Anonymous Attribute-Based Broadcast Encryption With Hidden Multiple Access Structures

Tran Viet Xuan Phuong
*Old Dominion University, ptran@ualr.edu*

Follow this and additional works at: [https://digitalcommons.odu.edu/cybersecurity-pubs](https://digitalcommons.odu.edu/cybersecurity-pubs)

Part of the [Digital Communications and Networking Commons](https://digitalcommons.odu.edu/cybersecurity-pubs), and the [Information Security Commons](https://digitalcommons.odu.edu/cybersecurity-pubs)

**Original Publication Citation**

This Article is brought to you for free and open access by the School of Cybersecurity at ODU Digital Commons. It has been accepted for inclusion in School of Cybersecurity Faculty Publications by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.
Anonymous attribute-based broadcast encryption with hidden multiple access structures

Tran Viet Xuan Phuong¹,²

Received: 2 November 2022 / Revised: 20 November 2023 / Accepted: 24 January 2024
© The Author(s) 2024

Abstract
Due to the high demands of data communication, the broadcasting system streams the data daily. This service not only sends out the message to the correct participant but also respects the security of the identity user. In addition, when delivered, all the information must be protected for the party who employs the broadcasting service. Currently, Attribute-Based Broadcast Encryption (ABBE) is useful to apply for the broadcasting service. (ABBE) is a combination of Attribute-Based Encryption (ABE) and Broadcast Encryption (BE), which allows a broadcaster (or encrypter) to broadcast an encrypted message, including a predefined user set and specified access policy to install the authorization mechanism. It is desirable to hide all the information when producing in the ciphertext, which has not been considered in the previous works of ABBE. Motivated by the above issue, we devise a solution to achieve anonymity for the ABBE scheme, which not only hides the access structures but also anonymizes the user’s identity. In this work, we propose two schemes as Anonymous Key Policy (AKP)-ABBE and Anonymous Ciphertext Policy (ACP)-ABBE with supporting multiple access structures by using OR/AND gates. Specifically, we present the generic constructions of AKP/ACP-ABBE on the building block of the Inner Product Encryption (IPE), which enables the hidden user’s identity and complex OR/AND-Gate access structure. We show that our proposed schemes are secured under the standard models.

Keywords Anonymous · Broadcast · Attribute based · Viète · OR/AND gates · Wildcard · Polynomial · Root

Mathematics Subject Classification 35A01 · 65L10 · 65L12 · 65L20 · 65L70

Communicated by K. Matsuura.

Tran Viet Xuan Phuong
ptran@ualr.edu

¹ School of Cybersecurity, Old Dominion University, Norfolk, VA 23529, USA
² Department of Computer Science, University of Arkansas at Little Rock, Little Rock, AR 72204, USA

Published online: 06 April 2024
1 Introduction

The resources of the broadcasting channel are protected by allowing only the authorized person to be accessible to these resources [27]. By installing the access control into this channel, the individual has the attributes that will be attested to participate in the system. In addition, the access control is set up not only by the identity’s user but also by the predicate of attributes (age, career, address, etc.). Currently, there are many broadcasting systems that integrate fine-grained access control into the authorization of user-accessible such as mobile pay-TV [35], 5G direct access via satellite [12], and Internet of Thing [24]. The incorporation of access control to the broadcasting systems not only controls the filtered users when using the service but also prevents from an unauthorized attempt to the system.

Among all the existing cryptographic tools, Attribute-Based Broadcast Encryption (ABBE) [22, 36] is well-suitable to construct an efficient mechanism. It creates the complicated access control that enables the broadcasting system. ABBE is a combination of Attribute-Based Encryption (ABE) [4, 6, 11, 16, 21, 28, 32] and Broadcast Encryption (BE) [5, 7, 14]. In the first proposal, the ABBE [22] allows the broadcaster to select groups of users defined by their attributes. This scheme is restricted the access policy for the group of users who satisfy the access policy can decrypt the ciphertext of broadcast encryption scheme. Technically, a user joint to the ABBE system is issued by a secret key $SK$ associated with a user identity $ID$ and a set of user’s attributes $L$. Then the broadcaster who launched the ABBE system creates a ciphertext $CT$, which is associated with a list $S$ of the user’s identity and an access policy $W$. In addition, the access policy $W$ is expressed by the predicate of the specified attributes. In the end, a user whose $SK$ can decrypt the ciphertext $CT$ if and only if the user $ID$ belongs to the set $S$ of valid user’s identity, and the user’s attributes $L$ satisfies access policy $W$.

Motivation The Pay TV system wants to uphold customer service by offering exclusive prices and benefits. The system selects promising customers to participate in this campaign. However, all the information, including the price, benefits, and customers, cannot be unveiled publicly. Only the authorized person can intercommunicate with the system to obtain this information. For example, the broadcaster encrypts the data associated with the group of $k$ customers $\{ID_1, ID_2, \ldots, ID_k\}$, and the access policy as “(Town A AND Age > 22 AND No home-phone line) OR (Town C AND Registered home-phone line)”’. Therefore, the broadcaster needs to protect all the information when public on the channel. Indeed, suppose the access policy is hidden when producing in the ciphertext. In that case, the competitor/the adversary can not extract the customer’s information and learn from Pay TV’s strategy to attract customers. Eventually, the customers who have satisfied the access structure can subscribe to their favorite channels. The existing ABBE schemes [10, 20, 30, 33, 36] have not considered the issue of hidden access policy when generating the ciphertext to deliver in the broadcasting channel.

Contribution In order to anonymize both the information of the group of $ID$ users and the access structures, this work proposes two Anonymous Key Policy Attribute Based Broadcast Encryption (AKP-ABBE) and Anonymous Ciphertext Policy Attribute Based Broadcast Encryption (ACP-ABBE) schemes. Our proposed schemes can hide the information of the group of $ID$ users and the access structures when delivering to the broadcasting system. The access structure is expressed by the predicate of positive and negative attributes, which are concatenated by the Boolean gates (AND, OR). Formally, both the descriptions of AKP-ABBE and ACP-ABBE are similar to KP-ABBE [2, 30] and CP-ABBE [2, 30]. To strengthen the anonymity, we devise the solution to adapt two schemes KP-ABBE, CP-ABBE with
OR/AND Gates with positive, negative attributes by exploiting the “attribute-hiding” Inner Product Encryption (IPE) \([1, 3, 9, 19, 23, 26, 29]\) to achieve the A-KP-ABBE and A-CP-ABBE. We then enable the generic constructions for AKP-ABBE, ACP-ABBE.

In AKP-ABBE, to generate the ciphertext, we input a set of indices \(S\) and an attribute list \(L\) containing positive, and negative attributes. We create the polynomial \(\mathcal{P}_S\) from all the \(n\) elements of set \(S\). In order to generate the coefficient of \(\mathcal{P}_S\), we apply the Viète theorem \([31]\) to compute all the coefficients \((a_n, a_{n-1}, \ldots, a_1, a_0)\) of polynomial by using all the elements of set \(S\). Additionally, we aggregate all the attributes in the list \(L\) into one value \(b\), then generate \((b^m, b^{m-1}, \ldots, 1)\), where \(m\) is the total attributes in \(L\). Subsequently, we produce the ciphertext by calling the IPE’s encryption with the input of \(\vec{v} = (a_n, a_{n-1}, \ldots, a_1, a_0, b^m, b^{m-1}, \ldots, 1)\) and message \(M\). In order to generate the secret key, we input a user ID and the complex access structure \(\mathcal{W} = (\text{AND}_{i \in \{1, \ldots, m\}} A_i)\) OR (\(\text{AND}_{i \in \{1, \ldots, m\}} A_i\)) OR \(\text{OR}(\text{AND}_{i \in \{1, \ldots, m\}} A_i))\). We encodes ID to integer value \(\vec{x}\), then, generate as \((x^n, x^{n-1}, x^{n-2}, \ldots, 1)\). Similarly, we create the polynomial \(\mathcal{P}_W\), and obtain \((b_m, b_{m-1}, \ldots, 1, b_0)\). We then produce the secret key by calling the IPE’s key generation with the input of \(\vec{x} = (x^n, x^{n-1}, x^{n-2}, \ldots, 1, b_n, b_{m-1}, \ldots, b_1, b_0)\). As a result, if the inner product of \((\vec{v}, \vec{x})\) equals zero, the IPE’s decryption will return the message \(M\). This means that the ID belongs to set \(S\), and the attribute list \(L\) satisfies the access structure \(\mathcal{W}\). Mathematically, \(x\) and aggregated \(b\) are the roots of polynomial \(\mathcal{P}_S\) and \(\mathcal{P}_W\), respectively.

