Efficient Privacy-Preserving Ciphertext-Policy Attribute Based-Encryption and Broadcast Encryption

Zhibin Zhou, *Member*, *IEEE*, Dijiang Huang, *Senior Member*, *IEEE*, and Zhijie Wang

Abstract—Ciphertext Policy Attribute-Based Encryption (CP-ABE) enforces expressive data access policies and each policy consists of a number of attributes. Most existing CP-ABE schemes incur a very large ciphertext size, which increases linearly with respect to the number of attributes in the access policy. Recently, Herranz et al. proposed a construction of CP-ABE with constant ciphertext. However, Herranz et al. do not consider the recipients' anonymity and the access policies are exposed to potential malicious attackers. On the other hand, existing privacy preserving schemes protect the anonymity but require bulky, linearly increasing ciphertext size. In this paper, we proposed a new construction of CP-ABE, named Privacy Preserving Constant CP-ABE (denoted as PP-CP-ABE) that significantly reduces the ciphertext to a constant size with any given number of attributes. Furthermore, PP-CP-ABE leverages a hidden policy construction such that the recipients' privacy is preserved efficiently. As far as we know, PP-CP-ABE is the first construction with such properties. Furthermore, we developed a Privacy Preserving Attribute-Based Broadcast Encryption (PP-AB-BE) scheme. Compared to existing Broadcast Encryption (BE) schemes, PP-AB-BE is more flexible because a broadcasted message can be encrypted by an expressive hidden access policy, either with or without explicit specifying the receivers. Moreover, PP-AB-BE significantly reduces the storage and communication overhead to the order of $\boldsymbol{O}(\log \boldsymbol{N})$, where $\boldsymbol{N}$ is the system size. Also, we proved, using information theoretical approaches, PP-AB-BE attains minimal bound on storage overhead for each user to cover all possible subgroups in the communication system.

Index Terms—Attribute-based encryption (ABE), privacy-preserving, ciphertext-policy, constant ciphertext length, broadcast encryption

1 Introduction

Ciphertext Policy Attribute-Based Encryption (CP-ABE) has been a very active research area in recent years [2], [4]-[6]. In the construction of CP-ABE, each attribute is a descriptive string and each entity may be tagged with multiple attributes. Many entities may share common attributes, which allows message encryptors to specify a secure data access policy over the shared attributes to reach a group of receivers. A decryptor's attributes need to satisfy the access policy in order to recover the message. These unique features make CP-ABE solutions appealing in many systems, where expressive data access control is required for a large number of users.

One major problem of existing CP-ABE schemes is bulky, linearly increasing ciphertext. In the CP-ABE schemes reported in [4], [6], and [2], the size of a ciphertext proliferates linearly with respect to the number of included attributes. For example, the message size in BSW CP-ABE [4] starts at about 630 bytes, and each additional attribute adds about 250-300 bytes.

Recently, Herranz et al. [1] proposed a CP-ABE that requires constant ciphertext size. However, it does not consider the anonymity of data recipients and the data access policies are attached to the ciphertext in plaintext form. Thus, passive attackers can track a user or infer the sensitivity of ciphertext by eavesdropping the access policies. In many environments, it is also critical to protect the access policies as well as the data content. For example, the access policy "General" **AND** "Pentagon" disclose the recipient's roles or positions and implies the sensitivities of the message. On the other hand, existing privacy preserving schemes [2], [3] protect the access policies but require large, linearly increasing ciphertext size. To the best of our knowledge, there is no work that can achieve privacy-preservation and constant ciphertext size at the same time.

In this paper, we propose a novel PP-CP-ABE construction, named *Privacy Preserving Constant-size Ciphertext Policy Attribute Based Encryption* (PP-CP-ABE), which enforces hidden access policies with wildcards and incurs constant-size conjunctive headers, regardless of the number of attributes. Each conjunctive ciphertext header only requires 2 bilinear group elements, which are bounded by 100 bytes in total. The actual size of the bilinear group depends on the chosen parameters for the cryptosystem. In our implementation, we use Type–D MNT curves with element compression [7]. To support disjunctive or more flexible access policies, multiple constant-size conjunctive headers can be attached to the same ciphertext message. It should be noted that we restricted each ciphertext header to be conjunctive in order to avoid ambiguity while preserving the receivers' anonymity. Moreover, PP-CP-ABE supports non-monotonic data access control policy. To the best of our knowledge, this is the first construction that

- *Z. Zhou is with Amazon, Seattle, WA 98109. E-mail: zhibin@amazon.com.*
- *D. Huang and Z. Wang are with the School of Computing Informatics Decision Systems Engineering, Arizona State University, Tempe, AZ 85287. E-mail: {Dijiang.Huang, wangzj}@asu.edu.*

Manuscript received 22 Jan. 2013; revised 18 Aug. 2013; accepted 22 Sep. 2013. Date of publication 07 Oct. 2013; date of current version 12 Dec. 2014. Recommended for acceptance by L. Imbert.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2013.200

achieves these properties, namely: privacy-preservation and constant-size conjunctive headers with wildcards.

Based on presented PP-CP-ABE, we further provide a new construction named as Privacy Preserving *Attribute Based Broadcast Encryption* (PP-AB-BE). In existing BE schemes, e.g., [8], a sender encrypts a message for a specified set of receivers who are listening on a broadcast channel. Each receiver in the specified set can decrypt the message while all other listeners cannot decrypt it even though they collude together. However, in large scale systems, identifying every receiver, and acquiring and storing their public keys are not easy tasks. For example, to broadcast a message to all CS students in a university, the encryptor needs to query the CS department roster and acquire the public key of every student in the roster; this process could be very expensive and time consuming.

Using PP-AB-BE, an encryptor has the flexibility to encrypt the broadcasted data either with or without the exact information of intended receivers. For example, Alice can specify a hidden access policy: "CS" **AND** "Student" to restrict the broadcast message to all CS students without specifying the receivers explicitly. Accordingly, Bob, who has attributes {"EE", "Student"}, cannot decrypt the data while Carol, who has attributes {"CS", "Student"} can access the data. Moreover, Alice can also encrypt the broadcasted message to any arbitrary set of receivers such as {"Bob", "Carol"}.

PP-AB-BE also significantly reduces the storage overhead compared to many existing BE schemes, where cryptographic key materials required by encryption or decryption increase linearly or sublinearly on the number of receivers. For example, in BGW scheme [8], the public key size is $O(N)$ or $O(N^{1/2})$, where N is the number of users in the system. PP-AB-BE reduces the key storage overhead problem by optimizing the organization of the attribute hierarchy. In a system with N users, the storage overhead is $O(\log N + m)$, where m is a constant number and $m \ll N$. We also proved from the information theoretical perspective that PP-AB-BE achieves storage lower bound to satisfy all possible subgroup formations, and thus it can be applied to storage constrained systems.

The most significant nature of this research article is that we present a fundamental and unified Privacy Preserving Attribute Based solution considering constraints on both communication and storage, and our solution is provably secure. It is also worth noting that our proposed PP-CP-ABE can be used to implement an identity-based encryption with wildcards (WIBE) [9] to achieve the first constant ciphertext size WIBE construction with privacy preserving features. In summary, the main contributions of this research are presented as follows:

- *PP-CP-ABE*: We construct an efficient Privacy Preserving Constant Ciphertext Policy Attribute Based Encryption (PP-CP-ABE) scheme that enforces hidden conjunctive access policies with wildcards in constant ciphertext size. To the best of our knowledge, this is the first construction that achieves these properties.
- *PP-AB-BE*: Based on PP-CP-ABE, we present a Privacy Preserving Attribute Based Broadcast Encryption (PP-AB-BE) scheme. Compared with existing BE schemes, PP-AB-BE is flexible as it uses both descriptive and non-descriptive attributes, which enables a user to specify the decryptors based on different abstraction levels, with or without exact information of intended receivers. Moreover, PP-AB-BE demands less storage overhead compared to existing BE schemes. We proved that our construction requires minimal storage to support all the possible user group formations for BE applications.

The rest of this paper is organized as follows. We first summarize the related work in Section 2. Then, in Section 3, we present system models used in this paper. We give detailed PP-CP-ABE construction in Section 4. In Section 5, we present the construction of PP-AB-BE and the storage analysis using an information theoretical approach. In Section 6, the performance of PP-AB-BE is presented through both theoretical analysis and experimental studies. Finally, we conclude our work in Section 7.

2 Related Works

Attribute Based Encryption (ABE) was first proposed as a fuzzy version of IBE in [10]. In CP-ABE [4], [5], [2], [11]-[14], each user's private key is associated with a set of attributes and each ciphertext is encrypted by an access policy. To decrypt the message, the attributes in the user private key need to satisfy the access policy. The key difference between identity and attribute is that identities are many-to-one mapped to users while attributes are many-to-many mapped to users. Thus, to simulate a constant size conjunctive header, one needs to encrypt the message using each receiver's identity and the size of ciphertext is linearly increasing.

