Efficient Certificateless Authentication and Key Agreement (CL-AK) for Grid Computing

Shengbao Wang[1,2], Zhenfu Cao[1], and Haiyong Bao[1]
(Corresponding author: Zhenfu Cao)
Department of Computer Science and Engineering, Shanghai Jiao Tong University[1]
1954 Huashan Road, Shanghai 200030, P. R. China
Center of Computing, Paobing Academy 451 Huangshan Road, Hefei 230031, P.R. China[2]
(Email: {shengbao-wang,cao-zf,bhy}@cs.sjtu.edu.cn)
(Received Sept. 27, 2006; revised and accepted Feb. 1, 2007)

Abstract

Most of the current security architectures for grid systems use conventional public key infrastructure (PKI) to authenticate grid members and to secure resource allocation to these members. Certificateless public-key cryptography (CL-PKC) has some attractive properties which seem to align well with the demands of grid computing. In this paper, we present a certificateless protocol for authentication and key agreement (CL-AK) which fits well with the Grid Security Infrastructure (GSI) and provides a more lightweight key management approach for grid users. We show that the newly proposed protocol is of great efficiency and practical. Moreover, we prove that it provides perfect forward secrecy plus all the other security attributes of authentication and key agreement protocols such as known-key secrecy and no key control.

Keywords: Certificateless AK (CL-AK), certificateless public key cryptography (CL-PKC), grid computing, mutual authentication, key agreement (AK)

1 Introduction

Grid computing [12, 13] has been proposed as a mechanism to provide access to more computational power and resources. A computational grid is a distributed computing system which consists of a large number of sites of computational resources from which a virtual organization (VO) of high performance services can be combined for use by demanding users [18]. These resource contributing sites usually form different trust domains. In order to gain secure access to the resource contributing sites, at a VO setting-up stage, a user proxy (UP) must conduct *mutual authentication* with a resource proxy (RP) that manages these resource contributing sites.

The current grid security standard, Grid Security Infrastructure (GSI) [14] employs the standard SSL Authentication Protocol (SAP) [15] to achieve mutual entity authentication between UP and RP. Hence, UP and RP have identity certificates which are under the organization of the standard certificate-based public key authentication infrastructure X.509 [16, 17]. However, the extensive use of certificates in the hierarchical PKI setting within a dynamic grid environment brings many problems to GSI. For example, the authors of GSI conceded that the security architecture in GSI has a poor scalability which limits the number of resource allocation sessions that a UP can make which in turn limits the degree of high-performance grid computing services available to a user [9].

In 2003, Al-Riyami and Paterson introduced and developed the notion of certificateless public key cryptography (CL-PKC) [1, 2]. CL-PKC is designed to overcome the inherit key escrow shortcomings of identity-based cryptography (IBC) [21] without introducing public key certificates and the management overheads that it entails. CL-PKC is a model for the use of public key cryptography that is intermediate between the identity-based and traditional public key infrastructure (PKI) approaches.

By making uses of certificateless public keys, we propose the first certificateless authentication and key agreement protocol (hereafter referred to as CL-AK) for grid computing based on the Diffie-Hellman key agreement protocol [11]. We aim at employing CL-PKC to provide greater flexibility to entities within GSI. Its certificateless property (i.e., in our protocol, no public key certificate is needed) should well match the dynamic characters of grid environments, bringing lightweight and flexible key management method for GSI than traditional PKI does.

The rest of this paper is structured as follows. In Section 2, we define some desirable security attributes of certificateless authentication and key agreement protocols for grid computing. Section 3 explains some fundamental concepts of certificateless public key cryptography and reviews the Al-Riyami-Paterson protocol. Then we present our proposed certificateless AK protocol (CL-AK) in Section 4. In Section 5, we give security discussions and

efficiency analysis. Finally, we draw our conclusions in Section 6.