On the other hand, ACP-ABBE is an inversion form of AKP-ABBE. A set of indices \(S\) and a complex access structure \(\mathcal{W}\) into a vector \(\vec{v}\), which is used for encryption. The user identity ID and user’s attributes L containing positive and negative symbols is transformed into another vector \(\vec{x}\), which is used in key generation. The decryption is successful if the ID belongs to set \(S\), and the attribute list \(L\) satisfies the access structure \(\mathcal{W}\).

Our proposed schemes utilize the IPE manner to achieve the hidden access structures. Hence, we apply the security proof of IPE scheme in \([19, 26]\) to prove that our AKP-ABBE and ACP-ABBE are secure in the standard model. We then compare with ABBE schemes to show our efficiency regarding hidden access structures and anonymity. Moreover, the generic constructions for AKP-ABBE, ACP-ABBE can be applied to many cryptography preliminaries to achieve the anonymous for ABBE schemes.

**Related work** Several ABBE schemes \([2, 18, 22, 30]\) have been proposed in the literature. In \([22]\), Lubicz and Sirvent proposed a CP-ABBE scheme which allows access policies to be expressed in disjunctive normal form, with the OR function provided by ciphertext concatenation. Attrapadung and Imai \([2]\) proposed two KP-ABBE and two CP-ABBE schemes, which are constructed by algebraically combining some existing BE schemes (namely, the Boneh–Gentry–Waters BE scheme \([7]\) and the Sahai–Waters BE scheme \([29]\)) with some existing ABE schemes (namely, the KP-ABE scheme by Goyal et al. \([16]\) and the CP-ABE scheme by Waters \([32]\)). Junod and Karlov \([18]\) also proposed a CP-ABBE scheme that supports boolean access policies with AND, OR and NOT gates. Junod and Karlov’s scheme achieved direct revocation by simply treating each user’s identity as a unique attribute in the attribute universe. In \([30]\) scheme has proposed CP-ABBE and KP-ABBE scheme, which is constant ciphertext size with AND Gates positive, negative attributes and wildcard. In addition, \([10]\) presented an efficient constant-size private key ciphertext-policy ABBE scheme for disjunctive normal form supporting fast decryption, and \([34]\) proposed an efficient ciphertext-policy attribute-based encryption scheme for partially hidden policy, direct revocation, and verifiable outsourced decryption. However, most of current ABBE schemes do not concern
about the anonymous access structures, which are essential when outsourcing the data in the broadcasting system.

Attribute-based encryption (ABE) [4, 6, 11, 16, 21, 28, 32], which was introduced by Sahai and Waters [28] and extensively studied in recent years [6, 16, 21, 32], provides a fine-grained access control of encrypted data. In a Ciphertext Policy Attribute Based Encryption (CP-ABE) system, the secret user key is associated with a set of attributes, and the ciphertext is associated with an access policy. The ciphertext can be decrypted by a secret key if and only if the attributes associated with the secret key satisfy the access policy. A Key Policy ABE (KP-ABE) the system can be defined in a similar way by swapping the positions of the attributes and the access policy. In BE setting, a center is allowed to broadcast a secret to any subset of privileged users out of a universe of size \( n \) so that conjunctions of \( k \) users not in the privileged set cannot learn the secret. Apart from this, several broadcast encryption schemes were adopted with many interesting problems as [7, 8, 13, 15, 17] with solutions for collusion resistance, trace, and revoke for BE.

1.1 Paper organization

We present the preliminaries and definitions in Sect. 2, which is followed by our generic constructions in Sect. 3 and our analyzing the security proof in Sect. 4. We discuss the extensions in Sect. 5, then give the comparisons in Sect. 6. The paper is concluded in Sect. 7.

2 Preliminaries

2.1 Bilinear map and its related assumptions

Let \( G \) and \( G_T \) be two multiplicative cyclic groups of same prime order \( p \). Let \( e: G \times G \rightarrow G_T \) be a bilinear map with the following properties:

1. Bilinearity: \( e(u^a, v^b) = e(u, v)^{ab} \) for any \( u, v \in G \) and \( a, b \in \mathbb{Z}_p \).
2. Non-degeneracy: \( e(g, g) \neq 1 \).

Definition 1 The Decisional Bilinear Diffie–Hellman (DBDH) problem in \( G \) is defined as follows: given a tuple \( (g, g^a, g^b, g^c, T) \in G^4 \times G_T \), decide whether \( T = e(g, g)^{abc} \) or \( T = e(g, g)^r \) where \( a, b, c, r \) are randomly selected from \( \mathbb{Z}_p \). An algorithm \( A \) has advantage \( \epsilon \) in solving the DBDH problem in \( G \) if

\[
\text{DBDH}_A(k) = \Pr[A(1^k, g, g^a, g^b, g^c, Z) = 1 \parallel Z = e(g, g)^{abc}] - \Pr[A(1^k, g, g^a, g^b, g^c, Z) = 1 \parallel Z = g^r] \leq \epsilon
\]

We say that the DBDH assumptions holds in \( G \) if \( \epsilon \) is negligible for any PPT algorithm \( A \).

Definition 2 The Decisional Linear (DLIN) problem in \( G \) defined as follows: given a tuple \( (g, g^a, g^b, g^{ac}, g^d, Z) \in G^5 \times G_T \), decide whether \( T = g^{b(c+d)} \) or \( Z \) in random in \( G \). An algorithm \( A \) has advantage \( \epsilon \) in solving the DLIN problem in \( G \) if

\[
\text{DLIN}_A(k) = \Pr[A(1^k, g, g^a, g^b, g^{ac}, g^d, Z) = 1 \parallel Z = g^{b(c+d)}] - \Pr[A(1^k, g, g^a, g^b, g^{ac}, g^d, Z = 1 \parallel Z = g^r] \leq \epsilon
\]

where \( a, b, c, d, r \in R \mathbb{Z}_p \). We say that the DLIN assumptions holds in \( G \) if \( \epsilon \) is negligible for any PPT algorithm \( A \).
2.2 Anonymous key-policy attribute based broadcast encryption definition

Let $\mathcal{U}$ denote the set of all user indices and $\mathcal{N}$ as the set of all user attributes. An Anonymous Key-Policy Attribute Based Broadcast Encryption (AKP-ABBE) scheme consists of four algorithms:

- **Setup($1^\lambda$)**: The setup algorithm takes the security parameter $1^\lambda$ as input and outputs the public parameters $\mathcal{P}K$, and a master key $\text{MSK}$.
- **Encrypt($M, S, L, \mathcal{P}K$)**: The encryption algorithm takes as input the public parameters $\mathcal{P}K$, a message $M$, a set of user index $S \subseteq \mathcal{U}$, a set of attributes $L \subseteq \mathcal{N}$, and outputs a ciphertext as CT.
- **KeyGen($ID, W, \text{MSK}, \mathcal{P}K$)**: The key generation algorithm takes as input the master key $\text{MSK}$, public parameters $\mathcal{P}K$, a user index $ID \in \mathcal{U}$, an access structure $W$, and outputs a user secret key $SK$.
- **Decrypt($CT, SK$)**: The decryption algorithm takes as input a ciphertext $CT$, a private key $SK$, then it outputs a message $M$ or an error symbol $\perp$.

**Security definition for AKP-ABBE** We define the Selective IND-CPA security for AKP-ABBE via the following game.

- **Init** The adversary commits to the challenge user indices $(S^*_0, S^*_1)$ and target attribute sets $(L^*_0, L^*_1)$.
- **Setup** The challenger runs the Setup algorithm and gives $\mathcal{P}K$ to the adversary.
- **Phase 1** The adversary queries for private keys with pairs of user index and access structure $(ID, W)$ following the cases:
  - $(L^*_0 \not\models W$ and $(L^*_1 \not\models W$) or $(ID \notin S^*_0$ and $ID \notin S^*_1$).
  - $(L^*_0 \models W$ and $(L^*_1 \models W$ and $(ID \in S^*_0$ and $ID \in S^*_1$).