In [15], the authors proposed a CP-ABE scheme with constant size conjunctive headers and constant number of pairing operations. It must be noted that they did not seek to address the issues of recipient anonymity. One drawback of their scheme does not support wildcards (or do-not-care) in the conjunctive access policies. To decrypt a ciphertext, the decryptor's attributes need to be identical to the access policy. In other words, the model is still one-to-one, i.e., an access policy is satisfied by one attribute list or ID, which makes the number of access policies increase exponentially. Thus, their scheme can be simply implemented using IBE schemes with same efficiency by using each user's attribute list as his/her ID. We should note that in a system with n attributes, the number of attribute combinations is 2^n. As the result, without using wildcards, there needs 2^n access policies to express all combinations. With wildcards, one can use a single access policy to express many combinations of attributes. Herranz et al. [1] proposed a more general construction of CP-ABE with constant ciphertext independently. Their proposed scheme achieves constant ciphertext with any monotonic threshold data access policy, e.g. n-of-n (AND), 1-of-n (OR) and m-of-n. However, compared with our proposed PP-CP-ABE, their scheme does not consider recipient anonymity as one of the design goals.

To protect the privacy of the access policy, KSW scheme [2], NYO scheme [3], RC scheme [13] and YRL1 scheme [16] were proposed, where the encryptor-specified access policy is hidden. Specifically, the attribute names in both [13] and [16] are explicitly disclosed in the access policy, while only the eligible attribute values are hidden. Also, YRL2 scheme was proposed in [17] based on BSW scheme [4] as a group key management scheme providing group membership anonymity. In [18], we proposed a novel alternative to the hidden

policy to preserve privacy efficiently. The main difference between our scheme and existing hidden policy attribute-based encryption schemes is PP-CP-ABE significantly reduced the size of ciphertext to a constant size, while all existing hidden policy solutions requires ciphertext that is linearly increasing on the number of attributes in the hidden policy.

It must be noted that the construction in this paper is developed from one of our earlier construction [14], where we proposed an ABE scheme with constant size ciphertext. The major improvements of in this paper are in 3 folds: 1) we introduce the privacy-preserving requirements for ABE and incorporate the privacy-preserving solutions into the previous approaches; 2) we present a PP-AB-BE with an information theoretical analysis to address its complexity; and 3) we conduct a comprehensive performance evaluation.

ABE can be used as a perfect cryptographic building block to realize Broadcast Encryption (BE), which was introduced by Fiat and Naor in [19]. The encrypter in the existing BE schemes need to specify the receiver list for a particular message. In many scenarios, it is very hard to know the complete receiver list and it is desirable to be able to encrypt without exact knowledge of possible receivers. Also, existing BE schemes [8], [20] can only support a simple receiver list. It is hard to support flexible, expressive access control policies. A broadcast encryption with an attribute based mechanism was proposed in [21], where an expressive attribute-based access policy replaces the flat receiver list. Also, in [22] and [5], the authors proposed to use a CP-ABE [4], [5] and flat-table [23] mechanism to minimize the number of messages and support expressive access policies. Compared with these works, our proposed scheme significantly reduces the size of ciphertext from linear to constant.

3 Models

In this section, we first describe how to use attributes to form a data access policy, followed by the concept of the broadcast encryption based on an attribute-based mechanism. Then we present the bilinear map, which is the building block of ABE schemes. Finally, we present the complexity assumption, which will be used for our security proof.

3.1 Attributes, Policy and Anonymity

Let $U = \{A_i\}_{i\in[1,k]}$ be the *Universe* of attributes in the system. Each A_i has three values: $\{A_i^+, A_i^-, A_i^*\}$. When a user u joins the system, u is tagged with an attribute list defined as follows:

Definition 1. *A user's attribute list is defined as $L = \{L[i]_{i\in[1,k]}\}$, where $L[i] \in \{A_i^+, A_i^-\}$ and k is the number of attributes in the universe.*

Intuitively, A_i^+ denotes the user has A_i; A_i^- denotes the user does not have A_i or A_i is not a proper attribute of this user. For example, suppose $U = \{A_1 = \text{CS}, A_2 = \text{EE}, A_3 = \text{Faculty}, A_4 = \text{Student}\}$. Alice is a student in CS department; Bob is a faculty in EE department; Carol is a faculty holding a joint position in EE and CS department. Their attribute lists are illustrated in Table I.

As the actual data access policy is hidden in the ciphertext header, effective measures are required to avoid ambiguity. In other words, when a decryptor receives a ciphertext header without knowing the access policy, he/she should NOT try a large number of access policies when performing decryption. To this end, we adopt a AND-gate policy construction so that each decryptor only needs to try once on each ciphertext header.

TABLE 1
Attribute Examples

Attributes	$L[1]$	$L[2]$	$L[3]$	$L[4]$
Description	CS	EE	Faculty	Student
Alice	A_1^+	A_2^-	A_3^-	A_4^+
Bob	A_1^-	A_2^+	A_3^+	A_4^-
Carol	A_1^+	A_2^+	A_3^+	A_4^-

TABLE 2
An Example of the Access Policies and Anonymized Policies

Attributes	$W[1]$	$W[2]$	$W[3]$	$W[4]$
Description	CS	EE	Faculty	Student
W_1	A_1^+	A_2^-	A_3^-	A_4^+
$\overline{W}_1$	✠	✠	✠	✠
W_2	A_1^+	A_2^-	A_3^*	A_4^*
$\overline{W}_2$	✠	✠	A_3^*	A_4^*

*where ✠ represents "*do not care*".

The hidden AND-gate access policy is defined as follows:

Definition 2. *Let $W = \{W[i]\}_{i\in[1,k]}$ be an AND-gate access policy, where $W[i] \in \{A_i^+, A_i^-, A_i^*\}$. We use the notation $L \models W$ to denote that the attribute list L of a user satisfies W, as:*

$$L \models W \Leftrightarrow W \subset L \bigcup \{A_i^*\}_{i\in[1,k]}.$$

A_i^+ or A_i^- requires the exact same attribute in the user's attribute list. As for A_i^*, it denotes a wildcard value, which means the policy does not care about the value of attribute A_i. Effectively, each user with either A_i^+ or A_i^- fulfills A_i^* automatically.

Accordingly, we also define an anonymized AND-gate policy that removes all identifying attribute values, i.e. $\{A_i^+, A_i^-\}$, except do-not-care values, i.e. A_i^*. Formally, we define an anonymized AND-gate policy as follows:

Definition 3. *Let $\overline{W} = W \bigcap_{\{A_i^*\}_{i\in[1,k]}}$ be an anonymized AND-gate access policy.*

We note that the do-not-care attribute values are included in the anonymized access policy. If we hide the wildcard attributes, the decryptor will need to guess 2^k possible access policies if there are k attributes in the policy, i.e., for each attribute, its value can be either A_i^* or the specific value (A_i^+ or A_i^-) assigned to the decryptor. This would make the scheme infeasible in terms of performance. Table II shows an example to specify an access policy W_1 for all CS students and an access policy W_2 for all CS people:

The anonymity policy is defined as *the state of being not identifiable within a set of subjects, i.e., the anonymity set*. As the access policy is one-to-many mapped to users, we can extend this definition of policy anonymity set of blinded policy as:

Definition 4. *The anonymity set of a blinded policy $\overline{W}$ is the set of access policies which are identically blinded to $\overline{W}$.*

Here, we briefly analyze the anonymity level of the blinded access policy. Firstly, if there are no wildcards in the original

access policy (hidden), the blinded policy $\overline{W}$ will be empty. In this case, the size of anonymity set is 2^k, as there are 2^k possible access policies blinded to $\overline{W}$. If there are j wildcards in the original access policy (hidden), the size of anonymity set is 2^{k-j}.

3.2 Broadcast with Attribute-Based Encryption

A broadcast encryption is usually applied in the scenario wherein a broadcaster sends messages to multiple receivers through an insecure channel. The broadcaster should be able to select a subset of users with certain policies from all receivers, and consequently only the eligible users are able to decrypt the ciphertexts and read the messages. It is possible that the number of all possible receivers are infinite, and the subset of privileged receivers changes dramatically in each broadcast based on the content of the message and the will of the broadcaster.

The notion of Attribute-based Encryption [10] can be utilized to address this problem. In ABE, all the possible receivers are ascribed by an attribute set. As such, the broadcaster can specify an expressive policy and select a group of privileged receivers defined by their attributes. Consequently, only the receivers whose attributes satisfy the policy embedded into the access structure are able to decrypt the ciphertexts transmitted through the unsecured broadcast channel.

3.3 Bilinear Maps

A pairing is a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$, where $\mathbb{G}_0$ and $\mathbb{G}_1$ are two multiplicative cyclic groups with large prime order p. The Discrete Logarithm Problem on both $\mathbb{G}_0$ and $\mathbb{G}_1$ is hard. Pairing has the following properties:

- *Bilinearity*:

$$e(P^a, Q^b) = e(P, Q)^{ab}, \quad \forall P, Q \in \mathbb{G}_0, \forall a, b \in \mathbb{Z}_p^*.$$

- *Nondegeneracy*:
  $e(g, g) \neq 1$ where g is the generator of $\mathbb{G}_0$.
- *Computability*:
  There exist an efficient algorithm to compute the pairing.

3.4 Complexity Assumption

The security of our proposed constructions is based on a complexity assumption called the Bilinear Diffie-Hellman Exponent assumption (BDHE) [24].

Let $\mathbb{G}_0$ be a bilinear group of prime order p. The K-BDHE problem in $\mathbb{G}_0$ is stated as follows: given the following vector of $2K + 1$ elements (Note that the $g^{\alpha^{K+1}}$ is not in the list):

$$(h, g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{\alpha^K}, g^{\alpha^{K+2}}, \ldots, g^{\alpha^{2K}}) \in \mathbb{G}_0^{2k+1}$$

as the input and the goal of the computational K-BDHE problem is to output $e(g, h)^{\alpha^{(K+1)}}$. We can denote the the set as:

$$Y_{g,\alpha,K} = \{g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{\alpha^K}, g^{\alpha^{K+2}}, \ldots, g^{\alpha^{2K}}\}.$$

Definition 5. (Decisional *K*-BDHE). *The decisional K-BDHE assumption is said to be held in $\mathbb{G}_0$ if there is no probabilistic polynomial time adversary who is able to distinguish*

$$\langle h, g, Y_{g,\alpha,K}, e(g, h)^{\alpha^{(K+1)}} \rangle$$

and

$$\langle h, g, Y_{g,\alpha,K}, e(g, h)^{R} \rangle$$

with non-negligible advantage, where $\alpha, R \in \mathbb{Z}_p$ and $g, h \in \mathbb{G}_0$ are chosen independently and uniformly at random.

4 PP-CP-ABE Construction

In this section, we present our construction of the PP-CP-ABE scheme.

4.1 PP-CP-ABE Construction Overview

The PP-CP-ABE scheme consists of four fundamental algorithms:

- $\mathbf{Setup}(\mathbf{1}^{\lambda}, \mathbf{k})$
  The **Setup** algorithm takes input of the security parameter 1^{λ} and the number of attributes in the system k. It returns public key PK and master key MK. The public key is used for encryption while the master key is used for private key generation.
- $\mathbf{KeyGen}(\mathbf{PK}, \mathbf{MK}, \mathbf{L})$
  The **KeyGen** algorithm takes the public key PK, the master key MK and the user's attribute list L as input. It outputs the private key of the user.
- $\mathbf{Encrypt}(\mathbf{PK}, \mathbf{W}, \mathbf{M})$
  The **Encrypt** algorithm takes the public key PK, the specified access policy W and the message M as input. The algorithm outputs ciphertext CT such that only a user with attribute list satisfying the access policy can decrypt the message. The ciphertext also associates the anonymized access policy $\overline{W}$.
- $\mathbf{Decrypt}(\mathbf{PK}, \mathbf{SK}, \mathbf{CT})$
  The **Decrypt** algorithm decrypts the ciphertext when the user's attribute list satisfies the access policy. It takes the public key PK, the private key SK of the user and the ciphertext CT, which only includes the anonymized access policy $\overline{W}$ as input. It returns a valid plaintext M if $L \models W$, where L is the user's attribute list and W is the access policy hidden from the ciphertext.

Boneh et al. proposed a broadcast encryption construction with constant ciphertext size in [8], where the broadcast encryptor uses the public key list corresponding to intended receivers to perform encryption. To make the ciphertext constant, each receiver's public key is multiplied together, assuming a multiplicative group structure. Thus, the resulting ciphertext is still an element on the group, i.e., the size of the ciphertext is constant. We use a similar strategy to achieve constant ciphertext in our proposed scheme.

In our construction, each public key is mapped to an attribute value, including A_i. To encrypt a message, the encryptor specifies an access policy W by assigning an attribute value ($A_i \in \{1, 0, *\}$) for each of the n attributes in the Universe and encrypts the message using public keys of the attribute values in the W. Each decryptor is generated as a set of private key components corresponding to his/her attribute list L. All the private key components of the same user are tied together by a common random factor to prevent collusion attacks.

4.2 Setup

Assuming there are k attributes $\{A_1, A_2, \cdots, A_k\}$ in the system, we have $K = 3k$ attribute values since each attribute A_i

TABLE 3
Mapping Attribute Values to Numbers

Attributes	A_1	A_2	A_3	$\cdots$	A_k
A_i^+	1	2	3	$\cdots$	k
A_i^-	$k+1$	$k+2$	$k+3$	$\cdots$	$2k$
A_i^*	$2k+1$	$2k+2$	$2k+3$	$\cdots$	$3k$

has 3 values: $\{A_i^+, A_i^-, A_i^*\}$. For ease of presentation, we map the attribute values to integer numbers as depicted in the Table 3.

Trusted Authority (TA) first chooses 2 bilinear groups $\mathbb{G}_0$ and $\mathbb{G}_1$ of prime order p (such that p is λ bits long) and a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$. TA then picks a random generator $g \in \mathbb{G}_0$ and a random $\alpha \in \mathbb{Z}_p$. It computes $g_i = g^{(\alpha^i)}$ for $i = 1, 2, \ldots, K, K+2, \ldots, 2K$ where $K = 3k$. Next, TA picks a random $\gamma \in \mathbb{Z}_p$ and sets $v = g^\gamma \in \mathbb{G}_0$. The public key is:

$$PK = (g, g_1, \ldots, g_K, g_{K+2}, \ldots, g_{2K}, v) \in \mathbb{G}_0^{2K+1}.$$

The master key $MK = \{\gamma, \alpha\}$ is guarded by the TA.

4.3 Key Generation

Each user u is tagged with the attribute list $L_u = \{L_u[i]_{i\in[1,k]}\}$ when joining the system, where $1 \leq L_u[i] \leq 2k$. The TA first selects k random numbers $\{r_i\}_{i\in[1,k]}$ from $\mathbb{Z}_p$ and calculate $r = \sum_{i=1}^{k} r_i$.

The TA computes $D = g^{\gamma r} = v^r$. For $\forall i \in [1, k]$, TA calculates $D_i = g^{\gamma(\alpha^{L_u}[i]+r_i)} = g_{L_u[i]}^{\gamma} \cdot g^{\gamma r_i}$ and $F_i = g^{\gamma(\alpha^{2k+i}+r_i)} = g_{2k+i}^{\gamma} \cdot g^{\gamma r_i}$.

The private key for user u is computed as:

$$SK_u = (D, \{D_i\}_{i\in[1,k]}, \{F_i\}_{i\in[1,k]}).$$

4.4 Encryption

The encrypter picks a random t in $\mathbb{Z}_p$ and sets the one-time symmetric encryption key $Key = e(g_K, g_1)^{kt}$. Suppose AND-gate policy is W with k attributes. Each attribute is either positive/negative or wildcards.

The encryptor first encrypts the message using symmetric key Key as $\{M\}_{Key}$. The encryptor also sets $C_0 = g^t$. Then, it calculates $C_1 = (v \prod_{j\in W} g_{K+1-j})^t$. Also, the encryptor anonymizes the access policy W by removing all attribute values except do-not-care values, i.e. A_i^*, and outputs $\overline{W} = W \bigcap_{\{A_i^*\}_{i\in[1,k]}}$.

Finally, the ciphertext is:

$$\begin{aligned} CT &= (\overline{W}, \{M\}_{Key}, g^t, (v \prod_{j\in W} g_{K+1-j})^t) \\ &= (\overline{W}, \{M\}_{Key}, Hdr), \end{aligned}$$

where the ciphertext header $Hdr = \{C_0, C_1\}$.

4.5 Decryption

Before performing decryption, the decryptor u has has little information about the access policy that is enforced over the ciphertext. Only if $L_u \models W$ can u successfully recover the valid plaintext and access policy. Otherwise, u can only get a random string which can be easily detected. Moreover, the access policy remain unknown to the unsuccessful decryptors.

First of all, u constructs a local guess of access policy, denoted as $\widetilde{W}$, as specified in Algorithm 1. Essentially, this algorithm constructs only one guess by replacing hidden attributes in the anonymized access policy $\overline{W}$ with the corresponding attribute values of the receiver. If the receiver satisfies the access policy, the Algorithm 1 will always produce the correct guess and the decryption will succeed. On the other hand, if a guess is not identical to the actual access policy, the decryption will fail and the decryptor does not need to try other guesses.

Algorithm 1 Construct local guess $\widetilde{W}$

Initialize $\widetilde{W} = \overline{W}$
for $i = 1$ to k **do**
  if $\overline{W}[i] ==$ ✠ **then**
    $\widetilde{W}[i] = L_u[i]$;
  end if
end for
return $\widetilde{W}$;

For $\forall i \in [1, k]$, u calculates the T_0 and T_1 as follows.

$$\begin{aligned} T_0 &= e(g_{\widetilde{W}[i]}, C_1) \\ &= e(g^{\widetilde{\alpha^{W[i]}}}, g^{t(\gamma+\sum_{j\in W}\widetilde{\alpha^{K+1-j}})}) \\ &= e(g, g)^{t\gamma\widetilde{\alpha^{W[i]}}+t\sum_{j\in W}\widetilde{\alpha^{K+1-j+W[i]}}}. \end{aligned}$$

And if $\widetilde{W}[i] \in L_u$, u computes:

$$\begin{aligned} T_1 &= e(D[i] \cdot \prod_{j\in\widetilde{W}, j\neq\widetilde{W}[i]} g_{K+1-j+\widetilde{W}[i]}, C_0) \\ &= e(g^t, g^{\widetilde{\gamma(\alpha^{W[i]}+r_i)}+\sum_{j\in\widetilde{W}, j\neq\widetilde{W}[i]}\widetilde{\alpha^{K+1-j+W[i]}}}) \\ &= e(g, g)^{t\widetilde{\gamma(\alpha^{W[i]}+r_i)}+t\sum_{j\in\widetilde{W}, j\neq\widetilde{W}[i]}\widetilde{\alpha^{K+1-j+W[i]}}}. \end{aligned}$$

Else, if $\widetilde{W}[i] \in \{A_i^*\}_{i\in[1,k]}$, u computes:

$$\begin{aligned} T_1 &= e(F[i] \cdot \prod_{j\in\widetilde{W}, j\neq\widetilde{W}[i]} g_{K+1-j+\widetilde{W}[i]}, C_0) \\ &= e(g^t, g^{\widetilde{\gamma(\alpha^{W[i]}+r_i)}+\sum_{j\in\widetilde{W}, j\neq\widetilde{W}[i]}\widetilde{\alpha^{K+1-j+W[i]}}}) \\ &= e(g, g)^{t\widetilde{\gamma(\alpha^{W[i]}+r_i)}+t\sum_{j\in\widetilde{W}, j\neq\widetilde{W}[i]}\widetilde{\alpha^{K+1-j+W[i]}}}. \end{aligned}$$

Then, we calculate

$$T_0/T_1 = e(g, g)^{-t\gamma r_i+t\alpha^{K+1}}.$$

After u calculates all k terms, we make a production of all the quotient terms and get:

$$e(g,g)^{-t\gamma(r_1+r_2+\cdots+r_k)+kt\alpha^{k+1}} = e(g,g)^{-t\gamma r+kt\alpha^{K+1}}.$$

u calculates:

$$e(D,C_0) = e(g,g)^{t\gamma r}.$$

Then, u produces these two terms and gets $Key = e(g,g)^{kt\alpha^{K+1}} = e(g_K,g_1)^{kt}$ and decrypts the message. If the decrypted message is valid, $\widetilde{W} = W$ and u decrypts the ciphertext successfully. Otherwise, u has no information on the W and the anonymity set of $\overline{W}$ does not change.

4.6 Security Analysis

We reduce Chosen Plaintext Attack (CPA) security of our proposed scheme to decisional K-BDHE assumption. We first define the decryption proxy to model collusion attackers.

Security Game for PP-CP-ABE

A CP-ABE scheme is considered to be secure against chosen CPA if no probabilistic polynomial-time adversaries have non-negligible advantages in this game.

Init: The adversary chooses the challenge access policy W and gives it to challenger.

Setup: The challenger runs the Setup algorithm and gives the adversary the PK.

Phase 1: The adversary submits L for a KeyGen query, where $L \not\models W$. The challenger answers with a secret key SK for L. This can be repeated adaptively

Challenge: The challenger runs Encrypt algorithm to obtain $\{< C_0, C_1 >, Key\}$. Next, the challenger picks a random $b \in \{0,1\}$. It sets $Key_0 = Key$ and picks a random Key_1 with same length to Key_0 in $\mathbb{G}_1$. It then gives $\{< C_0, C_1 >, Key_b\}$ to the adversary.

Phase 2: Same as Phase 1.

Guess: The adversary outputs its guess $b' \in \{0,1\}$ and it wins the game if $b' = b$.

Note that the adversary may make multiple secret key queries both before and after the challenge, which results in the collusion resistance in our proposed scheme. We also point out this CPA security game is called as selective ID security, because the adversary must submit a challenge access structure before the setup phase.

Theorem 1. *If a probabilistic polynomial-time adversary wins the CPA game with non-negligible advantage, then we can construct a simulator that distinguish a K-DBHE tuple with non-negligible advantage.*

Proof of Theorem 1. We reduce CPA security of our proposed scheme to decisional K-BDHE assumption. We first define the decryption proxy to model collusion attackers. □

Definition 6. (Decryption Proxy). *In order to model the collusion attacks, we define $2k$ decrypting proxies in the security game. Each decrypting proxy $p_i(r) = g^{\gamma(\alpha^i+r)}$, where $r \in \mathbb{Z}_p$ and $i \in \{1,\ldots,2k\}$, i.e., a private key component corresponding to a particular attribute value.*

In collusion attacks against access policy W, a user with attribute list $L \not\models W$ collude with $x \le k$ decryption proxies to attack the ciphertext. We call the colluding with x decryption proxy as x-collusion. Intuitively, x-collusion means the attacker needs x attributes values, say $\{i_1, i_2, \ldots, i_x\}$ to add to his attribute list L such that $L \cup \{i_1, i_2, \ldots, i_x\} \models W$. Note that 0-collusion means no decryption proxy is used and user does not collude.

Suppose that an adversary $\mathcal{A}$ wins the selective game for PP-CP-ABE with the advantage ε. Then, we can construct a Simulator $\mathcal{B}$ that breaks decisional K-BDHE assumption with the advantage $\max\{\varepsilon/2, (1-q/p)^l\varepsilon/2, (1-(1-(1-q/p)^l)^m)\ \varepsilon/2\}$. The simulator $\mathcal{B}$ takes an input a random decisional K-BDHE challenge

$$\langle h, g, Y_{g,\alpha,K}, Z\rangle,$$

where Z is either $e(g,h)^{\alpha^{(K+1)}}$ or a random element on $\mathbb{G}_0$. $\mathcal{B}$ now plays the role of challenger in the pre-defined CPA game:

Init: $\mathcal{A}$ sends to $\mathcal{B}$ the access policy W that $\mathcal{A}$ wants to be challenged.

Setup: $\mathcal{B}$ runs the **Setup** algorithm to generate PK. $\mathcal{B}$ chooses random $d \in \mathbb{Z}_p$ and generates:

$$v = g^d(\prod_{j\in W} g_{K+1-j})^{-1} = g^{d-\sum_{j\in W}\alpha^{K+1-j}} = g^{\gamma}.$$

The $\mathcal{B}$ outputs the PK as:

$$PK = (g, Y_{g,\alpha,K}, v) \in \mathbb{G}_0^{2K+1}.$$

Phase 1: The adversary $\mathcal{A}$ submits an attribute list L for a private key query, where $L \not\models W$. Otherwise, the simulator quits.

The simulator $\mathcal{B}$ first selects k random numbers $r_i \in \mathbb{Z}_p$ for $i = 1 \ldots k$ and set $r = r_1 + \cdots + r_k$. Then, $\mathcal{B}$ generates

$$\begin{aligned} D &= (g^d \prod_{j\in W}(g_{K+1-j})^{-1})^r \\ &= g^{(d-\sum_{j\in W}\alpha^{K+1-j})r} \\ &= g^{\gamma r}. \end{aligned}$$

Then, for $\forall i \in [1,k]$ and $W[i]! = L[i]$, $\mathcal{B}$ generates:

$$D_i = g_{L[i]}^d \prod_{j\in W}(g_{K+1-j+L[i]})^{-1} g^{ur_i} \prod_{j\in W}(g_{K+1-j})^{-r_i}.$$

Then, for $\forall i \in [1,k]$ and $W[i]! = A_i^*$, $\mathcal{B}$ generates:

$$F_i = g_{2k+i}^d \prod_{j\in W}(g_{K+1-j+2k+i})^{-1} g^{ur_i} \prod_{j\in W}(g_{K+1-j})^{-r_i}.$$

Note that each for each D_i or F_i is valid since:

$$D_i = (g^d(\prod_{j\in W} g_{K+1-j})^{-1})^{(\alpha^{L[i]}+r_i)} = g^{\gamma(\alpha^{L[i]}+r_i)},$$

and

$$F_i = (g^d(\prod_{j\in W} g_{K+1-j})^{-1})^{(\alpha^{2k+i}+r_i)} = g^{\gamma(\alpha^{2k+i}+r_i)}.$$

Challenge: The simulator $\mathcal{B}$ sets $< C_0, C_1 >$ as $< h, h^d >$. It then gives the challenge $\{< C_0, C_1 >, Z^k\}$ to $\mathcal{A}$.

To see the validity of challenge, $C_0 = h = g^t$ for some unknown t. Then:

$$\begin{aligned} h^d &= (g^d)^t \\ &= (g^d \prod_{j \in W} (g_{K+1-j})^{-1} \prod_{j \in W} (g_{K+1-j}))^t \\ &= (v \prod_{j \in W} (g_{K+1-j}))^t, \end{aligned}$$

and if $Z = e(g,h)^{\alpha^{(K+1)}}$, then $Z^k = Key$.

Phase 2: Repeat as *Phase 1*.

Guess: The adversary $\mathcal{A}$ output a guess b' of b. When $b' = 0$, $\mathcal{A}$ guesses that $Z = e(g,h)^{\alpha^{(K+1)}}$. When $b' = 1$, $\mathcal{A}$ guesses Z is a random element.

If Z is a random element, then the $\mathbf{Pr}[\mathcal{B}(h, g, Y_{g,\alpha,K}, Z) = 0] = \frac{1}{2}$.

Before considering the case when $Z == e(g,h)^{\alpha^{(K+1)}}$, we explain how we use decryption proxy in the proof. Each decryption proxy $p_i(r)$ simulates a legal private key component embedded with random number r. When calling $p_i(r)$, $\mathcal{A}$ passes a random r as a guess of the $r_{i'}$, which is the random number embedded in the D_i or F_i, where $i \in W$. As a matter of fact, the procedure of calling decryption proxy mimics the collusion of multiple users, who combine their private key components.

Lemma 1. *Suppose the $\mathcal{A}$ has issued q private queries and there is only 1 attribute $i \notin W$, $\mathcal{A}$ queries $p_i(r) l$ times. The possibility that the none of the queries returns a legal private key component of any q is $(1 - q/p)^l$.*

Proof of Lemma 1. The possibility that the one query does not return a legal private key component of any q is $1 - q/p$. Thus, if none of the l query succeed, the probability $\mathbf{Pr}[r \neq r_{i'}] = (1 - q/p)^l$, where r is the random number in decryption proxy, $r_{i'}$ is the random number embedded in the private key, q is the number of private key queries in phase 1 and phase 2, l is the number of calling decryption proxy with different r, and p is the order of $\mathbb{Z}_p$. □

Lemma 2. *Suppose the $\mathcal{A}$ has issued q private queries and there is m attributes violate the W, $\mathcal{A}$ queries each of the m decryption proxy $p_{i_1}(r_1), p_{i_2}(r_2), \ldots, p_{i_m}(r_m) l$ times. The possibility that the none of the queries returns a legal private key component of any q is $(1 - (1 - q/p)^l)^m$.*

Proof of Lemma 2. The probability that 1 decryption proxy fails is $\mathbf{Pr}[r \neq r_{i'}] = (1 - q/p)^l$. The probability that all the m decryption proxy successfully return legal components is $(1 - (1 - (q/p)^l))^m$. In the case of not all m succeed, the probability is $\mathbf{Pr}[r_{i_j} \neq r_{i'_j}, \exists j \leq m] = 1 - (1 - (1 - q/p)^l)^m$.

If $Z == e(g,h)^{\alpha^{(K+1)}}$, we consider the following cases:

- *0-Collusion*: If no decryption proxy is used, $\mathcal{A}$ has at least $\varepsilon/2$ advantage in breaking our scheme, then $\mathcal{B}$ has at least ε advantage in breaking K-BDHE, i.e.,

$$|\mathbf{Pr}[\mathcal{B}(h, g, Y_{g,\alpha,K}, Z) = 0] - \frac{1}{2}| \geq \varepsilon/2.$$

- *1-collusion* If 1 decryption proxy, say $p_i(r)$ is used, $\mathbf{Pr}[r \neq r_{i'}] = (1 - q/p)^l$, where r is the random number in decryption proxy, $r_{i'}$ is the random number embedded in the private key, q is the number of private key queries in phase 1 and phase 2, l is the number of calling decryption proxy with different r, and p is the order of $\mathbb{Z}_p$. Note that if $r = r_{i'}$, $\mathcal{A}$ can use $p_i(r)$ as a valid private key component to compromise the ciphertext.
- If the $\mathcal{A}$ has at least ε advantage in breaking our scheme, then $\mathcal{B}$ has at least $(1 - q/p)^l \varepsilon/2$ advantage in breaking K-BDHE.
- *m-collusion* If m decryption proxies, say

$$p_{i_1}(r_1), p_{i_2}(r_2), \ldots, p_{i_m}(r_m)$$

are used. The possibility that $\mathbf{Pr}[r_{i_j} \neq r_{i'_j}, \exists j \leq m] = (1 - (1 - (q/p)^l))^m$, where r_m is the random number in m decryption proxy $p_{i_m}(r_{i_m})$ for the private key component i_m, $r_{i'_m}$ is the random number generated for the $\mathcal{A}$, q is the number of private key queries in phase 1 or phase 2, l is the number of calling m decryption proxies with different r's, p is the order of $\mathbb{Z}_p$.

If the $\mathcal{A}$ has at least ε advantage in breaking our scheme, then $\mathcal{B}$ has at least $(1 - (1 - (1 - q/p)^l)^m)\varepsilon/2$ advantage in breaking K-BDHE.

This concludes the proof. □

5 Privacy Preserving Attribute-Based Broadcast Encryption

Based on our construction of PP-CP-ABE, we construct an efficient and flexible Broadcast Encryption (BE) scheme—Privacy Preserving Attribute Based Broadcast Encryption (PP-AB-BE), where the size of any single ciphertext is still constant.

Compared to existing BE schemes, using PP-AB-BE, encryptor does not need to store a large number of key materials, i.e., public key and private key. By carefully organizing the attributes in the system, we will show that the storage overhead of each user can be reduced from $O(N)$ to $O(\log N + m)$, where N is the number of users in the system and $m \ll N$ is the number of descriptive attributes in the system.

Also, in PP-AB-BE, an encryptor enjoys the flexibility of encrypting broadcast data using either a specific list of decryptors or an access policy without giving an exact list of decryptors.

5.1 PP-AB-BE Setup

In PP-AB-BE with N users, each user is issued an n-bit binary ID $b_0 b_1 \cdots b_n$, where b_i represents the i'th bit in the user's binary ID, where $n = \log N$. Accordingly, we can define n bit-assignment attributes $\{B_1, B_2, \ldots, B_n\}$. Each user is assigned n bit-assignment attribute values according to his/her ID. If the $b_i = 1$, he/she is assigned the B_i^+, if the $b_i = 0$, he/she is assigned the B_i^-. For example, in a system with 8 possible users, each user is assigned 3 bit-assignment attributes to represent the bit values in their ID, as illustrated in Fig. 1.

Given the $n = \log N$ the bit-assignment attributes, the TA generates $3n$ attribute values, i.e., bit-assignment attribute B_i has $\{B_i^+, B_i^-, B_i^*\}$ values.

In addition to the bit-assignment attributes, the TA also chooses m descriptive attributes for the system. These descriptive attributes present the real properties or features of an entity, which can be used to describe the decryptors' social or role features, e.g., "CS", "EE", "Student", "Faculty", etc. Each of the m descriptive attributes has $\{1, 0, *\}$ values.

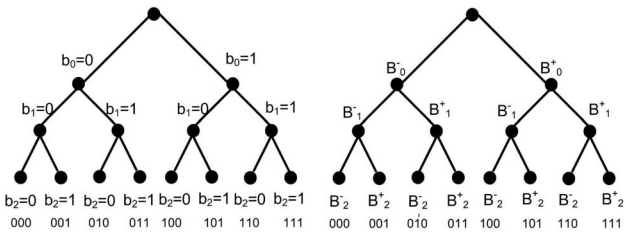


Fig. 1. An illustration of ID and bit-assignment attributes distribution.

With the $3n + 3m$ attribute values, the authority runs $\mathbf{Setup(n + m)}$ algorithm and generate public keys and private keys.

5.2 Broadcast Encryption

In order to control the access to the broadcasted message, the sender needs to specify an access policy using either the descriptive attributes or bit-assignment attributes. For example in the Table 4, if Alice wants to send a message to all CS students, she can specify the descriptive policy W_1 in the following table. Or if she wants to send a message to Bob and Carol, whose IDs are 100 and 101 respectively, she can use the bit-assignment policy W_2, which is equivalent to enumerating every receiver.

Here, we focus on how an encryptor can specify the list of receivers explicitly using n bit-assignment attributes. We first define some of the terms used in the following presentations:

- *Literal*: A variable or its complement, e.g., b_1, $\overline{b_1}$, etc.
- *Product Term*: Literals connected by AND, e.g., $\overline{b}_2 b_1 \overline{b}_0$.
- *Sum-of-Product Expression (SOPE)*: Product terms connected by OR, e.g., $\overline{b}_2 b_1 b_0 + b_2$.

Given the set of receivers S, the membership functions $f_S()$, which is in the form of SOPE, specifies the list of receivers:

$$f_S(b_1^u, b_2^u, \ldots, b_n^u) = \begin{cases} 1, & \text{iff } u \in S, \\ 0, & \text{iff } u \notin S. \end{cases}$$

For example, if the subgroup $S = \{000, 001, 011, 111\}$, then $f_S = \overline{b}_0\overline{b}_1\overline{b}_2 + \overline{b}_0\overline{b}_1 b_2 + \overline{b}_0 b_1 b_2 + b_0 b_1 b_2$.

Then, the broadcast encryptor runs the Quine-McCluskey algorithm [25] to reduce f_S to minimal SOPE f_S^{min}. The reduction can consider *do not care* values $*$ on those IDs that are not currently assigned to any receiver to further reduce the number of product terms in the membership function. For example, if $S = \{000, 001, 011, 111\}$, $f_S^{min} = \overline{b}_0\overline{b}_1 + b_1 b_2$.

Since f_S^{min} is in the form of SOPE, encryption is performed on each product term. That is, for each product term E in f_S^{min}, the encryptor specifies an AND-gate access policy W using the following rules:

1) For positive literal $b_i \in f_S^{min}$, set B_i^+ in the access policy W.
2) For negative literal $\overline{b}_i \in f_S^{min}$, set B_i^- in the access policy W.
3) Set B_i^* for the rest of bit-assignment attributes.

For each W, the encryptor uses $\mathbf{Encrypt(PK, W, M)}$ algorithm to encrypt the message. The total number of encrypted messages equals the number of product terms in f_S^{min}.

For example, if $S = \{000, 001, 011, 111\}$, $f_S^{min} = \overline{b}_0\overline{b}_1 + b_1 b_2$. The access policies W_1 and W_2 are shown in the following table:

TABLE 4
Sample Policies

	CS	EE	Student	Faculty	B_0	B_1	B_2
W_1	A_1^+	A_2^-	A_3^+	A_4^-	B_0^*	B_1^*	B_2^*
W_2	A_1^*	A_2^*	A_3^*	A_4^*	B_0^+	B_1^-	B_2^*

We can find that f_S^{min} contains 2 product terms. the message M for S can be encrypted into 2 ciphertexts with W_1 and W_2 respectively.

5.3 Information Theoretical Optimality

In this section, we present the optimality of PP-AB-BE through an information theoretical approach similar to the models in [26]. In Section 5.3.1, we proved that PP-AB-BE attains the information theoretical lower bound of storage requirements with $O(\log N)$ bit-assignment attributes. In Section 5.3.2, we also compared the BGW [8] BE scheme [8] and PP-AB-BE from information theoretical perspective.

5.3.1 Optimal Storage

To be uniquely identified, each user's ID should not be a prefix of any other user's ID, i.e. *prefix-free*. For example, suppose a user u' is issued an ID 00, which is prefix of u_1 with ID 000 and u_2 with ID 001. When an encryptor tries to reach u_1 and u_2, the minimized membership function is $M = \overline{B}_0\overline{B}_1$, which is also satisfied by u'. Similarly, it is also imperative that a user's bit-assignment attributes should not be a subset of any other user's attribute set. The prefix free condition is a necessary and sufficient condition for addressing any user with their bit-assignment attributes.

Theorem 2. *If we denote the number of bit-assignment attributes for a user u_i by l_i. For an broadcast encryption system with N users and satisfy the prefix-free condition, the set $\{l_1, l_2, \ldots, l_N\}$ satisfies the Kraft inequality:*

$$\sum_{i=1}^{N} 2^{-l_i} \leq 1.$$

Proof of Theorem 2. The proof is available in [27]. □

Assuming l_i bit-assignments are required to identify u_i and the probability to send a message to u_i is p_i, we can model the storage overhead as:

$$\sum_{i=1}^{N} p_i l_i. \tag{1}$$

Intuitively, this formation argues that the storage over head from a sender's perspective is the average number of bit-assignments required to address to any particular receiver. Thus, an optimization problem is formulated to minimize the storage overhead for a broadcast encryption system:

$$\min_{l_i} \sum_{i=1}^{N} p_i l_i,$$

s.t.

$$\sum_{i=1}^{N} 2^{-l_i} \leq 1.$$

This optimization problem is identical to the optimal code word-length selection problem [27] in information theory. Before giving the solution to this optimization problem, we define the entropy of targeting one user in our system:

Definition 7. *The entropy H of targeting a user is*

$$H = -\sum_{i=1}^{N} p_i \log p_i.$$

Theorem 3. *For a system of N users with prefix free distribution of bit-assignments, the optimal (i.e., minimal) average number of storage overhead required for a sender to address a receiver, written as $\sum_{i=1}^{N} p_i l_i$ can be given by the binary entropy*

$$H = -\sum_{i=1}^{N} p_i \log p_i.$$

Proof of Theorem 3. The theorem is equivalent to to optimal codeword-length selection problem and proof is available in [27].

Since the average number of bit-assignment attributes required for addressing one particular receiver is given by the entropy of targeting a user, we now try to derive the upper and lower bounds of the entropy:

$$\max_{p_i} -\sum_{i=i}^{N} p_i \log p_i$$

and

$$\min_{p_i} -\sum_{i=i}^{N} p_i \log p_i,$$

s.t.

$$\sum_{i=1}^{N} p_i = 1.$$

The upper bound $H_{max} = -\sum_{i=1}^{N} \frac{1}{N} \log N = \log N$ is yielded when $p_i = 1/N, \forall i \in \{1, 2, \ldots, N\}$, when each user has equal possibility to be addressed as the receiver. When there is no apriori information about the probability distribution of targeting one of the users, $l = H_{max} = \log_d N$ corresponds to the optimal strategy to minimize the average number of storage overhead required for each user. On the other hand, the lower bound $H_{min} = 0$ is achieved when $p_i = 1$ for $\exists i \in \{1, 2, \ldots, N\}$, which is an extreme case where there is no randomness and only one user is reachable.

5.3.2 Compare with BGW BE Scheme

If we denote our optimal bit-assignment attributes assignment to be minimalist, which requires the least number of bit-assignment attributes to identify each user. We can refer BGW scheme in [8] as maximalist. In BGW scheme, for a system with N users, each user is mapped to a unique public key. Given all N public keys, the number of combinations is $2^N - 1$, which equals the number of receiver subsets in the system. Thus, each encryptor needs maximal number of public keys to perform broadcast encryption.

To compare the minimalist and maximalist storage strategy, we can define The entropy of an attribute or a public key is defined as:

$$H(p) = p \log p^{-1} + (1-p) \log (1-p)^{-1}.$$

where p as the percentage of totals users who have this attributes or public key. We see the entropy of each attribute in minimalist strategy as $H(1/2) = 1$ since, for each particular attribute, exactly half of the users have it while the other half do not have it. On the other hand, the entropy of public key in maximalist strategy is $H(1/N) = (1/N)\log(N) + ((N-1)/N)\log(N/(N-1)) < 1$. Hence, we can conclude that minimalist strategy attains maximal binary entropy while the maximalist strategy attains minimal binary entropy.

6 System Performance Assessment

In this section, we analyze the performance of PP-AB-BE and compare it with several related solutions: subset-difference broadcast encryption scheme (Subset-Diff) [19], BGW [8], and FT implemented using CP-ABE (FT-ABE) [22]. We also compared some works in tree-based multicast group key distribution domain where a group controller removes some group members by selectively multicasting key update messages to all remaining members. Those solutions can be broadly divided into 2 categories: Flat-Table (FT) scheme [23] and Non-Flat-Table schemes, including OFT [28], LKH [29], ELK [30].

6.1 Communication Overhead

The complexity analysis of communication overhead for various schemes is summarized in Table 5. In Subset-Diff scheme, the communication overhead is $O(t^2 \cdot log^2 t \cdot \log N)$, with t as maximum number of colluding users to compromise the ciphertext. For BGW scheme, the message size is $O(N^{\frac{1}{2}})$ as reported in [8]. In ACP scheme, the size of message depends on the degree of access control polynomial, which equals the number of current receivers. Thus, the message size is $O(N)$.

For Non-flat-table tree-based multicast key distribution schemes such as OFT [28], LKH [29], ELK [30], etc., the communication overhead for removing members depends on the number of keys in the tree that need to be updated [31], [30]. In the case of removing a single member, $O(\log N)$ messages are required since the center needs to update $\log N$ auxiliary keys distributed to the removed member. Some tree-based schemes tried to optimize the number of messages to update all the affected keys in the case of multiple *leaves*. In ELK [30], which is known to be one of the most efficient tree-based schemes, the communication overhead for multiple *leaves* is $O(a - l)$, where $a \approx l \log N$ is the number of affected keys and l is the number of leaving members. Thus, the complexity can be written as $O(l \log N)$.

For flat-table tree-based scheme [23], the complexity of removing a single member is also $O(\log N)$. The main benefit of flat-table, however, is the minimal number of messages for batch removing multiple members. In fact, our scheme requires the same number of messages compared to flat-table schemes, thus they both achieved information theoretical optimality. However, flat-table is vulnerable to collusion

TABLE 5
Comparison of Communication Overhead and Storage Overhead in Different Broadcast Encryption Schemes and Group Key Management Schemes

Scheme	Communication Overhead		Storage Overhead	
	single receiver	multiple receivers	Center	User
PP-AB-BE	$O(1)$	$\approx O(\log N)$	N/A	$O(\log N + m)$
Subset-Diff	$O(t^2 \cdot log^2 t \cdot \log N)$	$O(t^2 \cdot log^2 t \cdot \log N)$	$O(N)$	$O(t \log t \log N)$
BGW_1	$O(1)$	$O(1)$	N/A	$O(N)$
BGW_2	$O(N^{\frac{1}{2}})$	$O(N^{\frac{1}{2}})$	N/A	$O(N^{\frac{1}{2}})$
Flat-Table	$O(\log N)$	$\approx O(\log N)$	$O(\log N)/O(N)$	$O(\log N)$
Flat-Table-ABE	$O(\log N)$	$\approx O(\log^2 N)$	$O(\log N)/O(N)$	$O(\log N)$
Non-Flat-Table-Tree	$O(\log N)$	$O(l \cdot \log N)$	$O(N)$	$O(\log N)$

N: the number of group members; l: the number of leaving members; t: maximum number of colluding users to compromise the ciphertext.

attacks. In [22], the authors proposed to implement flat-table using CP-ABE [4] to counter collusion attacks.

To control a set of receivers S using PP-AB-BE, the number of messages depends on the number of product terms in the f_S^{min}. In [32], the authors derived an upper bound and lower bound on the average number of product terms in a minimized SOPE. Experimentally, the average number of messages required is $\approx \log N$[22].

6.1.1 Number of Messages: Worst Cases

We examine some cases when maximal number of messages is required to reach multiple receivers.

Lemma 3 (Multiple Receivers Worst Case). *The worst case of reaching multiple receivers happens when both of following conditions hold: 1) the number of distinct receivers is $N/2$; 2) the Hamming distance between IDs of any two receivers is at least 2. In the worst case, the number of key updating messages is $N/2$.*

Proof of Lemma 3. Please refer to [23] for complete proof. □

In this case, the number messages is $N - N/2 = N/2$ using PP-AB-BE. However, we can see that the worst cases happens in extremely low probability:

Lemma 4 (Worst Case Possibility). *When communicating all subgroups with uniform opportunity, the worst case scenario happens with probability $\frac{1}{2^{N-1}}$.*

Proof of Lemma 4. In the worst case, the Hamming distance of IDs of $N/2$ receivers should be at least 2. As shown in the Karnaugh table in Fig. 2, each cell represents an ID. For any cell marked 0 and any cell marked 1, the Hamming distance is at least 2. Thus, the worst cases happens in two cases: (1) the encryptor wants to reach $N/2$ receivers marked 1 in Fig. 2; (2) the encryptor wants to reach $N/2$ receivers marked 0 in Fig. 2.

We also have the worst case for communicating with the majority of users.

Lemma 5 (Worst Case of Reaching N-2 Receivers). *When reaching $N-2$ receivers, the maximal number of messages required is $n = \log N$, when the Hamming distance between 2 non-receivers is n.*

Proof of Lemma 5. Please refer to [23]. □

6.1.2 Number of Messages: Average Case

To investigate the average case, we simulated PP-AB-BE in a system with 512 users and 1024 users, and the number of messages required are shown in Figs. 3 and 4 respectively. In the simulation, we consider the cases of 0%, 5%, 25%, 50% IDs are not assigned (i.e., *do not care* value). For each case, different percentages of receivers are randomly selected from the group. We repeat this 100 times to average the results. As shown in Figs. 3 and 4, PP-AB-BE achieves roughly $O(\log N)$ complexity, where the number of messages is bounded by $9 \log N$ for the 512-member group and $18 \log N$ for the 1024-member group.

b_0b_1 \ b_2b_3	00	01	11	10
00	1	0	1	0
01	0	1	0	1
11	1	0	1	0
10	0	1	0	1

Fig. 2. Worst cases of broadcast encryption to $N/2$ receivers.

6.1.3 Total Message Size

Finally, as shown in Figs. 5 and 6, we look into the message size of PP-AB-BE, with comparison to FT-CP-ABE [22]. As mentioned in [22], in FT-CP-ABE, the size of ciphertext grows linearly based on the increase of the number of attributes in the access policy [22], [4]. Experimentally, the message size in FT-CP-ABE starts at about 630 bytes, and each additional attribute adds about 300 bytes. In a system with 10 bit ID or 1024 users, the number of attributes using FT-CP-ABE ciphertext is at most 10 and the message size may be as large as $630 + 9 \cdot 300 = 3330$ bytes. Since the number of attributes in the access policy is bounded by $\log N$, we can conclude that the communication overhead of FT-CP-ABE is in the order of $O(\log^2 N)$. In PP-AB-BE, every ciphertext contains exactly 2 group member on $\mathbb{G}_0$. Empirically, the size of one element on $\mathbb{G}_0$ is about 128 bytes. Thus, the ciphertext header in PP-AB-BE is bounded within 300 bytes, which is significantly smaller than the ciphertext size reported in FT-CP-ABE [22]. Moreover, since the component C_0 in the ciphertext can be shared by multiple messages, we can further reduce the message size of PP-AB-BE with efficient communication protocol design.

6.2 Storage Overhead

In PP-AB-BE, there are $6 \log N + 1$ elements on $\mathbb{G}_0$ in the PK. Also, a user needs to store $m \ll N$ descriptive attributes. Thus, the storage overhead is $O(\log N + m)$, assuming a user

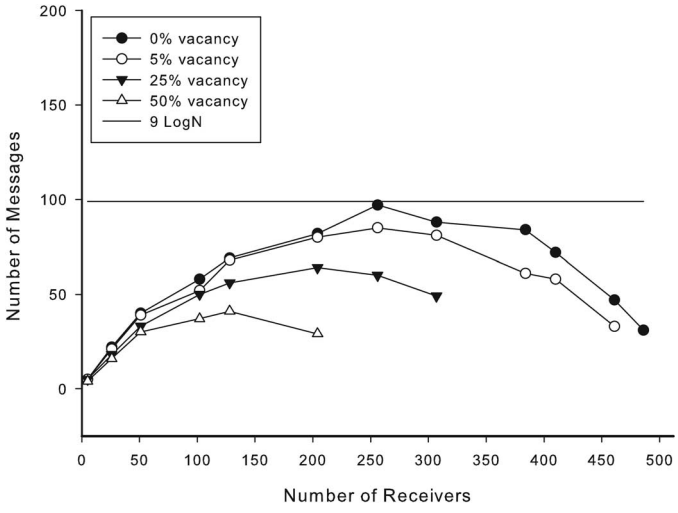


Fig. 3. Number of messages in a system with 512 users.

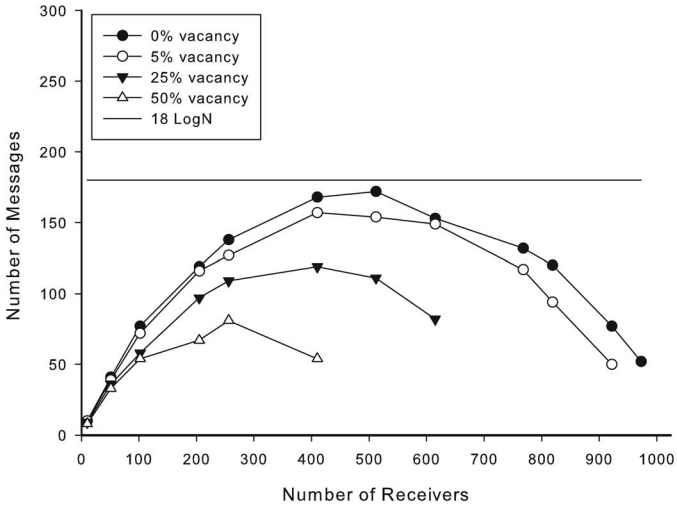


Fig. 4. Number of messages in a system with 1024 users.

does not store any IDs of other users. Although the broadcast encryptor may need the list of receivers' IDs along with the list of *do not care* IDs to perform boolean function minimization, we can argue that this does not incur extra storage overhead.

- The encryptors do not need to store the receiver's IDs after the broadcast; thus, the storage space can be released.
- The TA can periodically publish the minimized SOPE of all *do not care* IDs, which can be used by encryptors to further reduce number of messages.
- If IDs are assigned to users sequentially, i.e., from low to high, TA can simply publish the lowest unassigned IDs to all users, who can use the all higher IDs as *do not care* values.
- Even if a user needs to store N IDs, the space is merely $N \log N$ bits. If $N = 2^{20}$.
- If a broadcast encryptor cannot utilize *do not care* values to further reduce the membership function in SOPE form, the communication overhead might be a little higher. As shown in Figs. 3 and 4, the curve of 0% vacancy can also be used as number of messages required if a broadcast encryptor does not know the *do not care* IDs.

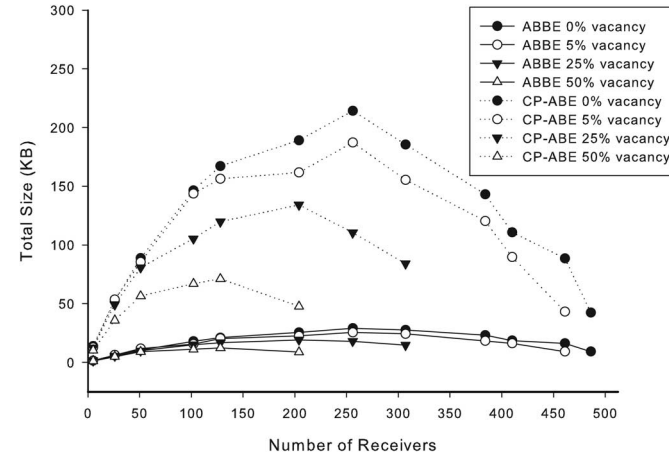


Fig. 5. Total size of messages in a system with 512 users.

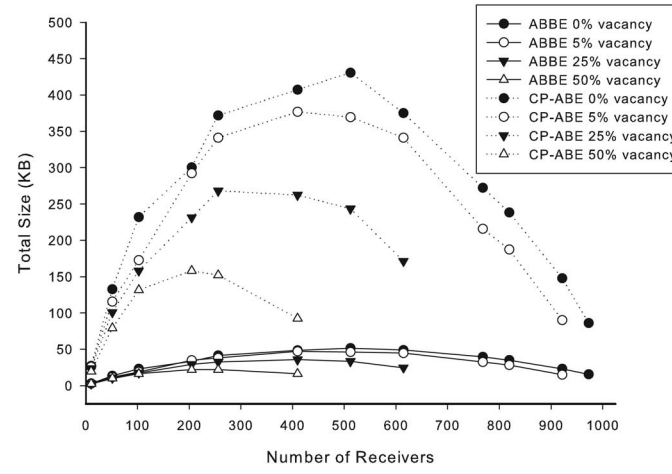


Fig. 6. Total size of messages in a system with 1024 users.

6.3 Computation Overhead

In this section, we compare the computation overhead of those asymmetric key based schemes and the summarized results are presented in Table 6. In ACP scheme, the author reports that the encryption needs $O(N^2)$ finite field operations when the sub-group size if N; in the BGW scheme, the encryption and decryption require $O(N)$ operations on the bilinear group, which are heavier than finite field operations [33]. In PP-AB-BE, each encryption requires $\log N$ operations on the $\mathbb{G}_0$, and the decryption requires $2 \log N + 1$ pairings and $\log N(\log N - 1) + \log N$ operations on $\mathbb{G}_0$ and $\log N$ operations on $\mathbb{G}_1$. Thus, the complexities of encryption and decryption are bounded by $O(\log N)$. Although the problem of minimizing SOPE is NP-hard, efficient approximations are widely known. Thus, PP-AB-BE is much more efficient than ACP and BGW when group size is large.

In Table 7, we summarize the computation overhead based on the benchmark evaluations for PP-CP-ABE operations. The benchmark was performed on a modern workstation which has a 3.0 GHz Pentium 4 CPU with 2 MB cache and 1.5 GB memory and runs Linux 2.6.32 kernel. In the performance evaluation, the Type- D curve [7] is used in the testing. We run each of the algorithm 100 times and the result is the average value. Since the Encryption algorithm only requires $\log N$

TABLE 6
Comparison of Computation Complexity in Different Broadcast Encryption Schemes

Scheme	Computation Overhead	
	Encryption	Decryption
PP-AB-BE	$O(\log N)$	$O(\log N)$
BGW	$O(M)$	$O(M)$
ACP	$O(M^2)$	$O(1)$

N: the number of group members; M: the number of receivers.

TABLE 7
Computation Overhead for 1024 and 4096 Group

	1024 group	4096 group
Encrypt (ms)	12	13
Decrypt (ms)	360	455

operations on the $\mathbb{G}_0$ group, the encryption time difference between 1024 group and 4096 group is very small. On the other hand, the decryption algorithm requires $2\log N + 1$ expensive pairings operations, and $\log N(\log N - 1) + \log N$ operations on the $\mathbb{G}_0$ group. Thus, the decryption on 4096 groups requires 4 more pairing operations than the decryption on 1024 group and each pairing requires around 20 ms in our experiment. Overall, the experiment results are consistent with our complexity analysis.

7 Conclusion and Future Work

In this paper, a Constant Ciphertext Policy Attribute Based Encryption (PP-CP-ABE) was proposed. Compared with existing CP-ABE constructions, PP-CP-ABE significantly reduces the ciphertext size from linear to constant and supports expressive access policies. Thus, PP-CP-ABE can be used in many communication constrained environments.

Based on PP-CP-ABE, we further proposed an Attribute Based Broadcast Encryption (PP-AB-BE) scheme that attains information theoretical minimal storage overhead. Thus, a storage restricted user can easily pre-install all required key materials to perform encryption and decryption. Through theoretical analysis and simulation, we compared PP-AB-BE with many existing BE solutions and we showed that PP-AB-BE achieve better trade-offs between storage and communication overhead.

The security of PP-CP-ABE is based on selective-ID attackers. One open problem is constructing constant CP-ABE that is secure against adaptive adversaries. Another limitation of this paper is the PP-CP-ABE is constructed and proved following BGW [8] model. We are looking for new constructions with equal or stronger security level. Also, in this paper, we only proved PP-AB-BE is minimalist in terms of storage overhead. We are working on more information theoretical analysis that takes into account both storage-communication overhead in BE schemes.

The future research of this work will have two directions: First, the presented solution only supports conjunctive access policy. An important improvement is to extend access policies to support more flexible forms, e.g., including disjunctive normal form and non-monotonic form. Second, the wildcard attribute is not hidden in the access policy to avoid ambiguity for the decryptor, an interesting enhancement is to support a complete hidden access policy without needing to identify the involved wildcard attributes.

Acknowledgments

The authors thank anonymous reviewers for their insightful comments to improve the quality of this article. This research is sponsored by ONR YIP Award N00014-10-1-0714 and ARO Research Grant W911NF-11-1-0191.

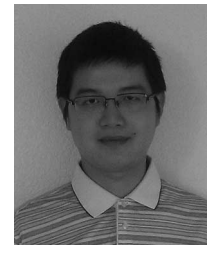
References

[1] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Proc. Public Key Cryptography (PKC)*, 2010, pp. 19-34.
[2] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Proc. Adv. Cryptology (EUROCRYPT)*, vol. 4965, 2008, pp. 146-162
[3] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proc. Appl. Cryptography Netw. Security*, vol. 5037, 2008, pp. 111-129.
[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy*, 2007, pp. 321-334.
[5] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th ACM Conf. Comput. Commun. Security*, 2007, pp. 456-465.
[6] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proc. Theory Cryptography*, 2007, pp. 535-554
[7] B. Lynn, "On the implementation of pairing-based cryptosystems," PhD dissertation, Stanford Univ., Stanford, CA [Online]. Available: http://crypto.stanford.edu/pbc/thesis.pdf, 2007.
[8] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proc. Adv. Cryptology (CRYPTO)*. Springer, 2005, pp. 258-275
[9] M. Abdalla, D. Catalano, A. Dent, J. Malone-Lee, G. Neven, and N. Smart, "Identity-Based encryption gone wild," in *Proc. Automata, Languages Program.*, 2006, pp. 300-311.
[10] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Adv. Cryptology—Eurocrypt*, vol. 3494, 2004, pp. 457-473
[11] R. Ostrovsky and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM Conf. Comput. Commun. Security*, 2007, pp. 195-203.
[12] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Proc. Automata, Languages Program.* Springer, 2008, pp. 579-591.
[13] S. Roy and M. Chuah, "Secure data retrieval based on ciphertext policy attribute-based encryption (CP-ABE) system for the DTNs," Tech. Rep., 2009.
[14] Z. Zhou and D. Huang, "On efficient ciphertext-policy attribute based encryption and broadcast encryption," in *Proc. 17th ACM Conf. Comput. Commun. Security*, 2010, pp. 753-755.
[15] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Proc. 5th Int. Conf. Inf. Security Practice Experience.* Springer-Verlag, 2009, pp. 13-23.
[16] S. Yu, K. Ren, and W. Lou, "Attribute-based content distribution with hidden policy," in *Proc. 4th Workshop Secure Netw. Protocols*, 2008, pp. 39-44.
[17] S. Yu, K. Ren, and W. Lou, "Attribute-based on-demand multicast group setup with membership anonymity," *Comput. Netw.*, vol. 54, no. 3, pp. 377-386, 2010.
[18] D. Huang, Z. Zhou, and Z. Yan, "Gradual identity exposure using attribute-based encryption," in *Proc. IEEE 2nd Int. Conf. Social Comput. (SocialCom)*, 2010, pp. 881-888.
[19] A. Fiat and M. Naor, "Broadcast encryption," in *Proc. Adv. Cryptology (Crypto93)*, vol. 773, 1994, pp. 480-491.
[20] C. Delerablée, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys," in *Proc. Pairing-Based Cryptography—Pairing*, 2007, pp. 39-59.

[21] D. Lubicz and T. Sirvent, "Attribute-based broadcast encryption scheme made efficient," in *Proc. 1st Int. Conf. Cryptology Africa (AFRICACRYPT)*. Springer, 2008, pp. 325-342.
[22] L. Cheung, J. Cooley, R. Khazan, and C. Newport, "Collusion-resistant group key management using attribute-based encryption," in *Proc. 1st Int. Workshop Group-Oriented Cryptographic Protocols*, 2007.
[23] I. Chang, R. Engel, D. Kandlur, D. Pendarakis, and D. Saha, "Key management for secure internet multicast using Boolean function minimization techniques," in *Proc. 18th Annu. Joint Conf. IEEE Comput. Commun. Soc. (Infocom)*, 1999, pp. 689-698.
[24] D. Boneh, X. Boyen, and E. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Proc. Adv. Cryptology (EUROCRYPT)*, 2005, pp. 440-456.
[25] E. McCluskey, "Minimization of Boolean functions," *Bell Syst. Tech. J.*, vol. 35, no. 5, pp. 1417-1444, 1956.
[26] R. Poovendran and J. Baras, "An information-theoretic approach for design and analysis of rooted-tree-based multicast key management schemes," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2824-2834, Nov. 2001.
[27] T. Cover and J. Thomas, *Elements of Information Theory*. Hoboken, NJ: Wiley, 2006.
[28] A. Sherman and D. McGrew, "Key establishment in large dynamic groups using one-way function trees," *IEEE Trans. Softw. Eng.*, vol. 29, no. 5, pp. 444-458, May 2003.
[29] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. Networking*, vol. 8, no. 1, pp. 16-30, Feb. 2000.
[30] A. Penrig, D. Song, and D. Tygar, "ELK, a new protocol for efficient large-group key distribution," in *Proc. IEEE Symp. Security Privacy*, 2001, pp. 247-262.
[31] J. Snoeyink, S. Suri, and G. Varghese, "A lower bound for multicast key distribution," *Comput. Netw.*, vol. 47, no. 3, pp. 429-441, 2005.
[32] T. Sasao, "Bounds on the average number of products in the minimum sum-of-products expressions for multiple-value input two-valued output functions," *IEEE Trans. Comput.*, vol. 40, no. 5, pp. 645-651, May 1991.
[33] D. Hankerson, S. Vanstone, and A. Menezes, *Guide to Elliptic Curve Cryptography*. New York, NY: Springer, 2004.

Zhibin Zhou (M'11) received the BS degree from Shanghai Jiao Tong University, China, in 2006, and the PhD degree from Arizona State University, Tempe, in 2011. His research interests include applied cryptography and mobile cloud computing. He is currently with Amazon.com.

Dijiang Huang (M'00–SM'11) received the BS degree from Beijing University of Posts & Telecommunications, China, in 1995. He received the MS and PhD degrees from the University of Missouri–Kansas City, in 2001 and 2004, respectively. He is an associate professor in the School of Computing Informatics and Decision System Engineering at the Arizona State University, Phoenix. His current research interests include computer networking, security, and privacy. He is an associate editor of the *Journal of Network and System Management* (JNSM) and an editor of IEEE COMMUNICATIONS SURVEYS & TUTORIALS. He has served as an organizer for many International conferences and workshops. His research is supported by NSF, ONR, ARO, NATO, and Consortium of Embedded System (CES). He is a recipient of ONR Young Investigator Program (YIP) Award.



Zhijie Wang received the BS and MS degrees from Beijing University of Posts & Telecommunications, China, in 2007 and 2010, respectively. He is currently working toward the PhD degree at Arizona State University, Phoenix. His research interests include wireless networking, applied cryptography, and cloud computing.