

Efficient Identity-based Authenticated Key Agreement Protocol with PKG Forward Secrecy

Shengbao Wang^{1,2}, Zhenfu Cao¹, and Feng Cao¹

(Corresponding author: Zhenfu Cao)

Department of Computer Science and Engineering, Shanghai Jiao Tong University¹

800 Dongchuan Road, Shanghai 200240, P. R. China (Email: cao-zf@cs.sjtu.edu.cn)

Center of Computing, Paobing Academy, 451 Huangshan Road, Hefei 230031, P. R. China²

(Received Sep. 27, 2006; revised and accepted Nov. 8, 2006)

Abstract

For an identity-based authenticated key agreement (ID-AK) protocol, PKG forward secrecy is the strongest notion of forward secrecy, which is about the security of previously established session keys after the master secret key of the Private Key Generator (PKG) is compromised. In this paper, we put forward a new identity-based authenticated key agreement protocol which achieves PKG forward secrecy. On its performance, we show that it is more computationally efficient than a previously proposed protocol of Chen and Kudla (called Protocol 2'). Furthermore, we examine other security attributes that our new protocol possesses one by one.

Keywords: Authenticated key agreement (AK), bilinear pairing, cryptographic protocol, identity-based cryptosystem, PKG forward secrecy (PKG-FS)

1 Introduction

Key agreement protocols (for two parties) are a fundamental building block for ensuring secure communications between two parties over an insecure network. Generally, key agreement protocols allow two communicating parties who never met in advance to establish a common secret key via public communication. The agreed secret key, which is usually called a *session key*, can then be used to create a confidential or integrity-protected communication channel between the parties.

The *implicit key authentication* (IKA) property requires that in a run of the key agreement, the two protocol principals are assured that no one aside from the intended partner in the communication can possibly learn the value of the final session key. While a stronger property, i.e. *key confirmation*, is achieved when a party is assured that her intended partner has actually computed the session key. A key agreement protocol that provides mutual implicit key authentication is called an *authenticated key agreement protocol* (or AK protocol). Farther, a

protocol that provides mutual key authentication as well as mutual key confirmation is called an *authenticated key agreement with key confirmation protocol* (or an AKC protocol) [3]. Beginning with the well-known Diffie-Hellman protocol [5], a large amount of key agreement protocols have been proposed (refer to [15] and Chapter 12.6 of [8] for comprehensive surveys).

This paper focuses on key agreement protocols in the public-key authentication model, wherein parties hold a public/private key pair. Note that if traditional (certificate-based) public-key cryptography is used, normally a public key infrastructure (PKI) will be required to be deployed to authenticate all the users' public keys and certificates are needed. In 1984, Shamir [12] proposed the concept of identity(ID)-based cryptography (IBC) in which each party's public key can be an arbitrary string (typically an identity string). Hence, no certificate is required to authenticate the public keys in IBC, thus reduces much of the overhead of key management. Following Boneh and Franklin's work on the ID-based encryption [2], many two-party ID-based key agreement protocols using bilinear pairings on elliptic curves have been proposed (e.g., [3, 6, 7, 11, 13]).

Our Contributions. In 2002, Chen and Kudla proposed some efficient and secure ID-based AK protocols using pairings. Among which Protocol 2' [3] achieves PKG forward secrecy with only 1 pairing evaluation for each party. So far, it is widely considered as the most efficient one. In this paper, we propose a more efficient identity-based AK protocol to surpass Protocol 2'. Our protocol is different from Protocol 2' in that both the form of the exchanged messages and the generation of the session-specific secret are unique. Our new construction results in an efficient alternative which reduces much computational cost.

The rest of this paper is proceed as follows. In Section 2, we briefly describe the desirable security attributes of AK(C) protocols, followed by definitions of bilinear pairings, bilinear Diffie-Hellman problem and some related computational assumptions. In Section 3, we review Chen

and Kudla's Protocol 2' from [3]. Our new proposal, which we call IDAK-PKG, is then presented in Section 4. In Section 5, security discussions of our IDAK-PKG protocol are given with heuristic arguments, followed by the performance comparison between the two protocols. Finally, we draw a brief conclusion in Section 6.

