\mathscr{A}$ will ask for $H_0(ID)$ before ID is used in any other queries. $\mathscr{C}$ will set several lists to store the queries and answers, all of the lists are initially empty.

- H_0 queries: $\mathscr{C}$ maintains the list L_0 of tuple (ID_i, A_i). When $\mathscr{A}$ makes a query $H_0(ID_i)$, $\mathscr{C}$ responds as follows.

  At the $j^{th} H_0$ query, $\mathscr{C}$ set $H_0(ID^*) = Y$. For $i \neq j$, $\mathscr{C}$ randomly picks a value $A_i \in Z_N^*$ and sets $H_0(ID_i) = A_i^b$, then the query and the answer will be stored in the list L_0.

- H_1 queries: $\mathscr{C}$ maintains the list L_1 of tuple (α_i, h_i). When $\mathscr{A}$ makes a query $H_1(\alpha_i)$. $\mathscr{C}$ randomly picks a value $h_i \in Z_b^*$ and sets $H_1(\alpha_i) = h_i$, then the query and the answer will be stored in the list L_1.

- H_2 queries: $\mathscr{C}$ maintains the list L_2 of tuple (β_i, k_i). When $\mathscr{A}$ makes a query $H_2(\beta_i)$. $\mathscr{C}$ randomly picks a value $k_i \in Z_b^*$ and sets $H_2(\beta_i) = k_i$, then the query and the answer will be stored in the list L_2.

- Key extraction queries: $\mathscr{C}$ maintains the list L_K of tuple (ID_i, D_i). When $\mathscr{A}$ makes private key extraction query for identity ID_i. If $ID_i = ID^*$, $\mathscr{C}$ fails and stops. Otherwise $\mathscr{C}$ finds the tuple (ID_i, A_i) in list L_0 and returns $D_i = A_i$ to $\mathscr{A}$.

- Delegate queries: When $\mathscr{A}$ submits ID_o, ID_p and m_w to challenger. $\mathscr{C}$ outputs a delegation as follows.

  If $ID_o \neq ID^*$, $\mathscr{C}$ gives a delegation by calling the delegate algorithm. Otherwise, $\mathscr{C}$ does as follows.

  1) Randomly selects $A \in Z_N^*$ and $h \in Z_b^*$.
  2) Computes $T = A^b Q_o^{-h}$ and $R = A$.
  3) Stores the relation $h = H_1(m_w, T)$. If collision occurs, repeats Steps (1)-(3).
  4) Outputs $\pi = (m_w, T, R)$ as the delegation.

- Proxy signature queries. When $\mathscr{A}$ submits a delegation $\pi = (m_w, T, R)$ message m to the challenger. $\mathscr{C}$ outputs an identity based proxy signature as follows.

  If $ID_p \neq ID^*$, $\mathscr{C}$ gives a signature by calling the proxy sign algorithm. Otherwise, $\mathscr{C}$ does as follow.

  1) Randomly selects $B \in Z_N^*$ and $k \in Z_b^*$.
  2) Computes $S = B^b Q_p^{-k}$ and $Z = RB$.
  3) Stores the relation $k = H_2(m, m_w, T, S)$. If collision occurs, repeats Steps (1)-(3).
  4) Outputs the proxy signature $\sigma = (m, m_w, T, S, Z)$.

Forge. $\mathscr{A}$ outputs a tuple $\{\pi^* = (m_w, T, R), ID_o\}$ or $\{\sigma^* = (m, m_w, T, S, Z), ID_o, ID_p\}$. There are three cases to consider:

Case 1. The final output is $\{\pi^* = (m_w, T, R), ID_o\}$ and the output satisfies the requirement of Case 1 as defined in UNF-IBPS game.

