Attribute-hiding Predicate Encryption with Equality Test in Cloud Computing

Jianfei Sun*, Yangyang Bao*, Xuyun Nie*, Hu Xiong*†
*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
† Guangxi Colleges and Universities Key Laboratory of cloud computing and complex systems, Guilin University of Electronic Technology, Guilin 541004, China

Abstract—Public key encryption with equality test (known as PKE-ET) enables anyone to perform equivalence test between two messages encrypted under distinct public keys. Attribute-hiding predicate encryption is a paradigm for public key encryption that supports both attribute-hiding and fine-grained access control. In this paper, we first initialize the concept of attribute-hiding predicate encryption with equality test (shorten as AH-PE-ET) by incorporating the notions of PKE-ET and PE, and then propose a concrete AH-PE-ET scheme. Inheriting the merits of predicate encryption, versatile access control can be achieved such that the ciphertexts and the secret key are respectively associated with the descriptive attributes x and the boolean functions f and decryption can only be done if $f(x)$ returns *true*. In the AH-PE-ET scheme, one data receiver can calculate a trapdoor using his/her private key and delivers this trapdoor to an untrusted cloud server, who in turn compares the ciphertexts from this receiver with other receivers' ciphertexts. During the comparison, the information about the trapdoor as well as the attributes associated with the ciphertexts will not be disclosed to this cloud server. Furthermore, it is also proven to be selectively secure against the chosen plaintext attack in the standard model under the decisional bilinear Diffie-Hellman assumption. Finally, the theoretical performance analysis and experimental simulation indicate the feasibility and practicability of our suggested scheme.

Index Terms—Cloud computing, flexible data search, privacy-preservation, predicate encryption with equality test, standard model.

1 Introduction

With the advent of big data era, massive amount of data are collected and analyzed in a variety of application scenarios including smart city [1], [2], intelligent transportation system (ITS) [3], [4] and smart grid [5]. To provide useful information to the individual and the society, the storage and processing of mass data have become an imperative task. Fortunately, cloud computing is a novel computing paradigm which enables ubiquitous access to infinite storage and computing resources at the price of minimal management cost. By considering the promising potential of cloud computing, individuals and enterprises are more inclined to remotely store and process their data with the support of cloud computing recently [6], [7].

Despite of the tremendous benefits of cloud computing, it is prudent to rethink the traditional approaches to maintain data privacy, integrity and reliability because the cloud server is usually provided by the untrusted commercial organization or corporation [8], [9], [10], [11]. Encryption-then-outsourcing is a common method to ensure the confidentiality of the data stored in the cloud server [12]. Particularly, public key encryption (PKE) [13], [14] has been widely used to ensure the confidentiality of the outsourced data. To share sensitive data with specific users securely, a data holder is able to encrypt these data under the public key of the desired receiver and deliver the corresponding ciphertext to the cloud server. In this way, only the designated receiver can access the data by performing decryption with his/her own private key. However, the data owner needs to perform the public-key encryption algorithm several times if the data needs to be shared with multiple users. To provide one-to-many encryption, the primitive of attribute-based encryption (ABE) [15] was proposed by Sahai and Waters as the extension of normal PKE. In the ABE mechanism, user's secret key and the ciphertext are respectively labelled with the descriptive attributes and the access policy. The user can decrypt the ciphertext provided the attributes associated with this user satisfy the access policy related to the ciphertext. With the support of ABE, the secure and flexible data sharing can be easily achieved in the cloud computing.

Search Functionality: Nevertheless, the lack of the search functionality may impede the further adoption of standard ABE in practical applications. When a user intends to access an encrypted data that he/she is interested in, a possible approach is to download all ciphertexts under his/her own attribute set. However, this approach is impractical and inefficient because massive data are stored in the cloud server. So it is necessary for the cloud server to provide flexible search functionality over the encrypted data. To solve this problem, Zheng *et al.* [16] formulated the primitive of attribute-based encryption with keyword search (ABE-KS). In an ABE-KS system, a user first combines his/her own secret key with a keyword to create the "keyword trapdoor". Then, the cloud server is delegated the capability of search functionality to retrieve the desired data encrypted under the same access policy. During the search process, cloud server could not obtain any information including the data and keyword. Although the ABE-KS scheme can be seen as a fine-grained method to solve the search problem, it is not good enough to provide more flexible search functionality for ciphertexts encrypted under different access policies.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2843565, IEEE Access

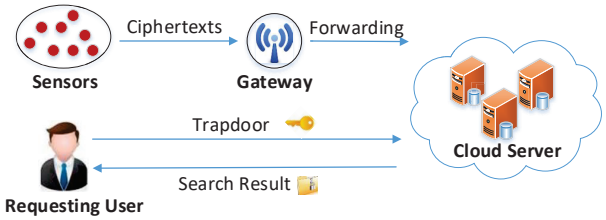


Fig. 1: An application of Cloud-assisted Internet of Things

Inspired by the concept of PKE-ET, a novel primitive named ABE with equality test (ABE-ET) [17] [18] was presented to support more flexible keyword search. In comparison to ABE-KS, ABE-ET can be used to judge whether the two ciphertexts encrypted under various access policies contain the same data.

Privacy Preserving: On the other hand, ABE-based access control mechanisms cannot preserve the privacy of user's attributes [19], [20]. To be more specific, although the outsourced data are stored in the form of ciphertext, the attributes associated with the ciphertext are clearly exposed to the malicious adversary. These attributes usually contain sensitive privacy information about users. This seriously disgraces the privacy of users if attribute disclosure happens. To address this problem, the primitive of predicate encryption (PE) [21], [22], [23], [24] was proposed to simultaneously support attribute hiding and fine-grained access control.

Gaps Between Search Functionality and Privacy Preserving: ABE-ET achieves the equality test for two ciphertexts under different access controls, but the two existing ABE-ET schemes [17] [18] can not support attribute hiding. This is because the sensitive attributes related to the ciphertexts are easily exposed to attackers or malicious users in ABE-ET. For another, as an extension of public key encryption, PE can achieve secure and flexible data sharing as well as attribute hiding. However, PE can not support search functionality over the encrypted data.

Guaranteeing the privacy of attributes and supporting flexible data search over ABE-type ciphertexts simultaneously have not been treated yet in literature. Motivated by the problem, we propose a novel primitive named attribute-hiding predicate encryption with equality test (AH-PE-ET), and then present a concrete AH-PE-ET scheme in this paper.

Fig. 1 presents an example of secure data sharing and searching mechanism based on the proposed AH-PE-ET scheme in the cloud-assisted Internet of Things (IoT). In the IoT system, all senors can first generate the encrypted sensing data $\mathcal{C}_0$ as well as the encrypted keywords $\mathcal{C}_i$ $(1 \leq i \leq n)$ with respect to the attribute set of IoT manager. And then, the sensors deliver the encrypted data and keywords through gateway to the cloud server. To retrieve the desired data, the requesting user employs his/her secret key to create a keyword trapdoor and delivers it to the cloud server. In this way, with the trapdoor of the requesting user, the cloud server can be delegated to test whether any of encrypted keyword $\mathcal{C}_i$ $(1 \leq i \leq n)$ equals to a specific keyword ciphertext $\mathcal{C}_{srch}$ encrypted under an attribute set of the requesting user. During the search process, the cloud server can not learn any information about data, trapdoor or attribute. Furthermore, the prohibitive computational overheads of keyword search for the requesting user are offloaded to the cloud server without interacting with the IoT manager.

