Science in China Series F: Information Sciences

© 2009 SCIENCE IN CHINA PRESS

Springer

www.scichina.com
info.scichina.com
www.springerlink.com

Perfect forward secure identity-based authenticated key agreement protocol in the escrow mode

WANG ShengBao[1,2], CAO ZhenFu[1†], CHENG ZhaoHui[3] & CHOO Kim-Kwang Raymond[4,5‡]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China;

[2] Computing Center, Artillery Academy of PLA, Hefei 230031, China;

[3] Independent Consultant, Room 1415 International Chamber of Commerce Building A, Shenzhen 518048, China;

[4] Australian Institute of Criminology, GPO Box 2944, Canberra ACT 2601, Australia;

[5] ARC Centre of Excellence in Policing and Security, Regulatory Institutions Network, Australian National University, Australia

The majority of existing escrowable identity-based key agreement protocols only provide partial forward secrecy. Such protocols are, arguably, not suitable for many real-word applications, as the latter tends to require a stronger sense of forward secrecy—perfect forward secrecy. In this paper, we propose an efficient perfect forward-secure identity-based key agreement protocol in the escrow mode. We prove the security of our protocol in the random oracle model, assuming the intractability of the Gap Bilinear Diffie-Hellman (GBDH) problem.

authenticated key agreement, perfect forward secrecy, bilinear pairing, provable security, modular security proof

1 Introduction

Key agreement protocols are fundamental to establishing secure communications between two or more parties over an insecure network. A key establishment protocol (including key agreement protocol) allows two or more communicating parties to establish a common secret (session) key via public communication channels (e.g., Internet). The established session key can then be used to create a confidential or integrity-protected communication channel between the parties. Authenticated key agreement (AK) protocols not only allow parties to compute the session key but also ensure the authenticity of the involved parties[1].

Authenticated key agreement protocols can be built using secret-key cryptography and/or public-key cryptography. If secret-key cryptography is used, then either a symmetric secret key or a shared password should be distributed before the key agreement protocol is executed. If traditional (certificate-based) public-key cryptography is used, a public key infrastructure (PKI) will typically be required to be deployed that allows authentication of registered users' public keys. In 1984, Shamir[2] proposed the concept of identity-based cryptography whereby each party's public key can

Received July 21, 2008; accepted October 28, 2008
doi: 10.1007/s11432-009-0135-4

† Corresponding author (email: cao-zf@cs.sjtu.edu.cn)

‡ The views and opinions expressed in this article are those of the author and should not be taken to represent, in whole or in part, the views of the Australian Government, the Australian Institute of Criminology, and the Australian National University. Research was undertaken in the author's personal capacity.

Supported in part by the National High-Tech Research & Development Program of China (Grant No. 2006AA01Z424), the National Natural Science Foundation of China (Grant Nos. 60673079, 60773086), and the National Basic Research Program of China (Grant No. 2007CB311201)

Citation: Wang S B, Cao Z F, Cheng Z H, et al. Perfect forward secure identity-based authenticated key agreement protocol in the escrow mode. Sci China Ser F-Inf Sci, 2009, 52(8): 1358–1370, doi: 10.1007/s11432-009-0135-4

be an arbitrary string (typically an identity string) and, hence, removes the need for certificates. This greatly simplifies the management of public keys in identity-based cryptosystems. Following the work of Boneh and Franklin[3] on identity-based encryption, several identity-based two-party key agreement schemes using bilinear pairings on elliptic curves have been proposed (see ref. [4]). Examples include:

- In 2002, Smart[5] proposed an ID-based key agreement protocol based on the ID-based encryption scheme in ref. [3]. However, Shim[6] and Chen and Kudla[7] independently showed that Smart's protocol only provides *half* forward secrecy. Chen and Kudla also proposed several ID-based key agreement protocols[7] (and the basic Chen–Kudla protocol is the only protocol in the suite that works in the escrow mode).

- In 2003, Shim[6] proposed an efficient ID-based key agreement protocol claiming to provide perfect forward secrecy, known-key secrecy, key-compromise impersonation (K-CI) resilience, and unknown key-share (UK-S) resilience (refer to section 2.2 for the definitions of all these terms). Sun and Hsieh[8], however, pointed out that Shim's protocol is vulnerable to a man-in-the-middle attack.

- In 2004, Ryu et al.[9] put forward a pairing-based protocol which has superb efficiency. A year later, Boyd and Choo[4] and Wang et al.[10] independently demonstrated that the protocol does not have the property of K-CI resilience. More recently in 2009, we showed that the protocol is, in fact, vulnerable to a reflection attack[11]. We then proposed an improved protocol and proved it secure in a widely accepted security model.

- In 2005, McCullagh and Barreto[12] proposed an ID-based key agreement protocol but was soon found out by Xie[13] that their protocol does not have K-CI resilience. An improved McCullagh–Barreto protocol presented in ref. [14] successfully removes the found security weakness but at the cost of losing PFS. Xie[15] also proposed an improved McCullagh–Barreto protocol, but Li et al.[16] later discovered that Xie's protocol still does not have K-CI resilience.

- Both Wang[17] and Yuan et al.[18] proposed a new ID-based key agreement protocol in 2005. The former achieves PFS in the escrow model while the latter (protocol) works only in the escrowless model.

Motivation 1. Secure key agreement protocol in the escrow mode
ID-based authenticated key agreement protocols may either work in the escrowed mode (i.e., the private key generator (PKG) is able to recover the session keys established by its users) or escrowless mode (i.e., the PKG is unable to recover the session keys established by its users). A majority of the ID-based key agreement protocols are designed to be escrowless due to privacy concerns. However, as noted in ref. [12], key escrow is desirable under certain circumstances especially in certain closed groups applications. For example, escrow is essential in situations where audit trail is a legal requirement, such as in secure communications applications in the health care profession.

It is possible to extend an escrowable protocol to work in the escrowless mode (and achieve PKG forward secrecy) by embedding a raw Diffie–Hellman protocol (see ref. [7] for further details). Perfect forward secure ID-based key agreement protocols are, however, uncommon in the escrow mode. As far as we are aware, only Wang's protocol[17] and Cheng et al.'s protocol[19] provide PFS and are escrowable.

Our first motivation is, therefore, to propose a more efficient protocol that is not only secure in the escrow mode but also provides perfect forward secrecy.

Motivation 2. Simplifying the proof—a modular approach
It is by now standard practice for protocol designers to provide security proof in widely accepted security models in order to assure protocol implementors of their security properties (see ref. [20]). As pointed out in a recent survey of two-party ID-based authenticated key agreement protocols[4], many existing protocols are not proven secure in the modular approach (e.g., the model of Canetti and Krawczyk[21]) and their proofs of security are often complicated and error-prone[20−23].

Kudla and Paterson[23,24] developed a modular technique for constructing security proofs for a large class of key agreement protocols using a slightly modified Smart's protocol as an example. Informally, their modular technique works in the following sequence.