Solve RSA problem. In fact, π^* is the signature on m_w by ID_o. By the forking lemma for generic signature scheme [3], two delegations can be generated: (m_w, T, R) and (m_w, T, R'). Where $h = H_1(m_w, T)$, $h' = H_1'(m_w, T)$. If $ID_o = ID^*$, RSA problem can be solved as follow: The relation becomes $(R'R^{-1})^b = Y^{h'-h} \bmod N$. Since $h, h' \in Z_b$, then $|h' - h| < b$. By the element b is a prime number, so it holds $\gcd(b, h' - h) = 1$. This means that there exist two integers c and d such that $cb + d(h' - h) = 1$. Finally, the value $X = (R'R^{-1})^d Y^c \bmod N$ is the solution of the given instance of the RSA problem. In effect, $X^b = (R'R^{-1})^{bd} Y^{bc} = Y^{d(h'-h)} Y^{bc} = Y^{cb+d(h'-h)} = Y$.

Probability. Let $q_{H_i} (i = 0, 1, 2)$, q_K, q_D and q_S be the number of $H_i (i = 0, 1, 2)$ queries, private key queries, delegating queries and proxy signing queries, respectively.
The probability that $\mathscr{C}$ does not fail during the queries is $\frac{q_{H_0} - q_K}{q_{H_0}}$. The probability that $ID_o = ID^*$ is $\frac{1}{q_{H_0} - q_K}$. So the combined probability is $\frac{q_{H_0} - q_K}{q_{H_0}} \cdot \frac{1}{q_{H_0} - q_K} = \frac{1}{q_{H_0}}$. Therefore, the probability of $\mathscr{C}$ to solve the RSA problem is: $\frac{\epsilon}{q_{H_0}}$.

Case 2. The final output is $\{\sigma^* = (m, m_w, T, S, Z), ID_o, ID_p\}$ and the output satisfies the requirement of Case 2 as defined in UNF-IBPS game.

Solve RSA problem. By the forking lemma for generic signature scheme [3], two proxy signatures can be generated: (m, m_w, T, S, Z) and (m, m_w, T, S, Z'). Where $k = k' = H_2(m, m_w, T, S)$, $h = H_1(m_w, T)$, $h' = H_1'(m_w, T)$, and $h \neq h'$. If $ID_o = ID^*$, RSA problem can be solved as follow: The relation becomes $(Z'Z^{-1})^b = Y^{h'-h} \bmod N$. Since $h, h' \in Z_b$, the that $|h' - h| < b$. By the element b is a prime number, so it holds $\gcd(b, h' - h) = 1$. This means that there exist two integers c and d such that $cb + d(h' - h) = 1$. Finally, the value $X = (Z'Z^{-1})^d Y^c \bmod N$ is the solution of the given instance of the RSA

problem. In effect, $X^b = (Z'Z^{-1})^{bd}Y^{bc} = Y^{d(h'-h)}Y^{bc} = Y^{cb+d(h'-h)} = Y$.

Probability. Probability of success is same as the probability in Case 1.

Case 3. The final output is $\{\sigma^* = (m, m_w, T, S, Z), ID_o, ID_p\}$ and the output satisfies the requirement of Case 3 as defined in UNF-IBPS game.

Solve RSA problem. By the forking lemma for generic signature scheme [3], two proxy signatures can be generated: (m, m_w, T, S, Z) and (m, m_w, T, S, Z'). Where $h = h' = H_1(m_w, T)$, $k = H_2(m, m_w, T, S)$, $k' = H_2'(m, m_w, T, S)$, and $k \neq k'$. If $ID_p = ID^*$, RSA problem can be solved as follow: The relation becomes $(Z'Z^{-1})^b = Y^{k'-k} \bmod N$. Since $k, k' \in Z_b$, then $|k' - k| < b$. By the element b is a prime number, so it holds $\gcd(b, k' - k) = 1$. This means that there exist two integers c and d such that $cb + d(k' - k) = 1$. Finally, the value $X = (Z'Z^{-1})^d Y^c \bmod N$ is the solution of the given instance of the RSA problem. In effect, $X^b = (Z'Z^{-1})^{bd}Y^{bc} = Y^{d(k'-k)}Y^{bc} = Y^{cb+d(k'-k)} = Y$.

Probability. Probability of success is same as the probability in Case 1.

□

5 Efficiency and Comparison

In this section, we compare the performance of our scheme with several other schemes. some notations are defined as follows:

P: a pairing operation.

E_M: a modular exponentiation.

M_P: a pairing-based scalar multiplication.

M_E: an ECC-based scalar multiplication.