2 Preliminaries

2.1 Desirable Security Attributes

From now on, we let Alice (A) and Bob (B) be two honest entities, i.e., legitimate entities who execute the steps of a protocol correctly. The most common desirable security attributes of an ID-based AK(C) protocol is listed as follows.

- *Known-key secrecy.* Suppose an established session key between A and B is disclosed, the adversary is unable to learn other established session keys between them. This attribute is also called *key independence*, which means that session keys of different sessions are computationally independent from each other.
- *PKG forward secrecy (PKG-FS).* If one party's private key is compromised, the secrecy of her past session keys should not be affected. If for an ID-based protocol, the master key known only to the private key generator (PKG) is compromised, the secrecy of previous session keys established by honest entities is not affected, then the protocol is said to be PKG forward secure. Obviously, PKG-FS implies *full* forward secrecy (which means that if long-term private keys of *both* the two entities are compromised, the secrecy of previous session keys is not affected). Note that this property is also known as the *master-key forward secrecy*.
- *Key-compromise impersonation (K-CI) resilience.* Suppose A 's private key is disclosed. Obviously, an adversary who knows this key can impersonate A to other entities (e.g. B). However, it is desired in some circumstances that this disclosure does not enable the adversary to impersonate *other* entities (e.g. B) to A .
- *Unknown key-share (UK-S) resistance.* Entity A should not be able to be coerced into sharing a key with entity C when in fact A believes that he is sharing the key with some entity B .
- *No key control.* Neither party nor an adversary should be able to dominate the generation of the final session key (or any portion of it).

For an AK protocol, it is desired that it has the following three performance attributes: a minimal number of passes (the number of messages exchanged in a run of the protocol), low communication overhead (total number of

bits transmitted), and low computational cost (the computational operations needed for A and B to finish a run of the protocol).

2.2 Bilinear Pairings

Here we briefly introduce some background knowledge of the bilinear pairing.

Let \mathbb{G}_1 denotes a cyclic additive group generated by an element P , whose order is a prime q , and \mathbb{G}_2 denotes a cyclic multiplicative group of the same prime order q . We assume that the discrete logarithm problem(DLP) in both \mathbb{G}_1 and \mathbb{G}_2 are hard.

Definition 1 (Admissible Pairing). An *admissible pairing* e is a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, which satisfies the following three properties:

- 1) *Bilinear:* If elements $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$, then $e(aP, bQ) = e(P, Q)^{ab}$;
- 2) *Non-degenerate:* There exists an element $P \in \mathbb{G}_1$ such that $e(P, P) \neq 1$;
- 3) *Computable:* If $P, Q \in \mathbb{G}_1$, one can compute $e(P, Q) \in \mathbb{G}_2$ in polynomial time.

The modified Weil and Tate pairings associated with supersingular elliptic curves are examples of such admissible pairings [2, 3].

2.3 Related Assumptions

The security of our proposed identity-based authenticated key agreement protocol is based on the Computational Diffie-Hellman (CDH) and Bilinear Diffie-Hellman (BDH) assumptions [3]:

Definition 2 (Diffie-Hellman Tuple). A *Diffie-Hellman tuple* in \mathbb{G}_1 . $(P, xP, yP, zP) \in \mathbb{G}_1^4$, for some x, y, z chosen at random from \mathbb{Z}_q^* satisfying $z = xy \pmod{q}$.

Definition 3 (Computational Diffie-Hellman (CDH) Problem). Given the first three elements from the four elements in a DH tuple, compute the fourth element.

Definition 4 (CDH Assumption). There exists no algorithm running in expected polynomial time, which can solve the CDH problem with non-negligible probability.

Definition 5 (Bilinear Diffie-Hellman (BDH) Problem). Let P be a generator of \mathbb{G}_1 . The BDH problem in $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$ is that given $(P, xP, yP, zP) \in \mathbb{G}_1^4$ for some x, y, z chosen at random from \mathbb{Z}_q^* , compute $W = e(P, P)^{xyz} \in \mathbb{G}_2$.

Definition 6 (BDH Assumption). No algorithm running in expected polynomial time can solve the BDH problem in $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$ with non-negligible probability.

3 Review of Protocol 2'

In this section, we briefly review the ID-based authenticated key agreement protocol due to Chen and Kudla (Protocol 2') [3]. A high level description of the protocol is depicted in Figure 1.

The protocol consists of two stages: Setup and Key Agreement.

