Attribute-Based Authenticated Key Exchange*

M. Choudary Gorantla, Colin Boyd, and Juan Manuel González Nieto

Information Security Institute, Faculty of IT, Queensland University of Technology
GPO Box 2434, Brisbane, QLD 4001, Australia
mc.gorantla@gmail.com, {c.boyd,j.gonzaleznieto}@qut.edu.au

Abstract. We introduce the concept of attribute-based authenticated key exchange (AB-AKE) within the framework of ciphertext-policy attribute-based systems. A notion of AKE-security for AB-AKE is presented based on the security models for group key exchange protocols and also taking into account the security requirements generally considered in the ciphertext-policy attribute-based setting. We also introduce a new primitive called encapsulation policy attribute-based key encapsulation mechanism (EP-AB-KEM) and then define a notion of chosen ciphertext security for EP-AB-KEMs. A generic one-round AB-AKE protocol that satisfies our AKE-security notion is then presented. The protocol is generically constructed from any EP-AB-KEM that achieves chosen ciphertext security. Finally, we propose an EP-AB-KEM from an existing attribute-based encryption scheme and show that it achieves chosen ciphertext security in the generic group and random oracle models. Instantiating our AB-AKE protocol with this EP-AB-KEM will result in a concrete one-round AB-AKE protocol also secure in the generic group and random oracle models.

Keywords: Attribute-based Key Exchange, Attribute-based KEM, Group Key Exchange.

1 Introduction

In a distributed collaborative system, it is often convenient for the members to communicate with the others in the system using attributes that describe their roles or responsibilities. These attributes are highly desirable if the members join/leave the system dynamically. Consider an Internet forum where the members are organized into user groups based on the members' skills or privileges. It is a natural requirement that the members of a user group should be able to establish secure communication with the other members belonging to particular user groups. The communication in these forums is generally carried out through initiating a thread or by posting messages within an existing thread. To enable authentic and confidential communication, the forum administrator may specify an access policy with the user groups being attributes. Obviously, only the

* This work has been supported in part by the Australian Research Council through Discovery Project DP0666065.

R. Steinfeld and P. Hawkes (Eds.): ACISP 2010, LNCS 6168, pp. 300–317, 2010.

members of the forum whose attributes (e.g. membership to user groups) satisfy the policy should be able to have read and/or write access to the thread.

In the above scenario, the members do not necessarily have to know the identity of the other members with whom they want to communicate. In fact, the administrator may be requested not to disclose the identity of a member to the others for privacy reasons. Any member whose attributes satisfy the policy specified by the administrator should be able to participate in the communication. Note that the communication can naturally be among a group of more than two members, since the defined policy may be satisfied by attributes of more than two members. Hence, an authenticated group key exchange protocol that facilitates attributes usage can be employed in this setting. We call such a protocol, an attribute-based authenticated key exchange (AB-AKE) protocol. Once a session key among the willing participants has been established via the key exchange protocol, it can be used for establishing secure communication among the participants.

We can further envisage applications for AB-AKE in interactive chat rooms and also in organizations with strict hierarchy like the military. In interactive chat rooms, each room may be associated with a policy defined with a set of interests being the attributes. Any member whose interests satisfy the policy of a chat room can have read and/or write access to it. Similarly, a policy over ranks (e.g., Sergeant, Lieutenant, Major, Colonel etc.) as attributes can be specified for the units in the military by another unit at a higher level in the hierarchy. All the units whose attributes satisfy the policy can establish secure communication among themselves through an AB-AKE protocol.

Attribute-based Encryption. Sahai and Waters [26] introduced the concept of attribute based encryption (ABE) as an extension to ID-based encryption [6], in which a set of descriptive attributes is regarded as an identity. Goyal et al. [19] further extended ABE and introduced two variants: key policy attribute based encryption (KP-ABE) and ciphertext policy attribute based encryption (CP-ABE). In a KP-ABE system, the private key of a party is associated with an access policy defined over a set of attributes while the ciphertext is associated with a set of attributes. A ciphertext can be decrypted by a party if the attributes associated with the ciphertext satisfy the policy associated the user's private key. A CP-ABE system can be seen as a complementary form to KP-ABE system, wherein the private key is associated with a set of attributes, while a policy defined over a set of attributes is attached to the ciphertext. A ciphertext can be decrypted by a party if the attributes associated with its private key satisfy the ciphertext's policy.

1.1 Contributions

In this paper, we introduce the concept of AB-AKE. We assume that each member willing to participate in an AB-AKE protocol is issued a private key for a set of attributes that he/she possesses. Our modelling of AB-AKE follows the framework of CP-ABE in that the attributes are associated with the private keys. We assume that the members are given an access policy which their attributes

have to satisfy for them to participate in the protocol. Alternatively, a common policy may be negotiated by the group members themselves. The protocol takes the access policy as input and computes messages for the other parties. Similar to CP-ABE systems, we may assume that the policy is attached to the protocol messages in an AB-AKE protocol, although this assumption is not necessary since each member knows the policy at the outset of the protocol. A member whose attributes satisfy the given policy can compute the session key from the incoming messages and (if exists) its own contribution.

While a complementary flavour of AB-AKE can be conceptualized based on KP-ABE systems, we do not explore this direction in this work. For the type of applications that we have discussed earlier, AB-AKE protocols based on CP-ABE systems suit well. AB-AKE can be seen as an extension of group key exchange (GKE) [9,23,22] with the additional expressiveness provided by the ciphertext-policy attribute-based systems. We define a notion of authenticated key exchange security (AKE-security) for AB-AKE by adapting a corresponding notion for GKE to the attribute-based setting. The property of collusion resistance considered by attribute-based systems [19,3,28] is naturally embedded into our AKE-security notion.

We then propose a generic one-round AB-AKE protocol that satisfies our AKE-security notion. The protocol is based on a type of attribute-based key encapsulation mechanism (KEM) that we call *encapsulation-policy attribute based KEM* (EP-AB-KEM). In an EP-AB-KEM, the attributes are associated to the private key of a party and access policy is attached to the encapsulation. We define a notion of chosen ciphertext security for EP-AB-KEM based on a corresponding notion considered for CP-ABE schemes.

Our AB-AKE protocol is generic in the sense that it can be instantiated using any EP-AB-KEM that satisfies chosen ciphertext security. We propose a chosen-ciphertext secure EP-AB-KEM based on the CP-ABE scheme of Bethencourt et al. [3] and using the generic technique of Boneh et al. [7]. While we apply the technique of Boneh et al. to the chosen plaintext secure EP-AB-KEM implicit in Bethencourt et al.'s scheme, we also make some non-trivial changes to adapt it to the attribute-based setting. The proposed EP-AB-KEM is then proven secure in the generic group and random oracle models. Incidentally, we are the first to model and construct EP-AB-KEMs, which are of independent interest.