Through PIV 3-GHZ processor with 512-MB memory and a Windows XP operation system. He et al. [7] obtained the running time for cryptographic operations. To achieve 1024-bit RSA level security, they use the Tate pairing defined over a supersingular curve $E/F_p : y^2 = x^3 + x$ with embedding degree 2, where q is a 160-bit Solinas prime $q = 2^{159} + 2^{17} + 1$ and p is a 512-bit prime satisfying $p + 1 = 12qr$. To achieve the same security level, they employed the parameter secp160r1 [23], where $p = 2^{160} - 2^{31} - 1$. The running times are listed in Table 1.

A simple method is used to evaluate the computation efficiency. For example, Wu et al.'s [25] scheme requires 6 pairing-based scalar multiplication operations and 8 pairing operations. So the resulting computation time is $6.38 \times 6 + 20.04 \times 8 = 198.60$.

Table 1: Cryptographic operation time (in milliseconds)

P	E_M	M_P	M_E
20.04	5.31	6.38	2.21

Based on the above parameter and ways, the detailed comparison results of several different IBPS schemes are illustrated in Table 2.

6 Conclusion

A proxy signature scheme allows a proxy signer to sign messages on behalf of an original signer within a given context. Most IBPS schemes currently known employ bilinear pairings. RSA is a key cryptography technique and provides various interfaces for the applied software in real-life scenarios. Although some good results were achieved in speeding up the computation of pairing function in recent years, the computation cost of the pairings is much higher than that of the exponentiation in a RSA group. In this paper, an IBPS scheme from RSA was proposed and the security was proved in the random oracle model. The scheme needs not pairings and it is more efficient than previous ones using bilinear pairings. Due to the good properties of our scheme, it should be useful for practical applications.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grants 61562012, 11261060, 61462016. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature scheme for delegation of signing rights," *IACR ePrint Archive*, 2003. (`http://eprint.iacr.org/2003/096/`)

[2] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal of Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[3] M. Bellare and G. Neven, "Multisignatures in the plain publickey model and a general forking lemma," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06)*, pp. 390–399, 2006.

[4] C. Gu and Y. Zhu, "Provable security of ID-based proxy signature schemes," in *Networking and Mobile Computing*, LNCS 3619, pp. 1277–1286, Springer, 2005.

[5] C. Gu and Y. Zhu, "An efficient ID-based proxy signature scheme from pairings," in *Proceedings of Inscrypt'07*, LNCS 4990, Springer, pp. 40–50, 2007.

Table 2: Comparison of several IBPS schemes

Scheme	Delegate	D-Verify	Proxy signing	P-Verify	Time
Wu et al. [25]	$2M_P$	$3P$	$4M_P$	$5P$	198.60
Gu et al. [5]	$4M_P$	$4M_P + P$	$2M_P$	$4M_P + P$	129.40
Ji et al. [11]	$3M_P$	$2P$	$3M_P$	$M_P + 2P$	124.82
He et al. [7]	M_E	$3M_E$	M_E	$6M_E$	24.31
Our scheme	$2E_M$	$2E_M$	$2E_M$	$2E_M$	42.48

[6] L. C. Guillou and J. J. Quisquater, "A paradoxical identity-based signature scheme resulting from zero-knowledge," in *Proceedings of Crypto'88*, LNCS 403, pp. 216–231, 1988.

[7] D. B. He, J. H. Chen, and J. Hu, "An ID-based proxy signature schemes without bilinear pairings," *Annual Telecommunications*, vol. 66, pp. 657–662, 2011.

[8] J. Herranz, "Identity-based ring signatures from RSA," *Theoretical Computer Science*, vol. 389, pp. 100–117, 2007.

[9] M. S. Hwang, E. J. L. Lu, and I. C. Lin, "A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem," *IEEE Transactions On Knowledge and Data Engineering*, vol. 15, no. 6, pp. 1552–1660, 2003.

[10] X. Hu, W. Tan, H. Xu, and J. Wang, "Short and provably secure designated verifier proxy signature scheme," *IET Information Security*, vol. 10, no. 2, pp. 69–79, 2013.