1. Prove that a protocol Π has the property of strong partnering.

2. Prove that a related protocol π is secure in a highly reduced security model.

3. The security proof for π in the reduced model is then translated into a security proof for Π in the full model using a Gap assumption[25].

This proof technique using the modular approach is easier to use (compared to the conventional approach) and, hence, forms the second motivation of our paper.

Our contributions

1. We propose an efficient ID-based AK protocol (with only a single online pairing computation) that works in the escrow mode. We also demonstrate that our proposed protocol achieves perfect forward secrecy without compromising on efficiency.

2. Using the modular technique of Kudla and Paterson, we prove that our proposed protocol is secure (for all the security properties except PFS) in the random oracle model, provided that the Gap Bilinear Diffie–Hellman (GBDH) problem is hard. Moreover, we prove in a conventional way that our protocol also achieves perfect forward secrecy, assuming the hardness of the standard Bilinear Die-Hellman (BDH) problem.

The remainder of this paper is structured as follows. In section 2, we briefly describe bilinear pairings, the computational problems and the corresponding complexity assumptions, the security model, and the Kudla–Paterson modular proof approach[24] required in this paper. We present our proposed ID-based authenticated key agreement protocol (hereafter referred to as E-IBAK) in section 3. In section 4, a detailed security proof in the random oracle model[26] of our proposed E-IBAK protocol is provided. Section 5 provides a performance comparison between several related ID-based protocols. We draw our conclusions in section 6.

2 Preliminaries

2.1 Bilinear pairings and GBDH probolem

Let $\mathbb{G}_1$ denotes an additive group of prime order q and $\mathbb{G}_2$ a multiplicative group of the same order. We let P denote a generator of $\mathbb{G}_1$. For us, an admissible pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties:

1. The map $\hat{e}$ is bilinear: given $Q, R \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$, we have $\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$.

2. The map $\hat{e}$ is non-degenerate: $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$.

3. The map $\hat{e}$ is efficiently computable.

Typically, the map $\hat{e}$ will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field. We refer to refs. [3, 27, 28] for a more all-around description of how these groups, pairings and other parameters should be chosen in practice for efficiency and security.

Definition 1 (Bilinear Diffie-Hellman (BDH) parameter generator)[3]**.** We say that a randomized algorithm $\mathcal{IG}$ is a BDH parameter generator if $\mathcal{IG}$ takes a security parameter $l > 0$, runs in time polynomial in l, and outputs the description of two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order q and the description of an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.

The security of the ID-based key agreement protocols in this paper are based on the difficulty of the following problems:

Definition 2 (Bilinear Diffie-Hellman (BDH) problem)**.** Let $\mathbb{G}_1$, $\mathbb{G}_2$, P and $\hat{e}$ be as above. The BDH problem in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ is as follows: Given $\langle P, aP, bP, cP \rangle$ with uniformly random choices of a, b, $c \in \mathbb{Z}_q^*$, compute $\hat{e}(P, P)^{abc} \in \mathbb{G}_2$.

We say that a probabilistic polynomial time (PPT) algorithm $\mathcal{B}$ has advantage ϵ in solving the BDH problem in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ if

$$\Pr[\mathcal{B}(P, aP, bP, cP) = \hat{e}(P, P)^{abc}] \geqslant \epsilon,$$

where the probability is measured over the random choices of $a, b, c \in \mathbb{Z}_q^*$ and the random bits of $\mathcal{B}$.

The above BDH problem has a decisional counterpart called the decisional bilinear Diffie-Hellman (DBDH) problem which is defined as follows.

Definition 3 (Decisional Bilinear Diffie-

Hellman (DBDH) problem). Let $\mathbb{G}_1$, $\mathbb{G}_2$, P and $\hat{e}$ be as above. The DBDH problem in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ is as follows: Given $\langle P, aP, bP, cP \rangle$ with uniformly random choices of a, b, $c \in \mathbb{Z}_q^*$, as well as $W \in \mathbb{G}_2$, determine if $\hat{e}(P,P)^{abc} = W$ (if it holds, then the tuple $\langle P, aP, bP, cP, W \rangle$ is called a BDH tuple).

Based on the BDH and DBDH problems, we can define a related Gap problem[25] as follows.

Definition 4 (Gap Bilinear Diffie-Hellman GBDH problem). Let $\mathbb{G}_1$, $\mathbb{G}_2$, P and $\hat{e}$ be as above. The GBDH problem in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ is as follows: Given $\langle P, aP, bP, cP \rangle$ with uniformly random choices of a, b, $c \in \mathbb{Z}_q^*$, as well as an oracle the solves the DBDH problem in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$, compute $\hat{e}(P,P)^{abc}$.

Informally, the BDH, DBDH and GBDH assumptions are that no PPT adversary has non-negligible advantage in solving the BDH, DBDH and GBDH problems, respectively.

2.2 Desirable security attributes

Let Alice (A) and Bob (B) be two honest entities, i.e., legitimate entities who execute the steps of a protocol correctly. Here we list up a number of desirable attributes of AK protocols which referred to refs. [1, 7].

- Known-key secrecy (K-KS). Each run of a key agreement between A and B should produce a unique secret session key. The compromise of one session key should not compromise the keys established in other sessions.

- Perfect forward secrecy (PFS). If long-term private keys of all entities are compromised, the secrecy of previous session keys established by honest entities is not affected.

- 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 that this disclosure does not allow the adversary to impersonate any other entity (e.g. B) to A.

- Unknown key-share (UK-S) resilience. Entity A cannot be coerced into sharing a key with entity B without A's knowledge, i.e., when A believes that the key is shared with some entity $C \neq B$, and B (correctly) believes the key is shared with A.

- No key control. Neither entity should be able to force the session key (or any portion of the session key) to a preselected value.

In addition to these security attributes, it would be desirable for a protocol to have low computational cost (the computing operations needed for A and B to finish a run of the protocol) and low communication overhead (which means that only a small amount of data is exchanged) for its practical use.

2.3 Security model for ID-based AK protocols—ID-mBJM

In this subsection, we present our refined formal security model for ID-based authenticated key agreement protocols. Kudla[23] proposed the so called ID-BJM model, which is an extension of the model of Blake-Wilson et al.[29] (known as the BJM model). In this paper, we extend a modified version of the BJM model to the ID-based setting which we call the ID-mBJM model. Following the approach of Choo et al.[30], we use the notion of session identifier SID (instead of matching conversation used in the BJM model and Kudla's ID-BJM model) in our partnership definition.

The model includes a set U of participants modeled by a collection of oracles (e.g., oracle $\Pi_{I,J}^n$ represents the nth instance of participant I carrying out a protocol session in the belief that it is communicating with another participant J). Each participant has a long-term ID-based public/private key pair, in which the public key is generated using her identity information and the private one is computed and issued secretly by a private key generator.