Then the challenger gives the adversary the corresponding secret key $SK$. Otherwise, it outputs $\perp$.
- **Challenge** The adversary submits the two messages $M_0, M_1$ to the challenger with respect to the challenge user indices $(S^*_0, S^*_1)$ and target attribute sets $(L^*_0, L^*_1)$. The challenger flips a random coin $\beta$ and passes the ciphertext $CT^* = \text{Encrypt}(\mathcal{P}K, M_\beta, L^*_\beta, S^*_\beta)$ to the adversary.
- **Phase 2** Phase 1 is repeated.
- **Guess** The adversary outputs a guess $\beta'$ of $\beta$.

**Definition 1** We say an AKP-ABBE scheme is selective IND-CPA secure if for any probabilistic polynomial time adversary

$$\text{Adv}_{\text{s-ind-cpa}}(\lambda) = |\Pr[\beta' = \beta] - 1/2|$$

is a negligible function of $\lambda$.

2.3 Anonymous ciphertext-policy attribute-based broadcast encryption definition

An Anonymous Ciphertext-Policy Attribute-Based Broadcast Encryption (ACP-ABBE) scheme consists of four algorithms:

- **Setup($1^\lambda$)**: The setup algorithm takes the security parameter $1^\lambda$ as input and outputs the public parameters $\mathcal{P}K$, and a master key $\text{MSK}$.
• Encrypt(M, §, W, PK): The encryption algorithm takes as input the public parameters PK, a message M, a set of user index S ⊆ U, and an access structure W, then outputs a ciphertext as CT.
• KeyGen(ID, L, MSK, PK): The key generation algorithm takes as input the master key MSK, public parameters PK, a user index ID ∈ U, and a set of attributes L ⊆ N, and outputs a user secret key SK.
• Decrypt(CT, SK): The decryption algorithm takes as input a ciphertext CT, and a private key SK, then outputs a message M or an error symbol ‘⊥’.

Security definition for ACP-ABBE We define the Selective IND-CPA security for ACP-ABBE via the following game.
• Init The adversary commits to the challenge user indices (S0, S1) and target access structures (W0, W1).
• Setup The challenger runs the Setup algorithm and gives PK to the adversary.
• Phase 1 The adversary queries for private keys with pairs of user index and a user attribute list (ID, L) following the cases:
  - (L ̸|= W0 and (L ̸|= W1) or (ID ̸∈ S0 and ID ̸∈ S1).
  - (L |= W0 and (L |= W1) or (ID ∈ S0 and ID ∈ S1).
• Challenge The adversary submits messages M0, M1 to the challenger with respect to the challenge user indices (S0, S1) and target access structures (W0, W1). The challenger flips a random coin β and passes the ciphertext CTβ = Encrypt(PK, Mβ, Wβ, Sβ) to the adversary.
• Phase 2 Phase 1 is repeated.
• Guess The adversary outputs a guess β’ of β.

Definition 2 We say a ACP-ABBE scheme is selective IND-CPA secure if for any probabilistic polynomial time adversary

\[ \text{Adv}_{cp}^{s-\text{IND-CPA}}(\lambda) = |\Pr[\beta’ = \beta] - 1/2| \]

is a negligible function of λ.

2.4 Inner product encryption

Let \( \Sigma \in \mathbb{Z} \) be the set of attributes involving vectors \( \vec{v} \) of dimension \( n \), and \( \mathcal{F} \) be the class of predicates involving inner-products over vectors \( \mathcal{F} = \{f_\vec{v}, \vec{v} \in \Sigma\} \) such that \( f_\vec{v}(\vec{x}) = 1 \) iff \( \langle \vec{v}, \vec{x} \rangle = 0 \). An inner-product encryption (IPE) scheme for the class of predicate \( \mathcal{F} \) over the set of attributes consists of four algorithms as follows:

• IPE.Setup(1^\lambda, n) on input a security parameter 1^\lambda and the vector length n = poly(\lambda), the algorithm outputs a public key PK and a master secret key MSK.
• IPE.Encrypt(M, PK, \vec{v} = (v_1, v_2, ..., v_n)): on input a message M, the public key PK, and a vector \( \vec{v} \in \Sigma^n \), it outputs a ciphertext CT.
• IPE.KeyGen(MSK, \vec{x} = (x_1, x_2, ..., x_n)): on input the master secret key MSK, a vector \( \vec{x} \in \Sigma \), the algorithm outputs a secret key SK.
• IPE.Decrypt(CT, SK): on input a secret key SK (w.r.t. a vector \( \vec{x} \)) and a ciphertext CT (w.r.t. a vector \( \vec{v} \)), if \( f_\vec{v}(\vec{x}) = 0 \), the algorithm outputs a message M; otherwise, it outputs ⊥.

Security model IPE scheme Following [19], we define the security, i.e., attribute-hiding property, of the IPE scheme. The security is defined by the following game interacted between an attacker \( A \) and a challenger \( C \). We assume that (\( \Sigma, \mathcal{F} \)) are given to both \( A \) and \( C \) in advance.
Anonymous attribute-based broadcast encryption

Fig. 1 Checking one root of P

\[ P(x) = ax^3 + bx^2 + cx + d \]

- **Init** \( A \) outputs two vectors \( \vec{v}, \vec{x} \in \Sigma \)
- **Setup** \( C \) runs Setup to obtain the public key \( \text{PK} \) and master secret key \( \text{MSK} \). \( A \) is given \( \text{PK} \).
- **Query Phase 1** \( A \) adaptively issues private key queries for any vectors \( \vec{v}_1, \ldots, \vec{v}_n \in \Sigma \), subject to the restriction that, \( \forall i, < \vec{v}_i, \vec{x}> = 0 \) if and only if \( < \vec{v}_i, \vec{x}> = 0 \). \( C \) responds with \( \text{SK}_{\vec{v}_i} \leftarrow \text{KeyGen}(\text{SK}, \vec{v}_i) \).
- **Challenge** \( A \) outputs two messages \( M_0, M_1 \) with equal length. If \( M_0 \neq M_1 \), then it is required that \( < \vec{v}, \vec{x}> \neq 0 \neq < \vec{x}, \vec{x}> \) for any \( \vec{x} \) appeared in Query Phase 1. \( C \) flips a random coin \( b \in \{0, 1\} \). If \( b = 0 \), \( C \) returns \( \text{CT} \leftarrow \text{Encryption}(\text{PK}, \vec{v}, M_0) \) to \( A \); otherwise, if \( b = 1 \), \( C \) returns \( \text{CT} \leftarrow \text{Encrypt}(\text{PK}, \vec{x}, M_1) \) to \( A \).
- **Query Phase 2** Phase 1 is repeatedly.
- **Guess** \( A \) outputs a guess bit \( b' \) and succeeds if \( b' = b \).

The advantage of \( A \) in this game is defined as
\[ \text{Adv}_A(\lambda) = \Pr[b' = b] - \frac{1}{2} \]

**Definition 3** We say that an IPE scheme is attribute-hiding if for all polynomial time adversaries \( A \), we have that \( \text{Adv}_A(\lambda) \) is negligible.

In fact, the challenge ciphertext is given to \( A \) as: if \( b = 0 \) then \( \text{CT} \leftarrow \text{Encrypt}(\text{PK}, \vec{v}, M_0) \) and if \( b = 1 \) then \( \text{CT} \leftarrow \text{Encrypt}(\text{PK}, \vec{x}, M_1) \). As well as similar \( \text{Adv}_A(\lambda) \) to the one above, we say that a IPE scheme is attribute-hiding if for all polynomial time adversaries \( A \), we have that \( \text{Adv}_A(\lambda) \) is negligible.