Setup. Suppose we have an admissible pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ as described in Section 2, where \mathbb{G}_1 and \mathbb{G}_2 are two groups with the same prime order q . The PKG follows the following steps:

- 1) picks an arbitrary generator $P \in \mathbb{G}_1$, a secret master key $s \in \mathbb{Z}_q^*$;
- 2) chooses a cryptographic hash function $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$;
- 3) publishes the system parameters $params = \langle \mathbb{G}_1, \mathbb{G}_2, e, q, P, H_1 \rangle$;
- 4) computes the private key $d_{ID} = sQ_{ID}$ for a user with the identity information ID , in which the user's public key is $Q_{ID} = H_1(ID)$;
- 5) distributes the private key d_{ID} to the user with the identity information ID via a secure channel.

Thus, each user's identity-based public/private key pair is defined as (Q_{ID}, d_{ID}) where $Q_{ID}, d_{ID} \in \mathbb{G}_1$.

Key Agreement. To agree upon a common session key, Alice and Bob each first randomly chooses an ephemeral private key, $a, b \in \mathbb{Z}_q^*$, and compute their corresponding ephemeral public keys as follows,

$$T_{A1} = aQ_A, T_{A2} = aP$$

and

$$T_{B1} = bQ_B, T_{B2} = bP.$$

They then exchange the above ephemeral public keys as depicted in Figure 1;

After the message exchange, the two participants follow the following steps:

- 1) Alice computes two secrets (K_{A1} and K_{A2}) as follows,

$$K_{A1} = aT_{B2} = abP, K_{A2} = e(d_A, aQ_B + T_{B1});$$
- 2) Symmetrically, Bob computes two secrets (K_{B1} and K_{B2}) as follows,

$$K_{B1} = bT_{A2} = abP, K_{B2} = e(d_B, bQ_A + T_{A1});$$
- 3) Alice computes the final session key sk as follows,

$$sk = H_2(K_{A1}, K_{A2}),$$

where $H_2 : G_1 \times G_2 \rightarrow \{0, 1\}^k$ is a key derivation function (KDF). (We denote by $|sk|$ the length of the binary representation of sk , hence $k = |sk|$.)

- 4) Analogously, Bob computes his final session key sk as follows,

$$sk = H_2(K_{B1}, K_{B2}).$$

Protocol Correctness. If Alice and Bob follow the protocol, they will successfully agree upon a common session key since we have:

$$K_{A2} = K_{B2} = e(Q_A, Q_B)^{s(a+b)}.$$

Chen and Kudla proved in a *restricted* security model of Bellare and Rogaway [1] the security attributes of Protocol 2', but the so-called formal security proof for this protocol does not even give guarantee of the most basic security attribute of AK protocols: known-key secrecy. Hence, as Wang stated in [14] recently, the security proof has limited value in practice. However, heuristic analysis does show that Protocol 2' has the following security attributes: mutual implicit key authentication, known-key secrecy, PKG forward secrecy (PKG-FS), no key-compromise impersonation, unknown key-share resilience, and *imperfect* key control (see discussions in Section 5).

In the next section, we will present a more efficient protocol, which reduces much of the computational cost.

4 New Identity-based AK Protocol

In this section, we describe our newly proposed ID-based authenticated key agreement protocol with PKG forward secrecy (hereafter referred to as IDAK-PKG). The **Setup** stage of our new protocol is identical to that of Protocol 2'. Hence here we merely describe the **Key Agreement** stage as follows.

Key Agreement. Likewise, we denote user Alice and Bob's public/private key pairs as (Q_A, d_A) and (Q_B, d_B) , respectively. To establish a shared session key, Alice and Bob each firstly generate an ephemeral private key (say a and $b \in \mathbb{Z}_q^*$), and compute the corresponding ephemeral public keys $T_{A1} = aQ_A, T_{A2} = aQ_B$ and $T_{B1} = bQ_B, T_{B2} = bQ_A$. They then exchange T_{A1}, T_{A2} and T_{B1}, T_{B2} as described in Figure 2.

After the message exchange, the two users do the following:

- 1) Alice computes the shared secret K_{AB} as follows (after receiving T_{B1} and T_{B2}):

$$K_{AB} = e(d_A + a \cdot T_{B2}, T_{A2} + T_{B1}),$$
 in which $T_{B1} = bQ_B, T_{B2} = bQ_A$ and $d_A = sQ_A$.
- 2) Bob computes the shared secret K_{BA} as follows (after receiving T_{A1} and T_{A2}):

$$K_{BA} = e(d_B + b \cdot T_{A2}, T_{B2} + T_{A1}),$$

in which $T_{A1} = aQ_A, T_{A2} = aQ_B$ and $d_B = sQ_B$.

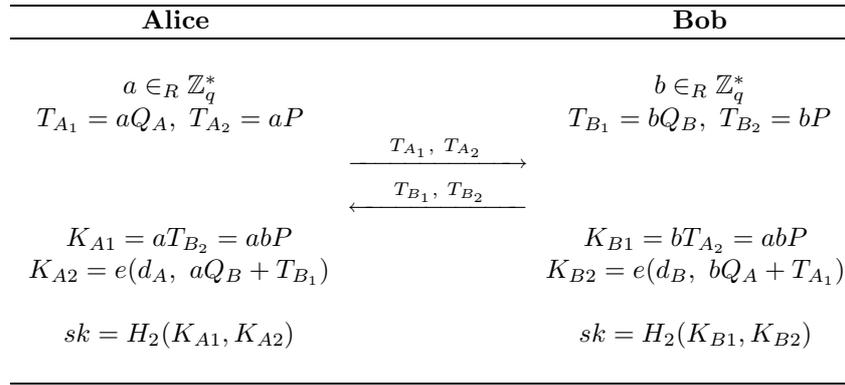


Figure 1: Protocol 2' of Chen and Kudla [3]

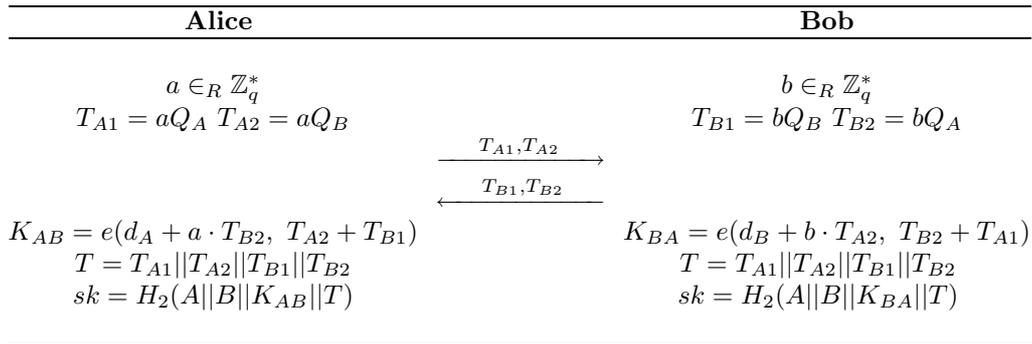


Figure 2: New ID-based AK protocol with PKG forward secrecy

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

$$\begin{aligned}
K_{AB} &= e(d_A + a \cdot T_{B2}, T_{A2} + T_{B1}) \\
&= e(d_A + a \cdot bQ_A, aQ_B + bQ_B) \\
&= e(Q_A, Q_B)^{(s+ab) \cdot (a+b)} \\
&= e((s+ab)Q_B, (a+b)Q_A) \\
&= e(d_B + b \cdot aQ_B, aQ_A + bQ_A) \\
&= e(d_B + b \cdot T_{A2}, T_{B2} + T_{A1}) \\
&= K_{BA}.
\end{aligned}$$

Thus, the two secret keys computed by Alice and Bob (K_{AB} and K_{BA}) are equal to each other, i.e., the two users successfully established a shared secret $K = K_{AB} = K_{BA}$ after running an instance of the protocol. The final shared secret session key is then $sk = H_2(A || B || K || T)$, where $H_2 : \{0, 1\}^* \rightarrow \{0, 1\}^k$ is a key derivation function (in which $k = |sk|$) and T is the protocol transcript. (In [4], Choo et al. suggest to include the transcript into the key derivation function to counter key replicating attacks.)

5 Security and Performance

We argue that the newly proposed IDAK-PKG protocol achieves PKG forward secrecy (PKG-FS) and almost all the other security attributes that Protocol 2' possesses. We address the security properties of our new protocol heuristically as follows.

- **PKG forward secrecy (PKG-FS):** The compromise of the master secret key, i.e. s (of PKG), gives no information about any previously established session keys. For any adversary (hereafter referred to as Eve) with knowledge of s , to derive a common shared secret $K = e(Q_A, Q_B)^{(s+ab)(a+b)}$, he computes as follows,

$$\begin{aligned}
K &= e(Q_A, Q_B)^{(s+ab)(a+b)} \\
&= e(Q_A, Q_B)^{sa+sb+ab(a+b)} \\
&= e(Q_A, Q_B)^{sa} e(Q_B, Q_A)^{sb} e(Q_A, Q_B)^{ab(a+b)} \\
&= e(Q_A, aQ_B)^s e(Q_A, bQ_B)^s e(abQ_A, (a+b)Q_B).
\end{aligned}$$

To compute the pairing value $e(abQ_A, (a+b)Q_B)$ from aQ_A, bQ_A, aQ_B and bQ_B , the values of the term abQ_A is required. Since all that Eve have on hand are $Q_A, Q_B, aQ_A, bQ_A, aQ_B$ and bQ_B , without the knowledge of a or b , it is computationally infeasible to recover the above value assuming the intractability of the computational Diffie-Hellman (CDH) problem in \mathbb{G}_1 . This shows that Eve is not able to compute an established secret K , hence the session key sk associated with it is still secure.

- **Known-key secrecy:** From the computation of the two users' shared secret, i.e. $K = e(Q_A, Q_B)^{(s+ab)(a+b)}$, we can see that each session key sk (takes K as input to the key

derivation function) depends on every particular ephemeral private key generated by the two user (i.e. a , b) and the master secret key of the PKG, i.e. s . Therefore, an adversary with some known session keys but no information on s , a , b , and d_A , d_B , is not able to derive even one new session key, since extracting the above shared secret K at least requires solve of the discrete logarithm problem (DLP) in \mathbb{G}_1 . In other words, our protocol provides the property of known-key secrecy.

- **K-CI resilience:** When the long-term private key of a user (e.g., Alice) is compromised, an adversary Eve, who comes into possession of Alice's private key d_A , is able to impersonate other entities to her.

That is to say, our new protocol is vulnerable to the key-compromise impersonation (K-CI) attack. We illustrate the attack as follows. Firstly, Eve picks randomly a $b \in \mathbb{Z}_q^*$, fabricates $T_{B1} = bQ_B$, $T_{B2} = bQ_A$ and then sends them to Alice. Upon receiving T_{B1} and T_{B2} , Alice computes the shared secret K_{AB} as $K_{AB} = e(d_A + a \cdot T_{B2}, aQ_B + T_{B1})$ as specified by the protocol. But Eve, equipped with d_A , aQ_A and aQ_B , is also able to compute K_{AB} . As a result, Eve can successfully impersonate Bob to Alice, this means that the new protocol does not have the property of K-CI resilience. However, in most circumstances the private key of a user is updated periodically, hence we argue that the damage of the above K-CI vulnerability can be limited to some considerably low extent.

- **UK-S resistance:** It is well-known that including the identities information of protocol participating parties in the key derivation function (KDF) can prevent potential unknown key-share (UK-S) attack. As for our protocol, we included A and B into the KDF function, so it achieves the property of UK-S resistance.
- **No key control:** In our protocol, the responder (Bob) will receive the transmitted data of the initiator (Alice) before he send out his own data, he can always gain an unfair advantage over his counterpart on controlling the value of the shared session key [9]. To achieve the property of no key control, as suggested in [9], we need to use commitments [10], which requires an extra round.

Performance Comparison. Both Protocol 2' and IDAK-PKG are role-symmetric. Here we compare the computational costs and communication overheads for each party (e.g., Alice) of the two protocols. Table 1 compares the performances (namely, computational and bandwidth efficiencies) of the two protocols with regard to elliptic curve point scalar multiplications, pairing evaluations and the transmitted data blocks.

As can be seen from Table 1, Protocol 2' requires Alice to compute 4 elliptic curve point scalar multiplications (i.e. aP , aQ_A , aQ_B and aT_{B1}), 1 point addition (i.e. $aQ_A + T_{A1}$) and 1 pairing evaluation. The protocol IDAK-PKG proposed here requires 3 point scalar multiplications (i.e. aQ_A , aQ_B and aT_{B2}), 2 point additions (i.e. $d_A + aT_{B2}$ and $T_{A2} + T_{B1}$) and 1 pairing evaluation. Among which the pairing evaluation is the most time-consuming operations that dominates the overall computational costs of the two protocols. We also note that a point addition is much quicker than a point scalar multiplication. Hence, with the same amount of pairing evaluations and only one more point addition, our new protocol improved much computational efficiency by decreasing 1 elliptic curve point scalar multiplication operation.

As to message bandwidth, our new protocol requires each protocol participant to distribute 2 elliptic curve point to its partner, which is the same as in Protocol 2'.

6 Conclusion

PKG forward secrecy (PKG-FS) is an important security property for identity-based authenticated key agreement protocols. Protocol 2' of Chen and Kudla makes use of the ephemeral Diffie-Hellman key agreement protocol to achieve PKG-FS. In this paper we aim at improving its efficiency by proposing a new protocol (IDAK-PKG). Our new protocol does not use the ephemeral Diffie-Hellman key agreement protocol and improves much computational cost.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable suggestions that improved this work. This work was supported in part 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] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Proceedings of Crypto'93*, LNCS 773, pp. 110-125, Springer-Verlag, New York, 1993.
- [2] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proceedings of Crypto'01*, LNCS 2139, pp. 213-229, Springer-Verlag, New York, 2001.
- [3] L. Chen and C. Kudla, "Identity based key agreement protocols from pairings," in *Proceedings of the 16th IEEE Computer Security Foundations Workshop*, pp. 219-213, IEEE Computer Society,

Table 1: Performance comparison between Protocol 2' and IDAK-PKG

↓Protocol / Items→	Pairing	Scalar multiplication	Message block
Protocol 2'	1	4	2 points
IDAK-PKG*	1	3	2 points

*One elliptic curve point addition is required additionally for our protocol.

2002. (See also Cryptology ePrint Archive, Report 2002/184.)

- [4] K. K. R. Choo, C. Boyd, and Y. Hitchcock, "On session key construction in provably secure protocols," in *Proceedings of Mycrypt'05*, LNCS 3715, pp. 116-131, Springer-Verlag, New York, 2005.
- [5] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.
- [6] H. Lee and Y. Lee, "Identity based key agreement from pairings," *Communication Korean Mathematics Society*, vol. 20, no. 4, pp. 849-859, 2005.
- [7] N. McCullagh and P. S. L. M. Barreto, "A new two-party identity-based authenticated key agreement," in *Proceedings of CT-RSA'05*, LNCS 3376, pp. 262-274, Springer-Verlag, New York, 2005.
- [8] A. Menezes, P. v. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
- [9] C. Mitchell, M. Ward, and P. Wilson, "Key control in key agreement protocols," *Electronics Letters*, vol. 34, no. 10, pp. 980-981, 1998.
- [10] M. Naor, "Bit commitment using pseudorandomness," in *Proceedings of Crypto'89*, LNCS 435, pp. 128-137, Springer-Verlag, New York, 1990.
- [11] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing," in *Proceedings of the 2000 Symposium on Cryptography and Information Security*, Okinawa, Japan, 2000.
- [12] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of Crypto'84*, LNCS 196, pp. 47-53, Springer-Verlag, New York, 1984.
- [13] N. P. Smart, "An identity based authenticated key agreement protocol based on the Weil pairing," *Electronics Letters*, vol. 38, no. 13, pp. 630-632, 2002.
- [14] Y. Wang, "Efficient identity-based and authenticated key agreement protocol," ePrint Archive, 2005. (<http://eprint.iacr.org/2005/108>)
- [15] S. B. Wilson and A. Menezes, "Authenticated diffie-hellman key agreement protocols," in *Proceedings of SAC'98*, LNCS 1556, pp. 339-361, Springer-Verlag, New York, 1999.

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.

Feng Cao received his B. S. degree in Department of Mathematics from Jiangxi Normal University in 2001 and M. S. degrees in College of Mathematics and Computer Science from Guangxi Normal University in 2004. Currently, he is a doctoral candidate in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His current research interests include cryptography and network security.