[11] H. Ji, W. Han, and L. Zhao, "An identity-based proxy signature from bilinear pairings," in *WASE International Conference on Information Engineering*, pp. 14–17, 2009.

[12] K. S. Kim, D. Hong, and I. R. Jeong, "Identity-based proxy signature from lattices," *Journal Of Communications and Networks*, vol. 15, no. 1, pp. 1–7, 2013.

[13] A. V. N. Krishna, A. H. Narayana, K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.

[14] B. Lee, H. Kim, and K. Kim, "Strong proxy signature and its applications," in *Proceedings of SCIS'01*, pp. 603–608, 2001.

[15] J. Y. Lee, J. H. Cheon, and S. Kim, "An analysis of proxy signatures: Is a secure channel necessary?" in *Proceedings of CT-RSA'03*, LNCS 2612, pp. 68–79, Springer, 2003.

[16] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature: Delegation of the power to sign messages," *IEICE Transactions on Fundamentals*, vol. E79-A, no. 9, pp. 1338–1353, 1996.

[17] T. Malkin, S. Obana, and M. Yung, "The hierarchy of key evolving signatures and a characterization of proxy signatures," in *Proceedings of EUROCRYPT'04*, LNCS 3027, pp. 306–322, Springer, 2004.

[18] S. Mashhadi, "A novel non-repudiable threshold proxy signature scheme with known signers," *International Journal of Network Security*, vol. 15, no. 4, pp. 274–279, 2013.

[19] T. Okamoto, A. Inomata, and E. Okamoto, "A proposal of short proxy signature using pairing," in *International Conference on Information Technology (ITCC'05)*, pp. 631–635, 2005.

[20] K. R. Santosh, C. Narasimham, and P. Shetty, "Cryptanalysis of multi-prime RSA with two decryption exponents," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 40–44, 2016.

[21] A. Shamir, "Identity-based cryptosystem and signature scheme," in *Advances in Cryptology (Crypto'84)*, LNCS 196, pp. 47–53, Springer, 1984.

[22] N. Tiwari and S. Padhye, "Provable secure multi-proxy signature scheme without bilinear maps," *International Journal of Network Security*, vol. 17, no. 6, pp. 736–742, 2015.

[23] The Certicom Corporation, *SEC2: Recommended Elliptic Curve Domain Parameters*, 2016. (`www.secg.org/collateral/sec2-final.pdf`)

[24] G. Wang, F. Bao, J. Zhou, and R. H. Deng, " Security analysis of some proxy signatures," in *Proceedings of ICISC'03*, LNCS 2971, pp. 305–319, Springer, 2003.

[25] W. Wu, Y. Mu, W. Susilo, J. Seberry, and X. Y. Huang, "Identity-based proxy signature from pairings," in *Proceedings of ATC'07*, LNCS 4610, pp. 22–31, Springer, 2007.

[26] J. Xu, Z. Zhang, and D. Feng, "ID-based proxy signature using bilinear pairings," in *PWorkshops on Parallel and Distributed Processing and Applications (ISPA'05)*, LNCS 3759, pp. 359–367, Springer, 2005.

Lunzhi Deng received his B.S. from Guizhou Normal University, Guiyang, China, in 2002; M.S. from Guizhou Normal University, Guiyang, China, in 2008; and Ph.D. from Xiamen University, Xiamen, China, in 2012. He is currently a Professor in the School of Mathematics Science, Guizhou Normal University, Guiyang, China. His recent research interests include algebra and cryptography.

Huawei Huang received his B.S. from Jiangxi Normal University, Nanchang, China, in 2001; M.S. from Jiangxi Normal University, Nanchang, China, in 2004; and

Ph.D. from Xidian University, Xi'an, China, in 2008. He is currently an Associate Professor in the School of Mathematics Science, Guizhou Normal University, Guiyang, China. His recent research interests include algebra and cryptography.

Yunyun Qu received his B.S. from Wuhan University, Wuhan, China, in 2005; M.S. from Southwest University, Chongqing, China, in 2009. He is currently an Associate Professor in the School of Mathematics Science, Guizhou Normal University, Guiyang, China. His recent research interests include number theory and cryptography.