### 2.5 Polynomial and roots

Consider that a polynomial \( P \) has degree \( n \) is defined as:
\[ P = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 \]  
(1)

We then extract the coefficients of \( P \) to create a vector \( \vec{v} \) as follows:
\[ \vec{v} = (a_n, a_{n-1}, \ldots, a_1, a_0) \]

In addition, we create the a vector \( \vec{x} \) by choosing a integer value \( x \) randomly as follows:
\[ \vec{x} = \left( x \cdot x \cdots x, \underbrace{x \cdots x, \ldots, x, 1}_{n}, \underbrace{x \cdots x, \ldots, x, 1}_{n-1} \right) \]

If \( \langle \vec{v}, \vec{x} \rangle = 0 \), then we conclude that \( x \) is a root of polynomial \( P \) (Fig. 1).
2.6 Consequence of Viète formula

We apply consequence of the Viète’s formula to reconstruct all the coefficients of $P$ in (1) as follows:

\[
\begin{align*}
\sum_{1 \leq i_1 < i_2 < \ldots < i_k \leq n} x_{i_1}x_{i_2} \cdots x_{i_k} &= (-1)^k \frac{a_{n-k}}{a_n} \\
\sum_{1 \leq i_1 < i_2 < \ldots < i_k \leq n} x_{i_1} = a_n \\
\sum_{1 \leq i_1 < i_2 < \ldots < i_k \leq n} x_{i_1}x_{i_2} \cdots x_{i_k} &= (-1)^n \frac{a_0}{a_n}
\end{align*}
\]

Generally, we write: $\sum_{1 \leq i_1 < i_2 < \ldots < i_k \leq n} x_{i_1}x_{i_2} \cdots x_{i_k} = (-1)^k \frac{a_{n-k}}{a_n}$ for $k = 1, 2, \ldots, n$. Apart from Sect. 2.4, we can rewrite the $\vec{v}$ as

$$\vec{v} = \left(1, -\frac{a_{n-1}}{a_n}, \ldots, (-1)^n \frac{a_1}{a_n}, (-1)^n \frac{a_0}{a_n}\right).$$

Then we have the $\vec{x} = (x^n, x^{n-1}, \ldots, x, 1)$. If $< \vec{v} \cdot \vec{x} > = 0$, then we conclude that $x$ is a root of polynomial $P$.

3 Generic constructions

3.1 AND/OR gates access structure

3.1.1 AND gates positive/negative attributes

Let $U = \{\text{Att}_1, \text{Att}_2, \ldots, \text{Att}_6\}$ be the universe of the attributes in the system. Each $\text{Att}_i$ is represented by a unique value $A_i$. When a user joins the system, the user is tagged with an attribute list defined as $S = \{S_1, S_2, \ldots, S_6\}$ where each symbol $S_i$ has two possible values: ‘+’ and ‘−’. Let $W = \{W_1, W_2, \ldots, W_6\}$ denote an AND-gate access policy where each symbol $W_i$ has two possible values: ‘+’ and ‘−’. We use the notation $S \models W$ to denote that the attribute list $S$ of a user satisfies $W$.

We illustrate the AND gates with positive/negative attribute by the following example. Suppose that $U = \{\text{Att}_1 = \text{CS}, \text{Att}_2 = \text{EE}, \text{Att}_3 = \text{Professor}, \text{Att}_4 = \text{Faculty}, \text{Att}_5 = \text{Student}, \text{Att}_6 = \text{Tutor}\}$. Alice is a student and tutor in the CS department; Bob is a faculty in the EE department; Carol is a faculty holding a joint position in the EE and CS departments. All attribute lists are expressed in Table 1. In addition, the access structure $W_1$ is designed to allow all the CS students and tutors in only CS departments to access to the system.
Table 2 List of attributes and AND then OR policies

<table>
<thead>
<tr>
<th>Attributes Description</th>
<th>Att_1</th>
<th>Att_2</th>
<th>Att_3</th>
<th>Att_4</th>
<th>Att_5</th>
<th>Att_6</th>
</tr>
</thead>
<tbody>
<tr>
<td>W_{11}</td>
<td>+</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>OR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W_{12}</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Observably, only Alice is the student/tutor of CS departments, which is attested to access to
the system since the Alice’s attributes satisfy the access structure \( W_1 \).

3.1.2 Multiple OR/AND gates

In this work, we consider the complex access structures, which are expressed the predicate
of attributes by both of the OR and AND gates.

Suppose that we have an access structures \( W_1 \) as follows:

\[
W_1 = ((\text{AND}_{i \in \{1, \ldots, m\}} A_i) \text{OR} (\text{AND}_{i \in \{1, \ldots, m\}} A_i) \text{OR} \ldots \text{OR} (\text{AND}_{i \in \{1, \ldots, m\}} A_i))
\]

as the Disjunctive Normal Form (DNF). Utilizing the set of attributes \( U = \{\text{Att}_1, \text{Att}_2, \ldots, \text{Att}_n\} \) in AND gate access structure, \( W_1 \) is expressed as:

\[
W_1 = ((\text{Att}_1 \text{AND Att}_3)) \text{OR} (\text{Att}_1 \text{AND Att}_6 \text{AND Att}_5)),
\]

Regarding the Table 2, we decouple \( W_1 \) into the two access structures \( W_{11} \) and \( W_{12} \). Then
if a user has the set of attributes satisfy \( W_{11} \) or \( W_{12} \), the he is valid to decrypt the message.

Next, we consider the Conjunctive Normal Form (CNF) access structures \( W_2 \) as follows:

\[
W_2 = ((\text{OR}_{i \in \{1, \ldots, m\}} A_i) \text{AND} (\text{OR}_{i \in \{1, \ldots, m\}} A_i) \text{AND} \ldots \text{AND} m-1 (\text{OR}_{i \in \{1, \ldots, m\}} A_i)).
\]

In practice, \( W_2 \) is expressed by the attributes in set \( U \) as:

\[
W_2 = ((\text{Att}_1 \text{OR} \text{Att}_2)) \text{AND} (\text{Att}_3 \text{OR} \text{Att}_4)).
\]

We then transform the \( W_2 \) in the other observation:

\[
W_2 = ((\text{Att}_1 \text{AND Att}_3) \text{OR} (\text{Att}_1 \text{AND Att}_4) \text{OR} (\text{Att}_2 \text{AND Att}_4))
\]

\[
\text{OR} (\text{Att}_2 \text{AND Att}_4))
\]

Regarding the Table 3, we interpret \( W_2 \) into the the set of access structures \( (W_{21}, W_{22}, W_{23}, W_{24}) \). Then if a user has the set of attributes satisfy \( W_{21} \) or \( W_{22} \) or \( W_{23} \)
or \( W_{24} \), then he is valid to decrypt the message. As a result, we realize that when a user joins
the system, the user is tagged with an attribute list defined as \( S = \{A_i\}_{i \in \{1, m\}} \). We conclude
the two statements as follows:

- \( S \models W_1 \), if the set attributes in \( S \) satisfies \textbf{one} of AND literals in \( W_1 \).
- \( S \models W_2 \), if the set attributes in \( S \) satisfies \textbf{all} of OR literals in \( W_2 \).
In this section, we represent the original of IPE scheme [25], which is a building block to construct our proposed work later.

**Setup**($1^k, n$): The setup algorithm first randomly generates $(g, \mathbb{G}, \mathbb{G}_T, p, e)$ and $n$ is the maximum length of vector. It then randomly selects $\gamma_1, \gamma_2, \theta_1, \theta_2, \{u_1, i\}_{i=1}^n, t_1, \{t_1, i\}_{i=1}^n, \{t_2, i\}_{i=1}^n, \{w_1, i\}_{i=1}^n, \{z_1, i\}_{i=1}^n, \{z_2, i\}_{i=1}^n$ in $\mathbb{Z}_p$ and $g_2$ in $\mathbb{G}$. Then it selects a random $\Delta \in \mathbb{Z}_p$ and obtains $(u_2, i)_{i=1}^n, \{w_2, i\}_{i=1}^n, w_2, u_2$ under the condition: $\Delta = \gamma_1 u_2, i - \gamma_2 u_1, i \Delta = \theta_1 w_2, i - \theta_2 w_1, i$.

For $i$ from 1 to $n$, it creates:

$U_{1, i} = g^{u_1, i}, U_{2, i} = g^{u_2, i}, W_{1, i} = g^{w_1, i}, W_{2, i} = g^{w_2, i}, T_{1, i} = g^{t_1, i}, T_{2, i} = g^{t_2, i},$

$Z_{1, i} = g^{z_1, i}, V_1 = g^{v_1}, V_2 = g^{v_2}, X_1 = g^{\theta_1}, V_2 = g^{\theta_2}$.

Next it sets $g_1 = g^\Delta, Y = e(g, g_2)$, and the public key $PK$ and master key $MSK$ as

$PK = (g, \mathbb{G}, \mathbb{G}_T, p, e, g_1, Y, \{U_{1, i}, U_{2, i}, T_{1, i}, T_{2, i}, W_{1, i}, W_{2, i}, Z_{1, i}, Z_{2, i}\}_{i=1}^n, \{V_i, X_i\}_{i=1}^n) \quad \text{MSK} = (g_2, \{u_1, i, u_2, i, t_1, i, t_2, i, w_1, i, w_2, i, z_1, i, z_2, i\}_{i=1}^n, \{v_i, x_i\}_{i=1}^n)$.