There is an active adversary (denoted by E) in the model modeled by a PPT Turing Machine which has access to all the participants' oracles[1]. Participant oracles only respond to queries by the adversary and do not communicate directly among themselves, i.e., there exists at least a benign adversary who simply passes messages between participants faithfully.

Definition of security in the model depends on

[1] If the proof is given in the random oracle model (ROM), then the adversary also has access to all the existing random oracles.

the notion of the partner oracles to any oracle being tested. We define partners by having the same session identifier (SID). Concretely, we define SID($\Pi^n_{I,J}$) as the concatenation of all messages that oracle $\Pi^n_{I,J}$ has sent and received.

Definition 5 (Partner)**.** Two oracles $\Pi^n_{I,J}$ and $\Pi^{n'}_{J,I}$ are said to be partner oracles if they have accepted with the same SID.

The security of a protocol is defined via a two-phase adaptive game (called the ID-mBJM game) between a challenger $\mathcal{C}$ that simulates a set of participant oracles running the protocol and the adversary E. $\mathcal{C}$ also simulates the PKG in this environment, and therefore generates the public parameters of the PKG and gives these to E. $\mathcal{C}$ also generates a master secret s from which it can generate a private key d_I from any given identity I.

In the first phase, the adversary E is allowed to issue the following queries in any order.

Send(I, J, n, M): E can send message M to oracle $\Pi^n_{I,J}$. The oracle executes the protocol and responds with an outgoing message m or a decision to indicate accepting or rejecting the session. Any incoming and outgoing message is recorded on its transcript. If $M = \lambda$ (denotes the null message), then the oracle initiates a protocol run.

Reveal($\Pi^n_{I,J}$): To respond to the query, oracle $\Pi^n_{I,J}$ returns the accepted session key (if any). Otherwise, returns a symbol $\perp$. Oracle $\Pi^n_{I,J}$ is then considered opened.

Corrupt(I): To respond to the query, $\mathcal{C}$ returns the private key d_I of the participant I. The participant is then considered corrupted.

Test($\Pi^n_{I,J}$): At some point, E can make a Test query to some fresh oracle $\Pi^n_{I,J}$ (see Definition 6 below). To answer the query $\mathcal{C}$ flips a fair coin $b \in \{0, 1\}$; if the answer is 0, then $\mathcal{C}$ outputs the agreed session key of the test oracle, otherwise outputs a randomly chosen value from the session key space.

During the second phase, E can continue issuing Send, Reveal and Corrupt queries to the oracles, except for revealing the target test oracle or its partner oracle (if any). Moreover, E is not allowed to corrupt participant J (assuming $\Pi^n_{I,J}$ is the test oracle).

Output: Finally, E outputs a prediction (b') on b. E wins the game if $b' = b$, and we define E's advantage (l is the security parameter) in winning the game as

$$Adv^E(l) = |\Pr[b' = b] - 1/2|.$$

Definition 6 (Fresh oracle)**.** An oracle $\Pi^n_{I,J}$ ($I \neq J$) is called fresh if it has accepted (and therefore holds a session key sk_i), it is not opened, J has not been corrupted, and there is no opened oracle $\Pi^{n'}_{J,I}$ which is a partner oracle of $\Pi^n_{I,J}$.

Remark 1. The above definition of fresh oracle is particularly defined to cover the security attribute of key-compromise impersonation resilience since it implies that the participant I could have been issued a Corrupt query[31].

Definition 7 (ID-mBJM secure protocol)**.** A protocol is a secure AK protocol in the ID-mBJM model if:

1. In the presence of the benign adversary (who faithfully relays messages between parties) on $\Pi^n_{I,J}$ and $\Pi^{n'}_{J,I}$, both oracles always accept holding the same session key, and this key is distributed uniformly on session key space.

2. $Adv^E(l)$ is negligible.

In the following, we briefly discuss the security properties (described in section 2.2) captured by the above security model.

• Known-key secrecy (K-KS). The property of known-key secrecy is implied by the definition of AK security (see Definition 7). Since E is allowed to make Reveal queries to any oracles except for the target Test oracle $\Pi^n_{I,J}$ and its partner oracle $\Pi^{n'}_{I,J}$ to obtain any session keys. Even with the knowledge of many other session keys, E's ability to distinguish between the session key held by $\Pi^n_{I,J}$ and a random number is still negligible. That is to say, the knowledge of any other session keys does not help E to deduce any information about the tested session key.

• Perfect forward secrecy (PFS). Definition 7 does not imply the property of perfect forward secrecy. This is because the model does not allow the Test query to be issued on an oracle with both participants corrupted and therefore does not model this type of attack.

- Key-compromise impersonation (K-CI) resilience. As mentioned above, the definition of fresh oracle implies the property of key-compromise impersonation resilience.

- Unknown key-share (UK-S) resilience. Definition 7 implies the unknown key-share resilience property. If ID_I establishes a session key with ID_J believes he is talking to ID_K, then there is an oracle $\Pi^n_{I,K}$ that holds this session key sk_{IK}. At the same time, there is an oracle $\Pi^{n'}_{J,I}$ that holds this session key sk_{IK}, for some n' (normally $n' = n$). Since $\Pi^n_{I,K}$ and $\Pi^{n'}_{J,I}$ are not partner oracles, the adversary can make a Reveal query to $\Pi^{n'}_{J,I}$ to learn this session key before asking a Test query to $\Pi^n_{I,K}$. Thus the adversary will succeed for this Test query challenge (i.e., the protocol is not secure) if the unknown key share attack is possible. By contradiction, a secure protocol in the model is resistant to the unknown key share attack.

- No key control. Definition 7 does not imply resilience to key control attacks that are launched by one of the protocol participants. However, key control attacks launched by an outside adversary are captured by the model. Otherwise, if manipulating message can control the session key bits, the outsider adversary E must have a non-negligible ability to distinguish between the session key held by $\Pi^n_{I,J}$ and a random number.

To model PFS, the definition of fresh oracle (refer to Definition 6) should be modified so that the the participants associated with the Test (fresh) oracle can also be corrupted. We define PFS as follows.

Definition 8 (perfect forward secrecy (PFS)). A protocol is said to have perfect forward secrecy (PFS) if any PPT adversary wins the ID-mBJM game with negligible advantage when it chooses an unopened oracle $\Pi^n_{I,J}$ which has an unopened partner oracle $\Pi^{n'}_{J,I}$ as the test oracle, and both oracles $\Pi^n_{J,I}$ and $\Pi^{n'}_{J,I}$ accepted and both participants I and J can be corrupted.

Note that as in ref. [32], here we refer to the practical notion of perfect forward secrecy that involves a benign adversary eavesdropping on a session of the protocol and then attempting to expose the key.

2.4 Modular proof technique for ID-based AK protocols

We borrow from refs. [11, 23] the review of the modular proof technique. For a more detailed description, we refer the reader to ref. [23]. Note that here we mainly focus on ID-based protocols.

As noted in refs. [23, 24], the modular proof technique only works on key agreement protocols that produce hashed session keys on completion of the protocol. In fact, this reliance on hashing is reasonable since it is quite common to use a key derivation function (KDF) to output a session key from a secret value established during a key agreement protocol, and the KDF is usually implemented via a hash function.

Definition 9 (session string). Suppose Π is a protocol that produces a hashed session key using the cryptographic hash function H. Then the session string for a particular oracle $\Pi^i_{I,J}$ is denoted $ss_{\Pi^i_{I,J}}$, and is defined to be the string which is hashed to produce the session key $sk_{\Pi^i_{I,J}}$. So we have that $sk_{\Pi^i_{I,J}} = H(ss_{\Pi^i_{I,J}})$.

Strong partnering. Suppose Π is a key agreement protocol. If there exists an adversary E, which when attacking Π in an ID-mBJM game defined in section 2.3 and with non-negligible probability in the security parameter l, can make some two oracles $\Pi^i_{I,J}$ and $\Pi^n_{J,I}$ accept holding the same session key when they are not partners, then we say that Π has weak partnering. If Π does not have weak partnering, then we say that it has strong partnering.

As shown in ref. [23], for a protocol Π to be ID-mBJM secure, it must have strong partnering. Since H is modeled as a random oracle, strong partnering can be ensured by including appropriate "partnering information" in the session string $ss_{\Pi^i_{I,J}}$, where partnering information is used to decide whether the two oracles are partners or not. In section 3, we will use the session identifier SID and the identities of the two parties as the partnering information of our new protocol.

Reduced games. A highly reduced game (called the cNR-ID-mBJM game) is used in the modular security proof. The reduced game is identical as the full ID-mBJM game defined in section

2.3 except that the adversary E is not allowed to make Reveal queries and to win the game, E must select an accepted fresh oracle on which to make a modified Test query at the end of its attack and output the session key held by this oracle. We define E's advantage, denoted $Adv^E(l)$, in the cNR-ID-mBJM game to be the probability that E outputs a session key sk such that $sk = sk_{\Pi^i_{I,J}}$ where $\Pi^i_{I,J}$ is the oracle selected by the adversary for the modified Test query. We define security in the reduced game as follows:

Definition 10 (cNR-ID-mBJM secure protocol[23])**.** A protocol Π is a secure key agreement protocol in the cNR-ID-mBJM model if:

1. In the presence of the benign adversary, two oracles running the protocol both accept holding the same session key, and the session key is distributed uniformly at random on session key space.

2. For any adversary E, $Adv^E(l)$ in the reduced game is negligible.

As part of the the proof technique, it will be necessary to prove that a related protocol π of protocol Π is secure in the above reduced game.

Related protocol π. The related protocol π of protocol Π is defied in the same way as Π except that the session key generated by π is defined to be the session string of Π rather than the hash of this string (i.e., $sk_{\pi^n_{I,J}} = ss_{\Pi^n_{I,J}}$). It is usually quite easy to establish a related protocol's security in the reduced game.

Definition 11 (session string decisional problem)**.** Given the public parameters, the transcript $T_{\Pi^n_{I,J}}$ of oracle $\Pi^n_{I,J}$, as well as the public keys of I and J and a string s, decide whether $s = ss_{\Pi^n_{I,J}}$, where $ss_{\Pi^n_{I,J}}$ is the session string of oracle $\Pi^n_{I,J}$.

The following result is at the heart of the modular proof technique that translates the weak security of a related weaker protocol into the security of the protocol in the full model.

Theorem 1 (Theorem 8.2 in ref. [23])**.** Suppose that key agreement protocol Π produces a hashed session key on completion of the protocol (via hash function H) and that Π has strong partnering. If the security of the related protocol π in the reduced game is probabilistic polynomial time reducible to the hardness of the computational problem of some relation f, and the session string decisional problem for Π is polynomial time reducible to the decisional problem of f, then the security of Π in the full model is probabilistic polynomial time reducible to the hardness of the Gap problem of f, assuming that H is a random oracle.

3 Proposed identity-based AK protocol

We now describe our proposed escrowable identity-based authenticated key agreement protocol with perfect forward secrecy (see Figure 1, hereafter referred to as E-IBAK). Our scheme employs the ID-based non-interactive key sharing protocol due to Sakai et al.[33] (hereafter referred to as the SOK protocol).

As with all other identity-based cryptosystems we assume the existence of a trusted PKG that is responsible for the generation and secure distribution of users' private keys. Our proposed key agreement protocol can be implemented using either the modified Weil or Tate pairing[3,12]. The proposed protocol comprises the following two stages:

Setup: Suppose we have an admissible pairing, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ 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:

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

Individual user's public/private key pair is, thus, defined as (Q_{ID}, d_{ID}) where $Q_{ID}, d_{ID} \in \mathbb{G}_1$.

Key agreement: We denote user Alice and Bob's public/private key pairs as (Q_A, d_A) and (Q_B, d_B), respectively. We assume that Alice and Bob both pre-compute and store the following SOK

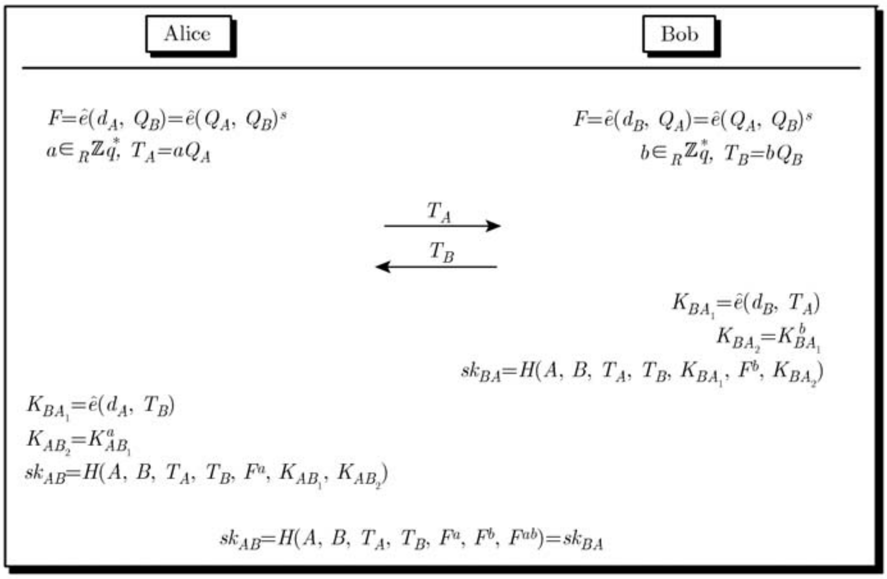
Alice
Bob
$F=\hat{e}(d_A, Q_B)=\hat{e}(Q_A, Q_B)^s$
$a\in_R \mathbb{Z}_q^*, T_A=aQ_A$
$F=\hat{e}(d_B, Q_A)=\hat{e}(Q_A, Q_B)^s$
$b\in_R \mathbb{Z}_q^*, T_B=bQ_B$
T_A
T_B
$K_{BA_1}=\hat{e}(d_B, T_A)$
$K_{BA_2}=K_{BA_1}^b$
$sk_{BA}=H(A, B, T_A, T_B, K_{BA_1}, F^b, K_{BA_2})$
$K_{BA_1}=\hat{e}(d_A, T_B)$
$K_{AB_2}=K_{AB_1}^a$
$sk_{AB}=H(A, B, T_A, T_B, F^a, K_{AB_1}, K_{AB_2})$
$sk_{AB}=H(A, B, T_A, T_B, F^a, F^b, F^{ab})=sk_{BA}$

Figure 1 Proposed protocol E-IBAK.

(non-interactively shared) secret[32]:

$$F = \hat{e}(d_A, Q_B) = \hat{e}(d_B, Q_A) = \hat{e}(Q_A, Q_B)^s.$$

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 $T_A = aQ_A$ and $T_B = bQ_B$, respectively. They then exchange T_A, T_B and compute the session key as described in Figure 1, where $H : \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{G}_2 \times \mathbb{G}_2 \rightarrow \{0,1\}^k$ is a key derivation function (in which $k = |sk|$).

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

$$K_{AB_1} = \hat{e}(d_A, T_B) = \hat{e}(d_A, bQ_B) = F^b,$$
$$K_{BA_1} = \hat{e}(d_B, T_A) = \hat{e}(d_B, aQ_A) = F^a,$$

and $K_{AB_2} = K_{BA_2} = F^{ab}$.

Thus, the two session keys computed by Alice and Bob are

$$sk_{AB} = sk_{BA} = H(A, B, T_A, T_B, F^a, F^b, F^{ab}).$$

Escrow. The protocol E-IBAK has the escrow function, namely the PKG can recover all the session keys using the master secret key s and other public data such as T_A and T_B. We prove this as follows.

$$F^a = \hat{e}(Q_A, Q_B)^{sa} = \hat{e}(T_A, Q_B)^s,$$
$$F^b = \hat{e}(Q_A, Q_B)^{sb} = \hat{e}(Q_A, T_B)^s,$$
$$F^{ab} = \hat{e}(Q_A, Q_B)^{sab} = \hat{e}(T_A, T_B)^s.$$

The protocol is message independent and role symmetric, which means that each party performing the same operations and thus incurring the same computational cost. In the next section we will prove that our protocol E-IBAK achieves the ID-mBJM security (see Definition 10) as well as perfect forward secrecy.

4 Security proof

We prove the security (i.e. ID-mBJM security plus PFS) of our new protocol E-IBAK in stages. We first give a basic identity-based protocol, E-IBAK', which does not provide perfect forward secrecy, and prove that it is ID-mBJM secure using the Kudla–Paterson modular technique. We then prove that the protocol E-IBAK is also secure in the ID-mBJM model and provides perfect forward secrecy. The only reason for describing the protocol E-IBAK' is to make the presentation easier to follow.

Protocol E-IBAK' is almost identical to protocol E-IBAK except that the final session key is computed as

$$sk_{AB} = H'(A, B, T_A, T_B, F^a, F^b),$$

where $H' : \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{G}_2 \rightarrow \{0,1\}^k$ is a key derivation function. In other words, without the value F^{ab} being part of the session string. With the description of the ID-mBJM model in section 2.3, we now state:

Theorem 2 (ID-mBJM security of E-IBAK')**.** If H' and H_1 are random oracles and the GBDH problem (for the pair of groups $\mathbb{G}_1$ and $\mathbb{G}_2$) is hard, then E-IBAK' is a secure key agreement protocol.

We now prove Theorem 2 in three steps. We first show that protocol E-IBAK' has strong partnering. Secondly, we prove that the related protocol π of E-IBAK' is secure in the cNR-ID-mBJM model (see Definition 10). Lastly, we show that the session string decisional problem (see Definition 11) of E-IBAK' is reducible to the DBDH problem.

Lemma 1 (Strong Partnering of E-IBAK')**.** Protocol E-IBAK' has strong partnering in the random oracle model.

Proof. The partnering information, namely the protocol transcript as well as the IDs of the two participants are included in the session string. Recall that we model H' as a random oracle, thus if two oracles accept the same session key, then they are partners except for negligible probability.

Lemma 2 (cNR-ID-mBJM security of π)**.** The related protocol π is secure in the cNR-ID-mBJM model, assuming the BDH problem is hard (for the pair of groups $\mathbb{G}_1$ and $\mathbb{G}_2$) and provided that H_1 is a random oracle.

Proof. Condition 1 follows from the correctness of the protocol π. Since H' is a random oracle, sk is distributed uniformly at random on $\{0,1\}^k$. In the following, we show that Condition 2 is also satisfied.

For a contradiction, assume that the adversary E has non-negligible advantage ϵ in winning the cNR-ID-mBJM game, making at most q_1 queries to H_1. Let q_S be the total number of the oracles that E creates, i.e., for any oracle Π^n_{AB}, $n \in \{1,\dots,q_S\}$. We shall slightly abuse the notation Π^n_{AB} to refer to the nth one among all the q_S participant instances in the game, instead of the nth instance of participant A.

We show how to construct a simulator S that uses E as a sub-routine to solve the BDH problem with non-negligible probability. Given input of the two groups $\mathbb{G}_1$, $\mathbb{G}_2$, the bilinear map $\hat{e}$, a generator P of $\mathbb{G}_1$, and a triple of elements xP, yP, $zP \in \mathbb{G}_1$ with $x, y, z \in \mathbb{Z}_q^*$ where q is the prime order of $\mathbb{G}_1$ and $\mathbb{G}_2$, S's task is to compute and output the value $\hat{e}(P,P)^{xyz}$.

The algorithm S selects a random integer v from $\{1,\dots,q_1\}$ and a random integer w from $\{1,\dots,q_S\}$ and works by interacting with E as follows:

Setup: S sets the PKG's master public key to be xP. S will also simulate all oracles required during the game. S controls the H_1 random oracle. S starts E, and answers all E's queries as follows.

$\boldsymbol{H_1(ID_i)}$**:** S simulates the random oracle H_1 by keeping a list of tuples $\langle r_i, ID_i, Q_i\rangle$ which is called the H_1-list. When the H_1 oracle is queried with an input $ID_i \in \{0,1\}^*$, S responds as follows.

• If ID_i is already on the H_1-list in the tuple $\langle r_i, ID_i, Q_i\rangle$, then S outputs Q_i.

• Otherwise, if ID_i is the vth distinct H_1 query, then the oracle outputs $Q_i = yP$ and adds the tuple $\langle \perp, ID_i, Q_i\rangle$ to the H_1-list.

• Otherwise S selects a random $r_i \in \mathbb{Z}_q^*$ and outputs $Q_i = r_iP$, and then adds the tuple $\langle r_i, ID_i, Q_i\rangle$ to the H_1-list.

We assume that J is the vth distinct participant created in the game.

Corrupt(ID_i)**:** Upon receiving the Corrupt query on input ID_i, S simulates as follows.

• If $ID_i \neq J$, S returns the long-term private key d_i.

• Otherwise, S aborts the game (Event 1).

Send(A,B,t,M): S answers the queries as follows.

• If $t \neq w$, S randomly samples $\xi_t \in \mathbb{Z}_q^*$ and responds with $\xi_t Q_A$ where $Q_A = H_1(A)$.

• Otherwise,

– If $B \neq J$, aborts the game (Event 2).

– Otherwise answer zP.

Test$(\Pi^t_{A,B})$: At some point in the game, E will issue a unique Test query. If E does not choose the guessed oracle $\Pi^w_{A,B}$ to issue the query, then S aborts the game (Event 3).

Output: At the end of the game, the algorithm E outputs a session key of the form (U,V,a,b,c,d) where $U, V \in \{0,1\}^*$, $a, b \in \mathbb{G}_1$ and $c, d \in \mathbb{G}_2$.

Solving the BDH problem: If $\Pi^w_{A,B}$ was an initiator oracle, then S outputs c as its guess for the value $\hat{e}(P,P)^{xyz}$, otherwise S outputs d as its guess.

Now we calculate the probability that the game does not abort. If the wth oracle was chosen as the test oracle and it supposes to establish a session key with party J, then by the rules of the model Events 1, 2 and 3 would not happen. We have

$$\Pr[S \text{ does not abort}] \geqslant \frac{1}{q_S q_1}.$$

Note that participant J has the public key $Q_J = yP$ and private key $d_J(= xyP)$. Given a message zP, part of the agreed secret is $\hat{e}(xyP, zP)$. So if the adversary computes the correct session key with non-negligible probability ϵ, then S answers the BDH problem correctly with probability with $\epsilon/(q_S q_1)$ (which is non-negligible in the security parameter l), contradicting to the hardness of the BDH problem.

Lemma 3 (Session string decisional problem of E-IBAK')**.** The session string decisional problem of protocol E-IBAK' is reducible to the DBDH problem.

Proof. Recall the session string of protocol E-IBAK' is of the form $(A, B, T_A, T_B, F^a, F^b)$ with $T_A = aQ_A$, $F^a = \hat{e}(d_B, T_A) = \hat{e}(sQ_B, T_A)$, s being the master secret key and P, sP being the public parameters, then we see $\langle P, sP, Q_B, T_A, F^a\rangle$ (similarly, $\langle P, sP, Q_A, T_B, F^b\rangle$) is a BDH tuple. This implies that the session string decisional problem of protocol E-IBAK' is reducible to the DBDH problem.

Proof of Theorem 2. The theorem follows directly from Lemmas 1, 2, 3 and Theorem 1.

Now we are ready to prove our main result—the security of our newly proposed protocol E-IBAK.

Theorem 3 (security of protocol E-IBAK)**.** Protocol E-IBAK

i) is secure in the ID-mBJM model, assuming the GBDH problem (for the pair of groups $\mathbb{G}_1$ and $\mathbb{G}_2$) is hard and provided that H and H_1 are random oracles, and

ii) has the property of perfect forward secrecy (PFS), assuming the BDH problem (for the pair of groups $\mathbb{G}_1$ and $\mathbb{G}_2$) is hard and provided that H and H_1 are random oracles.

Proof. i) This follows directly from Theorem 2, since it is easy to see that any successful attack on protocol E-IBAK can be immediately converted to a successful attack on protocol E-IBAK'.

ii) According to our definition of perfect forward secrecy (see Definition 8), we require that when E chooses an oracle $\Pi^n_{I,J}$ as the test oracle, this oracle must indeed have a partner oracle $\Pi^{n'}_{J,I}$.

The proof follows along similar lines to the proof of Lemma 2. For a contradiction, we assume that the adversary E can win the game with non-negligible advantage ϵ by creating at most q_S oracles and making q_H queries to the H random oracle. We show how to construct a simulator S that uses E as the sub-routine to solve the BDH problem with non-negligible probability. Identical to the proof of Lemma 2, the input of S are the two groups $\mathbb{G}_1$, $\mathbb{G}_2$, the bilinear map $\hat{e}$, a generator P of $\mathbb{G}_1$, and a triple of elements xP, yP, $zP \in \mathbb{G}_1$ with $x, y, z \in \mathbb{Z}^*_q$ where q is the prime order of $\mathbb{G}_1$ and $\mathbb{G}_2$, its task is to compute and output the value $\hat{e}(P,P)^{xyz}$.

The algorithm S selects two random integers u, v from $\{1, \ldots, q_S\}$ (assuming $u < v$) and works by interacting with E as follows:

Setup: S sets the PKG's master public key to be xP. S will also simulate all oracles required during the game. S controls two random oracles H_1 and H. S starts E, and answers all E's queries as follows.

$\boldsymbol{H_1(ID_i)}$**:** S simulates the oracle H_1 by keeping a list of tuples $\langle r_i, ID_i, Q_i\rangle$ which is called the H_1-list. When the H_1 oracle is queried with an input $ID_i \in \{0,1\}^*$, S responds as follows.

– If ID_i is already on the H_1-list in the tuple $\langle r_i, ID_i, Q_i\rangle$, then S outputs Q_i.

– Otherwise S selects a random $r_i \in \mathbb{Z}^*_q$ and outputs $Q_i = r_i P$, and then adds the tuple $\langle r_i, ID_i, Q_i\rangle$ to the H_1-list.

$H(ID_i, ID_j, T_i, T_j, A_i, B_i, C_i)$: S simulates the random oracle H by keeping an H-list with tuples of the form $\langle ID_i, ID_j, T_i, T_j, A_i, B_i, C_i, k_i\rangle$. If the requested input is already on the list, then the corresponding k_i is returned, otherwise a random

$k_i \in \{0,1\}^k$ is responded and a new entry is inserted into the list.

Corrupt(ID_i): Upon receiving the Corrupt query on input ID_i, S outputs the corresponding long-term private key $d_i = r_i xP$.

Send(A, B, t, M): S answers all Send queries as follows;

• When $t = u$, if oracle $M \neq \lambda$, then abort (Event 1), otherwise return yP.

• When $t = v$, if $M \neq yP$, then abort (Event 2), otherwise return zP.

• When $t \neq u, v$, randomly sample $\xi_t \in \mathbb{Z}_q^*$, return $\xi_t H_1(A)$.

Reveal$(\Pi^t_{A,B})$: S outputs the appropriate session key to answer the query. However, if E reveals oracle $\Pi^u_{A,B}$ or $\Pi^v_{A,B}$, then S aborts (Event 3). Note that given ξ_t, the input message and the private key d_A, S is able to compute the session secret.

Test$(\Pi^t_{A,B})$: S aborts (Event 4) if the guessed oracle $\Pi^u_{A,B}$ or $\Pi^v_{A,B}$ is not chosen. Otherwise, S randomly picks a value β from the session key space and responds to E with β.

Output: At the end of the game, E returns its guess.

Solving the BDH problem: S randomly picks a tuple of the form $\langle I, J, T_I, T_J, A_h, B_h, C_h\rangle$ (for some h) from the H-list and returns C_h as the response to the BDH challenge.

Now we calculate the probability that S does not abort, namely Events 1, 2, 3 and 4 do not happen. By the rule of the game, if the test session is between the uth and vth oracles, then the simulation goes through. The probability that the simulator has chosen the right session is $1/q_S^2$, because a randomly chosen oracle is the initiator of the test session is $1/q_S$ and similarly another randomly chosen oracle is the responder of the test session is also $1/q_S$. We have

$$\Pr[S \text{ does not abort}] \geqslant 1/q_S^2.$$

According to the simulation of the Send query, the test oracle $\Pi^u_{I,J}$ must have obtained the value $T_J = zP$ from its partner oracle $\Pi^v_{J,I}$. The oracle should hold a session key of the form $H(I, J, T_I, T_J, A_h, B_h, \hat{e}(d_I, T_J)^{y/r_I})$, in which $\hat{e}(d_I, T_J)^{y/r_I} = \hat{e}(xr_I P, zP)^{y/r_I} = \hat{e}(P, P)^{xyz}$.

Let Q be the event that the session string of the test oracle has been queried to H. Because of the construction of the session string and the use of session identifier to define partner oracles, we can easily prove that if Q happens with non-negligible probability in the random oracle model, it must be caused by a query issued by the adversary (see the detailed argument in ref. [34]). Because H is a random oracle, we have $\Pr[E \text{ wins}|\bar{Q}] = 1/2$. Then

$$\begin{aligned}\Pr[E \text{ wins}] &= \Pr[E \text{ wins}|\bar{Q}]\Pr[\bar{Q}] \\ &\quad + \Pr[E \text{ wins}|Q]\Pr[Q] \\ &\leqslant \Pr[E \text{ wins}|\bar{Q}]\Pr[\bar{Q}] + \Pr[Q] \\ &= \frac{1}{2}\Pr[\bar{Q}] + \Pr[Q] \\ &= \frac{1}{2} + \frac{1}{2}\Pr[Q].\end{aligned}$$

$$\begin{aligned}\Pr[E \text{ wins}] &= \Pr[E \text{ wins}|\bar{Q}]\Pr[\bar{Q}] \\ &\quad + \Pr[E \text{ wins}|Q]\Pr[Q] \\ &\geqslant \Pr[E \text{ wins}|\bar{Q}]\Pr[\bar{Q}] \\ &= \frac{1}{2}\Pr[\bar{Q}] \\ &= \frac{1}{2} - \frac{1}{2}\Pr[Q].\end{aligned}$$

It follows that $\Pr[Q] \geqslant 2|\Pr[E \text{ wins}] - 1/2| = 2\epsilon$. Combining all the above results, we have that S solves the BDH problem with probability at least $2\epsilon/(q_S^2 q_H)$ (which is non-negligible in the security parameter l), contradicting to the hardness of the BDH problem.

5 Comparison with existing escrowable protocols

Here we summarize the security properties and performances of our E-IBAK protocol and several other previously published escrowable protocols in Table 1. Note that:

• Since all listed protocols offer the basic security properties (i.e., known-key secrecy, unknown key-share resilience, key-compromise impersonation resilience and no key control), we will restrict our comparison to only the forward secrecy property.

• In practice, pre-computation is often carried out prior to the execution of the protocol for better performance. We, therefore, compare only the on-line computation complexity of these protocols.

Table 1 Comparisons of escrowable key agreement protocols (with pre-computation)

Protocols	$\mathbb{P}$	$\mathbb{M}$	$\mathbb{E}$	$\mathbb{A}$	Bandwidth	PFS	Extensible^a)
Chen–Kudla[7]	1	0	0	1	1 point	×	✓
Smart's[5]	1	0	0	0	1 point	×	✓
MB-2[14]	1	0	0	0	1 point	×	✓
Wang's[17]	1	1	0	2	1 point	✓	✓
Cheng et al.'s[19]	1	1	1	1	1 point	✓	✓
E-IBAK	1	0	1	0	1 point	✓	✓

a) This indicates that if the protocol is extensible to work in the escrowless mode.

In Table 1, ✓ and × denote that the property holds and does not hold in the protocol respectively. We also use the following symbols to explain the computation complexity of each protocol. For simplicity, we only count these computationally expensive operations:

– $\mathbb{P}$: pairing.

– $\mathbb{M}$: scalar point multiplication in $\mathbb{G}_1$.

– $\mathbb{E}$: exponentiation in $\mathbb{G}_2$.

– $\mathbb{A}$: point addition in in $\mathbb{G}_1$.

From Table 1, we observe that only Wang's protocol[17], Cheng et al.'s protocol[19] and our proposed protocol E-IBAK achieve perfect forward secrecy (PFS) in the escrow mode. Our protocol is, however, more efficient than the other two protocols especially when we take into consideration that certain computations can be performed off-line.

Finally, it is worth noting that the escrowable protocols listed in Table 1 can be extended to work in the escrowless mode, using the simple idea due to Chen and Kudla[7] by embedding a raw Diffie–Hellman protocol[35] (interested readers are referred to ref. [7] for the details). This reflects the flexibility of these key agreement protocols.

6 Conclusions

Perfect forward secrecy (PFS) is an important security property for authenticated key agreement protocols (in both escrow and escrowless modes). We presented an identity-based authenticated key agreement protocol and proved it secure in the escrow mode. We demonstrated that our proposed protocol provides perfect forward secrecy without compromising on computational efficiency. We also proved the security of our proposed protocol in a widely accepted model yet with a simpler modular proof using the modular technique due to Kudla and Paterson[24].

The authors would like to thank Liqun Chen and the anonymous reviewers for their invaluable comments and suggestions.

1 Blake-Wilson S, Menezes A. Authenticated Diffie-Hellman key agreement protocols. In: Proc of SAC 1998, LNCS vol. 1556. New York: Springer-Verlag, 1999. 339–361

2 Shamir A. Identity-based cryptosystems and signature schemes. In: Proc of CRYPTO 1984, LNCS vol. 196. New York: Springer-Verlag, 1984. 47–53

3 Boneh D, Franklin M. Identity-based encryption from the Weil pairing. In: Proc of CRTPTO 2001, LNCS vol. 2139. New York: Springer-Verlag, 2001. 213–229

4 Boyd C, Choo K -K R. Security of two-party identity-based key agreement. In: Proc of MYCRYPT 2005, LNCS vol. 3715. New York: Springer-Verlag, 2005. 229–243

5 Smart N P. An identity based authenticated key agreement protocol based on the Weil pairing. Electron Lett, 2002, 38(13): 630–632

6 Shim K. Efficient ID-based authenticated key agreement protocol based on the Weil pairing. Electron Lett, 2003, 39(8): 653–654

7 Chen L, Kudla C. Identity based key agreement protocols from pairings. In: Proc of the 16th IEEE Computer Security Foundations Workshop. New York: IEEE Computer Society, 2002. 219–213 (See also Cryptology ePrint Archive, Report 2002/184.)

8 Sun H, Hsieh B. Security analysis of Shim's authenticated key agreement protocols from pairings. Cryptology ePrint Archive, Report 2003/113, 2003. Available at http://eprint.iacr.org/2003/113.

9 Ryu E K, Yoon E J, Yoo Y Y. An efficient ID-based authenticated key agreement protocol from pairings. In: Proc of NETWORKING 2004, LNCS vol. 3042. New York: Springer-Verlag, 2004. 1458–1463

10 Wang S, Cao Z, Bao H. Security of an efficient ID-based authenticated key agreement protocol from pairings. In: Proc of ISPA'05 Workshops, LNCS vol. 3759. New York: Springer-Verlag, 2005. 342–349

11 Wang S, Cao Z, Choo K -K R, et al. An improved identity-based key agreement protocol and its security proof. Inf Sci, 2009, 179(3): 307–318

12 McCullagh N, Barreto P S L M. A new two-party identity-based authenticated key agreement. In: Proc of CT-RSA 2005, LNCS vol. 3376. New York: Springer-Verlag, 2005. 262–274

13 Xie G. Cryptanalysis of Noel McCullagh and Paulo S. L. M.Barreto's two-party identity-based key agreement. Cryptology ePrint Archive, Report 2004/308, 2004. Available at

http://eprint.iacr.org/2004/308.

14 McCullagh N, Barreto P S L M. A new two-party identity-based authenticated key agreement. Cryptology ePrint Archive, Report 2004/122, 2004. Available at http://eprint.iacr.org/2004/122. (Updated paper of [11].)

15 Xie G. An ID-based key agreement scheme from pairing. Cryptology ePrint Archive, Report 2005/093, 2005. Available at http://eprint.iacr.org/2005/093

16 Li S, Yuan Q, Li J. Towards security two-part authenticated key agreement protocols. Cryptology ePrint Archive, Report 2005/300, 2005. Available at http://eprint.iacr.org/2005/300

17 Wang Y. Efficient identity-based and authenticated key agreement protocol. Cryptology ePrint Archive, Report 2005/108, 2005. Available at http://eprint.iacr.org/2005/108

18 Yuan Q, Li S. A new efficient ID-based authenticated key agreement protocol. Cryptology ePrint Archive, Report 2005/309, 2005. Available at http://eprint.iacr.org/2005/309

19 Cheng Z, Chen L, Comley R, Tang Q. Identity-based key agreement with unilateral identity privacy using pairings. In: Proc of ISPEC 2006, LNCS vol. 3903. New York: Springer-Verlag, 2006. 202–213

20 Choo K -K R, Boyd C, Hitchcock Y. Errors in computational complexity proofs for protocols. In: Proc of ASIACRYPT 2005, LNCS vol. 3788. New York: Springer-Verlag, 2005. 624–643

21 Canetti R, Krawczyk H. Analysis of key-exchange protocols and their use for building secure channels. In: Proc of EUROCRYPT'01, LNCS vol. 2045. New York: Springer-Verlag, 2001. 453–474

22 Choo K-K R. Key Establishment: proofs and refutations. Ph.D. Thesis. Brisbane: Queensland University of Technology. (Available at http://adt.library.qut.edu.au/adt-qut/public/adt-QUT20060928.114022.)

23 Kudla C. Special signature schemes and key agreement protocols. PhD Thesis, Royal Holloway University of London, 2006

24 Kudla C, Paterson K G. Modular security proofs for key agreement protocols. In: Proc of ASIACRYPT'05, LNCS vol. 3788. New York: Springer-Verlag, 2005. 549–565

25 Okamoto T, Pointcheval D. The Gap-problems: a new class of problems for the security of cryptographic schemes. In: Proc of PKC 2001, LNCS vol. 1992. New York: Springer-Verlag, 2002. 104–118

26 Bellare M, Rogaway P. Entity authentication and key distribution. In: Proc of CRYPTO 1993, LNCS vol. 773. New York: Springer-Verlag, 1993. 110–125

27 Barreto P S L M, Kim K Y, Lynn B. Efficient algorithms for pairing-based cryptosystems. In: Proc CRYPTO 2002, LNCS vol. 2442. New York: Springer-Verlag, 2002. 354–368

28 Galbraith S D, Harrison K, Soldera D. Implementing the Tate pairing. In: Proc of ANTS-V, LNCS vol. 2369. New York: Springer-Verlag, 2002. 324–337

29 Blake-Wilson S, Johnson C, Menezes A. Key agreement protocols and their security analysis. In: Proc of the sixth IMA International Conference on Cryptography and Coding, LNCS vol. 1355. New York: Springer-Verlag, 1997. 30–45

30 Choo K -K R, Boyd C, Hitchcock Y, et al. On session identifiers in provably secure protocols: The Bellare-Rogaway three-party key distribution protocol revisited. In: Proc of SCN 2004, LNCS vol. 3352. New York: Springer-Verlag, 2005. 351–366

31 Cheng Z, Nistazakis M, Comley R, et al. On the indistinguishability-based security model of key agreement protocols—simple cases. In: Proc of ACNS 2004 (technical track). (The full paper available on Cryptology ePrint Archive, Report 2005/129)

32 Krawczyk H. HMQV: A high performance secure Diffie-Hellman protocol. In: Proc of Crypto 2005, LNCS vol. 3621. New York: Springer-Verlag, 2005. 546–566

33 Sakai R, Ohgishi K, Kasahara M. Cryptosystems based on pairing. In: Proc of the 2000 Symposium on Cryptography and Information Security. Okinawa, Japan, 2000

34 Cheng Z, Chen L. On security proof of McCullagh-Barreto's key agreement protocol and its variants. Int J Secur Netw, 2007, 2(3/4): 251–259

35 Diffie W, Hellman M E. New directions in cryptography. IEEE Trans Inf Theory, 1976, 22(6): 644–654