An AB-AKE protocol satisfying our AKE-security provides implicit authentication that is similar to the corresponding notion considered for normal key exchange protocols. Particularly, our AKE-security notion ensures each protocol participant that no other party apart from parties who satisfy the given policy can possibly learn the value of the session key. Note that an EP-AB-KEM cannot achieve this property since it does not provide any sender authentication. Consequently, the receivers in EP-AB-KEM whose attributes satisfy the policy have no way of knowing whether the sender actually satisfies the same policy or not. For example, if we use an EP-AB-KEM in a user group, any one can post a message that is encrypted with the symmetric of the EP-AB-KEM. Alternatively, if the message is encrypted with a session key derived from an AB-AKE protocol

the readers will get the assurance that only someone with valid attribute set has posted the message.

Our generic construction of AB-AKE can be seen as an extension of the protocols of Boyd et al. [8] and Gorantla et al. [17] to the attribute-based setting. One disadvantage of our protocol is that it cannot provide forward secrecy. However, for some of the applications that we have discussed forward secrecy may not be necessary. For example, in an Internet forum the administrator may like to moderate the content posted in the user groups or in the military a unit at a higher rank would like to monitor the communication among the units at the same or a lower rank. In such scenarios, an AB-AKE protocol without forward secrecy will be useful since any party with the right attribute set will be able to recover the session key and consequently the messages encrypted with it. Nevertheless, forward secrecy is generally a highly desirable property for key exchange protocols. Hence, we also sketch constructions of AB-AKE protocols that can achieve forward secrecy.

1.2 Related Work

The concept of fuzzy secret handshake proposed by Ateniese et al. [1] seems closely related to our modelling of AB-AKE. However, there are a few important differences: In AB-AKE, we allow policies specified by the members to be very expressive consisting of several threshold gates, while fuzzy secret handshake only considers a single threshold gate. In a (fuzzy) secret handshake protocol, if a member do not satisfy the attributes specified by another member, the attributes of none of the members can be learned by the other member. On the other hand, in an AB-AKE protocol, if a member does not satisfy the policy specified by the other members, the members do not know anything about the attributes of the other members except what can be inferred by the policies attached to the protocol messages. Although both the properties look similar, we emphasize that an AB-AKE protocol would not hide the affiliation of the members even if the protocol was not successful [20]. Note that the property of "affiliation hiding" is the main requirement for (fuzzy) secret handshakes. Finally, the fuzzy secret handshake protocol of Ateniese et al. considers only two party setting, while our protocol naturally operates in a group setting.

In independent work, Steinwandt and Corona [27] proposed a two-round attribute-based group key exchange protocol that achieves forward secrecy. Their protocol uses the GKE protocol of Bohli et al. [5] as the base protocol and replaces the public key signature scheme in Bohli et al. with an attribute-based signcryption scheme to authenticate the protocol messages. Recently, Birkett and Stebila [4] introduced the concept of predicate-based key exchange which encompasses key policy attribute-based key exchange. However, their security model considers key exchange between only two parties.

1.3 Organization

Section 2 presents a security model for EP-AB-KEM and also proposes a chosen ciphertext secure EP-AB-KEM. We define a security model for AB-AKE in

Section 3 and present a generic one-round AB-AKE protocol in Section 4. In Section 5, we outline how to construct AB-AKE protocols with forward secrecy.

2 Encapsulation Policy Attribute-Based KEM

We first give a formal definition of security for EP-AB-KEM. As in the earlier attribute-based systems [19,3], we review the definition of an access structure and use it in the security model. Later, we present a concrete EP-AB-KEM based on the CP-ABE scheme of Bethencourt et al. [3].

Definition 1 (Access Structure [2])**.** Let $\{U_1, \cdots, U_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{U_1, \cdots, U_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{U_1, \cdots, U_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{U_1, \cdots, U_n\}} \setminus \{\phi\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

In our EP-AB-KEM and later in the protocol, each party is assumed to possess a set of attributes. A policy over a set of attributes is specified through an access structure $\mathbb{A}$. Hence, $\mathbb{A}$ contains the authorized sets of attributes i.e., $\mathbb{A} \subseteq 2^{\{S_1, \cdots, S_n\}} \setminus \{\phi\}$ for a given set of attributes $\{S_1, \cdots, S_n\}$. As in the CP-ABE of Bethencourt et al., we consider only monotonic access structures. In the rest of the paper, by an access structure we mean a monotonic one.

A EP-AB-KEM consists of five polynomial-time algorithms:

Setup: takes the security parameter k and the attribute universe description $\mathbb{U}$ as inputs. The public parameters PK and the master key MK are the outputs.

Encapsulation: takes as input the public parameters PK and an access structure $\mathbb{A}$ over the attribute universe $\mathbb{U}$. It outputs an encapsulation C and a symmetric key K such that only a user with attributes satisfying $\mathbb{A}$ can recover K from C. Similar to the CP-ABE schemes, we assume that the encapsulation implicitly contains $\mathbb{A}$.

KeyGen: takes as input the master key MK, the public parameters PK and a set of attributes S of a user that gives a description of the user's private key. The output is the user's private key SK.

Decapsulation: takes as input the public parameters PK, an encapsulation C and a private key SK corresponding to a set of attributes S. The algorithm outputs either a symmetric key K or $\perp$.

We also define an optional delegation algorithm, which allows a user with attribute sets S and a corresponding secret key SK to derive a secret key for another set of attributes $\tilde{S}$ such that $\tilde{S} \subseteq S$.

Delegate: takes as input the public parameters PK, a secret key SK corresponding to a set of attributes S and a set $\tilde{S} \subseteq S$. It outputs a secret key $\tilde{SK}$ for the attribute set $\tilde{S}$.

For an EP-AB-KEM to be considered valid, it is required that for any key SK corresponding to an attribute set S, if S satisfies $\mathbb{A}$ and if $(K, C) \leftarrow \mathsf{Encapsulation}(PK, \mathbb{A})$, then $\mathsf{Decapsulation}(PK, C, SK) = K$.

2.1 Security Model

Bethencourt et al. [3] defined the notion of indistinguishability under chosen plaintext attack (IND-CPA) for CP-ABE schemes. In this section, we adapt their notion and extend it to define a notion of indistinguishability under chosen ciphertext attacks (IND-CCA) for EP-AB-KEM. The security notion is formally defined as follows.