**Encrypt**($PK, \vec{v}, M$): The encryption algorithm chooses random $s_1, s_2, \alpha, \beta \in \mathbb{Z}_p$ and creates the ciphertext as follows:

$C_m = M \cdot Y^{s_2}, C_A = g^{s_2}, C_B = g_1^{s_1},$

$\{C_{1, i}, C_{2, i}\} = \{U_{1, i}^{s_1} T_{1, i}^{s_2} V_1^\alpha, U_{2, i}^{s_1} T_{2, i}^{s_2} V_2^\alpha\},$

$\{C_{3, i}, C_{4, i}\} = \{W_{1, i}^{s_1} Z_{1, i}^\beta X_1, W_{2, i}^{s_1} Z_{2, i}^\beta X_2\}$,

where $\vec{v} = (v_1, \ldots, v_n)$, then ciphertext $CT$ is set as:

$CT = (C_m, C_A, C_B, \{C_{1, i}, C_{2, i}, C_{3, i}, C_{4, i}\}_{i=1}^n)$.

**KeyGen**($PK, \vec{x}, MSK$): The key generation algorithm chooses randomly $r_1, i_1, r_2, i_2$ for $i = 1$ to $n$, and $f_1, f_2, r_1, r_2 \in \mathbb{Z}_p$, and then creates the secret key as follows:

$\{K_{1, i}, K_{2, i}\} = \{g^{-\gamma_2 r_1, i} g^{f_1 x_i u_2, i}, g^{\gamma_1 r_1, i} g^{-f_1 x_i u_1, i}\},$

$\{K_{3, i}, K_{4, i}\} = \{g^{-\theta_2 r_2, i} g^{f_2 x_i w_2, i}, g^{\theta_1 r_2, i} g^{-f_2 x_i w_1, i}\},$

$K_A = g_2 \cdot \prod_{i=1}^n K_{1, i} K_{2, i} K_{3, i} K_{4, i}$,

$K_B = \prod_{i=1}^n g^{-\gamma_2 r_1, i} g^{-\theta_2 r_2, i}$.
where \( \vec{x} = (x_1, \ldots, x_n) \), the secret key is set as:

\[
\text{SK} = (K_A, K_B, \{K_{1,i}, K_{2,i}, K_{3,i}, K_{4,i}\})_{i=1}^n.
\]

**Decrypt**(SK, CT): The decryption algorithm returns

\[
\frac{C_m}{e(C_A, K_A) e(C_B, K_B) \prod_{i=1}^n e(C_{j,i}, K_{j,i})} = \frac{M}{e(g, g)^{\sum_{i=0}^{n} v_i \cdot r_i \cdot (f_1 a + f_2 b \Delta)}}.
\]

Therefore, the message \( M \) will be returned iff \( (\vec{v}, \vec{x}) = 0 \) meaning the attributes list in user key SK satisfies the access policy in the ciphertext CT.

Following the description of the above Multiple OR/AND gate access structures and the original IPE construction, we present two Anonymous Key Policy Attribute Based Broadcast Encryption schemes with OR/AND Gate with positive, negative attributes in access structure.

### 3.3 Generic construction of AKP-ABBE from IPE

In our AKP-ABBE scheme, we only consider two values, positive, negative, of attributes. In order to construct, we desire an \((n + m)\)-dimensional IPE scheme, where \( n \) is the number of set indices, and \( m \) is the maximum number of access structures. In this scheme, we present the construction of DNF access structure since the CNF form can converse to the DNF.

Let \( U \) denote the set of all user indices, and \( N \) as the set of all user attributes and given an IPE scheme with four algorithms: (IPE.Setup, IPE.KeyGen, IPE.Enc, IPE.Dec), we construct an AKP-ABBE scheme with the corresponding four algorithms Setup, KeyGen, Encrypt, Decrypt, which we elaborate as follows:

**Setup** \((1^k)\): The algorithm chooses a suitable encoding \( \tau_1 \) sending each of the \( n \) indices \( \text{ID} \in \mathbb{N} \) onto an element \( \tau_1(\text{ID}) = x_1 \in (\mathbb{Z}/p\mathbb{Z})^* \), and choose \( t_1, \ldots, t_n \) randomly in \( \mathbb{Z}_p \).

It runs IPE.Setup \((1^k, n + m)\) with \( m \) as the number of attributes to construct to access structure, and outputs public parameters PK and a master key MSK.

**Encrypt**(PK, M, S, L): The algorithm inputs a user index set \( S = \{\text{ID}_a, \text{ID}_b, \text{ID}_c, \ldots, \text{ID}_s\} \subseteq U \), and message \( M \), attribute list \( L \). The algorithm transforms \((S, L)\) into \( \vec{v} \) as:

The user index set is input as \( S = \{\text{ID}_a, \text{ID}_b, \text{ID}_c, \ldots, \text{ID}_s\} \subseteq U \). We denote \( \Delta \) as the total number elements in set \( S \), then the algorithm applies the Viète’s formula to compute:

\[
\begin{align*}
\tau_1(\text{ID}_a) + \tau_1(\text{ID}_b) + \tau_1(\text{ID}_c) + \ldots + \tau_1(\text{ID}_s) &= a_\Delta \\
(\tau_1(\text{ID}_a)\tau_1(\text{ID}_b) + \tau_1(\text{ID}_a)\tau_1(\text{ID}_c) + \ldots + \tau_1(\text{ID}_a)\tau_1(\text{ID}_s)) &+ \tau_1(\text{ID}_{\Delta-1})\tau_1(\text{ID}_s) &= a_{\Delta-1} \\
\ldots & \ldots \\
\tau_1(\text{ID}_a)\tau_1(\text{ID}_b)\tau_1(\text{ID}_c) \ldots \tau_1(\text{ID}_s) &= a_0 
\end{align*}
\]

The algorithm converts an attribute user list \( L \) by generating:

\[
\text{If } \begin{cases} \text{Att}_i \text{ is } + & : r'_i = t_i \\ \text{Att}_i \text{ is } - & : r'_i = t_{2i} \end{cases}
\]

\( \text{Adv} \)} Springer
Then set \( b = \sum_{\text{Att} \in L} r'_i \), and it computes based on \( b \):

\[
\begin{align*}
    b_m &= b^m \\
    b_{m-1} &= b^{m-1} \\
    b_{m-2} &= b^{m-2} \\
    &\vdots \\
    b_0 &= 1
\end{align*}
\]

The \( \vec{v} \) is produced as

\[
\vec{v} = (1_0, a_\Delta, a_{\Delta-1}, \ldots, a_0, b_m, \ldots, 1_m).
\]

Then it runs \( CT \leftarrow \text{IPE.Enc}(PK, \vec{v}, M) \), and output the ciphertext \( CT \).

**KeyGen**(*MSK, ID, W = (W_1 OR \ldots OR W_m)*): Suppose that a user joins the system with the a given user identity \( ID \) and the access structure \( W = (W_1 OR \ldots OR W_m) \), the algorithm inputs \((ID, W)\), and transforms them into a vector \( \vec{z} \) by generating:

It encodes \( ID \) by \( \tau_1(ID) = x_{ID} \in (\mathbb{Z}/p\mathbb{Z})^* \). Then, we compute \( x_{ID} \) as the one of the roots of polynomial degree \( n \):

\[
\begin{align*}
    a'_n &= x_{ID}^n \\
    a'_{n-1} &= x_{ID}^{n-1} \\
    &\vdots \\
    a'_0 &= 1
\end{align*}
\] (4)

Next, the access structure \( W \) is interpreted as:

\[
W = ((\text{AND}_{i \in \{1,\ldots,m\}} A_i) \text{ OR } \text{AND}_{i \in \{1,\ldots,m\}} A_i) \text{ OR } \ldots \text{ OR } (\text{AND}_{i \in \{1,\ldots,m\}} A_i)).
\]

Then the algorithms computes as follows:

Each \( W_i \), If \( \begin{cases} \text{Att} \_ j \text{ is } + : r_j = t_1 \_ i \text{,} \\ \text{Att} \_ j \text{ is } - : r_j = t_2 \_ i \end{cases} \)

Then set \( W_i = \sum_{\text{Att} \_ j \in W_i} r_j \).

Next apply the Viète’s formula as (2) to computes the whole access structure \( W \):

\[
\begin{align*}
    W_1 + W_2 + \ldots + W_m &= b'_{m-1} \\
    W_1W_2 + W_1W_3 + \ldots + W_{m-1}W_m &= b'_{m-2} \\
    &\vdots \\
    W_1W_2 \ldots W_m &= b'_0
\end{align*}
\] (5)

The \( \vec{z} \) is produced as

\[
\vec{z} = (a'_n, a'_{n-1}, \ldots, 1_n, 1_m, b'_{m-1}, \ldots, b'_0).
\]

Then it runs \( SK \leftarrow \text{IPE.KeyGen}(PK, \vec{z}, MSK) \), and output the secret key \( SK \).

**Decrypt**(SK, CT) the algorithms runs \( \text{IPE.Dec}( CT, SK) \) and outputs the message \( M \) iff \( <\vec{v}, \vec{z}> == 0 \).
Correctness for the vector $\mathbf{vec} = (1_{t_0}, a_{\Delta}, a_{\Delta-1}, \ldots, a_0, b_m, \ldots, b_0)$ corresponding to the set user indices $S$ and attribute list $L$ in the ciphertext $CT$ and the vector $\bar{z} = (a'_n, a'_{n-1}, \ldots, a_0, b'_m, \ldots, b'_0)$ corresponding to the secret key component $SK$ in the AKP-ABBE, we have:

$$
\sum_{i=0}^{n+m} v_i \cdot x_i = \sum_{i=0}^{n} v_i \cdot x_i + \sum_{i=n+1}^{n+m} v_i \cdot x_i
$$

$$
= \sum_{i=0}^{n} a_i \cdot x_i^i + \sum_{i=n+1}^{n+m} b_{i-n} \cdot b_{i-n}'
$$

$$
= (1 \cdot x_0^n + a_\Delta \cdot x_\Delta^n + \ldots + a_0 \cdot 1_n)
$$

$$
+ \left( \left( \sum_{\text{Att}_i \in L} r_i' \right)^m \cdot 1_m + \left( \sum_{\text{Att}_i \in L} r_i' \right)^{m-1} \cdot b_{m-1}' + \ldots + b_0' \right)
$$

$$
= (x_\Delta^n + [\tau_1 (ID_a) + \ldots + \tau_1 (ID_s)] \cdot x_\Delta^{n-1} + \ldots + [\tau_1 (ID_a) \ldots \tau_1 (ID_s)])
$$

$$
+ \left( \left( \sum_{\text{Att}_i \in L} r_i' \right)^m + \left( \sum_{\text{Att}_i \in L} r_i' \right)^{m-1} \cdot \left[ \sum_{i=1}^{m} \sum_{j=1}^{W_i} r_j \right]
$$

$$
+ \ldots + \left[ \prod_{i=1}^{m} \sum_{j=1}^{W_i} r_j \right] \right).
$$

If $\sum_{i=0}^{n+m} v_i \cdot x_i = 0$, the algorithm return $M$. This means that the ID user is belongs to the set of indices $S$, and the attribute list $L$ satisfies the user’s access structures $W$. Otherwise, the algorithms return $\bot$.

Theorem 1 Our AKP-ABBE scheme is secure under the standard assumption if the underlying IPE is secure under the standard assumption.

3.4 Generic construction of ACP-ABBE from IPE

The ACP-ABBE scheme is a dual form of AKP-ABBE.

3.4.1 Main scheme

Given an IPE scheme with four algorithms: (IPE.Setup, IPE.KeyGen, IPE.Enc, IPE.Dec), we construct an ACP-ABBE scheme with the corresponding four algorithms: Setup, KeyGen, Encrypt, Decrypt) as follows:

- **Setup($1^k$):** The algorithm chooses a suitable encoding $\tau_1$ sending each of the $n$ indices $ID \in \mathbb{N}$ onto an element $\tau_1 (ID) = x_1 \in (\mathbb{Z}/p\mathbb{Z})^*$, and choose $t_1, \ldots, t_\Delta$ randomly in $\mathbb{Z}_p$. It runs IPE.Setup($1^k, n + m$) with $m$ as the number of attributes to construct to access structure, and outputs public parameters $PK$ and a master key $MSK$.

- **Encrypt($PK, M, S, W = (W_1 \text{ OR } \ldots \text{ OR } W_m)$):** The algorithm inputs a user index set $S = \{ID_a, ID_B, ID_C, \ldots, ID_s\} \subseteq U$, and message $M$, the access structure $W = (W_1 \text{ OR } \ldots \text{ OR } W_m)$. The algorithm transforms $(S, W)$ into $\bar{v}$ as:
The user index set is input as \( S = (I_D^a, I_D^b, I_D^c, \ldots, I_D^s) \subseteq U \). We denote \( \Delta \) as the total number elements in set \( S \), then the algorithm applies the Viète's formula to compute:

\[
\begin{align*}
\tau_1(I_D^a) + \tau_1(I_D^b) + \tau_1(I_D^c) + \ldots + \tau_1(I_D^s) &= a_\Delta \\
(\tau_1(I_D^a)\tau_1(I_D^b) + \tau_1(I_D^a)\tau_1(I_D^c) + \ldots + \tau_1(I_D^a)\tau_1(I_D^s)) &+ \ldots \\
&\vdots \\
\tau_1(I_D^a)\tau_1(I_D^b)\tau_1(I_D^c)\ldots \tau_1(I_D^s) &= a_0
\end{align*}
\]

Next, the access structure \( W \) is interpreted as:

\[
W = ((\text{AND}_{i \in \{1, \ldots, m\}} \text{Att}_i) \text{ OR } (\text{AND}_{i \in \{1, \ldots, m\}} \text{Att}_i) \text{ OR } \ldots \text{ OR } (\text{AND}_{i \in \{1, \ldots, m\}} \text{Att}_i)).
\]

Then the algorithm computes as follows:

\[
\text{Each } W_i, \text{ If } \begin{cases} 
\text{Att}_j \text{ is } +: r_j = t_i, \\
\text{Att}_j \text{ is } -: r_j = t_{2i}
\end{cases}
\]

Then set \( W_i = \sum_{\text{Att}_j \in W_i} r_j \).

Next apply the Viète's formula as (2) to computes the whole access structure \( W \):

\[
\begin{align*}
W_1 + W_2 + \ldots + W_m &= b_{m-1} \\
W_1W_2 + W_1W_3 + \ldots + W_{m-1}W_m &= b_{m-2} \\
&\vdots \\
W_1W_2\ldots W_m &= b_0
\end{align*}
\]

Then it produces a vector:

\[
\vec{v} = (1_0, a_\Delta, a_{\Delta-1}, \ldots, a_0, 1_m, b_{m-1}', \ldots, b_0')
\]

Then it runs IPE.Enc(PK, \( \vec{v}, M \)), and output the ciphertext CT.

**KeyGen**(MSK, ID, L): Suppose that a user joins the system with the a given user identity ID and his attribute list L, the algorithm inputs (ID, L), and transforms them into a vector \( \vec{z} \) by generating:

It encodes ID by \( \tau_1(I_D) = x_{ID} \in (\mathbb{Z}/p\mathbb{Z})^* \). Then, we compute \( x_{ID} \) as the one of the roots of polynomial degree \( n \):

\[
\begin{align*}
\tau_1(I_D) &= a_n \\
\tau_1(I_D^a) &= a_{n-1} \\
\tau_1(I_D^b) &= a_{n-2} \\
&\vdots \\
\tau_1(I_D^s) &= a_0
\end{align*}
\]

The algorithm converts an attribute user list L by generating:

\[
\text{If } \begin{cases} 
\text{Att}_i \text{ is } +: r_i' = t_i, \\
\text{Att}_i \text{ is } -: r_i' = t_{2i}
\end{cases}
\]
Then set \( b = \sum_{\text{Att}i \in L} r_i' \), and it computes based on \( b \):

\[
\begin{align*}
    b_m &= b^m \\
    b_{m-1} &= b^{m-1} \\
    b_{m-2} &= b^{m-2} \\
    \cdots \\
    b_0 &= 1
\end{align*}
\]

We then produce a vector:

\( \vec{z} = (a_n', a_{n-1}', \ldots, b_m, \ldots, b_1) \)

Then it runs \( \text{IPE.KeyGen}(\text{PK}, \vec{z}, \text{MSK}) \), and output the secret key \( \text{SK} \).

**Decrypt** (CT, SK): the algorithm inputs the ciphertext CT and the user’s secret key SK, then it runs \( \text{IPE.Dec}(\text{CT}, \text{SK}) \) and outputs the message \( M \) iff \( \vec{v}, \vec{z} \neq 0 \). Otherwise, the algorithms the symbol \( \bot \).

**Correctness:** for the vector \( \vec{v} = (1, 0, a_\Delta, a_{\Delta-1}, \ldots, a_0, 1_m, b_m', \ldots, b_0) \) corresponding to the set user indices \( S \) and access structure \( W \) embedded in the ciphertext \( CT \) and the vector \( \vec{z} = (a_n', a_{n-1}', \ldots, b_m, \ldots, b_1) \) corresponding to the secret key component \( \text{SK} \) in the ACP-ABBE., we have:

\[
\sum_{i=0}^{n+m} v_i \cdot x_i = \sum_{i=0}^{n} v_i \cdot x_i + \sum_{i=n+1}^{n+m} v_i \cdot x_i \\
= \sum_{i=0}^{n} a_i \cdot x_i^\Delta + \sum_{i=n+1}^{n+m} b_i' \cdot x_{i-n} \\
= (1 \cdot x_0^\Delta + a_\Delta \cdot x_1^{n-1} + \ldots + a_0 \cdot x_n) \\
+ \left( \left( \sum_{\text{Att}i \in L} r_i' \right)^m \cdot 1_m + \left( \sum_{\text{Att}i \in L} r_i' \right)^{m-1} \cdot b_m' + \ldots + b_0 \right)
\]

If \( \sum_{i=0}^{n+m} v_i \cdot x_i = 0 \), the algorithm return \( M \). This means that the ID user is belongs to the set of indices \( S \), and the user attribute list \( L \) satisfies the access structures \( W \). Otherwise, the algorithms return \( \bot \).

**Construction of secret keys** We assume \( \sum_{\text{Att}i \in L} y_1 \neq \sum_{\text{Att}i \in L'} y_1 \) in both of AKP-ABBE and ACP-ABBE.

If there exist \( L \) and \( L' \) such that \( \sum_{\text{Att}i \in L} y_1 = \sum_{\text{Att}i \in L'} y_1 \), a user with attribute list \( L \) can decrypt a ciphertext associated with \( W \), where \( L' \not| W \) and \( L \models W \).

Hence the assumption holds with overwhelming probability:

\[
\frac{p(p-1)(p-(N-1))}{p^n} > \frac{(p-N+1)^N}{p^N} = \left( 1 - \frac{N-1}{p} \right)^N > 1 - \frac{N(N-1)}{p} > 1 - \frac{N^2}{p},
\]

where \( p \) is the prime number which chosen in the first step, \( N = \prod_{i=1}^{2^n} y_i \). If each secret key \( y_i \) is chosen at random from \( \mathbb{Z}_p \), then our assumption is natural. Then, the advantage of \( A \) in this game is defined as \( \text{Adv} \cdot (1 - \frac{N^2}{p}) \).
Theorem 2 Our ACP-ABBE scheme is secure under the standard assumption if the underlying IPE is secure under the standard assumption.

4 Security analysis

Our AKP-ABBE and ACP-ABBE utilize the IPE manner to achieve the hidden access structures. Indeed the access structure and the user index set are transformed into the vector. In this part, we choose AKP-ABBE to elaborate the security analysis. Hence, in order to prove that our AKP-ABBE scheme is access structure hiding, we apply the indistinguishability, in which the adversary cannot distinguish two vectors \( \vec{v} \) and \( \vec{x} \). These two vectors correspond to \((S_0^*, L_0^*)\) and \((S_1^*, L_1^*)\), respectively, which have been used to generate the two ciphertexts \( M_0 \) and \( M_1 \).

Based on these above games, we apply the security proof of [19] to our Theorems 1 and 2 directly. To prove the AKP-ABBE be secured in the indistinguishable chosen plaintext attack, we consider two cases \( M_0 = M_1 \) and \( M_0 \neq M_1 \):

- \( M_0 = M_1 \), we only consider the following game sequence from Game_1 to Game_5. In this case, we prove the property of attribute hiding.
- \( M_0 \neq M_1 \), we consider the whole proof from Game_0 to Game_6.

We then present a description of each game, where the challenge ciphertexts \( CT_1, \ldots, CT_6 \) are generated by the IPE’s encryption scheme:

- Game_0: The challenge ciphertext \( CT_0 \) is generated under \((\vec{v}, \vec{v})\) and \( M_0 \).
- Game_1: The challenge ciphertext \( CT_1 \) is generated under \((\vec{v}, \vec{v})\) and a random message \( R \).
- Game_2: The challenge ciphertext \( CT_2 \) is generated under \((\vec{v}, \vec{0})\) and a random message \( R \).
- Game_3: The challenge ciphertext \( CT_3 \) is generated under \((\vec{v}, \vec{x})\) and a random message \( R \).
- Game_4: The challenge ciphertext \( CT_4 \) is generated under \((\vec{0}, \vec{x})\) and a random message \( R \).
- Game_5: The challenge ciphertext \( CT_5 \) is generated under \((\vec{x}, \vec{x})\) and a random message \( R \).
- Game_6: The challenge ciphertext \( CT_6 \) is generated under \((\vec{x}, \vec{x})\) and message \( M_1 \).

PROOF Suppose that the adversary commits to the challenge user indices \( S_0^* = (ID_{0a}^*, ID_{0b}^*, ID_{0c}^*, \ldots, ID_{0s}^*) \) and \( S_1^* = (ID_{1a}^*, ID_{1b}^*, ID_{1c}^*, \ldots, ID_{1s}^*) \subseteq U \), and the target attribute sets \( L_0^* = (\text{Att}_{01}^*, \ldots, \text{Att}_{0m}^*) \) and \( L_1^* = (\text{Att}_{11}^*, \ldots, \text{Att}_{1m}^*) \) at the beginning of the game.

The \( \vec{v} \) is produced of \( S_0^* = (ID_{0a}^*, ID_{0b}^*, ID_{0c}^*, \ldots, ID_{0s}^*) \), and \( L_0^* = (\text{Att}_{01}^*, \ldots, \text{Att}_{0m}^*) \) by using (2), (3) from the original construction.

The \( \vec{x} \) is produced of \( S_1^* = (ID_{1a}^*, ID_{1b}^*, ID_{1c}^*, \ldots, ID_{1s}^*) \), and \( L_1^* = (\text{Att}_{11}^*, \ldots, \text{Att}_{1m}^*) \) by using (2), (3) from the original construction.

We also note that in the query phase the adversary is issued the SK corresponding to the access structure \( W \) and the user identity \( ID \). It is also considered that the SK is related to \( \vec{y} \), where he \( \vec{y} \) is produced of the access structure \( W \) and the user identity \( ID \) by using (4), (5) from the original construction.

We use the above sequence of hybrid games to prove that the adversary cannot win the original security game with the non-negligible security. We begin with game Game_0.

Indistinguishability between Game_0 and Game_4: If the adversary obtain the secret key SK corresponding to the access structure \( W \) and the user identity \( ID \) satisfying such that \( (L_0^* \models W \)
and \((ID \in S_0^a)\) (meanwhile \(<\vec{v}, \vec{y}\rangle = 0\), then the challenge ciphertext is generated correctly. We consider that the challenge ciphertext is distributed in Game_4.

On the other hand, if the adversary obtains the secret key \(SK\) with corresponding to the access structure \(W\) and the user identity \(ID\) where \((L_0^a \not\models W\) and \((ID \not\in S_0^a)\) (meanwhile \(<\vec{v}, \vec{y}\rangle = 0\), then the challenge ciphertext component \(C_m\) of IPE scheme is a random element in \(G\) regardless of the random choice, while the rest of the challenge ciphertext are generated in an original way. Then we consider that the challenge ciphertext is distributed in Game_1.

**Indistinguishability between Game_1 and Game_2**

If the adversary obtains the secret key \(SK\) with corresponding to the access structure \(W\) and the user identity \(ID\) where \((L_0^a \not\models W\) and \((ID \not\in S_0^a)\), or \((L_0^a \models W\) and \((ID \not\in S_0^a)\) or \((L_0^a \not\models W\) and \((ID \in S_0^a)\) (meanwhile \(<\vec{v}, \vec{y}\rangle = 0\), or \((L_0^a \models W\) and \((ID \in S_0^a)\) (meanwhile \(<\vec{v}, \vec{y}\rangle = 0\), then the challenge ciphertext is generated correctly. We consider that the challenge ciphertext is distributed in Game_1.

On the other hand, if the adversary obtains the secret key \(SK\) with corresponding to the access structure \(W\) and the user identity \(ID\) by relaxed generation, then the two challenge ciphertext components \(C_{3,i}\) and \(C_{4,i}\) are the random elements in \(G\) regardless of the random choice, while the rest of the challenge ciphertext is generated in an original way. Then we consider that the challenge ciphertext is distributed in Game_2.

**Indistinguishability between Game_2 and Game_3**

If the adversary obtain the secret key \(SK\) corresponding to the access structure \(W\) and the user identity \(ID\) satisfying such that \((L_0^a \models W\) and \((L_1^a \models W)\) and \((ID \in S_0^a)\) and \((ID \in S_1^a)\) (meanwhile \(<\vec{v}, \vec{y}\rangle = <\vec{x}, \vec{y}\rangle = 0\), then the challenge ciphertext is generated correctly. We consider that the challenge ciphertext is distributed in Game_2.

On the other hand, if the adversary did not obtain the secret key \(SK\) corresponding to the access structure \(W\) and the user identity \(ID\) satisfying the constrain of \((L_0^a \models W\) and \((L_1^a \models W)\) and \((ID \in S_0^a)\) and \((ID \in S_1^a)\) (meanwhile \(<\vec{v}, \vec{y}\rangle = <\vec{x}, \vec{y}\rangle = 0\), then the two challenge ciphertext components \(C_{3,i}\) and \(C_{4,i}\) are the random elements in \(G\) regardless of the random choice, while the rest of the challenge ciphertext are generated in an original way. Then we consider that the challenge ciphertext is distributed in Game_3.

Due to the symmetric observation, the rest of the proof is similar to the above proofs:

- the indistinguishability between Game_3 and Game_4 can be proved in the same way as for Game_2 and Game_3;
- the indistinguishability between Game_4 and Game_5 can be proved in the same way as for Game_1 and Game_2;
- the indistinguishability of Game_5 and Game_6 can be proved in the same way as for Game_0 and Game_1.

The ACP-ABBE is proved secure under standard assumption by the similar arguments of AKP-ABBE, where \(\vec{v}\) is produced of \(S_0^a = (ID_{0a}^1, ID_{0b}^1, ID_{0c}^1, ..., ID_{0i}^1)\), and \(W_0^a = (W_{01}^a, ..., W_{0m}^a)\) by using (4), (5) from the original construction. In addition, the \(\vec{x}\) is produced of \(S_1^a = (ID_{1a}^1, ID_{1b}^1, ID_{1c}^1, ..., ID_{1i}^1)\), and \(W_1^a = (W_{11}^a, ..., W_{1m}^a)\) by using (6), (7) from the original construction. It is also considered that the \(SK\) is related to \(\vec{y}\), where he \(\vec{y}\) is produced of the access structure \(W\) and the user identity \(ID\) by using (8), (9) from the original construction.

Our proposal utilizes the work of [26] as a building block to construct the AKP-ABBE and ACP-ABBE schemes. Inherently, the strategy of our security proof is also argued as [19, 26], in which we directly apply the DBDH and DLIN assumption as [26] to prove our AKP-ABBE and ACP-ABBE be secured in the standard assumption. Therefore, by underlying the
Table 4  Comparisons in AKP-ABBE

<table>
<thead>
<tr>
<th>KP-ABBE</th>
<th>CipherText</th>
<th>Decryption</th>
<th>Access structure</th>
<th>Attributes</th>
<th>Complexity assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>[2]</td>
<td>$O(k_{\text{max}} + r)$</td>
<td>$O(k_{\text{max}} + r)$</td>
<td>LSSS</td>
<td>+ Attributes</td>
<td>q-BDHE</td>
</tr>
<tr>
<td>[30]</td>
<td>$O(m + 2)$</td>
<td>$O(3)$</td>
<td>AND gates</td>
<td>$\pm$ w/ wildcard</td>
<td>q-BDHE</td>
</tr>
<tr>
<td>AKP-ABBE</td>
<td>$O(n + m)$</td>
<td>$O(n + m)$</td>
<td>OR/AND gates</td>
<td>$\pm$ Attributes w/ wildcard</td>
<td>DBDH, D-linear</td>
</tr>
</tbody>
</table>

secured IPE of [26] under the standard assumption, we conclude that our AKP-ABBE and ACP-ABBE schemes are secure under the standard assumption.

5 Extensions

We extend how our proposed scheme can also achieve the Anonymous ABBE, which access structure supports AND Gates with positive, negative, wildcard [means “don’t care” (i.e., both positive and negative attributes are accepted)]:

Firstly, we choose the suitable encoding $\tau_2$ sending each of the $m$ attributes $\text{Att} \in U$ onto an element $\tau_2(\text{Att}) = x_2 \in (\mathbb{Z}/p\mathbb{Z})^*$

\[
\begin{cases}
\text{Att} \text{ is } +: & b_{2i-1} = \tau_2(\text{Att}_i), b_{2i} = -1 \\
\text{Att} \text{ is } -: & b_{2i-1} = -\tau_2(\text{Att}_i), b_{2i} = -1 \\
\text{Att} \text{ is } *: & b_{2i-1} = 0, b_{2i} = 0
\end{cases}
\]  

(10)

then we generate the $\vec{v}$ as:

\[
\vec{v} = (b_1, \ldots, b_m)
\]

For an attribute user list $L$, it computes:

\[
\begin{cases}
\text{att} \text{ is } +: & b'_{2i-1} = 1; b'_{2i} = \tau_2(\text{att}_i) \\
\text{att} \text{ is } -: & b'_{2i-1} = 1; b'_{2i} = -\tau_2(\text{att}_i)
\end{cases}
\]

(11)

, then we generate the $\vec{z}$ as:

\[
\vec{v} = (b'_1, \ldots, b'_m)
\]

If $\langle \vec{v}, \vec{z} \rangle = 0$, we conclude that $L \models W$.

6 Comparisons

In this section, we give a comparison among ABBE schemes in Tables 5 and 4. The schemes are compared in terms of the order of the underlying group, ciphertext size, decryption cost, access structure, and complexity assumption. In the table, $N$—number of clauses in a policy, $M$—maximum number of attributes in the given clause, $k$—number of attributes for a given user, $r$—number of revoked users, $k_{\text{max}}$—maximum number of attributes in access structure, $n$—total of the user’s identity, $m$—number of universe attributes.

As can be seen in Tables 4 and 5, our encryption and decryption are linear depending on the size of the user’s indices and the size of the access structure. In fact, both two proposed

\[\copyright\ \text{Springer}\]
schemes implement the IPE’s encryption to produce the ciphertext, and invoke the IPE’s decryption to recover the message. In addition, the cost of IPE scheme relies on the input of vectors. Our access structures are designed with flexibility by employing both AND/OR gates with negative/positive attributes and wildcards. This idea is well-suitable in practice, where the architecture of access control always requires multiple authorizations. In terms of security proof, both KP-ABBE and CP-ABBE can be proved in the standard assumptions, such as DBDH and DLinear assumptions. Specifically, we highlight that our proposed ABBEs can achieve anonymity due to the inherence of attribute hiding from the IPE scheme. Therefore, our ABBE schemes achieve anonymity with multiple access structures.

7 Conclusion

This paper proposes two new constructions of Anonymous Attribute-Based Broadcast Encryption as AKP-ABBE and ACP-ABBE for complex access structure by considering the OR/AND Gates with positive, and negative attributes. We present our proposed schemes in generic constructions, achieving anonymity. We also proved the security of our schemes be secured in the standard model. One open problem is to construct our AKP/ACP-ABBE schemes that have constant ciphertext and secret key, and we leave it as our future work.

Acknowledgements This work is supported by the Commonwealth Cyber Initiative (CCI).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.
To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.
References

Anonymous attribute-based broadcast encryption


Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.