1.1 Related Work

Public key Encryption with keyword search (PKE-KS) first was proposed by Boneh *et al.* in [28]. PKE-KS enables any user to achieve the equivalence test over the ciphertext encrypted with the same public key. Nevertheless, there is a restriction that user can not conduct search functionality for ciphertexts under various public keys. In order to address this problem, Yang *et al.* [27] initially presented public key encryption with equality test (PKE-ET). This primitive supports a flexible search functionality of two data encrypted with various public keys. Since then, many research results that focus on PKE-ET have been put forward in [16], [26], [31], [32], [33]. Despite the excellent performance of PKE-ET, there still exists the problem of public key certificate management, which severely limits the efficiency in practice. By introducing the concept of identity-based encryption (IBE) [29], [30] , Ma proposed the primitive of identity-based encryption with equality test (IBE-ET) [34]. Compared with PKE-ET, IBE-ET leverages the complicated public key certificate management. Inspired by Ma's achievement [34], fruitful research works concentrating on IBE-ET have been introduced in recent years. To resist the inner keywords guessing attack in single server scenario, Wu *et al.* [35] presented a dual server IBE-ET construction. Lee al et. [36] proposed a semi-generic construction of IBE-ET to provide a sophisticated security requirement that achieves the IND-ID-CCA security level. Recently, an efficient IBE-ET construction was introduced by Wu *et al.* [37] and proved to be more efficient and practical than Ma's work [34]. Combined KP-ABE with equality test, Zhu *et al.* [18] put forward a novel KP-ABE-ET scheme. In this scheme, the authors presented a mechanism that determines whether the same plaintext data encrypted under distinct attribute sets is contained in two different ciphertexts. Meanwhile, CP-ABE scheme with equality test (CP-ABE-ET) was suggested by Wang *et al.* [17] to achieve the functionality that checks the equivalence between two data under different access policies. Further, the CP-ABE-ET scheme was proven to be more efficient than Zhu's scheme [18]. Even so, all the previously formulated schemes did not consider the importance of attribute-hiding.

As a new paradigm of public key encryption, predicate encryption (PE) was first proposed by Katz *et al.* [21] to provide attribute hiding and payload hiding property. Besides, PE supports a flexible access control. Unfortunately, PE can not enable the authorized cloud server to perform a flexible equality test. To date, there has not been any proposed scheme that supports the functionality of equality test and sensitive-attribute privacy preservation at the same time. Therefore, in this paper, by incorporating AH-PE [22] with PKE-ET, an attribute-hiding predicate encryption with equality test (AH-PE-ET) scheme is proposed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2843565, IEEE Access

1.2 Our Contributions

In this paper, a concrete construction of attribute-hiding PE with equality test (AH-PE-ET) featured with rigorous security proof has been proposed. With our suggested AH-PE-ET scheme, it supports flexible search functionality on PE-type ciphertexts as well as privacy-preservation of trapdoor and attributes. The main contribution of this paper is summarized as three-fold as below.

1) We first incorporate the idea of PKE-ET into AH-PE-based setting to enjoy the best-of-the-two-worlds. Specifically, AH-PE-ET enables a semi-reliable entity (such as cloud server) to conduct an equivalence test on AH-PE-type ciphertexts under various access policies. Meanwhile, both the trapdoor and attribute can not be learned by the cloud server.
2) We then propose a concrete AH-PE-ET scheme, which features with constant pairing computations and minimal costs for decryption and test. Compared to the schemes in [17], [18], our scheme supports attribute-hiding as well as more expressive access control. Besides, our introduced scheme is rigorously proven to be selectively secure against the chosen plaintext attack under decisional bilinear Diffie-Hellman (DBDH) assumption in the standard model.
3) We finally give the theoretical analysis and experimental simulation that indicates the feasibility and practicability of our suggested scheme.

1.3 Organization

In section 2, some preliminaries are presented such as bilinear map, the underlying assumption, the formal definition of attribute-hiding predicate encryption with equality test (AH-PE-ET) and security model. In section 3, the concrete construction of AH-PE-ET scheme is shown. In section 4, we introduce the security proof of our proposed scheme under DBDH assumption in the standard model by slightly modifying the predicate encryption schemes [22] [38]. In section 5, the performance comparison of existing ABE-ETs and our AH-PE-ET scheme is described. A conclusion for this paper is summarized in section 6.

2 PRELIMINARIES

This section briefly presents a bilinear map, hardness assumption, formal definition, security model and system model which will be used throughout the whole paper.

2.1 Bilinear Map

Definition 1 (Bilinear Map). *$\mathbb{G}_1$, $\mathbb{G}_2$ are two bilinear groups of the same prime order p in case there exists a computable map* $\mathrm{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ *with the following properties:*

- $\mathbb{G}_1$ and $\mathbb{G}_2$ are two cyclic multiplicative groups.
- Bilinearity: For all $u, v \in \mathbb{G}_1$ and all $r, s \in \mathbb{Z}_p^*$, $e(u^r, v^s) = e(u, v)^{rs}$.
- Non-degeneracy: Given a generator g of the group $\mathbb{G}_1$, $e(g, g)$ is not equal to the identity element of the group $\mathbb{G}_2$.

2.2 Hardness Assumption

Definition 2 (DBDH). *Given one five-tuple $(g, g^u, g^v, g^w, \mathcal{D}) \in \mathbb{G}_1^4 \times \mathbb{G}_2$, where g is a generator of $\mathbb{G}_1$, u, v, w are randomly picked from $\mathbb{Z}_p^*$, the decisional bilinear Diffie-Hellman problem is to decide whether $\mathcal{D}$ is a random element or $\mathcal{D} = e(g, g)^{uvw}$. The advantage of an adversary $\mathcal{A}$ in solving the DBDH problem can be defined as follow:*

$$Adv^{DBDH}=|\mathsf{Pr}[\mathcal{D} = e(g, g)^{uvw}] - \mathsf{Pr}[\mathcal{D} = R_1 \xleftarrow{R} \mathbb{G}_2]|.$$

Definition 3 (Twin-DBDH). *Given one seven-tuple $(g, g^u, g^v, g^w\ , g^\vartheta, \mathcal{D}_1, \mathcal{D}_2) \in \mathbb{G}_1^5 \times \mathbb{G}_2^2$, where g is a generator of $\mathbb{G}_1$, u, v, w, ϑ are picked from $\mathbb{Z}_p^*$, the twin decisional bilinear Diffie-Hellman problem is to decide whether $\mathcal{D}_1, \mathcal{D}_2$ are two random elements or $\mathcal{D}_1 = e(g, g)^{uvw}, \mathcal{D}_2 = e(g, g)^{uw\vartheta}$. The advantage of an adversary $\mathcal{A}$ in solving the DBDH problem can be defined as follow:*

$$Adv^{DBDH}=|\mathsf{Pr}[\mathcal{D}_1 = e(g, g)^{uvw}, \mathcal{D}_2 = e(g, g)^{uw\vartheta}] - \mathsf{Pr}[\mathcal{D}_1 = R_1 \xleftarrow{R} \mathbb{G}_2, \mathcal{D}_2 = R_2 \xleftarrow{R} \mathbb{G}_2]|.$$

Generally, the twin-DBDH problem is weaker than the DBDH problem, but it is practically as hard as DBDH problem. The twin-DBDH problem is different from decisional twin bilinear Diffie-Hellman inversion problem (DBDHI) proposed by Chen *et. al* [40] and variations of the twin-DBDH problem proved by Zhu *et. al* [18].

Theorem 1. *The twin-DBDH problem is as hard as the DBDH problem.*

Proof: The proof of this theorem consists of two directions. First, we prove that twin-DBDH $\Leftarrow$ DBDH. We assume that there exists an algorithm $\mathcal{A}_1$ that solves the DBDH problem in a probabilistic polynomial time. Given an oracle $\mathcal{A}_1$, on input $(g, g^u, g^v, g^w\ , g^\vartheta, \mathcal{D}_1, \mathcal{D}_2) \in \mathbb{G}_1^5 \times \mathbb{G}_2^2$, it decides whether $\mathcal{D}_1, \mathcal{D}_2$ are two random elements of $\mathbb{G}_2$ or $\mathcal{D}_1 = e(g, g)^{uvw}, \mathcal{D}_2 = e(g, g)^{uw\vartheta}$.

Then, $\mathcal{A}_2$ constructs a five-tuple $(g, g^u, g^v, g^w, \mathcal{D})$. Then $\mathcal{A}_2$ calls $\mathcal{A}_1$. The $\mathcal{A}_1$ checks whether $\mathcal{D} = e(g, g)^{uvw}$ or $\mathcal{D}$ is a random value of $\mathbb{G}_2$.

If $\mathcal{A}_1$ outputs "Yes", this implies $\mathcal{D} = e(g, g)^{uvw}$. That is to say, it checks that the input is a DBDH instance. Thus, $\mathcal{A}_2$ responds "Yes".

If $\mathcal{A}_1$ outputs "No", this implies $\mathcal{D}$ is a random value of $\mathbb{G}_2$. Thus, $\mathcal{A}_2$ responds "No".

Next, we give the proof to present that DBDH $\Leftarrow$ twin-DBDH. To prove BDH $\Leftarrow$ twin-DBDH, we assume that there exists an algorithm $\mathcal{A}_1'$ that solves the twin-DBDH problem in a probabilistic polynomial time. We construct an algorithm $\mathcal{A}_2'$ as follows. On input $(g, g^u, g^v, g^w, \mathcal{D})$, it determines that whether $\mathcal{D} = e(g, g)^{uvw}$ or $\mathcal{D}$ is a random value of $\mathbb{G}_2$.

$\mathcal{A}_2'$ randomly picks ϑ, computes g^ϑ and constructs a seven-tuple $(g, g^u, g^v, g^w,\ g^\vartheta, \mathcal{D}_1, \mathcal{D}_2)$. Then, $\mathcal{A}_2'$ calls $\mathcal{A}_1'$. The $\mathcal{A}_1'$ checks whether $\mathcal{D}_1, \mathcal{D}_2$ are two random elements of $\mathbb{G}_2$ or $\mathcal{D}_1 = e(g, g)^{uvw}, \mathcal{D}_2 = e(g, g)^{uw\vartheta}$.

If $\mathcal{A}_1'$ outputs "Yes", this implies $\mathcal{D}_1 = e(g, g)^{uvw}, \mathcal{D}_2 = e(g, g)^{uw\vartheta}$. That is to say, it checks that the input is a double-DBDH instance. Thus, $\mathcal{A}_2'$ responds "Yes".

If $\mathcal{A}_1'$ outputs "No", this implies $\mathcal{D}_1, \mathcal{D}_2$ are two random elements. Thus, $\mathcal{A}_2'$ responds "No".

2.3 Formal definition of our PE-ET scheme

Our proposed AH-PE-ET scheme is comprised of six algorithms: Setup, KeyGen, Trapdoor, Encrypt, Decrypt and Test. These algorithms are defined as follows:

- Setup(λ): Produce the master secret key MSK, the public parameter PP based on a security parameter λ.
- KeyGen(PP, MSK, $\overrightarrow{x}$): Create the decryption secret key DSK for users based on the public parameter PP, the master key MSK and a predicate vector $\overrightarrow{x}$.
- Trapdoor(PP, DSK, $\overrightarrow{x}$): Generate the trapdoor TD for users based on the public parameter PP, the decryption key DSK and an attribute vector $\overrightarrow{x}$.
- Encrypt(PP, M, $\overrightarrow{y}$): Produce the ciphertext CT based on the public parameter PP, a plaintext message M and the predefined attribute vector $\overrightarrow{y}$.
- Decrypt(CT,DSK): Decipher the ciphertext CT using the decryption secret key DSK.
- Test($\mathsf{CT_A}$,$\mathsf{TD_A}$, $\mathsf{CT_B}$,$\mathsf{TD_B}$): Decide whether M_A in $\mathsf{CT_A}$ is the same with M_B in $\mathsf{CT_B}$ using the trapdoor $\mathsf{TD_A}$ and the trapdoor $\mathsf{TD_B}$.

2.4 Security Model

Definition 4. *The proposed AH-PE-ET scheme is selective chosen-plaintext security via the following game between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$.*

- **Init**: A challenge attribute vector $\overrightarrow{y}$ is picked by the adversary $\mathcal{A}$.
- **Setup**: The security parameter λ first is taken as input and then the Setup algorithm is executed by $\mathcal{B}$ to produce the master key MSK and the public parameter PP which is delivered to the adversary $\mathcal{A}$.
- **Phase 1&2**: The adversary $\mathcal{A}$ chooses a predicate vector $\overrightarrow{x}$ and makes secret key queries & trapdoor queries to generate a decryption secret key DSK and a trapdoor TD. After receiving the vector $\overrightarrow{x}$, such that $\langle \overrightarrow{x}, \overrightarrow{y} \rangle \neq 0$, $\mathcal{B}$ creates corresponding secret key DSK and corresponding trapdoor TD for the adversary $\mathcal{A}$.
- **Challenge**: After obtaining two messages M_0 and M_1 with equal length from the adversary $\mathcal{A}$, the challenger $\mathcal{B}$ replies the adversary $\mathcal{A}$ with the challenge ciphertext CT by running Encrypt(PP, $\overrightarrow{y}$, M_β).
- **Guess**: The adversary $\mathcal{A}$ outputs a guess $\beta' \in \{0, 1\}$ for β and wins the game if $\beta = \beta'$.

3 Concrete Construction

Our formulated AH-PE-ET scheme comprises six different algorithms: Setup, KeyGen, Trapdoor, Encrypt, Decrypt and Test. The detailed descriptions of our construction are elaborated as follows.

Setup(λ): Taking as input the security parameter λ, this algorithm generates the master public parameter PP and the master secret key MSK as follows:

1) Choose a bilinear map group $\mathbb{BM} = (\mathbb{G}_1, \mathbb{G}_2, p, g, e)$.
2) Select two different hash functions $\mathcal{H}_1$ and $\mathcal{H}_2$, where $\mathcal{H}_1 : \mathbb{G}_2 \to \mathbb{G}_1 \times \mathbb{Z}_p^*$, $\mathcal{H}_2 : \mathbb{G}_2 \to \mathbb{G}_1$.
3) Pick $\gamma, \theta, \sigma, \alpha_1, \ldots, \alpha_n \in \mathbb{Z}_p^*$ randomly and compute

$$g_0 = g^\gamma, g_1 = g^{\alpha_1}, \ldots, g_n = g^{\alpha_n},$$

$$u = e(g, g)^\sigma, v = e(g, g)^\theta.$$

4) Return PP $= (\mathbb{BM}, g_0, g_1, \ldots, g_n, u, v)$ and keep MSK $= (\alpha_1, \ldots, \alpha_n, \gamma, \theta, \sigma)$.

KeyGen(PP, MSK, $\overrightarrow{x}$): Taking as input the public parameter PP, the master secret key MSK and a predicate vector $\overrightarrow{x} = (x_1, \ldots, x_n)$, this algorithm produces the decryption secret key DSK as follows:

1) Select $s_1, s_2, r_1 \in \mathbb{Z}_p^*$, and compute

$$sk_1 = g^\theta \prod_{i=1}^{n} (g_i)^{x_i s_1} g_0^{r_1}, sk_2 = g^{s_1}, sk_3 = g^{r_1},$$

$$sk_1' = g^\sigma \prod_{i=1}^{n} (g_i)^{x_i s_2} g_0^{r_1}, sk_2' = g^{s_2}.$$

2) Set DSK $= (sk_1, sk_2, sk_3, sk_1', sk_2', x_1, \ldots, x_n)$.

Trapdoor(PP, DSK, $\overrightarrow{x}$): Taking as input the public parameter PP, the decryption secret key DSK, and the predicate vector $\overrightarrow{x}$, this algorithm produces trapdoor TD as follows:

1) Set $td_1 = sk_1', td_2 = sk_2', td_3 = sk_3$.
2) Output TD $= (td_1, td_2, td_3, x_1, \ldots, x_n)$.

Encrypt(PP, M, $\overrightarrow{y}$): Taking as input the public parameter PP, a plaintext message $M \in \mathbb{G}_1$ and an attribute vector $\overrightarrow{y} = (y_1, \ldots, y_n)$, this algorithm creates the ciphertext CT as follows:

1) Choose $t, z, \tau \in \mathbb{Z}_p^*$ and compute

$$C_M = M \| z \oplus \mathcal{H}_1(v^t), C_M' = M^z \cdot \mathcal{H}_2(u^t), C_0 = g^t,$$

$$C_0' = g_0^t, C_0'' = g^z, C_i = g^{y_i \tau} (g_i)^t,$$

where i varies from 1 to n.

2) Return CT $= (C_M, C_M', C_0, C_0', C_0'', C_i)$.

Decrypt(CT,SK): Taking as input the ciphertext CT, the decryption secret key DSK, this algorithm recovers the plaintext message M by executing the following steps:

1) Compute

$$V = e(\prod_{i=1}^{n} (C_i)^{x_i}, sk_2)^{-1} \cdot e(C_0, sk_1) \cdot e(C_0', sk_3)^{-1},$$

$$U = e(\prod_{i=1}^{n} (C_i)^{x_i}, sk_2')^{-1} \cdot e(C_0, sk_1') \cdot e(C_0', sk_3)^{-1}.$$

2) If $\sum_{i=1}^{n} x_i y_i = 0$, calculate $M \| z = C_M \oplus \mathcal{H}_1(V)$.
3) If $C_0'' = g^z$ and $C_M' / M^z = \mathcal{H}_2(U)$, the plaintext message M can be recovered.

Test($\mathsf{CT_A}$,$\mathsf{TD_A}$,$\mathsf{CT_B}$,$\mathsf{TD_B}$): Taking as input A's ciphertext $\mathsf{CT_A}$, A's trapdoor $\mathsf{TD_A}$ and B's ciphertext $\mathsf{CT_B}$, B's trapdoor $\mathsf{TD_B}$, this algorithm computes as follows to decide whether $M_A = M_B$:

$$Q_A = \frac{C_{M,A}'}{e(\prod_{i=1}^{n} (C_{i,A})^{x_{i,A}}, td_{2,A})^{-1} e(C_{0,A}, td_{1,A}) e(C_{0,A}', td_{3,A})^{-1}},$$

$$Q_B = \frac{C_{M,B}'}{e(\prod_{i=1}^{n} (C_{i,B})^{x_{i,B}}, td_{2,B})^{-1} e(C_{0,B}, td_{1,B}) e(C_{0,B}', td_{3,B})^{-1}},$$

and returns 1 such that $e(C_{0,A}'', Q_B) = e(Q_A, C_{0,B}'')$. Otherwise, it outputs 0. Note that users who can recover the plaintext are allowed to deliver the trapdoor to the third party.

4 SECURITY PROOF

Theorem 2. *The proposed AH-PE-ET scheme is correct.*

Proof: The above theorem is equivalent to the correctness of both the Decrypt function and the Test function. For the decryption, the ciphertexts and the secret key are respectively associated with the access vector $\overrightarrow{y}$ and the descriptive attribute vector $\overrightarrow{x}$, users can perform decryption successfully if the descriptive attribute vector matches the access vector. That is to say, the inner product $\langle \overrightarrow{x}, \overrightarrow{y} \rangle = 0$. For the equivalence test, after receiving the submitted trapdoors of two different users, the cloud server that is on behalf of users can execute Test algorithm to conduct the equivalence test on various ciphertexts. We can see that:

$$
\begin{aligned}
V &= e(\prod_{i=1}^{n}(C_i)^{x_i}, sk_2)^{-1} \cdot e(C_0, sk_1) \cdot e(C_0', sk_3)^{-1} \\
&= e(\prod_{i=1}^{n}(g^{y_i\tau} \cdot g^{\alpha_i t})^{x_i}, g^{s_1})^{-1} \cdot e(g^t, g^{\theta} \prod_{i=1}^{n}(g^{\alpha_i})^{x_i s_1} g_0^{r_1}) \cdot \\
&\quad e(g_0^t, g^{r_1})^{-1} \\
&= e(\prod_{i=1}^{n} g^{x_i y_i \tau}, g^{s_1})^{-1} \cdot e(\prod_{i=1}^{n} g^{\alpha_i x_i t}, g^{s_1})^{-1} \cdot \\
&\quad e(g^t, g^{\theta}) \cdot e(g^t, \prod_{i=1}^{n} g^{\alpha_i x_i s_1}) \cdot e(g^t, g_0^{r_1}) \cdot e(g_0^t, g^{r_1})^{-1} \\
&= e(g,g)^{-\tau s_1 \sum_{i=1}^{n} x_i y_i} \cdot e(g,g)^{\theta t}. \\
U &= e(\prod_{i=1}^{n}(C_i)^{x_i}, sk_2')^{-1} \cdot e(C_0, sk_1') \cdot e(C_0', sk_3)^{-1} \\
&= e(\prod_{i=1}^{n}(g^{y_i\tau} \cdot g^{\alpha_i t})^{x_i}, g^{s_2})^{-1} \cdot e(g^t, g^{\sigma} \prod_{i=1}^{n}(g^{\alpha_i})^{x_i s_2} g_0^{r_1}) \cdot \\
&\quad e(g_0^t, g^{r_1})^{-1} \\
&= e(\prod_{i=1}^{n} g^{x_i y_i \tau}, g^{s_2})^{-1} \cdot e(\prod_{i=1}^{n} g^{\alpha_i x_i t}, g^{s_2})^{-1} \cdot \\
&\quad e(g^t, g^{\sigma}) \cdot e(g^t, \prod_{i=1}^{n} g^{\alpha_i x_i s_2}) \cdot e(g^t, g_0^{r_1}) \cdot e(g_0^t, g^{r_1})^{-1} \\
&= e(g,g)^{-\tau s_2 \sum_{i=1}^{n} x_i y_i} \cdot e(g,g)^{\sigma t}.
\end{aligned}
$$

If $\sum_{i=1}^{n} x_i y_i = 0$, then $M \| z = C_M \oplus \mathcal{H}_1(V)$. Additionally, if $C_0^{''} = g^z$ and $C_M'/M^z = \mathcal{H}_2(U)$, the plaintext message M can be recovered. Since the correctness of Test is similar to Decrypt, so here we omit it.

Theorem 3. *The proposed AH-PE-ET scheme is selectively CPA-secure in the standard model under the twin DBDH problem.*

Proof: Assume that our AH-PE-ET scheme can be broken by the adversary $\mathcal{A}$ with non-negligible advantage ϵ, then another algorithm $\mathcal{B}$ can be constructed to solve the twin DBDH problem by interacting games with $\mathcal{A}$ with non-negligible advantage. Based on input a seven-tuple $(g, g^a, g^b, g^c, g^d, T_1, T_2) \in \mathbb{G}_1^5 \times \mathbb{G}_2^2$, $\mathcal{B}$ calls $\mathcal{A}$ and simulates the game to determine whether $T_1 = e(g,g)^{abc}, T_2 = e(g,g)^{acd}$ or T_1, T_2 are two random elements of $\mathbb{G}_2$.

Init: In this phase, the challenge attribute vectors $\overrightarrow{y} = (y_1, \ldots, y_n)$ picked by the adversary $\mathcal{A}$.

Setup: Algorithm $\mathcal{B}$ produces the public parameter PP and the master key MSK by randomly picking $\gamma', \varsigma, \alpha_1', \ldots, \alpha_n' \in \mathbb{Z}_p^*$, two hash functions $\mathcal{H}_1 : \mathbb{G}_2 \to \mathbb{G}_1 \times \mathbb{Z}_p^*$, $\mathcal{H}_2 : \mathbb{G}_2 \to \mathbb{G}_1$ as follows:

$$
\begin{aligned}
\overrightarrow{y_\eta}, g_0 = g^{\gamma'}, g_1 = (g^a)^{-\varsigma y_1} g^{\alpha_1'} = g^{-a\varsigma y_1 + \alpha_1'}, \\
\ldots, g_n = (g^a)^{-\varsigma y_n} g^{\alpha_n'} = g^{-a\varsigma y_n + \alpha_n'}, \\
u = e(g^a, g^d) = e(g,g)^{\sigma}, v = e(g^a, g^b) = e(g,g)^{\theta}.
\end{aligned}
$$

Which implies that

$$
\begin{aligned}
\overrightarrow{y_\eta}, \theta = ab, \sigma = ad, \gamma = \gamma', \\
\alpha_1 = -a\varsigma y_1 + \alpha_1', \ldots, \alpha_n = -a\varsigma y_n + \alpha_n'.
\end{aligned}
$$

After that, $\mathcal{B}$ delivers $\mathsf{PP} = (g, g_0, g_1, \ldots, g_n, u, v\ \mathcal{H}_1, \mathcal{H}_2)$ to the adversary $\mathcal{A}$ and keeps $\mathsf{MSK} = (\alpha_1, \ldots, \alpha_n)$ securely under his/her control.

Phase 1&2: After receiving the query $\overrightarrow{x} = (x_1, \ldots, x_n)$ such that $\langle \overrightarrow{x}, \overrightarrow{y} \rangle \neq 0$, $\mathcal{B}$ creates corresponding decryption secret key for $\mathcal{A}$. Specifically, $\mathcal{B}$ randomly selects $s_1', s_2', r_1' \in \mathbb{Z}_p^*$ and defines the decryption secret key $\mathsf{DSK} = (sk_1, sk_2, sk_3, sk_1', sk_2', x_1, \ldots, x_n)$ as follows:

$$
\begin{aligned}
sk_1 = \prod_{i=1}^{n}((g^a)^{-\varsigma y_i} g^{\alpha_i'})^{x_i s_1'} \cdot (g^b)^{\frac{\alpha_i' x_i}{\varsigma V}} (g^{\gamma'})^{r_1'}, \\
sk_2 = g^{s_1'} (g^b)^{\frac{1}{\varsigma V}}, \\
sk_3 = g^{r_1'}, \\
sk_1' = \prod_{i=1}^{n}((g^a)^{-\varsigma y_i} g^{\alpha_i'})^{x_i s_2'} \cdot (g^d)^{\frac{\alpha_i' x_i}{\varsigma V}} (g^{\gamma'})^{r_1'}, \\
sk_2' = g^{s_2'} (g^d)^{\frac{1}{\varsigma V}}.
\end{aligned}
$$

Here $V = \langle \overrightarrow{x}, \overrightarrow{y} \rangle$. To clearly understand how to create the DSK, we denote $s_1 = s_1' + \frac{b}{\varsigma V}, r_1 = r_1'$. After that, we calculate

$$
\begin{aligned}
sk_1 &= \prod_{i=1}^{n}((g^a)^{-\varsigma y_i} g^{\alpha_i'})^{x_i s_1'} \cdot (g^b)^{\frac{\alpha_i' x_i}{\varsigma V}} \cdot (g^{\gamma'})^{r_1'} \\
&= \prod_{i=1}^{n} g^{-a x_i y_i \varsigma s_1'} g^{-ab\varsigma x_i y_i \frac{1}{\varsigma V}} g^{ab\varsigma x_i y_i \frac{1}{\varsigma V}} g^{\alpha_i' x_i s_1'} g^{\frac{b\alpha_i' x_i'}{\varsigma V}} (g^{\gamma'})^{r_1'} \\
&= \prod_{i=1}^{n} (g^{-a\varsigma y_i})^{x_i (s_1' + \frac{b}{\varsigma V})} (g^{\alpha_i'})^{x_i (s_1' + \frac{b}{\varsigma V})} g^{ab\varsigma x_i y_i \frac{1}{\varsigma V}} (g^{\gamma'})^{r_1'} \\
&= g^{ab} \prod_{i=1}^{n} (g^{-a\varsigma y_i} g^{\alpha_i'})^{x_i (s_1' + \frac{b}{\varsigma V})} (g^{\gamma'})^{r_1'} \\
&= g^{\theta} \prod_{i=1}^{n} (g_i)^{x_i s_1} g_0^{r_1},
\end{aligned}
$$

where $g^{ab\varsigma x_i y_i \frac{1}{\varsigma V}} = g^{ab}$.

In above similar derivation process, $\mathcal{B}$ also chooses $s_2' \in \mathbb{Z}_p^*$ and computes

$$
\begin{aligned}
sk_1' &= \prod_{i=1}^{n}((g^a)^{-\varsigma y_i} g^{\alpha_i'})^{x_i s_2'} \cdot (g^d)^{\frac{\alpha_i' x_i}{\varsigma V}} \\
&= g^{\sigma} \prod_{i=1}^{n} (g_i)^{x_i s_2},
\end{aligned}
$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2843565, IEEE Access

where $s_2 = s_2' + \frac{d}{\varsigma V}$.

As aforementioned computations, the decryption secret key $\mathsf{DSK} = (sk_1, sk_2, sk_3, sk_1', sk_2', x_1, \ldots, x_n)$ can be represented as

$$\begin{aligned} sk_1 &= g^{\theta} \prod_{i=1}^{n} (g_i)^{x_i s_1} g_0^{r_1}, \\ sk_2 &= g^{s_1}, \\ sk_3 &= g^{r_1}, \\ sk_1' &= g^{\sigma} \prod_{i=1}^{n} (g_i)^{x_i s_2} g_0^{r_1}, \\ sk_2' &= g^{s_2}. \end{aligned}$$

Thus, the trapdoor $\mathsf{TD} = (td_1, td_2, td_3, x_1, \ldots, x_n)$ can be also created as follows:

$$\begin{aligned} td_1 &= sk_1', \\ td_2 &= sk_2', \\ td_3 &= sk_3. \end{aligned}$$

Challenge: After obtaining two messages M_0 and M_1 with equal length from $\mathcal{A}$, the algorithm $\mathcal{B}$ replies $\mathcal{A}$ with the challenge ciphertext $\mathsf{CT} = (M_\beta \| z' \oplus \mathcal{H}_1(T_1), M_\beta^{z'} \cdot \mathcal{H}_2(T_2), g^c, (g^c)^{\gamma}, g^{z'}, (g^c)^{\alpha_1'}, \ldots, (g^c)^{\alpha_n'})$. To be more specific, let $t = c, z = z', \tau = a\varsigma c$, we can get $\mathsf{CT} = (M_\beta \| z \oplus \mathcal{H}_1(T_1), M_\beta^z \cdot \mathcal{H}_2(T_2), g^t, (g^{\gamma})^t, g^z, g^{y_1 \tau}(g_1)^t, \ldots, g^{y_n \tau}(g_n)^t)$. Note that $(g^c)^{\alpha_i'} = g^{\alpha_i' c - a\varsigma y_i c + a\varsigma y_i c} = g^{y_i \tau}(g_i)^t$.

Guess: The adversary $\mathcal{A}$ replies a guess $\beta' \in \{0, 1\}$ on β. If $\beta = \beta'$, then $\mathcal{B}$ returns 1 to guess $T_1 = e(g, g)^{abc}, T_2 = e(g, g)^{acd}$. Otherwise, $\mathcal{B}$ returns 0 to guess that T_1, T_2 are two random numbers of $\mathbb{G}_2$.

5 Comparison Evaluation

This section presents the comparisons of the existing ABE-ETs and our proposed AH-PE-ET in terms of storage and communication cost, computation cost, functionality, security level and hardness assumption.

In Table 1, the third, fourth, fifth, sixth rows present the comparisons of storage and communication overheads for public parameter, ciphertext, decryption secret key, trapdoor, respectively. The seventh, eighth, ninth rows show the comparisons of computation costs for encryption algorithm, decryption algorithm, test algorithm, respectively. The tenth, eleventh, twelfth and thirteenth rows describe whether the table-listed schemes support the functionalities of flexible access control, keyword search, equality test and attribute hiding, respectively. The fourteenth row represents whether the security proof can be rigorously proven in the standard model. The fifteenth row suggests which security level can be reached for the listed schemes. The final row describes the hardness assumption that the security proof needs to base upon.

Here, we assume the number of user attributes m equals to the dimension n of attribute vector. Consider an extreme case that the access policy contains n wildcards, we can get the number of wildcards equals to the number of user attributes. In this way, we can know $n' = m = n \leq m'$. As presented in Table 1, we can find the public parameter size in [17], [18] is linear with the amount of attributes in the system, whereas our scheme linearly grows with the dimension of attribute vector. According to the above-mentioned premises, it is easy to observe that the public parameter size of our scheme is smaller than that of the schemes [17], [18]. For the ciphertext size and trapdoor size, the scheme [18] and our scheme follow a linear relationship with the number of attributes and the dimension of attribute vector respectively, and the scheme [17] has constant size on both the ciphertext and trapdoor. For the decryption secret key size, the sizes in [17], [18] and our scheme increase linearly with the number of attributes, the amount of wildcards and the dimension of attribute vector, respectively. We can easily observe that the other two schemes need $2n$ elements in $\mathbb{G}_1$ and $6n'$ elements in $\mathbb{G}_2$ respectively, whereas our scheme only has 3 elements in $\mathbb{G}_1$. Obviously, our scheme has a smaller size of decryption secret key.

For the computational complexities of encryption, decryption and test algorithms, Zhu *et al.*'s scheme [18] needs $(2m' + 3)$ exponentiation computations in $\mathbb{G}_1$ for encryption, $2m$ pairing computations and $(2m + 2)$ exponentiation computations in $\mathbb{G}_1$ for decryption, and $2m'$ pairing computations and $2m'$ exponentiation computations in $\mathbb{G}_1$ for test. Wang *et. al*'s scheme [17] needs $(2m' + 11)$ exponentiation computations in $\mathbb{G}_1$ for encryption, 12 pairing computations, $(8n' + 1)$ exponentiation computations in $\mathbb{G}_1$ and 4 exponentiation computations in $\mathbb{G}_2$ for decryption, and 14 pairing computations, $8n'$ exponentiation computations in $\mathbb{G}_1$ and 4 exponentiation computations in $\mathbb{G}_1$ for test. Our scheme needs $(2n + 4)$ exponentiation computations in $\mathbb{G}_1$ and 2 exponentiation computations in $\mathbb{G}_2$ for encryption, 3 pairing computations and n exponentiation computations in $\mathbb{G}_1$ for decryption, 5 pairing computations and n exponentiation computations in $\mathbb{G}_1$ for test. Obviously, compared to the schemes [17], [18], our scheme completely outperforms these two schemes on both decryption and test algorithms and has a similar computation overhead on encryption algorithm.

With regards to the functionality of all the schemes in the listed table, the other two schemes and our proposed scheme in Table 1 support flexible access control, keyword search and equality test on ciphertexts. However, both Zhu *et al.*'s scheme [18] and Wang *et. al*'s scheme [17] cannot support the privacy-preserving of attributes while our formulated scheme could achieve this property. As depicted in Table 1, the scheme [18] can be one-way secure against chosen-ciphertext attack (OW-CCA) under the BDH assumption in the random oracle model. The scheme [17] can achieve an indistinguishability under chosen plaintext attack security (IND-CPA) under the DL assumption without the random oracle model. Our scheme could also achieve the IND-CPA security under the DBDH assumption in the standard model.

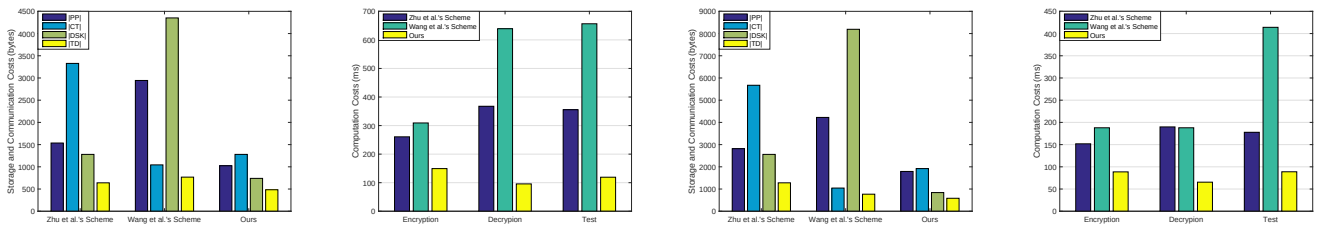
In order to show the practical performance comparisons, we simulate the above-listed two ABE-ETs schemes and our AH-PE-ET scheme based on ABE toolkit and Pairing Based Cryptography (PBC) library [39]. Specifically, these experiments are simulated in C++ on windows 10 64 bits operation system with Inter(R) Core(TM) i7-7700 CPU @3.60 GHz and 8GB RAM. To attain the 80-bit security level, our simulations is tested based on a 20-byte elliptic curve group constructed on the curve $y^2 = x^3 + x$ over a 64-byte finite

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2843565, IEEE Access

TABLE 1: Comparisons of the existing ABE-ETs and our AH-PE-ET

		KP-ABE-ET (Zhu et al. [18])	CP-ABE-ET (Wang et al. [17])	AH-PE-ET (Our Scheme)
Size of	PP	m'$\|\mathbb{G}_1\|$+2$\|\mathbb{G}_2\|$	(m'+7)$\|\mathbb{G}_1\| + 6\|\mathbb{G}_2\|$	n$\|\mathbb{G}_1\| + 2\|\mathbb{G}_2\|$
	CT	(2m'+4)$\|\mathbb{G}_1\|$+2$\|\mathbb{Z}_p^*\|$	8$\|\mathbb{G}_1\|$+$\|\mathbb{Z}_p^*\|$	(n+3)$\|\mathbb{G}_1\|$+2$\|\mathbb{G}_2\|$
	DSK	2m$\|\mathbb{G}_1\|$	$4\|\mathbb{G}_1\|$+$6n'\|\mathbb{G}_2\|$	5$\|\mathbb{G}_1\|$+n$\|\mathbb{Z}_p^*\|$
	TD	$m\|\mathbb{G}_1\|$	6$\|\mathbb{G}_1\|$	3$\|\mathbb{G}_1\|$+n$\|\mathbb{Z}_p^*\|$
Comp of	Enc	(2m'+3)Exp_1	(2m'+11)Exp_1	(2n+4)Exp_1+2Exp_2
	Dec	2mPairing+(2m+2)Exp_1	12Pairing+(8n'+1)Exp_1+4Exp_2	3Pairing+nExp_1
	Test	2mPairing+2mExp_1	14Pairing+8n'Exp_1+4Exp_2	5Pairing+nExp_1
Fun	FAC	✓	✓	✓
	KS	✓	✓	✓
	ET	✓	✓	✓
	AH	×	×	✓
SM		×	✓	✓
SL		OW-CCA	IND-CPA	IND-CPA
Assumption		BDH	DLIN	DBDH

‡ KP-ABE-ET: Key-policy attribute-based encryption with equality test, CP-ABE-ET: Ciphertext-policy attribute-based encryption with equality test, AH-PE-ET: Attribute-hiding predicate encryption with equality test, Comp: Computational cost, Fuc: Functionality, SL: Security level, SM: Standard model, Enc: Encryption algorithm, Dec: Decryption algorithm, Test: Test algorithm, PP: Public parameter, CT: Ciphertext, DSK: Decryption secret key, TD: Trapdoor, KS: Keyword search, ET: Equality test, FAC: Flexible access control, AH: Attribute hiding, Exp_1: Exponentiation computation in group $\mathbb{G}_1$, Exp_2: Exponentiation computation in group $\mathbb{G}_2$, Pairing: Pairing, $|\mathbb{G}_1|$, $|\mathbb{G}_2|$: Size of one element in $\mathbb{G}_1$, $\mathbb{G}_2$, $|\mathbb{Z}_p^*|$: Size of a random number in $\mathbb{Z}_p^*$, m': the number of attributes in access policy, m: the number of attribute in attribute set, n': the number of wildcards in access policy, n: the dimension of attribute vector, OW-CCA: One-Way secure against Chosen-Ciphertext Attack, IND-CPA: Indistinguishability under Chosen Plaintext Attack, BDH: Bilinear Diffie-Hellman, DLIN: Decisional Linear, DBDH: Decisional bilinear Diffie-Hellman.



(a) n'=m=n=5, m'=10. (b) n'=m=n=5, m'=10. (c) n'=m=n=10, m'=20. (d) n'=m=n=10, m'=20.

Fig. 2: (a) & (c): Comparison of the storage and communication costs. (b) & (d): Comparison of the computation overheads

field. We could obtain $|\mathbb{Z}_p^*| = 20$ bytes, $|\mathbb{G}_1| = |\mathbb{G}_2| = 128$ bytes. In our simulation, the computation costs of each pairing, each exponentiation computation in $\mathbb{G}_1$ and each exponentiation computation in $\mathbb{G}_2$ are 11.712ms, 6.062ms, 1.899ms, respectively.

For ease of comparison, we set the number of attributes in the system to two times the number of attributes in access policy. In Fig. 2(a) and Fig. 2(c), we make a comparison of the storage and communication costs in [17], [18] and our scheme when n'=m=n=5, m'=10 and n'=m=n=10, m'=20. We can easily observe that the storage and communication costs in our scheme are much less than that in [17], [18] in terms of the public parameter size, decryption secret size, trapdoor size. That is because the sizes in other schemes are linear with the amount of attributes and our scheme linearly grows with the dimension of attribute vector. As illustrated in table 1, it's easy to find that the number of group elements in our ciphertext part is much less than that in [17], [18]. Moreover, as depicted in Fig. 2(b) and Fig.2(d) when n'=m=n=5, m'=10 and n'=m=n=10, m'=20, we can also find that our scheme is more lightweight than the other two schemes [17], [18]. This is because our scheme uses less pairing operations and has similar exponentiation operations compared to the schemes [17], [18]. Overall, our scheme is more efficient than the other two schemes on both storage and communication costs and computation overheads. Furthermore, compared to the other two schemes [17], [18], our scheme can achieve the privacy preservation of user attributes. Therefore, our scheme is more feasible and practical for the real scenarios.

6 Conclusion

In this paper, a novel AH-PE-ET scheme named attribute-hiding predicate encryption with equality test is formulated to provide the privacy preservation of user attributes and flexible search capability on ciphertexts simultaneously. With our introduced scheme, data user, who features with a set of attributes, can delegate the capability of equivalence test to the cloud server for determining whether two expected ciphertexts contain the same plaintext message without leaking any attribute privacy and trapdoor privacy. To the best of our knowledge, this proposed scheme is the first such scheme which meanwhile deals with these issues on both privacy protection of user attributes and flexible data search. Additionally, the rigorous security proof is clearly state to prove that our scheme is IND-CPA secure in the standard

model under decisional bilinear Diffie-Hellman assumption. Finally, we present theoretical comparisons and experimental simulations of the existing ABE-ETs and our AH-PE-ET scheme to indicate the feasibility and practicability of our proposed scheme. Future work contains building one novel scheme that achieves the IND-CCA2 security in standard model based on our present scheme.

7 Acknowledgement

We are greatly grateful to the associate editor and reviewers for their invaluable suggestions. This work was supported in part by the National Science Foundation of China (No. 61370026 and U1401257), Science and Technology Project of Guangdong Province (No. 2016A010101002), 13th Five-Year Plan of National CryptographyDevelopment Fund for Cryptographic Theory of China (MMJJ20170204), Fundamental Research Funds for the Central Universities (No. ZYGX2016J091) and Guangxi Colleges Universities Key Laboratory of cloud computing and complex systems, the Neijiang Science and Technology Incubating Project (No. 170676) and the Major International (Regional) Joint Research Project of China National Science Foundation under grant No. 61520106007. The corresponding author is Hu Xiong (xionghu.uestc@gmail.com).

References

[1] H. Menouar, I. Guvenc, K. Akkaya, et al, "UAV-Enabled Intelligent Transportation Systems for the Smart City: Applications and Challenges", *IEEE Communications Magazine,* vol. 55, no. 3, pp. 22-28, 2017.

[2] R. Petrolo, V. Loscri, N. Mitton, "Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms", *IEEE Transactions on Emerging Telecommunications Technologies,* vol. 28, no. 1, 2017.

[3] R. Brydia, S. Turner, W. Eisele, et al, "Development of intelligent transportation system data management", *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1625, pp. 124-130, 1998.

[4] M. Chaturvedi, S. Srivastava, "Multi-modal design of an intelligent transportation system", *IEEE Transactions on Intelligent Transportation Systems,* vol. 18, no. 8, 2017.

[5] X. Li, Q. Huang, D. Wu, "Distributed Large-Scale Co-Simulation for IoT-Aided Smart Grid Control", *IEEE Access,* vol. 5, pp. 19951-19960, 2017.

[6] S. Sakr, A. Liu, D. M. Batista, M. Alomari, "A Survey of Large Scale Data Management Approaches in Cloud Environments", *IEEE Communications Surveys and Tutorials*, vol. 13, no. 3, pp. 311-336, 2011.

[7] H. Xiong, J. Sun,"Comments on "Verifiable and Exculpable Outsourced Attribute-Based Encryption for Access Control in Cloud Computing"", *IEEE Transactions on Dependable and Secure Computing,* vol. 14, no. 4, pp. 461-462, 2017.

[8] D. Chen, H. Zhao, "Data security and privacy protection issues in cloud computing", *IEEE International Conference on Computer Science and Electronics Engineering (ICCSEE),* vol. 1, pp. 647-651, 2012.

[9] A. Squicciarini, S. Sundareswaran, D. Lin, "Preventing information leakage from indexing in the cloud", *IEEE International Conference on Cloud Computing (CLOUD),* pp. 188-195, 2010.

[10] C. Wang, Q. Wang, K. Ren, et al., "Toward secure and dependable storage services in cloud computing", *IEEE Transactions on Services Computing,* vol. 5, no. 2, pp. 220-232, 2012.

[11] K. Yang, X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing", *IEEE Transactions on parallel and distributed systems,* vol. 24, no. 9, pp. 1717-1726, 2013.

[12] Y. Zhang, X. Chen, J. Li, D. Wong, and H. Li, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing", *Information Sciences,* vol. 379, pp. 42-61, 2017.

[13] R. Canetti, S. Halevi, J. Katz, "A forward-secure public-key encryption scheme", *International Conference on the Theory and Applications of Cryptographic Techniques. Springer,* pp. 255-271, 2003.

[14] M. Ma, D. He, N. Kumar, et al., "Certificateless searchable public key encryption scheme for industrial internet of things", *IEEE Transactions on Industrial Informatics*, 2017.

[15] A. Sahai, B. Waters, "Fuzzy Identity-Based Encryption", *International Conference on Theory and Applications of Cryptographic Techniques,* Springer, LNCS 3494, pp. 457-473, 2005.

[16] Q. Zheng, S. Xu, G. Ateniese, "VABKS: Verifiable attribute-based keyword search over outsourced encrypted data", *33rd Annual IEEE International Conference on Computer Communication (INFOCOM 2014),* pp. 522-530, IEEE, 2014.

[17] Q. Wang, L. Peng, H. Xiong, et al., "Ciphertext-Policy Attribute-based Encryption with Delegated Equality Test in Cloud Computing", *IEEE Access,* vol. 6, pp. 760-771, 2018.

[18] H. Zhu, L. Wang, H. Ahmad, et al., "Key-Policy Attribute-Based Encryption With Equality Test in Cloud Computing", *IEEE Access,* vol. 5, pp. 20428-20439, 2017.

[19] Y. Zhang, D. Zheng, and R. Deng,"Security and privacy in smart health: efficient policy-hiding attribute-based access control," *IEEE Internet of Things Journal,* 2018, doi:10.1109/JIOT.2018.2825289.

[20] Q. Jiang, M. Khan, X. Lu, J. Ma, D. He, "A privacy preserving three-factor authentication protocol for e-Health clouds", *The Journal of Supercomputing*, vol. 72, no. 10, pp. 3826-3849, 2016.

[21] J. Katz, A. Sahai, B. Waters., "Predicate encryption supporting disjunctions, polynomial equations, and inner products," *J. Cryptology* vol. 26, no. 2, pp. 191-224. 2008

[22] I. Kim, S. Hwang, J. Park et al,"An Efficient Predicate Encryption with Constant Pairing Computations and Minimum Costs", *IEEE Transactions on Computers (TOC),* vol. 65, no. 10, pp. 2947-2958, 2016.

[23] H. Wee, "Attribute-Hiding Predicate Encryption in Bilinear Groups, Revisited," *Theory of Cryptography Conference*, Springer, LNCS 106777, pp. 206-233, 2017.

[24] X. Wang, F. Xhafa, W. Cai, et al., "Efficient privacy preserving predicate encryption with fine-grained searchable capability for Cloud storage", *Computers & Electrical Engineering,* pp. 56, no. 871-883, 2016.

[25] K. Liang, W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage", *IEEE Transactions on Information Forensics and Security (TIFS),* vol. 10, no. 9, pp. 1981-1992, 2015.

[26] J. Li, X. Lin, Y. Zhang, et al., "KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage", *IEEE Transactions on Services Computing (TSC),* vol. 10, no. 5, pp. 715-725, 2017.

[27] G. Yang, C. Tan, Q. Huang, et al., "Probabilistic public key encryption with equality test", *International Conference on Topics in Cryptology (CT-RSA 2010)*, LNCS 5985, Springer, pp. 119-131, 2010.

[28] B.Dan, G. