Provably Secure Identity-Based Authenticated Key Agreement Protocols Without Random Oracles

Shengbao Wang[1], Zhenfu Cao[1], and Kim-Kwang Raymond Choo[2]

[1] Department of Computer Science and Engineering,
Shanghai Jiao Tong University,
800 Dongchuan Road, Shanghai 200240, China
{shengbao-wang,cao-zf}@cs.sjtu.edu.cn

[2] Independent Security Researcher,
Canberra, Australia
raymond.choo.au@gmail.com

Abstract. We present the first identity-based key agreement protocol that is provably secure without random oracles (namely in the standard model). It is inspired on a new identity-based encryption scheme first proposed by Gentry. We show how this key agreement can be used in either escrowed or escrowless mode. We also give a protocol which enables users of separate private key generators to agree on a shared secret key. All our proposed protocols have comparable performance to all known protocols that are secure in the random oracle model.

Keywords: identity-based cryptography, authenticated key agreement, bilinear pairings, standard model

1 Introduction

Key agreement protocols are of fundamental importance for communications between two parties over an insecure network. Informally, *authenticated key agreement* (AK) protocols not only allow parties to compute a *session key* known only to them but also ensure authenticity of the parties [5]. This secret session key can then be used to provide privacy and data integrity during subsequent sessions.

In 1976, Diffie and Hellman [15] proposed the first key agreement protocol in their seminal paper that also marked the birth of public-key cryptography (PKC). If in a protocol one party is assured that no other party aside from the designated party (or parties) may gain access to the particular established secret key, then the key agreement protocol is said to provide *implicit key authentication* (IKA). An authenticated key agreement protocol provides mutual IKA between (or among) parties. A key agreement protocol provides key confirmation (of B to A) if A is assured that B actually possesses the session key. An AK protocol that provides mutual key confirmation is called an authenticated key agreement with key confirmation protocol (or an AKC protocol). Key agreement protocols employ private or public-key cryptography. In this paper, we shall only consider two-party key agreement protocols in the public-key setting.

The idea of *identity(ID)-based cryptography* was first introduced by Shamir in 1984 [26]. The basic idea behind an ID-based cryptosystem is that end users can choose an arbitrary string, for example their email addresses or other online identifiers, as their public key. This eliminates much of the overhead associated with key management [21]. In 2001, Boneh and Franklin [2] gave the first fully functional solution for ID-based encryption (IBE), which is an variant of the ElGamal [16] encryption scheme,

using the pairing on elliptic curves. Since then, abundant ID-based AK protocols with or without signatures/encryptions using pairings have also been suggested (e.g., [27,28,14,25,13]).

Motivation. The random oracle has been a popular technique in provable security before and after its formalization by Bellare and Rogaway [8]. To the best of our knowledge, all the existing secure *ID-based* authenticated key agreement protocols are proven in the random oracle model, including protocols from (to name a few) [14,21,12,29,6]. However, security in the random oracle model does not imply security in the real world. On the other hand, proofs in the model *without* random oracles (i.e. in the standard model) clearly imply that the scheme cannot be broken without violating the security assumption, thus a protocol is more reliable, if we do not use ideal functions such as random oracles. This is the main motivation of our work.

Most identity-based key agreement protocols have the inherent property of *session key escrow*, namely the private key generator (PKG) can recover all the session key agreed by its users. As noted in [21], this property is either acceptable, unacceptable, or desired depending on the circumstances. For example, key escrow is essential in situations where confidentiality as well as an audit trail is a legal requirement, for instance in confidential communication in the health care profession. However, there are also other situations, such as personal communications, where it would be advantageous to turn key escrow off [21] to ensure the users' privacy, though users must trust their PKGs with their long-term private keys.

In addition, key agreement between different networks/domains may be desirable in some situation. For example, for the encrypted VoIP to work on a global scale, there simply must be compatibility between networks, and therefore key agreement between separate networks/domains (i.e., with different PKGs) is important.

Our Contributions. Based on the IBE system of Gentry [18], we put forward a new signature-/encryption-less ID-based AK protocol.

Our contributions in this paper are:

- The first provably-secure ID-based key agreement protocol in the standard model that can be instantiated with *or* without session key escrow.
- An efficient ID-based key agreement that allows users across separate private key generators (PKGs) to establish a shared session key.

Our Design Strategy. Signature-/encryption-less key agreement protocols have numerous advantages, and namely from the efficiency point of view. They are thus well-suited for some constrained environments [20]. To our knowledge, the common strategy of constructing an signature-/encryption-less *authenticated* key agreement protocol using the idea behind the ElGamal [16] public key encryption scheme was first suggested by Matsumoto, Takashima and Imai in 1986 [23]. The authors designed several authenticated Diffie-Hellman key agreement protocols (i.e., to provide the original Diffie-Hellman protocol with key authentication), which is well-known as the MTI key agreement family. In particular, the MTI/A0 protocol is derived from the ElGamal encryption in such a way that we name it as a mutual "`Encryption and Decryption`" mechanism (refer to Appendix A for details). Similar but later proposals are the protocol of Goss [19], KEA [24] authenticated key agreement designed by NSA in 1994 (declassified in 1998) and Protocol 4 from Blake-Wilson et al. [4].

Although this strategy has already been used for many times in the research community, for the first time, we make it explicit, and show its effectiveness again by proposing several new ID-based AK protocols using pairings.

Paper Organization. The rest of this paper is structured as follows. In the next section, we give the necessary technical backgrounds. Section 3 reviews Gentry's ID-based encryption scheme. In Section 4, we present our new protocols and briefly

discuss their efficiencies. In Section 5, we prove the security of our proposed protocol without using random oracles. We draw some conclusions in Section 6. Finally, a high-level description of the common design strategy is given in Appendix A.

2 Technical Backgrounds

2.1 Security Requirements

Security attributes for AK(C) protocols have been identified in several previous work [5,22,7,14]. We briefly explain the security attributes as follows (refer to [5,22] for more detailed discussions):

- **Known-key secrecy.** Suppose an established session key between two entities is disclosed, the adversary is unable to learn other established session keys.
- **Perfect forward secrecy (PFS).** If both long-term secret keys of two entities (i.e. the protocol principals) are disclosed, the adversary is unable to derive old session keys established by that two entities.
- **PKG forward secrecy (PKG-FS).** If in an ID-based key agreement protocol, the master key known only to the PKG is disclosed, the adversary is unable to derive old session keys established by that two entities. Note this attribute implies that the PKG is not able to passively escrow any session key of its users.
- **Key-compromise impersonation (K-CI) resilience.** Assume that entities A and B are two principals. Suppose $A's$ secret key is disclosed. Obviously, an adversary who knows this secret key can impersonate A to other entities (e.g. B). However, it is desired that this disclosure does not allow the adversary to impersonate other entities (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 the two protocol principals (A and B) can predetermine any portion of the shared session key being established between them.

The main desirable *performance attributes* include low computational overhead, a minimal number of passes (the number of messages exchanged in a run of the protocol), and low communication overhead (total number of bits transmitted).

2.2 Security Model for ID-Based AK Protocols

In this subsection, we review the formal security model for ID-based authenticated key agreement protocols due to Chen, Cheng and Smart [12]. Their model is an adapted version of the model of Blake-Wilson et al. [4], which in turn is an extension of the Bellare-Rogaway model [9] to the public key setting.

The model includes a set U of participants, each one is modeled by an oracle, e.g., $\Pi^n_{I,J}$ would model a participant I carrying out a protocol session in the belief that it is communicating with another participant J for the n-th time (i.e. the n-th run of the protocol between I and J). Each participant has a long-term ID-based long-term 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, who is modeled by a probabilistic polynomial time Turing Machine and has access to all the participants'

oracles as well as the random oracles in the game. 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 1 (Matching Conversation). *An oracle $\Pi^n_{I,J}$ is said to have engaged in a matching conversation with another oracle $\Pi^{n'}_{J,I}$ if every message that one oracle sends out is subsequently delivered to another oracle, with the response to this message being returned to the first oracle as the next message on its transcript.*

The security of a protocol is defined via a two-phase game between a challenger $\mathcal{C}$ and the adversary E. In the first phase, the adversary E is allowed to issue the following queries in any order.

Send: E can send message M to $\Pi^n_{I,J}$. 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 $M = \phi$ (denotes the null message), then the oracle initiates a protocol run.

Reveal: This query asks the oracle to reveal whatever session key it currently holds. An oracle is called *revealed* if it has responded to a Reveal query.

Corrupt: This query asks a participant to reveal the long term private key. A participant is called corrupted if it has responded to a Corrupt query.

Test: At some point, E can make a Test query to some *fresh oracle* (see Definition 2 below). E receives either the session key or a random value from a particular oracle. Specifically, to answer the query the fresh oracle flips a fair coin $b \in \{0, 1\}$; if the answer is 0 it outputs the agreed session key, and if the answer is 1 it outputs a random element of $\mathbb{G}_2$.

In the second phase, E can continue making Send, Reveal and Corrupt queries to the oracles, except that E is constrained not to reveal the tested oracle or its matching oracle (if any), and E cannot corrupt the two tested participants.

Output: Finally, E output 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

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

Definition 2 (Fresh Oracle). *An oracle $\Pi^n_{I,J}$ is called fresh if it has accepted (and therefore holds a session key sk_i), it is not revealed, J has been corrupted, and there is no revealed oracle $\Pi^{n'}_{J,I}$ with which it has had a matching conversation.*

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 [12].

Definition 3 ([4]). *A protocol is a secure AK protocol if:*

1. *In the presence of the* benign adversary *(who simply relays messages between parties without modification) 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; and if for every adversary E:*
2. *If uncorrupted oracles $\Pi^n_{I,J}$ and $\Pi^{n'}_{J,I}$ have matching conversations then both oracles accept and hold the same session key;*
3. *$Advantage^E(l)$ is negligible.*

In the following, we briefly discuss the attributes that the above definitions of a secure AK achieves. Recall the security attributes which were mentioned above, and here we examine them one by one.

- *Known-key secrecy.* The property of known-key secrecy is implied by the above definitions of AK security. Since E is allowed to make Reveal queries to any oracles except for the tested oracle $\Pi^n_{I,J}$ and its matching oracle $\Pi^{n'}_{J,I}$ 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).* The definition does not imply the property of perfect forward secrecy. This is because the model does not allow the adversary to make queries of corrupted oracles and therefore does not model this type of attack.
- *Key-compromise impersonation (K-CI) resilience.* As mentioned above, the definition of fresh oracle imply the key compromise impersonation property.
- *Unknown key-share (UK-S) resilience.* The definition also imply the unknown key-share resilience property. If ID_I establishes a session key with ID_J though he believes that 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 $n' = n$). During an unknown key share attack, the user ID_K may not know this session key. Since $\Pi^n_{I,K}$ and $\Pi^{n'}_{J,I}$ are not matching 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.* The above definition of AK security 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. In the model, all participants are assumed to be honest participants. If the protocol is not attacked (i.e., can be proven secure in the model), then we can be assure that the session key established is distributed uniformly at random in the session key space. Otherwise, the adversary E must have a non-negligible ability to distinguish between the session key held by $\Pi^n_{I,J}$ and a random number.

2.3 Bilinear Pairings

In this section, we describe in a more general format the basic definition and properties of the pairing: more details can be found in [2,18].

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

Definition 4. *An admissible pairing e is a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, which satisfies the following three properties:*

1. Bilinear*: If $u, v \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}^*_p$, then $e(u^a, v^b) = e(u, v)^{ab}$;*
2. Non-degenerate*: $e(g, g) \neq 1$;*
3. Computable*: If $u, v \in \mathbb{G}_1$, one can compute $e(u, v) \in \mathbb{G}_2$ in polynomial time.*

2.4 Complexity Assumptions

The security of Gentry's IBE system [18] is based on a complexity assumption that they call *the truncated augmented bilinear Diffie-Hellman exponent assumption* (the truncated q-ABDHE). We recall the truncated decision q-ABDHE problem as follows (refer to [18] for detailed description).

Truncated Decision q-ABDHE Problem. Given a vector of $q+3$ elements

$$(g', g'^{\alpha^{q+2}}, g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^q}) \in \mathbb{G}_1^{q+3}$$

as input (here $\alpha \in \mathbb{Z}_p$), an algorithm $\mathcal{B}$ that outputs $b \in \{0,1\}$ has advantage ϵ in solving the truncated decision q-ABDHE if

$$\begin{aligned} |\Pr[\mathcal{B}(g', g'^{\alpha^{q+2}}, g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^q}, e(g^{\alpha^{q+1}}, g')] = 0 \\ -\Pr[\mathcal{B}(g', g'^{\alpha^{q+2}}, g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^q}, Z) = 0]\ | \geq \epsilon \end{aligned}$$

where the probability is over the random choice of generators g, $g' \in \mathbb{G}_1$, the random choice of $\alpha \in \mathbb{Z}_p$, the random choice of $Z \in \mathbb{G}_2$, and the random bits consumed by $\mathcal{B}$.

The security of our escrowless version of the key agreement protocol is based on the following variant of the truncated q-ABDHE problem, with two additional elements g and g^{α} being given as input. We note that this modified problem is believed to be as hard as the original one.

Modified Truncated q-ABDHE Problem. Given a vector of $q+5$ elements

$$(g', g'^{\alpha^{q+2}}, g, g^{\alpha}, t, t^{\alpha}, t^{\alpha^2}, ..., t^{\alpha^q}) \in \mathbb{G}_1^{q+5}$$

as input, an algorithm $\mathcal{B}$ that outputs $b \in \{0,1\}$ has advantage ϵ in solving the *modified* truncated decision q-ABDHE if

$$\begin{aligned} |\Pr[\mathcal{B}(g', g'^{\alpha^{q+2}}, g, g^{\alpha}, t, t^{\alpha}, t^{\alpha^2}, ..., t^{\alpha^q}, e(t^{\alpha^{q+1}}, g')] = 0 \\ -\Pr[\mathcal{B}(g', g'^{\alpha^{q+2}}, g, g^{\alpha}, t, t^{\alpha}, t^{\alpha^2}, ..., t^{\alpha^q}, Z) = 0]\ | \geq \epsilon \end{aligned}$$

where the probability is over the random choice of generators g, $g', t \in \mathbb{G}_1$, the random choice of $\alpha \in \mathbb{Z}_p$, the random choice of $Z \in \mathbb{G}_2$, and the random bits consumed by $\mathcal{B}$.

Definition 5 ((*Modified*) Truncated Decision (t, ϵ, q)-ABDHE Assumption). *We say that the truncated decision (t, ϵ, q)-ABDHE assumption (resp. the modified truncated decision (t, ϵ, q)-ABDHE assumption) holds in $\mathbb{G}_1$ if no t-time algorithm has advantage at least ϵ in solving the truncated decision q-ABDHE problem (resp. the modified truncated decision q-ABDHE problem) in $\mathbb{G}_1$.*

3 Review of Gentry's IBE Scheme

In this section, we review the first construction of Gentry from [18], which is a chosen-plaintext secure ElGamal-type IBE scheme (proved in the standard model).

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups of prime order p, and let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be the bilinear pairing. The IBE system works as follows.

Setup: The PKG chooses two random generators $g, h \in \mathbb{G}_1$ and a random $\alpha \in \mathbb{Z}_p$, calculates $g_1 = g^\alpha \in \mathbb{G}_1$. It sets the public *params* as $\langle g, g_1, h\rangle$ and the *master-key* as α.
Key Generation: To generate a private key for identity $ID \in \mathbb{Z}_p$, the PKG generates a random $r_{ID} \in \mathbb{Z}_p$, and outputs the private key as $d_{ID} = \langle r_{ID}, h_{ID}\rangle$, where $h_{ID} = (hg^{-r_{ID}})^{1/(\alpha - ID)}$. (The PKG ensures that $ID \neq \alpha$ and it always assigns identical r_{ID} for a given identity ID.)
Encryption: The sender picks randomly a $s \in \mathbb{Z}_p$, using the receiver's identity ID, sets the ciphertext to be (to encrypt message $m \in \mathbb{G}_2$)

$$C = (g_1^s g^{-s \cdot ID}, e(g,g)^s, m \cdot e(g,h)^{-s}).$$

Decryption: To decrypt ciphertext $C = (u, v, w)$, the decrypter of the identity ID computes

$$m = w \cdot e(u,\ h_{ID}) \cdot v^{r_{ID}}.$$

Consistence: The recipient can correctly decrypt C to get m since

$$\begin{aligned} & e(u,\ h_{ID}) \cdot v^{r_{ID}} \\ = {} & e(g^{s(\alpha - ID)}, h^{1/(\alpha - ID)} g^{-r_{ID}/(\alpha - ID)}) \cdot e(g,g)^{s r_{ID}} \\ = {} & e(g,h)^s. \end{aligned}$$

4 Proposed ID-Based Key Agreement Protocols

In this section, we propose three new ID-based authenticated key agreement protocols based on Gentry's IBE scheme (refer to Section 3).

4.1 An ID-Based Key Agreement Protocol with Escrow

Our first protocol (named as Protocol I) does not provide PKG forward secrecy (or master-key forward secrecy), i.e., when the maser key α of the PKG is compromised, an adversary who gets it can recover all the users' past session keys. Equivalently, this means that the PKG can *escrow* all the session keys (refer to [14,21] for more details). Clearly, for any ID-based key agreement protocol, perfect forward secrecy (PFS) is implied by the PKG forward secrecy. Whereas, our second protocol (named as Protocol II) provides the PKG forward secrecy. Both of our protocols are two-pass protocol with *mutual* implicit key authentication (IKA). We note that it is readily to extend our protocols into 3-pass AKC protocols, using the common method given in [4,14].

As with all the other ID-based AK protocols we assume the existence of a PKG that is responsible for the creation and secure distribution of users' private keys.

Protocol I consists of three stages, i.e. `Setup`, `Key Generation` and `Key Agreement`. The `Setup` and `Key Generation` stages are identical to that of Gentry's IBE scheme [18].

Suppose two principals Alice and Bob are about to agree on a session key (we denote their identity $Ident$ as A and B, respectively), we follow previous notations and hereafter, let $g_{Ident} = g_1 g^{-ID_{Ident}}$ and $g_T = e(g,g)$, where $Ident \in \{A, B\}$. We denote the participant's private key as $\langle r_{ID}, h_{ID}\rangle$. The `Key Agreement` stage is as follows.

Key Agreement. To establish a shared session key, Alice and Bob each firstly generates an ephemeral private key (say x and $y \in \mathbb{Z}_p$), and compute the corresponding ephemeral public keys $T_{11} = g_B^x$, $T_{12} = g_T^x$ and $T_{21} = g_A^y$, $T_{22} = g_T^y$. They then

Alice (A)		**Bob** (B)
$x \in_R \mathbb{Z}_p$ $T_{11} = g_B^x$, $T_{12} = g_T^x$		$y \in_R \mathbb{Z}_p$ $T_{21} = g_A^y$, $T_{22} = g_T^y$
	$\xrightarrow{T_1 = T_{11}\|\|T_{12}}$	
	$\xleftarrow{T_2 = T_{21}\|\|T_{22}}$	
$K_{AB} = [e(T_{21}, h_A) \cdot (T_{22})^{r_A}] \cdot e(g,h)^x$		$K_{BA} = [e(T_{11}, h_B) \cdot (T_{12})^{r_B}] \cdot e(g,h)^y$
$sk_A = H_2(A\|\|B\|\|T_1\|\|T_2\|\|K_{AB})$		$sk_B = H_2(A\|\|B\|\|T_1\|\|T_2\|\|K_{BA})$

Fig. 1. Protocol I

exchange $T_1 = T_{11}||T_{12}$ and $T_2 = T_{21}||T_{22}$ as described in Figure 1 (where the symbol $''||''$ denotes concatenation).

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

1. Alice computes the shared secret K_{AB} as follows:

$$K_{AB} = [e(T_{21},\ h_A) \cdot (T_{22})^{r_A}] \cdot e(g,h)^x.$$

2. Bob computes the shared secret K_{BA} as follows:

$$K_{BA} = [e(T_{11},\ h_B) \cdot (T_{12})^{r_B}] \cdot e(g,h)^y.$$

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

$$\begin{aligned}
K_{AB} &= e(T_{21},\ h_A) \cdot (T_{22})^{r_A} \cdot e(g,h)^x \\
&= e(g_A^y, (g^{-r_A}h)^{1/(\alpha - ID_A)}) \cdot (g_T^y)^{r_A} \cdot e(g,h)^x \\
&= e(g^{y(\alpha - ID_A)}, (g^{-r_A}h)^{1/(\alpha - ID_A)}) \cdot (g_T^y)^{r_A} \cdot e(g,h)^x \\
&= e(g^y, g^{-r_A}h) \cdot (g_T^y)^{r_A} \cdot e(g,h)^x \\
&= e(g^y, g^{-r_A}h) \cdot e(g,g)^{yr_A} \cdot e(g,h)^x \\
&= e(g^y, g^{-r_A}h) \cdot e(g^y, g^{r_A}) \cdot e(g,h)^x \\
&= e(g^y, g^{-r_A}hg^{r_A}) \cdot e(g,h)^x \\
&= e(g^y, h) \cdot e(g,h)^x \\
&= e(g,h)^{x+y}.
\end{aligned}$$

Analogously, we can get $K_{BA} = e(g,h)^{x+y}$. 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 $sk = H_2(A||B||T_1||T_2||K)$, where $H_2 : \{0,1\}^* \rightarrow \{0,1\}^k$ is a *key derivation function* (in which $k = |sk|$). Note here we include the transcript (T_1 and T_2) in the key derivation function to resist the potential *key replicating attack* (whereby an adversary is somehow able to manipulate the shared secret K using his own contributions while he still does not know the value of K) [11].

Efficiency. Protocol I is *role symmetric*, which means that each party performing the same operations. Protocol I has comparable computational efficiency to those protocols from [14,21] (which are only proven secure in the random oracle model). In Protocol I, each participant has to generate a random number, perform one exponentiations in $\mathbb{G}_1$, two exponentiations in $\mathbb{G}_2$, and compute two pairings (which are the most expensive operations in the protocol and one of these pairings can be precomputed). We leave out multiplication in $\mathbb{G}_1$ and pairing multiplication in $\mathbb{G}_2$, and hashing as they are fast to compute compared to the other principle operations.

Compared with the Chen-Kudla protocol [14] and the McCullagh-Barreto protocol [21], Protocol I is less efficient on the message bandwidth since two message blocks (as opposed to only one) need to be distributed by each user.

Escrow. The escrow property derives from the PKG's ability to recover the shared session key, since with the knowledge of all the two users' private keys, the PKG is also able to calculate $e(g,h)^x$ and $e(g,h)^y$ using the publicly transmitted data.

4.2 An ID-Based Key Agreement Protocol Without Escrow

In this subsection, we show how to turn off the session key escrow property of the above protocol by making a slight modification to the `Setup` and `Key Generation` algorithms to Gentry's IBE system. Similar to the idea used in [14], we calculate an extra *Diffie-Hellman* shared key from the two participants' *ephemeral* contributions. However, unlike the protocols in [14], Protocol II does not bring additional communication cost. We now introduce Protocol II, which has PKG forward secrecy (or, master-key forward secrecy).

Setup: Compared with the original `Setup` algorithm, the new public parameter has one more generator (denoted as t) of group $\mathbb{G}_1$. Now the PKG chooses three random generators $g, h, t \in \mathbb{G}_1$ and a random $\alpha \in \mathbb{Z}_p$, calculates $g_1 = g^\alpha \in \mathbb{G}_1$. It sets the public *params* as $< g, g_1, h, t >$ and the *master-key* as α.

Key Generation: To generate a private key for identity $ID \in \mathbb{Z}_p$, the PKG generates a random $r_{ID} \in \mathbb{Z}_p$, and outputs the private key as $d_{ID} = \langle r_{ID}, h_{ID} \rangle$, where $h_{ID} = (ht^{-r_{ID}})^{1/(\alpha - ID)}$. (Again, the PKG ensures that $ID \neq \alpha$ and it always assigns identical r_{ID} for a given identity ID.)

Suppose that Alice and Bob are about to agree on a session key (we denote their identity *Ident* as A and B, respectively), we let $g_{Ident} = g_1 g^{-ID_{Ident}}$ and $t_T = e(g,t) \in \mathbb{G}_2$[1]. The `Key Agreement` stage is as follows.

Key Agreement. To establish a shared session key, Alice and Bob each firstly generates an ephemeral private key (say x and $y \in \mathbb{Z}_p$), and compute the corresponding ephemeral public keys $T_{11} = g_B^x$, $T_{12} = t_T^x$ and $T_{21} = g_A^y$, $T_{22} = t_T^y$. They then exchange $T_1 = T_{11}||T_{12}$ and $T_2 = T_{21}||T_{22}$ and compute the session key as described in Figure 2, with extra operations (in contrast to Protocol I) underlined.

Protocol Correctness: Both Alice and Bob will compute the same session key since we have the following equations:

[1] Note that in Protocol I, the corresponding component is $g_T = e(g,g)$ instead.

Alice (A)		**Bob** (B)
$x \in_R \mathbb{Z}_p$ $T_{11} = g_B^x$, $T_{12} = t_T^x$		$y \in_R \mathbb{Z}_p$ $T_{21} = g_A^y$, $T_{22} = t_T^y$
	$\xrightarrow{T_1 = T_{11}\|\|T_{12}}$	
	$\xleftarrow{T_2 = T_{21}\|\|T_{22}}$	
$K_{AB_1} = [e(T_{21}, h_A) \cdot (T_{22})^{r_A}] \cdot e(g,h)^x$		$K_{BA_1} = [e(T_{11}, h_B) \cdot (T_{12})^{r_B}] \cdot e(g,h)^y$
$\underline{K_{AB_2} = T_{22}^x = t_T^{xy}}$		$\underline{K_{BA_2} = T_{12}^y = t_T^{xy}}$
$sk_A = H_2(A\|\|B\|\|T_1\|\|T_2\|\|K_{AB_1}\|\|\underline{K_{AB_2}})$		$sk_B = H_2(A\|\|B\|\|T_1\|\|T_2\|\|K_{BA_1}\|\|\underline{K_{BA_2}})$

Fig. 2. Protocol II

$$
\begin{aligned}
K_{AB_1} &= e(T_{21},\ h_A) \cdot (T_{22})^{r_A} \cdot e(g,h)^x \\
&= e(g_A^y, (ht^{-r_A})^{1/(\alpha - ID_A)}) \cdot (t_T^y)^{r_A} \cdot e(g,h)^x \\
&= e(g^{y(\alpha - ID_A)}, (ht^{-r_A})^{1/(\alpha - ID_A)}) \cdot (t_T^y)^{r_A} \cdot e(g,h)^x \\
&= e(g^y, ht^{-r_A}) \cdot (g_T^y)^{r_A} \cdot e(g,h)^x \\
&= e(g^y, ht^{-r_A}) \cdot e(g,t)^{yr_A} \cdot e(g,h)^x \\
&= e(g^y, ht^{-r_A}) \cdot e(g^y, t^{r_A}) \cdot e(g,h)^x \\
&= e(g^y, ht^{-r_A} t^{r_A}) \cdot e(g,h)^x \\
&= e(g^y, h) \cdot e(g,h)^x \\
&= e(g,h)^{x+y} \\
&= K_{BA_1},
\end{aligned}
$$

$$
\begin{aligned}
K_{AB_2} &= T_{22}^x \\
&= e(g,t)^{xy} \\
&= K_{BA_2}.
\end{aligned}
$$

Efficiency. Protocol II is also role symmetric. Obviously, compared with Protocol I, Protocol II only requires one more exponentiations in $\mathbb{G}_2$ for each participant.

Protocol II has exactly the same communication efficiency as Protocol I.

Escrowless. Notice that with α, the PKG is able to calculate g^x (resp. g^y) from $T_{11} = g^{x(\alpha - ID_B)}$ (resp. $T_{21} = g^{y(\alpha - ID_A)}$). Protocol II avoids session key escrow since the PKG is still *not* able to compute $t_T^{xy} = e(g,t)^{xy}$. We note that calculating the keying term t_T^{xy} from t_T^x and t_T^y involves solving the Computational Diffie-Hellman Problem (CDHP) over the group $\mathbb{G}_2$.

4.3 Key Agreement Between Users of Separate PKGs

We now look at ID-based key agreement protocols between users of separate PKGs. The first such protocol was suggested by Chen and Kudla in [14], and [21] also gives

an efficient construction. Based on our new protocol given above, we propose another ID-based key agreement protocol across separate PKGs. Again, we note that this protocol can be instantiated in escrowed or escrowless mode.

For key agreement to be feasible between users of distinct PKGs, it is required that the PKGs use the globally agreed domain parameters. Since elliptic curves, suitable group generator points (i.e., g and h) and other cryptographic tools such as hash functions, have been standardised for security applications, for example in the NIST FIPS standards, it is therefore reasonable to assume the availability of standard pairing-friendly curves as well [21]. Once these group generator points and curves have been agreed upon, each PKG can generate their own random master secret key (i.e. α).

Suppose that Alice is a user of the private key generator PKG_A (with a a master secret α_A), and Bob is a user of the private key generator PKG_B (with a master secret α_B). Therefore, Alice's private key is generated by PKG_A with α_A and Bob's private key is generated by PKG_B with α_B. Assume that Alice (resp. Bob) has obtained an authentic copy of the public key of PKG_B (resp. PKG_A). Alice and Bob now perform the authenticated key agreement depicted in Figure 3.

Alice (A)		**Bob** (B)
$x \in_R \mathbb{Z}_p$ $T_{11} = (g^{\alpha_B} g^{-ID_B})^x$, $T_{12} = g_T^x$		$y \in_R \mathbb{Z}_p$ $T_{21} = (g^{\alpha_A} g^{-ID_A})^y$, $T_{22} = g_T^y$
	$\xrightarrow{T_1 = T_{11} \| T_{12}}$	
	$\xleftarrow{T_2 = T_{21} \| T_{22}}$	
$K_{AB} = [e(T_{21}, h_A) \cdot (T_{22})^{r_A}] \cdot e(g,h)^x$		$K_{BA} = [e(T_{11}, h_B) \cdot (T_{12})^{r_B}] \cdot e(g,h)^y$
$sk_A = H_2(A\|\|B\|\|T_1\|\|T_2\|\|K_{AB})$		$sk_B = H_2(A\|\|B\|\|T_1\|\|T_2\|\|K_{BA})$

Fig. 3. Protocol III

On Escrow. Correctness of Protocol III can be easily checked. In this protocol, neither PKG_A nor PKG_B can escrow the established session keys. But if the two PKGs work together (or collude) they still can passively escrow all their users' session keys via $K = [e(T_{21}, h_A) \cdot (T_{22})^{r_A}] \cdot [e(T_{11}, h_B) \cdot (T_{12})^{r_B}]$, since the two users' private keys $\langle r_A, h_A \rangle$ and $\langle r_B, h_B \rangle$ are known to their PKGs.

Note that it is straightforward to turn off the session key escrow property of Protocol III, simply in a parallel way to the construction of Protocol II from Protocol I.

Efficiency. Protocol III has identical computational and communication efficiencies with Protocol I.

5 Security Proof

Since the above three protocols have almost the same structures, it is sufficient to only consider the security of the basic protocol, Protocol I. We leave the proof of the security of Protocol II and Protocol III to the reader. We note that Protocol II can

be proven secure using the *modified* truncated decision q-ABDHE assumption (see Section 2.4).

With the description of the security model in Section 2.2, we now state:

Theorem 1. *If the truncated decision q-ABDHE assumption holds for the pair of groups $\mathbb{G}_1$ and $\mathbb{G}_2$, then Protocol I is a secure AK protocol.*

Proof. Our proof owes a lot to the security proof of Gentry's first IBE scheme from [18].

Condition 1 follows from the assumption that the two oracles follow the protocol and E is benign. In this case, both oracles accept (since they both receive correctly formatted messages from the other oracle) holding the same key sk (since $K_{AB} = K_{BA}$ by the bilinearity of the pairing and the matching conversation).

Condition 2 follows from the fact that if the two oracles are uncorrupted, then they cannot be impersonated, and if they have had matching conversations then each has received properly formatted messages from the other. So they will both accept holding the same session key sk. In the following, we show that the Condition 3 is also satisfied.

For a contradiction, assume that the adversary E has non-negligible advantage ϵ in guessing the value of b in time t after the Test query. We show how to construct a simulator S that uses E as an oracle to solves the truncated decision q-ABDHE problem with non-negligible advantage ϵ'. Given input of the two groups $\mathbb{G}_1$, $\mathbb{G}_2$, the bilinear map e, and a random truncated decision q-ABDHE challenge $(g', g'^{\alpha^{q+2}}, g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^q}, Z)$, $S's$ task is to distinguish whether Z is $e(g^{\alpha^{q+1}}, g')$ or a random element of $\mathbb{G}_2$.

The algorithm S works by interacting with E as follows:

Setup: This stage is identical to that of [18]. S generates a random polynomial $f(x) \in \mathbb{Z}_p[x]$ of degree q. It sets $h = g^{f(\alpha)}$, computing h from $(g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^q})$. It sets the public key as $(g, g_1 = g^{\alpha}, h)$ and sends it to E. Clearly, this public key has a distribution identical to that in the actual construction. Note that the master key is α, which is unknown to S.

Corrupt queries: S simulates the Corrupt query on input ID_i as follows. If $g^{ID_i} = g_1$, then we have $ID_i = \alpha$, S can uses α to solve the truncated decision q-ABDHE problem immediately. Else, let $F_{ID}(x)$ denote the $(q-1)$-degree polynomial $(f(x) - f(ID))/(x - ID)$. S sets the private key $\langle r_{ID_i}, h_{ID_i} \rangle$ to be $(f(ID_i), g^{F_{ID_i}(\alpha)})$. This is a valid private key for ID_i, since $g^{F_{ID_i}(\alpha)} = g^{(f(\alpha)-f(ID_i))/(\alpha - ID_i)} = (hg^{-f(ID_i)})^{1/(\alpha - ID_i)}$, as required. In order to keep the secrecy of the polynomial $f(x)$, we require that the total number of the Corrupt query is less than q (the degree of $f(x)$). Since $f(x)$ is a uniformly random polynomial, this private key appears to E to be correctly distributed.

Send queries: Assume S guesses that the oracle $\Pi^s_{J,I}$ is to be asked the Test query by the adversary E. Without loss of generality, we let $\Pi^s_{J,I}$ be the responder oracle. S answers all Send queries as specified for a normal oracle, i.e., for the first Send query to an oracle, S takes a random value in $\mathbb{Z}_p$ to form its contribution, except that if E asks a Send query to the matching oracle of $\Pi^s_{J,I}$, namely $\Pi^n_{I,J}$ for some n. Let $f_2(x) = x^{q+2}$ and $F_{2,ID_J}(x) = (f_2(x) - f_2(ID_J))/(x - ID_J)$, which is a polynomial of degree $q+1$. S will compute T_1 and T_2 for $\Pi^n_{I,J}$ as follows : (Note that T_1 and T_2 are intended for its matching oracle, $\Pi^s_{J,I}$.)

$$T_1 = g'^{f_2(\alpha) - f_2(ID_J)},$$

$$T_2 = Z \cdot e(g', \prod_{l=0}^{q} (g^{\alpha^l})^{F_{2,ID_J,l}}),$$

where $F_{2,ID_J,l}$ is the coefficient of x^l in $F_{2,ID_J}(x)$.

S picks randomly a $y \in \mathbb{Z}_p$ and computes the session key K_{JI} for $\Pi^s_{J,I}$ as follows:

$$K_{JI} = [e(T_1, h_J) \cdot T_2^{r_J}] \cdot e(g,h)^y.$$

Let $\lambda = (\log_g g')F_{2,ID_j}(\alpha)$. If $Z = e(g^{\alpha^{q+1}}, g')$, then $T_1 = g^{\lambda(\alpha - ID_J)}$, $T_2 = e(g,g)^\lambda$, and $e(T_1, h_J) \cdot T_2^{r_J} = e(g,h)^\lambda$. Thus $K_{JI} = e(g,h)^{\lambda+y}$ is a valid session key for participants J and I under randomness of λ and y. Since $\log_g g'$ is uniformly random, λ is uniformly random, and so K_{JI} is a valid, appropriately-distributed challenge for E.

Reveal queries: If the query is ask to the oracle $\Pi^n_{I,J}$ or its matching oracle $\Pi^s_{J,I}$, S aborts with failure. Otherwise, it gives the session key hold by the oracle to E.

Test query: At some point in the simulation, E will ask a Test query of some oracle. If E does not choose the guessed oracles $\Pi^s_{J,I}$ (or its matching oracle, $\Pi^n_{I,J}$) to ask the Test query, then S aborts with failure. However if E does pick $\Pi^s_{J,I}$ (or $\Pi^n_{I,J}$) for the Test query, S gives K_{JI} to E.

Guess: After the Test query, the algorithm E outputs its guess $b' \in \{0,1\}$.

We claim that if S does not abort during the simulation then E's view is identical to its view in the real attack. Furthermore, if S does not abort, then $|\Pr[b' = b] - 1/2| > \epsilon$, where the probability is over all random coins used by S and E.

Solving the truncated decision q-ABDHE problem: S simply forward the output of E to its challenger of the truncated decision q-ABDHE problem.

If $Z = e(g^{\alpha^{q+1}}, g')$, then the simulation is perfect, and E will guess the bit b correctly with probability $\epsilon + 1/2$. Else, Z is uniformly random, and thus T_1 and T_2 are uniformly random and independent element of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Hence the value of K_{JI} is uniformly random and independent from E's view. In this case, E has no advantage in guessing the bit b, i.e., E will guess the bit b correctly with probability $1/2$.

We are now ready to calculate the S's advantage ϵ' in solving the truncated decision q-ABDHE problem:

$$\begin{aligned} &|\Pr[\mathcal{B}(g', g'^{\alpha^{q+2}}, g, g^\alpha, g^{\alpha^2}, ..., g^{\alpha^q}, e(g^{\alpha^{q+1}}, g')] = 0 \\ -\Pr[\mathcal{B}(g', g'^{\alpha^{q+2}}, g, g^\alpha, g^{\alpha^2}, ..., g^{\alpha^q}, Z) = 0] &|\geq 1/2 + \epsilon - 1/2 = \epsilon. \end{aligned}$$

The probability that S does not abort equals to the probability of its right guess of the tested oracle, which is as least $1/q$. So, we have $\epsilon' = \epsilon/q$.

The time-complexity of S is identical to that of the proof in [18]. In the simulation, S's operation is dominated by computing $g^{F_{ID}(\alpha)}$ in response to E's Corrupt query on ID, where $F_{ID}(x)$ is a polynomial of degree $q-1$. Each such computation requires $O(q)$ exponentiations in $\mathbb{G}_1$. The time-complexity of S is therefore $t + O(t_{exp} \cdot q^2)$. □

6 Conclusion

We presented a new identity-based authenticated key agreement protocol inspired on Gentry's IBE system. We showed that the new protocol can be instantiated in either escrowed or escrowless mode, and also give a protocol which is suitable for circumstances where there are separate private key generators. Finally, we proved that the security of the proposed protocol (in its basic form) is relative to the decision truncated Augmented Bilinear Diffie-Hellman Exponent (the decision truncated q-ABDHE) problem without using random oracles.

Acknowledgments

We would like to thank the anonymous reviewers of Pairing'07 for their helpful comments and suggestions that improved this work. The first author would also like to thank Caroline Belrose for her valuable suggestions on designing secure ID-based AK protocols without random oracles. 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 No. 60673079.

References

1. S.S. Al-Riyami and K.G. Paterson. Certificateless public key cryptography. In *Proc. of ASIACRYPT 2003*, LNCS vol. 2894, pp. 452-473. Springer-Verlag, 2003.
2. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Proc. of CRYPTO 2001*, LNCS vol. 2139, pp. 213-229. Springer-Verlag, 2001.
3. C. Boyd and R. Choo. Security of two-party identity-based key agreement. In *Proc. of Mycrypt 2005*, LNCS vol. 3715, pp.229-243. Springer-Verlag, 2005.
4. S. Blake-Wilson, D. Johnson, and A. Menezes. Key agreement protocols and their security analysis. In *Proc. of 6th IMA International Conference on Cryptography and Coding*, LNCS vol. 1355, pp. 30-45. Springer-Verlag, 1997.
5. S. Blake-Wilson and A. Menezes. Authenticated Diffie-Hellman key agreement protocols. In *Proc. of SAC 1998*, LNCS vol. 1556, pp. 339-361. Springer-Verlag, 1999.
6. C. Boyd, W. Mao, and K. Paterson. Key agreement using statically keyed authenticators. In *Proc. of ACNS 2004*, LNCS vol. 3089, pp. 248-262. Springer-Verlag,, 2004
7. C. Boyd and A. Mathuria. Protocols for Authentication and Key Establishment. Springer-Verlag, June 2003.
8. M. Bellare and P. Rogaway, Random oracles are practical: a paradigm for designing effiient protocols, In *Proc. of First ACM Conference on Computer and Communications Security*, pp.62-73, ACM press, 1993.
9. M. Bellare and P. Rogaway. Entity authentication and key distribution. In *Proc. of CRYPTO 1993*, LNCS vol. 773, pp. 110-125. Springer-Verlag, 1994.
10. M. Burmester. On the risk of opening distributed keys. In *Proc. of CRYPTO 1994*, LNCS vol. 839, pp. 308-317. Springer-Verlag, 1994.
11. K.-K. R. Choo, C. Boyd, and Y. Hitchcock. On session key construction in provably secure protocols. In *Proc. of MYCRYPT 2005*, LNCS vol. 3715, pp. 116-131. Springer-Verlag, 2005.
12. L. Chen, Z. Cheng, and N. P. Smart. Identity-based key agreement protocols from pairings. Cryptology ePrint Archive, Report 2006/199.
13. Y. J. Choie, E. Jeong and E. Lee. Efficient identity-based authenticated key agreement protocol from pairings. *Journal of Applied Mathematics and Computation*, 162(1), pp. 179-188, 2005.
14. L. Chen and C. Kudla. Identity based key agreement protocols from pairings. In *Proc. of the 16^{th} IEEE Computer Security Foundations Workshop*, pp. 219-213. IEEE Computer Society, 2002.
15. W. Diffie, M.E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6), pp.644 - 654, 1976.
16. T. ElGamal. A public key cryptosystem and signature scheme based on discrete logarithms. *IEEE Trans. Info. Theory*, 31(4), pp. 469-472, 1985.
17. C. Gentry. Certificate-based encryption and the certificate-revocations problem. In *Proc. of EUROCRYPT 2003*, LNCS vol. 2656, pp. 272- 291. Springer-Verlag, 2003.
18. C. Gentry. Practical identity-based encryption without random oracles. In *Proc. of EUROCRYPT 2006*, LNCS vol. 4004, pp. 445-464. Springer-Verlag, 2006.
19. K. C. Goss. Cryptographic method and apparatus for public key exchange with authentication. US Patent 4,956,863, September 1990.
20. S. Kunz-Jacques and David Pointcheval. About the Security of MTI/C0 and MQV. In *Proc. of SCN 2006*. LNCS vol. 4116, pp. 156-172. Springer-Verlag, 2006.

21. N. McCullagh and P.S.L.M. Barreto. A new two-party identity-based authenticated key agreement. In *Proc. of CT-RSA 2005*, LNCS vol. 3376, pp. 262-274. Springer-Verlag, 2005.
22. A. Menezes, P. van Oorschot and S. Vanstone. Handbook of Applied Cryptography, pp. 237-238. CRC Press, 1997.
23. T. Matsumoto, Y. Takashima and H. Imai. On seeking smart public-key distribution systems. *Trans. IECE of Japan*, E69, pp.99-106, 1986.
24. NIST, SKIPJACK and KEA Algorithm Specification, http://csrc.nist.gov/encryption /skipjack/skipjack.pdf, 1998.
25. E.K. Ryu, E.J. Yoon, and K.Y. Yoo. An efficient ID-based authenticated key agreement protocol from pairings. In *Proc. of NETWORKING 2004*, LNCS vol. 3042, pp. 1458-1463. Springer-Verlag, 2004.
26. A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. of CRYPTO 1984*, LNCS vol. 196, pp. 47-53. Springer-Verlag, 1984.
27. N. Smart. An ID-based authenticated key agreement protocol based on the Weil pairing. *Electron. Lett.,* 38(13), pp. 630-632, 2002.
28. K. Shim. Efficient ID-based authenticated key agreement protocol based on Weil pairing. *Electron. Lett.*, 39(8), pp. 653-654, 2003.
29. Y. Wang. Efficient identity-based and authenticated key agreement protocol. Cryptology ePrint Archive, Report 2005/108.
30. G. Xie. An ID-based key agreement scheme from pairing. Cryptology ePrint Archive, Report 2005/093.
31. S. Wang, Z. Cao. Escrow-free certificate-based authenticated key agreement protocol from pairings. *Wuhan University Journal of Natural Sciences*, vol. 12(1), pp. 063-066, 2007.

A A Design Strategy for ID-Based AK Protocols

Here we describe in detail a design strategy for ID-based authenticated key agreement protocols, using any ElGamal-type ID-based encryption system.

To illustrate the ideas behind the above mentioned strategy, we first recall the so-called ElGamal one-pass *unilateral* key agreement protocol that was first given in Chap. 12 of [22]. We quote it as follows.

> *ElGamal key agreement is a Diffie-Hellman variant providing a one-pass protocol with unilateral key authentication (of the intended recipient to the originator), provided the public key of the recipient is known to the originator a priori. While related to ElGamal encryption, the protocol is more simply Diffie-Hellman key agreement wherein the public exponential of the recipient is fixed and has verifiable authenticity (e.g., is embedded in a certificate).*

Informally, the protocol proceeds as follows. The sender A forms a *shared* secret using her random input r_A in combination with B's long-term public key y_B by computing $K = y_B^{r_A}$. On receipt of the ephemeral public key $T_A = r_A P$, the receiver B is able to reconstruct the session key $K = T_A^{x_B}$.

The two-pass MTI/A0 protocol can be seen as a parallel execution of ElGamal one-pass key agreement protocol. It yields session keys with *mutual* (implicit) key authentication against passive attacks. As in ElGamal one-pass key agreement, A sends to B a single message, resulting in the shared key K. B independently initiates an analogous protocol with A, resulting in the shared key K'. A and B then output the KK' as their agreed session key.

Note that although the original MTI/A0 protocol is of certain security weaknesses, e.g., it doesn't provide perfect forward secrecy and is vulnerable to some active attacks such as unknown key-share attacks [7], triangle attacks [10], the elegant idea behind it is very useful in Diffie-Hellman authenticated key agreement protocol design.

The above idea can be applied to the design of pairing-based AK protocols. Inspired by the above design strategy, Al-Riyami and Paterson [1] gave the first certificateless authenticated key agreement protocol based on their certificateless public key encryption (CL-PKE) scheme (also appears in [1]) in 2003. Similarly, Wang and Cao [31] proposed the first certificate-based authenticated key agreement protocol in 2004, using the ideas from the certificate-based encryption (CBE) scheme of Gentry [17].

So far, we are ready to define a general framework, named as *the general MTI/A0 protocol* (GMP), for the design of ID-based AK protocols based on an ElGamal-type IBE scheme.

Definition 6. General MTI/A0 Protocol (GMP). *Suppose we have an ElGamal-type IBE scheme and two users (Alice and Bob) want to agree on a shared session key. They does the following:*

1. Alice (Bob) firstly generates her (his) ephemeral private key, then computes her (his) ephemeral public key PK_{eph} using the public parameters of the system, finally she (he) sends PK_{eph} to Bob (Alice).
2. Alice (Bob) uses the ephemeral private key generated in Step 1 to compute *the ElGamal encryption session key* K_{En}.
3. After receiving the ephemeral public key, they each calculate *the ElGamal decryption session key* K_{De}, using their own long-term private keys.
4. Alice (Bob) computes her (his) final session key sk as follows. (Where the symbol "$*$" stands for a commutative operation, e.g., multiplication, addition, or bitwise XOR.)
$$sk = K_{En} * K_{De}.$$

Remark 1. The two users are able to sucessfully establish a shared session key after the above GMP.