2 Security Attributes of CL-AK Protocols

In this section, we define some desired security attributes for CL-AK protocols for the grid environment. All these security attributes have their counterparts in the context of traditional PKC. We borrow our definitions from [6, 19]. These attributes can be vital in excluding realistic attacks in the open grid environment. Suppose a UP and a RP want to mutually authenticate each other and at the same time agree on a common shared secret session key and assuming that all the long-term private keys of UP and RP are kept secret properly, hence we excluded those impersonation attacks relating to the compromise of the long-term private key, e.g. key-compromise impersonation attack [6]. Now we define the security attributes of the CL-AK protocol rum between them as follows:

(Implicit) Mutual Authentication. In a authentication and key agreement protocol, mutual entity authentication is the property whereby one entity (e.g., UP) is assured that no other entity aside the specifically identified other entities (i.e. RP) may gain access to a particular secret session key. Mutual entity authentication is independent of the actual possession of such key by the other entities. For this reason, it is sometimes referred to more precisely as *implicit* mutual entity authentication.

Known-key security. A CL-AK protocol for grid computing is known-key secure if it still achieves its goal in the face of an adversary who has learned some previous session keys between UP and RP.

Forward secrecy. A CL-AK protocol enjoys forward secrecy if, when the long-term private keys of one (UP or RP) or all the entities (UP and RP) are compromised, the secrecy of previous session keys is not affected. *Full forward secrecy* refers to the scenario when all the long term private keys of UP and RP are compromised. Specially, in the CL-PKC setting, i.e., for the CL-AK protocols, we define an extra security attributes: *perfect* forward secrecy (PFS). A CL-AK protocol achieves PFS if, when even the master private key of trusted third party (TTP) of the system is compromised, the secrecy of previously established session keys is not affected.

No key control. It should not be possible for UP or RP to force any portion of the session key to be equal to a preselected value.

It is also desirable that a CL-AK protocol has high *bandwidth efficiency*, which means that only a small amount of data is transmitted between UP and RP to gain mutual entity authentication and agreement on a session key. Two important computational attributes are low *computation cost* and the ability to perform precomputation.

3 Certificateless Public Key Cryptography (CL-PKC)

3.1 Bilinear Pairings

Since Boneh and Frankin's breakthrough work on IBC [7], the elliptic curve pairings have also brought many other interesting applications to authentication and key agreement protocols (e.g., [8]). The basic concept of pairing is outlined as follows.

Let G_1, G_2 be two groups of the same prime order q. We view G_1 as an additive group of points on certain elliptic curves and G_2 as a multiplicative group throughout the paper. A pairing is a computable bilinear map $e : G_1 \times G_1 \rightarrow G_2$ satisfying the following properties [7]:

Bilinearity: If elements $P, Q \in G_1$ and $a, b \in Z_q^*$, then $e(aP, bQ) = e(P, Q)^{ab}$;

Non-degeneracy: There exists an element $P \in G_1$ such that $e(P, P) \neq 1$;

Computability: If $P, Q \in G_1$, one can compute $e(P, Q) \in G_2$ in polynomial time.

The modified Weil pairings [5, 7] and Tate pairing [4] on supersingular elliptic curves are examples of such bilinear maps, for which the Bilinear Diffie-Hellman (BDH) problem is believed to be hard. Informally, the BDH problem is stated as follows: let P be a generator of G_1. Given $(P, xP, yP, zP) \in G_1^4$ for some x, y, z chosen at random from Z_q^*, to compute $W = e(P, P)^{xyz} \in G_2$. We refer readers to [7] and [8] for further details on pairings and the BDH problem.

3.2 CL-PKC

Certificateless public key cryptography (CL-PKC) was first introduced by Al-Riyami and Paterson [1, 2] in 2003. Here we briefly review some basic knowledge about CL-PKC, interested readers can refer to [1, 2] for more comprehensive decryptions. Following the publication of [1], many proposals for certificateless schemes have been proposed [3, 22].

The main consideration for CL-PKC is the perceived problem of managing certificates and associated keys within conventional PKI and the key escrow problem of IBC. In an CL-PKC cryptosystem, there is *no* public key certificate to authenticate the public key of a user. Users' partial private keys are generated and distributed by a Trusted Authority (TA) in possession of a system master secret s. This TA roughly corresponds to the Certificate Authority/ Registration Authority (CA/RA) combination in conventional PKI. CL-PKC eliminates the need for public key certificates and the key escrow problems

that IBC brings. Thus it presents a more lightweight approach to deploying public key cryptography.

System Setup Phase. The TA does the following:

1) First randomly picks an arbitrary generator $P \in G_1$, a secret master key $s \in Z_q^*$ and computes its public key as $P_{Pub} = sP$;
2) Then chooses a cryptographic hash function $h : \{0,1\}^* \rightarrow G_1$;
3) Publishes the system parameters $params = \langle G_1, G_2, e, P, P_{Pub}, h \rangle$;
4) Computes the partial private key $S_{ID} = sQ_{ID}$ for a user with the identity information ID, in which the user's partial public key is $Q_{ID} = h(ID)$. (For example, Alice's partial private key from TA is $S_A = sQ_A$, where $Q_A = h(ID_A)$);
5) Finally, TA distributes the partial private key S_{ID} to the user with the identity information ID via a secure channel. After the above steps, Alice and Bob get their partial private key S_A and S_B, respectively.

User Setup Phase. A user (Alice) does the following to set up her public/private key pair (For simplicity of description, here we only describe a simplified version of user setup phase which is suitable for grid computing.):

1) She firstly chooses a $x_A \in Z_q^*$ as her own-chosen partial private keys;
2) Then computes $P_A = x_A P_{Pub} = x_A sP$ as her public key;
3) Publishes her public key via an open directory that all users in the system have access to.

After the above two setup phases, when another user (Bob) wants to send a message to Alice, he must obtain Alice's public key. However, no authentication of this public key is necessary and no public key certificate is required.

3.3 The Al-Riyami-Paterson CL-AK Protocol

Al-Riyami and Paterson also gave the first certificateless authentication and key agreement (CL-AK) protocol in [2]. Here we briefly review their protocol (hereafter referred to as the AP's CL-AK) [2]. The AP's CL-AK protocol consists of two phases: *Setup* and *Key Agreement*.

Setup Phase. Entities Alice and Bob who wish to agree a key first follow the User Setup Phase described in Section 3.2. We denote the two users' partial private keys as $S_A = sQ_A$ and $S_B = sQ_B$, in which their corresponding partial public key are $Q_A = h(ID_A)$ and $Q_B = h(ID_B)$, respectively.

We also denote Alice and Bob's public keys as P_A and P_B, where $P_A = x_A P_{Pub} = x_A sP$ and $P_B = x_B P_{Pub} = x_B sP$, respectively.

Key Agreement Phase. Alice and Bob each chooses random values $a, b \in Z_q^*$. Given these initializations, the protocol is as follows:

Protocol messages:

A $\longrightarrow$ B: T_A=aP
B $\longrightarrow$ A: T_B=bP.

After the above messages are exchanged, Alice computes

$$K_{AB} = e(Q_B, P_B)^a \cdot e(x_A S_A, T_B),$$

and Bob computes K_{BA} as follows:

$$K_{BA} = e(Q_A, P_A)^b \cdot e(x_B S_B, T_A).$$

It is easy to verify the following equations:

$$\begin{aligned} K_{AB} &= e(Q_B, P_B)^a \cdot e(x_A S_A, T_B) \\ &= e(Q_B, x_B sP)^a \cdot e(x_A S_A, bP) \\ &= e(x_B sQ_B, aP) \cdot e(x_A sQ_A, bP) \\ &= e(x_B S_B, T_A) \cdot e(Q_A, P_A)^b \\ &= K_{BA}. \end{aligned}$$

Hence, $K = K_{AB} = K_{BA}$ is a key shared between Alice and Bob; To ensure forward security, the authors use the shared key $H(K, abP)$ as the final established session key between the two users, where H is a suitable hash function.

The protocol uses two passes and is bandwidth-efficient. But each party needs to compute 2 expensive pairings. Notice that the pairing $e(Q_{ID}, P_{ID})$ can be precomputed, but the other one, i.e. $e(x_{ID} S_{ID}, T_{ID'})$ has to be computed on-line (where ID denotes UP and RP's identity).

4 Our New CL-AK Protocol for Grid Computing

In this section, we present our new CL-AK protocol for UP and RP to mutually authenticate each other and to share a common secret session key simultaneously for subsequent secure communications.

Similar to the AP's CL-AK [2], a grid trusted authority (GTA) is required to generate partial private keys for users, using their unique identity (ID) to derive the associate partial public keys. UP and RP both have their own chosen partial private/public key pairs. This initial registration is what can be called a user single-sign-on (SSO) session [18]. In an SSO session, GTA conducts a thorough identity validation on UP and RP.

We denote UP and RP's own chosen partial private/public key pairs as (x_U, P_U) and (x_R, P_R), respectively, in which $P_U = x_U P_{Pub} = x_U sP$ and $P_R = x_R P_{Pub} = x_R sP$. They publish their partial public keys

(P_U and P_R) via a public open directory. Note again that in CL-PKC, no public key certificate is needed to guarantee the authenticity of the user's own chosen partial public keys. Our new protocol also consists of two stages:

Setup Stage. his stage is identical as the Setup Phase of the AP's protocol.

Key Agreement Stage. To mutually authenticate each other and establish a shared session key, UP and RP each firstly generates an ephemeral private key (say a and $b \in Z_q^*$), and computes the corresponding ephemeral public key $T_U = aP$ and $T_R = bP$, respectively. They then exchange T_U and T_R as illustrated in Figure 1.

User Proxy(UP)		Resource Proxy(RP)
$a \in_R Z_q^*$		$b \in_R Z_q^*$
$T_U = aP$		$T_R = bP$
	$\xrightarrow{T_U}$	
	$\xleftarrow{T_R}$	
$K_{UR} = H(k_{UR}\|\|x_U P_R\|\|aT_R)$		$K_{RU} = H(k_{RU}\|\|x_R P_U\|\|bT_U)$

Figure 1: CL-AK protocol for grid computing

After the message exchange, UP and RP do the following:

1) UP computes the shared secret key K_{UR} as follows:

$$K_{UR} = H(k_{UR}||x_U P_R||aT_R),$$

in which $k_{UR} = e(S_U, Q_R)$, $x_U P_R = x_R x_U P$ and H is a predetermined key derivation function (a hash function) of the two users.

2) RP computes the shared secret key K_{RU} as follows:

$$K_{RU} = H(k_{RU}||x_R P_U||bT_U),$$

in which $k_{RU} = e(S_R, Q_U)$ and $x_R P_U = x_R x_U P$.

Protocol Correctness. By the bilinearity of the pairing, we can get the following equations:

$$\begin{aligned} k_{UR} &= e(S_U, Q_R) \\ &= e(sQ_U, Q_R) \\ &= e(Q_U, sQ_R) \\ &= e(Q_U, S_R) \\ &= k_{RU}. \end{aligned}$$

Combining with the equation $aT_R = bT_U = abP$ (Actually, this is exactly an instance of Diffie-Hellman key agreement protocol [11]) and non-interactive shared secret $x_U x_R P$, the two secret keys (K_{UR} and K_{RU}) computed by UP and RP (and only by them) are equal to each other, i.e., UP and RP successfully authenticated each other (in other words, gained implicity *mutual entity authentication* between them) and established a shared secret session key after running an instance of the new CL-AK protocol.

5 Analysis of Security and Performance

In this section, we argue that the newly proposed protocol achieves all the security goals defined in Section 2, i.e., UP and RP establish secure mutual authentication and key agreement by running an instance of the CL-AK protocol. After the security discussions, we also give an analysis on the computation cost and communication overhead for UP and RP. Finally, we give an efficiency comparison between our CL-AK protocol and the AP's protocol [2].

5.1 Security Analysis

We now examine our CL-AK protocol in relation to the security attributes defined in Section 2, and informal arguments are provided to support our claims:

- **Forward secrecy.**
  - *Full forward secrecy:* The compromise of both party's long-term partial private key, i.e. x_U, S_U of UP and x_R, S_R of RP, gives no information about any previously established session keys. Suppose Eve knows all the long-term partial private keys x_{ID} and S_{ID} (in whice $ID \in \{U, R\}$), to extract a past session key, he must compute $abP \in G_1$ from P, aP and bP. Without the knowledge of a and b, this is exactly an instance of the computational Diffie-Hellman(CDH) problem in G_1 that Eve is not able to solve.
  - *Perfect forward secrecy:* Suppose at a moment the master key s known only to the GTA is compromised. Since the established session keys are computed with the ephemeral private keys of UP and RP, Eve still have to solve the CDH problem in G_1 to reveal the session key. This means that our protocol has perfect forward secrecy(PFS).
- **Known-key secrecy.** Each run of the protocol between UP and RP shall produce a unique session key which depends on every particular ephemeral private key a and $b \in Z_q^*$ of UP and RP. Even if the adversary Eve has learned some other session keys, he can not compute the keying point $abP \in G_1$ from them aP and bP, because when he has no access to a and b, he faces the computational Diffie-Hellman problem which is believed to have no polynomial time

Table 1: Computational and bandwidth efficiency comparisons

Item ↓ / Protocol →	AP's	Ours
Pairing	1	**0**
Point multiplication	3	2
Bandwidth	1	1

algorithm to compute. Hence our protocol has the property of known-key security.

- **No Key control:** As has been pointed out in [20], if in a key agreement protocol the responder (RP) will receive the key component of the initiator (accordingly, UP) before he send out his own component, he can always gain an unfair advantage over his counterpart on controlling the value of the shared session key. Therefore, like most of the existing protocol, our protocol does not posses the full key control property. To avoid this weakness, as suggested in [20], we need to use commitments, which require an extra round.

Finally, we note that one may include the protocol transcripts and the identity of the protocol participants in the key derivation function (i.e. the hash function H) to resist some potential attacks, e.g., unknown-key share attacks, replay attacks and key-replicating attacks [6, 8, 10].

5.2 Performance Analysis

Now we analyze the performance (in relation to computational and bandwidth efficiency) of our CL-AK. The operations that dominate the processing time of each participate in a protocol execution are pairing evaluation and elliptic curve point scalar multiplication. Furthermore, compared with scalar multiplication, pairing evaluation is far more time-consuming. Here we only consider pairing evaluations and scalar multiplications in G_1. Table 1 compares the computation costs and bandwidth efficiencies (i.e. communication overheads) of the two protocols (each for one party, since each party of a particular protocol has the same computation and communication overheads).

In our CL-AK protocol, UP and RP can have the pairing evaluation $e(Q_U, Q_R)^s$ and the scalar multiplication $x_U x_R P$ pre-computed (i.e., computed off-line) and stored. So, the number of pairing evaluation on-line in our protocol can be reduced to 0. As has been pointed out in Section 3, while with pre-computation in consideration, for each run of AP's protocol, one party still has to compute a pairing value on-line. One can see from the table that our protocol requires the same number scalar multiplications with the AP's protocol, and the communication costs of the two protocols are the same (both require one point in G_1 to be distributed by one party), but our protocol decreases the number of on-line pairing evaluation to 0.

6 Conclusions

Recently, the concept of certificateless public key cryptography was put forward, which is intermediate between the identity-based and traditional public key infrastructure (PKI) approaches and brought significant impacts on key management technology. Based on the Diffie-Hellman protocol, we put forward a new certificateless authentication and key agreement protocol (CL-AK) between user proxy (UP) and resource proxy (RP) in the grid computing setting. We showed that our protocol is more efficient than that of Al-Riyami and Paterson. Our proposal improves the performance for the current GSI authentication scheme to a considerable extent.

Our present work can be seen as the first step towards integrating certificateless public key cryptography (CL-PKC) into the grid environment to bring more lightweight key management approach for grid users. Also, our newly proposed CL-AK protocol eliminates on-line expensive pairing evaluation and can be seen as a significate improvement on the performance of certificateless authentication and key agreement protocols so that it becomes more practical for grid computing.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that improved the presentation of this paper. This work was supported partially by the National High Technology Development Program of China under Grant No. 242006AA01Z424 and the National Natural Science Foundation of China under Grant Nos. 60673079 and 60572155.

References

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," *Proceedings of ASIACRYPT 2003*, LNCS 2894, pp. 452-473, Springer-Verlag, 2003.

[2] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," *Cryptology ePrint Archive, Report 2003/126*, 2003. (http://eprint.iacr.org/)

[3] S. S. Al-Riyami and K. G. Paterson. "CBE from CL-PKE: A generic construction and efficient schemes," *Proceedings PKC 2005*, LNCS 3386, pp. 398-415, Springer-Verlag, Berlin, 2005.

[4] P. Barreto, H. Y. Kim, B. Bynn and M. Scott, "Efficient algorithms for pairing-based cryptosystems," *Proceedings of CRYPTO 2002*, LNCS 2442, pp. 354-368, Springer-Verlag, 2002.

[5] P. Barreto, B. Lynn, and M. Scott, "On the selection of pairing-friendly groups," *Proceedings of SAC 2003*, LNCS 3006, pp. 17-25, Springer-Verlag, 2004.

[6] S. Blake-Wilson and A. Menezes, "Authenticated Diffie-Hellman key agreement protocols," *Proceedings of SAC 1998*, LNCS 1556, pp. 339-361, Springer-Verlag, 1999.
[7] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *Proceedings of CRYPTO 2001*, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.
[8] L. Chen and C. Kudla, "Identity based key agreement protocols from pairings," *Proceedings of the 16^{th} IEEE Computer Security Foundations Workshop*, pp. 219-213, IEEE Computer Society, 2002.
[9] L. Chen, H. W. Lim, and W. Mao, "User-friendly grid security architecture and protocols," *Proceedings of the 13th International Workshop on Security Protocols*, Cambridge, UK, 2005.
[10] K. K. R. Choo, C. Boyd, and Y. Hitchcock, "On session key construction in provably secure protocols," *Proceedings of MYCRYPT 2005*, LNCS 3715, pp. 116-131, Springer-Verlag, 2005.
[11] W. Diffie, and M. E. Hellman. "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.
[12] I. Foster, "The Grid: A new infrastructure for 21st century science," *Physics Today*, vol. 55, no. 2, pp. 42-47, 2002.
[13] I. Foster, and C. Kesselman, *The Grid 2: Blueprint for a new computing infrastructure*, Elsevier, San Francisco, 2004.
[14] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A security architecture for Computational Grids," *Proceedings of 5th ACM Conference on Computer and Communications Security*, pp. 83-92, 1998.
[15] A. O. Freier, P. Karlton, and P. C. Kocher, *The SSL Protocol*, Version 3.0, INTERNET-DRAFT, Nov. 1996.
[16] R. Housley, W. Polk, W. Ford, and D. Solo *Internet X.509 Public Key Infrastructure Certificate And Certificate Revocation List (CRL) Profile*, The Internet Engineering Task Force (IETF), RFC 3280, 2002.
[17] ITU-T, *Recommendation X.509 the Directory- Authentication Framework*, International Telecommunication Union, Geneva, Switzerland. 1993.
[18] W. Mao, *An Identity-Based Non-Interactive Authentication Framework For Computational Grids*, HP Laboratories Bristol, HPL-TR-2004-96, 2004.
[19] A. Menezes, P. v. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
[20] C. Mitchell, M. Ward, and P. Wilson, "Key control in key agreement protocols," *Electronics Letters*, vol. 34, no. 10, pp. 980-981, 1998.
[21] A. Shamir, "Identity-based cryptosystems and signature schemes," *Proceedings of CRYPTO 1984*, LNCS 196, pp. 47-53, Springer-Verlag, 1984.
[22] D. H. Yum, and P. J. Lee, "Generic construction of certificateless encryption," *Proceedings of ICCSA 2004*, LNCS 3043, pp. 802-811, Springer-Verlag, 2004.

Shengbao Wang received his B. S and M. S in computer science in 2000 and 2003 respectively, and is currently a doctoral candidate in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. His main research interests are applied cryptography and network security.

Zhenfu Cao is the professor and the doctoral supervisor of Computer Software and Theory at Department of Computer Science and Engineering of Shanghai Jiao Tong University. His main research areas are number theory and modern cryptography, theory and technology of information security etc. He is the gainer of Ying-Tung Fok Young Teacher Award (1989), the First Ten Outstanding Youth in Harbin (1996), Best Ph. D thesis award in Harbin Institute of Technology (2001) and the National Outstanding Youth Fund in 2002.

Haiyong Bao received his B. S. and M. S. in engineering in 2000 and 2003 respectively, both from Department of Automation, China University of Mining and Technology. He received his PhD degree in computer science from Department of Computer Science and Engineering, Shanghai Jiao Tong University in 2006. His current research interests include public key cryptography and network information security.