Definition 2. An EP-AB-KEM is IND-CCA secure if the advantage of any probabilistic polynomial time adversary $\mathcal{A}^{\mathsf{cca}}$ in the following game is negligible in the security parameter k.

Setup: The challenger runs the Setup algorithm and returns PK to $\mathcal{A}^{\mathsf{cca}}$.

Phase 1: $\mathcal{A}^{\mathsf{cca}}$ issues Extract and Decap queries as follows:

- Extract: This query can be issued multiple times with sets of attributes $S_1, \cdots, S_{q_1}$ as input. The challenger returns a private key corresponding to each input attribute set. We do not require the input attribute sets to be distinct.
- Decap: This query is issued with an encapsulation C and an attribute set S as inputs. Note that C implicitly contains an access structure $\mathbb{A}$ defined over the attribute universe $\mathbb{U}$. The challenger executes the Decapsulation algorithm on C using a private key corresponding to S and returns the output of Decapsulation to $\mathcal{A}^{\mathsf{cca}}$.

Challenge: At the end of **Phase 1**, $\mathcal{A}^{\mathsf{cca}}$ gives an access structure $\mathbb{A}^*$ defined over $\mathbb{U}$ to the challenger. The challenger first chooses a bit b. It then runs the Encapsulation algorithm with $\mathbb{A}^*$ as input and generates a symmetric key–encapsulation pair (K_1, C^*). It then sets K_0 to be a random key drawn from the probability distribution of the symmetric key. The tuple (K_b, C^*) is returned to $\mathcal{A}^{\mathsf{cca}}$ as the challenge. A trivial restriction on the adversary's choice of $\mathbb{A}^*$ is that none of the attributes sets $S_1, \cdots, S_{q_1}$ passed as input to Extract queries in **Phase 1** should satisfy $\mathbb{A}^*$.

Phase 2: $\mathcal{A}^{\mathsf{cca}}$ is allowed to execute in the same way as in **Phase 1** with the following restrictions: (1) none of the attribute sets $S_{q_1+1}, \cdots, S_q$ passed as input to Extract queries in **Phase 2** satisfy $\mathbb{A}^*$ and (2) a Decap query with C^* as input in combination with an attribute set S^* that satisfies $\mathbb{A}^*$ is not allowed.

Guess: The goal of $\mathcal{A}^{\mathsf{cca}}$ is to guess whether the key K_b is encapsulated within C^* or not. $\mathcal{A}^{\mathsf{cca}}$ finally outputs a guess bit b'. It wins the game if $b' = b$. The advantage of $\mathcal{A}^{\mathsf{cca}}$ is given as $Adv_{\mathcal{A}^{\mathsf{cca}}} = |2 \cdot \Pr[b' = b] - 1|$.

Existing security notions for CP-ABE schemes also consider the weaker *selective model* where $\mathcal{A}^{\mathsf{cca}}$ declares the challenge access structure $\mathbb{A}^*$ before the **Setup** phase. Similarly, a corresponding model for EP-AB-KEMs can be defined.

Similar to earlier CP-ABE schemes [3,11,28], we have not explicitly modelled the delegation mechanism in the security model for EP-AB-KEMs. However, we require that for a given set of attributes, a secret key output by the Delegate algorithm will have identical distribution to the one output by the KeyGen algorithm. In particular, the Decapsulation algorithm using a private key SK should work in the same way irrespective of SK being an output of KeyGen or Delegate. Our security model for EP-AB-KEMs suffices in the presence of an adversary who may obtain delegated private keys since such queries can be simulated using Extract queries.

Remark 1. In Definition 2, $\mathcal{A}^{\mathsf{cca}}$ is allowed to issue multiple Extract queries with attribute sets as input such that none of the individual sets S_i satisfy the challenge access structure $\mathbb{A}^*$. Hence, similar to earlier definitions of attribute-based encryption schemes, our definition also takes care of collusion resistance. An EP-AB-KEM satisfying the above definition ensures that from the private keys of S_i's, $\mathcal{A}^{\mathsf{cca}}$ cannot construct a private key corresponding to another attribute set S^* such that S^* satisfies $\mathbb{A}^*$.

Hybrid CP-ABE. An EP-AB-KEM satisfying the above IND-CCA security notion can be combined with any IND-CCA secure data encapsulation mechanism to construct an IND-CCA secure CP-ABE scheme [12,13]. We describe the hybrid construction and prove its security in the full version of this paper [16].

2.2 A Chosen Ciphertext Secure EP-AB-KEM

Bethencourt et al. [3] first proposed a construction of a CP-ABE scheme. Their scheme was shown IND-CPA secure assuming generic group and random oracle models. Later, many CP-ABE schemes [18,11,28] have been proposed and shown IND-CPA secure without assuming generic group or random oracle models, but analyzed only in the selective model of security. Recently, Lewko et al. [24] proposed a fully secure CP-ABE scheme in the standard model using composite order bilinear groups.

We now construct an IND-CCA secure EP-AB-KEM based on the CP-ABE scheme of Bethencourt et al. The idea is to enhance the security of the IND-CPA secure EP-AB-KEM that is implicit in Bethencourt et al.'s CP-ABE scheme. For this purpose, the techniques of Fujisaki and Okamoto [15,14] and Canetti et al. (CHK) [10] can be applied in the random oracle and standard models respectively. As remarked by Bethencourt et al., IND-CCA security for CP-ABE (and correspondingly for EP-AB-KEM) schemes can be achieved by a straightforward application of the Fujisaki-Okamoto technique.

Bethencourt et al. also suggested that the delegation mechanism of their CP-ABE scheme can be leveraged to achieve IND-CCA security using the CHK transform. However, we observe that applying the CHK transform to CP-ABE schemes (similarly to EP-AB-KEMs) is slightly more involved. Specifically, contrary to the approach followed by KP-ABE schemes, IND-CCA security for CP-ABE schemes cannot be achieved by directly leveraging the delegation mechanism. We later discuss why this is so and then present an IND-CCA secure

EP-AB-KEM by making a few changes to the Setup and Encapsulation algorithms derived from Bethencourt et al.'s CP-ABE scheme. Although the CHK technique can be used to achieve IND-CCA security in the standard model, our EP-AB-KEM will only be secure assuming generic groups and random oracles since the base CP-ABE scheme also assumes the same. Finally, we choose the scheme of Bethencourt et al. because it is secure in the fully adaptive model (i.e., non-selective model). Alternatively, one could derive an EP-AB-KEM secure in the fully adaptive model from the CP-ABE scheme of Lewko et al. [24]. In the full version [16], we discuss the necessity of an EP-AB-KEM to be secure in the adaptive model for constructing AB-AKE protocols.

The IND-CCA secure scheme first generates a one-time key pair (sk, vk) for a signature scheme with the condition that the verification key is of the same length as the length of an attribute in the attribute universe $\mathbb{U}$. Let $\mathbb{A}$ be the access structure given as input to the EP-AB-KEM. We now construct a more restrictive access structure $\mathbb{A}' = \mathbb{A}$ AND vk and execute the CPA-secure EP-AB-KEM under $\mathbb{A}'$. The resulting encapsulation is then signed using the one-time signing key sk. The encapsulation of the CCA-secure EP-AB-KEM contains the encapsulation generated by the underlying CPA-secure EP-AB-KEM, the signature generated on it and the verification key vk. The recipient first checks the signature using vk and then executes the CPA-secure KEM's decapsulation algorithm under $\mathbb{A}'$ to extract the symmetric key.

While the above informal description of our construction directly follows the CHK technique, the tricky part in the context of EP-AB-KEM (or CP-ABE) is to empower the recipient with a private key corresponding to the attributes that satisfy the modified access structure $\mathbb{A}'$. The recipient may already possess attributes that satisfy $\mathbb{A}$. However, since the verification key vk is one-time and chosen randomly for each execution of EP-AB-KEM, the recipient cannot be issued with a private key that can decrypt messages encrypted under $\mathbb{A}' = \mathbb{A}$ AND vk. This problem cannot be addressed by the delegation mechanism in an EP-AB-KEM (or CP-ABE) scheme since it can be used to derive private key corresponding to an attribute set S' from the one corresponding to S only if $S' \subseteq S$. But, we have an additional attribute in the form of vk. Note that this is not a problem in the KP-ABE system since it naturally allows a party with a private key corresponding to an access structure $\mathbb{A}$ to derive private keys corresponding to access structures that are more restrictive than $\mathbb{A}$.

To address the above problem, we make modifications to the Setup and Encapsulation algorithms derived from the CP-ABE scheme of Bethencourt et al. [3]. Our EP-AB-KEM now enables a recipient with private key for attributes that satisfy $\mathbb{A}$ to decapsulate an encapsulation created under $\mathbb{A}'$, irrespective of the choice of vk by the sender. As in the CP-ABE scheme of Bethencourt et al., an access structure $\mathbb{A}$ is represented in the form of an access tree $\mathcal{T}$.

Access Tree. Let $\mathcal{T}$ be a tree representing an access structure. Each interior node of $\mathcal{T}$ represents a threshold gate, while each leaf node is described by an attribute. Let num_x be the number of children of a node x and let k_x be its threshold value. We have $0 \leq k_x \leq num_x$. A threshold gate associated to an

internal node with threshold value k_x outputs true if at least k_x of its children output true. If the threshold gate represented by an interior node is an AND gate then $k_x = num_x$ and if the gate is OR, $k_x = 1$. The threshold value for each leaf node x is defined to be $k_x = 1$. The parent of a node x in the tree $\mathcal{T}$ is denoted by the function $\mathsf{parent}(x)$, while the attribute of a leaf node x is denoted by $\mathsf{att}(x)$. The children of each interior node are numbered from 1 to num_x. The function $\mathsf{index}(x)$ returns such a number associated with a node x. We assume that the index values are uniquely assigned in an arbitrary manner for a given access structure.

Satisfying an access tree. Let r be the root of an access tree $\mathcal{T}$. The subtree of $\mathcal{T}$ rooted at a node x is denoted by $\mathcal{T}_x$. If a set of attributes γ satisfy the access tree $\mathcal{T}_x$, it is denoted as $\mathcal{T}_x(\gamma) = 1$. The function $\mathcal{T}_x(\gamma)$ is computed recursively as follows: If x is an interior node, for each children x' of x, $\mathcal{T}_{x'}(\gamma)$ is evaluated. $\mathcal{T}_x(\gamma)$ returns 1 if and only if at least k_x children of x return 1. If x is a leaf node, $\mathcal{T}_x(\gamma)$ returns 1 if and only if $\mathsf{att}(x) \in \gamma$.

Let $\mathbb{G}_0$ and $\mathbb{G}_1$ be two multiplicative groups of prime order p and g be an arbitrary generator of $\mathbb{G}_0$. Let $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$ be an admissible bilinear map. The Lagrange's coefficient $\Delta_{i,S}$ for $i \in \mathbb{Z}_p$ and a set S of elements in $\mathbb{Z}_p$ is defined as: $\Delta_{i,S} = \Pi_{j \in S, j \neq i} \frac{x-j}{i-j}$.

$\mathsf{Setup}(k)$. It chooses the groups $\mathbb{G}_0$, $\mathbb{G}_1$ and defines a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$. It also selects $\alpha, \beta_1, \beta_2 \in \mathbb{Z}_p$ such that $\beta_1 \neq \beta_2$, $\beta_1 \neq 0$ and $\beta_2 \neq 0$. The public key is

$$PK = \left(\mathbb{G}_0, \mathbb{G}_1, e, g, h_1 = g^{\beta_1}, f_1 = g^{1/\beta_1}, h_2 = g^{\beta_2}, f_2 = g^{1/\beta_2}, e(g,g)^{\alpha}\right).$$

The master key MK is $(\beta_1, \beta_2, g^{\alpha})$.

$\mathsf{Encapsulation}(PK, \mathcal{T})$. This algorithm generates an encapsulation and a symmetric key under the access tree $\mathcal{T}$ using the public key PK. It first executes the KeyGen algorithm of the signature scheme and obtains a one-time key pair (sk, vk). Let $\mathbb{A}$ be the access structure represented by $\mathcal{T}$. The algorithm now constructs a new access tree $\mathcal{T}'$ for the access structure ($\mathbb{A}$ AND vk) as follows: Let R be the root node of $\mathcal{T}$. The root node R' of the new tree $\mathcal{T}'$ is set as the AND gate with $\mathcal{T}$ as its subtree and the verification key vk as a leaf node attached to R'.

The algorithm now generates a polynomial q_x for each node x in the tree $\mathcal{T}'$ in a top-down approach as follows: Starting from the root node R', for each node x in the tree set the degree d_x of the polynomial associated with x to be $k_x - 1$ i.e., the degree of the polynomial is one less than the threshold value associated with the node x. The algorithm starts from the root node and first chooses a random $s \in \mathbb{Z}_p$. Then it chooses $d_{R'}$ other points randomly to define the polynomial $q(R')$. For any node x other than the root, it sets $q_x(0) = q_{\mathsf{parent}(x)}(\mathsf{index}(x))$ and chooses d_x other points randomly to define the polynomial $q(x)$.

Let Y be the set of leaf nodes in the subtree $\mathcal{T}$ rooted at R. The only other leaf node in the tree $\mathcal{T}'$ is the one that describes the verification key vk. The algorithm proceeds as follows:

1. $K = e(g,g)^{\alpha s}$.
2. $C_1 = h_1^s$.
3. $\forall y \in Y: \;\; C_y = g^{q_y(0)}, C'_y = H(\mathsf{att}(y))^{q_y(0)}$.
4. $C_{vk} = h_2^{q_{vk}(0)}$, $C'_{vk} = H(vk)^{q_{vk}(0)}$.
5. Let $\mathcal{C} = (\mathcal{T}', C_1, C_y, C'_y, C_{vk}, C'_{vk})$, $\forall y \in Y$. Compute a signature $\sigma = \mathsf{Sig}_{sk}(\mathcal{C})$.

The final encapsulation $C = (\mathcal{C}, vk, \sigma)$.

KeyGen(MK,PK,S). The key generation algorithm takes as input the master key MK and a set of attributes S and outputs a private key corresponding to S. It chooses $r, r_{vk} \in \mathbb{Z}_p$ and $r_j \in \mathbb{Z}_p$ for each $j \in S$. The private key is computed as:

$$SK = (D = g^{(\alpha+r)/\beta_1},\; E = g^{r/\beta_2},\; \forall j \in S: \;\; D_j = g^r \cdot H(j)^{r_j},\; D'_j = g^{r_j}).$$

Delegate($SK, PK, \tilde{S}$). It takes as input a secret key SK corresponding to a set of attributes S and another set $\tilde{S} \subseteq S$. The key SK is of the form $SK = (D, E, \forall j \in S : D_j, D'_j)$. The algorithm chooses $\tilde{r}$ and $\tilde{r}_k \forall k \in \tilde{S}$. The new key for $\tilde{S}$ is generated as:

$$\tilde{SK} = (\tilde{D} = D f_1^{\tilde{r}},\; \tilde{E} = E f_2^{\tilde{r}},\; \forall k \in \tilde{S}: \;\; \tilde{D}_k = D_k g^{\tilde{r}} H(k)^{\tilde{r}_k},\; \tilde{D}'_k = D'_k g^{\tilde{r}_k}).$$

Decapsulation(SK, PK, C). Upon receiving an encapsulation C, the decryptor first parses the access tree $\mathcal{T}'$. It then extracts the subtree $\mathcal{T}$ rooted at R from $\mathcal{T}'$. Note that this can be easily done since the node that describes the verification key as an attribute can be identified with the help of the verification key vk sent in the encapsulation. The algorithm first verifies the signature σ on C using the verification key vk. If the verification succeeds, it proceeds as follows:

$$\begin{aligned} F_{vk} = \frac{e(C_{vk}, H(vk) \cdot g^{r/\beta_2})}{e(C'_{vk}, h_2)} &= \frac{e(C_{vk}, g^{r/\beta_2}) \cdot e(C_{vk}, H(vk))}{e(C'_{vk}, h_2)} \\ &= \frac{e(h_2^{q_{vk}(0)}, g^{r/\beta_2}) \cdot e(h_2^{q_{vk}(0)}, H(vk))}{e(H(vk)^{q_{vk}(0)}, h_2)} \\ &= e(g^{\beta_2 \cdot q_{vk}(0)}, g^{r/\beta_2}) = e(g,g)^{r q_{vk}(0)}. \end{aligned} \tag{1}$$

A recursive algorithm DecryptNode($\mathcal{C}, SK, x$) that takes as input $\mathcal{C}$, a private key SK associated with a set of attributes S and a node x from the subtree $\mathcal{T}$ is then executed as below:

If x is a leaf node, then let $i = \mathsf{att}(\mathsf{x})$. If $i \notin S$, then DecryptNode($\mathcal{C}, SK, x$) = $\perp$. Otherwise it is defined as follows:

$\mathsf{DecryptNode}(\mathcal{C}, SK, x) = \frac{e(D_i, C_x)}{e(D'_i, C'_x)} = \frac{e(g^r \cdot H(i)^{r_i}, g^{q_x(0)})}{e(g^{r_i}, H(i)^{q_x(0)})} = e(g,g)^{rq_x(0)}$.
If x is an interior node then $\mathsf{DecryptNode}(\mathcal{C}, SK, x)$ proceeds as follows: For all nodes z that are children of x, the algorithm $\mathsf{DecryptNode}(\mathcal{C}, sk, z)$ is called. The output is stored as F_z. Let S_x be an arbitrary k_x-sized set of child nodes z such that $F_z \neq \perp$. If no such set exists, the function returns $\perp$. Otherwise, the decapsulation algorithm proceeds as follows:

$$\begin{aligned}
F_x &= \prod_{z \in S_x} F_z^{\Delta_{i,S'_x}(0)}, \text{ where } i = \mathsf{index}(z), S'_x = \{\mathsf{index}(z) : z \in S_x\} \\
&= \prod_{z \in S_x} (e(g,g)^{r \cdot q_z(0)})^{\Delta_{i,S'_x}(0)} \\
&= \prod_{z \in S_x} (e(g,g)^{r \cdot q_{\mathsf{parent(z)}}(\mathsf{index}(z))})^{\Delta_{i,S'_x}(0)} \\
&= \prod_{z \in S_x} (e(g,g)^{r \cdot q_x(i) \cdot \Delta_{i,S'_x}(0)} \\
&= (e(g,g)^{r \cdot q_x(0)}.
\end{aligned}$$

Finally, the decapsulation algorithm calls the $\mathsf{DecryptNode}$ algorithm on the node R, which is the root of the subtree $\mathcal{T}$. If $\mathcal{T}$ is satisfied by the attribute set S, then we have $F_R = \mathsf{DecryptNode}(\mathcal{C}, SK, R) = e(g,g)^{r \cdot q_R(0)}$. We now compute $F_{R'}$ from F_{vk} and F_R using polynomial interpolation as follows:

$$\begin{aligned}
F_{R'} &= \prod_{x \in \{R, vk\}} F_x^{\Delta_{\mathsf{index}(x), \{R, vk\}}} \\
&= e(g,g)^{r \cdot q_{R'}(0)} \\
&= e(g,g)^{rs}.
\end{aligned}$$

Let $A = e(g,g)^{rs}$. The symmetric key is recovered as

$$\frac{e(C_1, D)}{A} = \frac{e(h_1^s, g^{(\alpha+r)\beta_1})}{e(g,g)^{rs}} = \frac{e(g,g)^{s(\alpha+r)}}{e(g,g)^{rs}} = e(g,g)^{\alpha s} = K. \tag{2}$$

Note that in Equation 1, we implicitly verify that the one-time verification key has not been replaced. If vk was replaced the symmetric key computed in Equation 2 would be $\perp$. Alternatively, the verification check can be done explicitly at the cost of an additional pairing operation. In the full version [16], we show that the proposed EP-AB-KEM is IND-CCA secure in the generic group and random oracle models.

3 Attribute-Based Authenticated Key Exchange

An AB-AKE protocol consists of three polynomial-time algorithms: Setup, KeyGen and KeyExchange. The Setup and KeyGen algorithms are identical to those defined for EP-AB-KEM in Section 2. Each party in the AB-AKE protocol

executes the KeyExchange algorithm which initially takes as input PK, an access structure $\mathbb{A}$ and a private key for a set of attributes S. If S satisfies $\mathbb{A}$, KeyExchange proceeds as per specification and may generate outgoing messages and also accept incoming messages from other parties as inputs. The output of KeyExchange is either a session key κ or $\perp$.

Communication Model. Let $\mathcal{U}= \{U_1, \cdots, U_n\}$ be a set of n users. The protocol may be executed among any subset $\tilde{\mathcal{U}} \subseteq \mathcal{U}$ of size $\tilde{n} \geq 2$. We assume that each user has a set of descriptive attributes. Let SK_i be the private key corresponding to an attribute set S_i of user U_i. We assume that an access structure $\mathbb{A}$ is given as input to all the users. Note that this $\mathbb{A}$ may be specified by a higher level protocol. Alternatively, the users can run an interactive protocol to negotiate a common access structure $\mathbb{A}$. We also assume that all the users execute the protocol honestly. If a user U_i wants to establish a session key with respect to an access structure $\mathbb{A}$, it first checks whether its attribute set S_i satisfies $\mathbb{A}$ or not i.e., checks if $S_i \in \mathbb{A}$. U_i proceeds with the protocol execution only if S_i satisfies $\mathbb{A}$. Thus, any user U_j with attribute set S_j that satisfies $\mathbb{A}$ is a potential participant in the key exchange protocol. The set of parties whose individual attributes satisfy $\mathbb{A}$ can compute a common session key.

An AB-AKE protocol π executed among $\tilde{n} \leq n$ users is modelled as a collection of $\tilde{n}$ programs running at the $\tilde{n}$ parties. Each instance of π within a party is defined as a session and each party may have multiple such sessions running concurrently. Let π_i^j be the j-th run of the protocol π at party $U_i \in \tilde{\mathcal{U}}$. Each protocol instance at a party is identified by a unique session ID. We assume that the session ID is derived during the run of the protocol. The session ID of an instance π_i^j is denoted by sid_i^j. An instance π_i^j enters an *accepted* state when it computes a session key sk_i^j. Note that an instance may terminate without ever entering into an accepted state. The information of whether an instance has terminated with acceptance or without acceptance is assumed to be public.

Note that there may be more than one party whose attributes satisfy $\mathbb{A}$, hence we consider group setting for AB-AKE. We define partnership in AB-AKE protocol as follows: A set of $\tilde{n}$ instances at $\tilde{n}$ different parties $\tilde{\mathcal{U}} \subseteq \mathcal{U}$ are called partners if

1. they all have the same session ID; **and**
2. the attributes of each $U_i \in \tilde{\mathcal{U}}$ satisfy $\mathbb{A}$.

An AB-AKE protocol is called *correct* if the instances at the parties in $\tilde{\mathcal{U}}$ are partnered and output identical session keys in the presence of a passive adversary.

Adversarial Model. The communication network is assumed to be fully controlled by the adversary, which schedules and mediates the sessions among all the parties. The adversary is allowed to insert, delete or modify the protocol messages. We also assume that it is the adversary that may select the protocol participants from the set $\mathcal{U}$. While the adversary may not know the attribute

set that a user possesses, it can initiate an instance of the AB-AKE protocol with an access structure of its choice. In addition to controlling the message transmission, the adversary is allowed to ask the following queries.

- $\mathsf{Send}(\pi_i^j, m)$ sends a message m to the instance π_i^j. If the message is $\mathbb{A}$, the instance π_i^j is initiated with the access structure $\mathbb{A}$. Otherwise, the message is processed as per the protocol specification. The response of π_i^j to any Send query is returned to the adversary.
- $\mathsf{RevealKey}(\pi_i^j)$ If π_i^j has accepted, the adversary is given the session key sk_i^j established at π_i^j.
- $\mathsf{Corrupt}(S_i)$ This query returns a private key SK_i corresponding to the attribute set S_i.
- $\mathsf{Test}(\pi_i^j)$ A random bit b is secretly chosen. If $b = 1$, the adversary is given sk_i^j established at π_i^j. Otherwise, a random value chosen from the session key probability distribution is given. Note that a Test query is allowed only on an accepted instance.

Definition 3 (Freshness)**.** Let $\mathbb{A}$ be the access structure for an instance π_i^j. π_i^j is called fresh if the following conditions hold: (1) the instance π_i^j or any of its partners has not been asked a $\mathsf{RevealKey}$ query **and** (2) there has not been a $\mathsf{Corrupt}$ query on an input S_i such that S_i satisfies $\mathbb{A}$.

Definition 4 (AKE-security)**.** An adversary $\mathcal{A}_{\mathsf{ake}}$ against the AKE-security notion is allowed to make Send, $\mathsf{RevealKey}$ and $\mathsf{Corrupt}$ queries in Stage 1. $\mathcal{A}_{\mathsf{ake}}$ makes a Test query to an instance π_i^j at the end of Stage 1 and is given a challenge key K_b as described above. It can continue asking queries in Stage 2. Finally, $\mathcal{A}_{\mathsf{ake}}$ outputs a bit b' and wins the AKE-security game if (1) $b' = b$ **and** (2) the Test instance π_i^j remains fresh till the end of $\mathcal{A}_{\mathsf{ake}}$'s execution. Let $\mathsf{Succ}_{\mathcal{A}_{\mathsf{ake}}}$ be the event that $\mathcal{A}_{\mathsf{ake}}$ wins the AKE-security game. The advantage of $\mathcal{A}_{\mathsf{ake}}$ in winning this game is $\mathsf{Adv}_{\mathcal{A}_{\mathsf{ake}}} = |2 \cdot \Pr[\mathsf{Succ}_{\mathcal{A}_{\mathsf{ake}}}] - 1|$. A protocol is called AKE-secure if $\mathsf{Adv}_{\mathcal{A}_{\mathsf{ake}}}$ is negligible in the security parameter k for any polynomial time $\mathcal{A}_{\mathsf{ake}}$.

Remark 2. By allowing the adversary to reveal the private keys corresponding to attribute sets which individually do not satisfy the given access structure $\mathbb{A}^*$ in the test session, our definition naturally considers collusion resistance. In other words, any number of parties whose individual attribute sets do not satisfy $\mathbb{A}^*$ may collude among themselves and try to violate the AKE-security of the protocol. An AB-AKE protocol satisfying our AKE-security notion will still remain secure against such collusion attacks.

4 A Generic One-Round AB-AKE Protocol

We now present a simple generic AB-AKE protocol based on IND-CCA secure EP-AB-KEM. Informally, each party executes an EP-AB-KEM in parallel and combines the symmetric key it has generated with the symmetric keys extracted

Computation

Each U_i executes an EP-AB-KEM on the input $(PK, \mathcal{T})$ where PK is the public parameters and $\mathcal{T}$ is the access tree that represents an access structure $\mathbb{A}$. As a result, a symmetric key and encapsulation pair (K_i, C_i) is obtained.

$$(K_i, C_i) \leftarrow \mathsf{Encapsulation}(PK, \mathcal{T}).$$

Broadcast

Each U_i broadcasts the generated encapsulation C_i.

$$U_i \rightarrow * : \ C_i.$$

Key Computation

1. Each U_i executes the decapsulation algorithm using its private key SK_i on each of the incoming encapsulations C_j and obtains the symmetric keys K_j, for $j \neq i$.

$$K_j \leftarrow \mathsf{Decapsulation}(sk_i, C_j) \text{ for each } j \neq i.$$

2. Each U_i then computes the session ID as the concatenation of all the outgoing and incoming messages exchanged i.e. $\mathsf{sid} = (C_1 \| \cdots \| C_{\tilde{n}})$, where $\tilde{n}$ is the number of protocol participants.
3. The session key κ is then computed as

$$\kappa \ = f_{K_1}(\mathsf{sid}) \oplus f_{K_2}(\mathsf{sid}) \oplus \cdots \oplus f_{K_{\tilde{n}}}(\mathsf{sid})$$

where f is a pseudorandom function.

Fig. 1. A Generic One-round AB-AKE Protocol

from the incoming messages to establish a common session key. Our construction is an extension of the one-round protocols of Boyd et al. [8] and Gorantla et al. [17] to the attribute-based setting. Figure 1 presents our generic one-round AB-AKE protocol.

At the beginning of the protocol each party is given an access structure $\mathbb{A}$ represented via an access tree $\mathcal{T}$. The protocol uses an EP-AB-KEM scheme (Setup, Encapsulation, KeyGen, Decapsulation). Each U_i is issued a private key SK_i corresponding to the attributes set S_i that it possesses. Each party U_i who has attribute set S_i satisfying the access structure $\mathbb{A}$ runs the Encapsulation algorithm and obtains a symmetric key-encapsulation pair (K_i, C_i). The parties broadcast the encapsulations to the other parties. Upon receiving the encapsulations, each party runs the Decapsulation algorithm using its private key on each of the incoming encapsulations and extracts the symmetric keys. The number of protocol participants $\tilde{n}$ can be derived based on the number of input messages received within a prescribed time period. The session key is finally computed by each party from the symmetric key that it has generated and all the symmetric keys decapsulated from the incoming encapsulations.

A pseudorandom function f is applied to derive the session key. We assume that the output of the Decapsulation algorithm can be directly used as a seed for f. Otherwise, we will have to extract and then expand the randomness from the output of the Decapsulation algorithm as done by Boyd et al. [8].

Theorem 1. *The AB-AKE protocol in Fig. 1 is AKE-secure as per Definition 4 assuming that the underlying EP-AB-KEM is IND-CCA secure. The advantage of* $\mathcal{A}_{\mathsf{ake}}$ *is*

$$Adv_{\mathcal{A}_{\mathsf{ake}}} \leq \tilde{n} \cdot \frac{q_s^2}{|C|} + q_s \cdot (\tilde{n} \cdot Adv_{\mathcal{A}^{\mathsf{prf}}} + Adv_{\mathcal{A}^{\mathsf{cca}}})$$

where $\tilde{n}$ *is the number of parties in the protocol,* q_s *is the number of sessions* $\mathcal{A}_{\mathsf{ake}}$ *is allowed to activate,* $|C|$ *is the size of the ciphertext space,* $Adv_{\mathcal{A}^{\mathsf{cca}}}$ *is the advantage of a polynomial adversary* $\mathcal{A}^{\mathsf{cca}}$ *against the IND-CCA security of the underlying EP-AB-KEM and* $Adv_{\mathcal{A}^{\mathsf{prf}}}$ *is the advantage of a polynomial adversary* $\mathcal{A}^{\mathsf{prf}}$ *against the pseudorandomness of the pseudorandom function* f*.*

The proof of the above theorem is given in the full version [16].

Concrete Instantiation. From the EP-AB-KEM proposed in Section 2.2, a concrete AB-AKE protocol can be directly realized. It follows from the security of the EP-AB-KEM and the generic AB-AKE protocol that the instantiated protocol is AKE-secure in the generic group and the random oracle models.

5 Extensions

The security model in Section 3 is concerned only about the basic notion of AKE-security without forward secrecy. Forward secrecy is one of the most important security attributes for key exchange protocols since it limits the damage of long-term key exposure. A key exchange protocol with forward secrecy ensures that even if the long-term key of a party is exposed, all the past session keys established using that long-term key will remain uncompromised.

Forward secrecy seems to be more important for AB-AKE protocols than in the case of normal key exchange protocols. To see why, let us assume that the adversary obtains the private key of a user U_i who possesses a set of attributes S_i. If an AB-AKE protocol does not achieve forward secrecy, then the adversary can compromise all the protocol sessions which have been established with access structures that can be satisfied by S_i. Note that the party U_i does not even have to participate in any of these sessions. We now define a notion of freshness that takes forward secrecy into account.

5.1 AKE-Security with Forward Secrecy

Definition 5 (FS-Freshness)**.** Let $\mathbb{A}$ be the access structure for an instance π_i^j. π_i^j is called fs-fresh if the following the conditions hold: (1) the instance π_i^j or any its partners has not been asked a RevealKey query **and** (2) there has not been a Corrupt query on an input S_i before π_i^j or its partner instances have terminated, such that S_i satisfies $\mathbb{A}$.

Definition 5 can be coupled with the AKE-security notion in Definition 4 to arrive at AKE-security notion with forward secrecy for AB-AKE protocols.

5.2 Constructing AB-AKE Protocols with Forward Secrecy

Our one-round AB-AKE protocol can be modified to achieve AKE-security with forward secrecy for two-party and three-party settings using known techniques. For a two-party AB-AKE protocol with forward secrecy, one can use the technique of Boyd et al. [8] where ephemeral Diffie-Hellman public keys are appended with the encapsulations. Similarly, for a three-party AB-AKE protocol with forward secrecy, the protocol of Joux [21] can be executed in the same round with our EP-AB-KEM based protocol. The session keys in both the protocols will include ephemeral Diffie-Hellman key components, which ensure forward secrecy. However, the protocols will achieve *weak forward secrecy*, wherein the adversary has to remain passive during protocol execution. The security of the resulting two-party and three-party AB-AKE protocols will depend on the hardness of the computational Diffie-Hellman and bilinear Diffie-Hellman problems respectively along with the security of the underlying AB-AKE protocol (the security of the latter has been proven already).

Constructing AB-AKE protocols in the more general group setting needs more than one round. The compiler of Katz and Yung (KY) [23] turns an unauthenticated group key exchange protocol into an authenticated one. The compiler uses a public key based signature as an "authenticator" for this purpose. One may adapt the KY compiler to the attribute-based setting by replacing the normal public key based signature with an attribute-based signature [25]. The resulting compiler can then be applied to the two-round unauthenticated Burmester and Desmedt (BD) protocol [9] to achieve a three-round AB-AKE protocol with forward secrecy. Since the session key established by the BD protocol is ephemeral it achieves forward secrecy, where as the attribute-based KY compiler provides authentication. Although the attribute-based version of the KY compiler can be constructed with necessary changes to the KY compiler, it may not be straightforward. We leave this construction for future work.

6 Conclusion

We have initiated the concept of AB-AKE in the ciphertext-policy attribute-based system. Our modelling of AB-AKE assumes that each party has a set of attributes and a corresponding private key. A policy is defined (or negotiated) for each execution of the protocol and the parties satisfying the policy can establish a common shared key by executing the protocol. In the security model for AB-AKE, we have considered only outsider adversaries. Our security model can be extended by considering insider attackers who try to impersonate other protocol participants [22].

We have also introduced the concept of EP-AB-KEM. We then proposed a one-round generic AB-AKE protocol based on IND-CCA secure EP-AB-KEMs. For concrete instantiation of this protocol, we have presented an EP-AB-KEM

and shown it secure under the IND-CCA notion in the generic group and random oracle models. As a consequence, a concrete AB-AKE protocol based on this EP-AB-KEM would also be secure in the generic group and random oracle models.

References

1. Ateniese, G., Kirsch, J., Blanton, M.: Secret Handshakes with Dynamic and Fuzzy Matching. In: Proceedings of the Network and Distributed System Security Symposium–NDSS 2007. The Internet Society (2007)
2. Beimel, A.: Secure Schemes for Secret Sharing and Key Distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)
3. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334. IEEE Computer Society, Los Alamitos (2007)
4. Birkett, J., Stebila, D.: Predicate-Based Key Exchange. Cryptology ePrint Archive, Report 2010/082 (2010); To appear at ACISP 2010, `http://eprint.iacr.org/2010/082`
5. Bohli, J.M., Gonzalez Vasco, M.I., Steinwandt, R.: Secure group key establishment revisited. Int. J. Inf. Sec. 6(4), 243–254 (2007)
6. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
7. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. SIAM J. Comput. 36(5), 1301–1328 (2007)
8. Boyd, C., Cliff, Y., González Nieto, J.M., Paterson, K.G.: One-Round Key Exchange in the Standard Model. International Journal of Applied Cryptography 1(3), 181–199 (2009)
9. Burmester, M., Desmedt, Y.: A Secure and Efficient Conference Key Distribution System (Extended Abstract). In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 275–286. Springer, Heidelberg (1995)
10. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
11. Cheung, L., Newport, C.: Provably secure ciphertext policy ABE. In: CCS 2007: Proceedings of the 14th ACM conference on Computer and communications security, pp. 456–465. ACM, New York (2007)
12. Cramer, R., Shoup, V.: Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. SIAM J. Comput. 33(1), 167–226 (2004)
13. Dent, A.W.: A Designer's Guide to KEMs. In: Paterson, K.G. (ed.) Cryptography and Coding 2003. LNCS, vol. 2898, pp. 133–151. Springer, Heidelberg (2003)
14. Fujisaki, E., Okamoto, T.: How to Enhance the Security of Public-Key Encryption at Minimum Cost. In: Imai, H., Zheng, Y. (eds.) PKC 1999. LNCS, vol. 1560, pp. 53–68. Springer, Heidelberg (1999)
15. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)
16. Gorantla, M.C., Boyd, C., González Nieto, J.M.: Attribute-based Authenticated Key Exchange. Cryptology ePrint Archive, Report 2010/084 (2010), `http://eprint.iacr.org/2010/084`

17. Gorantla, M.C., Boyd, C., González Nieto, J.M., Manulis, M.: Generic One Round Group Key Exchange in the Standard Model. In: 12th International Conference on Information Security and Cryptology–ICISC 2009. Springer, Heidelberg (2009)
18. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded Ciphertext Policy Attribute Based Encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
19. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security–CCS 2006, pp. 89–98. ACM, New York (2006)
20. Jarecki, S., Liu, X.: Private Mutual Authentication and Conditional Oblivious Transfer. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 90–107. Springer, Heidelberg (2009)
21. Joux, A.: A One Round Protocol for Tripartite Diffie-Hellman. In: Bosma, W. (ed.) ANTS 2000. LNCS, vol. 1838, pp. 385–394. Springer, Heidelberg (2000)
22. Katz, J., Shin, J.S.: Modeling insider attacks on group key-exchange protocols. In: Proceedings of the 12th ACM Conference on Computer and Communications Security–CCS 2005, pp. 180–189. ACM, New York (2005)
23. Katz, J., Yung, M.: Scalable Protocols for Authenticated Group Key Exchange. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 110–125. Springer, Heidelberg (2003)
24. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. Cryptology ePrint Archive, Report 2010/100 (2010); To appear at EUROCRYPT 2010, http://eprint.iacr.org/2010/110
25. Maji, H., Prabhakaran, M., Rosulek, M.: Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. Cryptology ePrint Archive, Report 2008/328 (2008), http://eprint.iacr.org/2008/328
26. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
27. Steinwandt, R., Corona, A.S.: Attribute-based group key establishment (unpublished manuscript)
28. Waters, B.: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. Cryptology ePrint Archive, Report 2008/290 (2008), http://eprint.iacr.org/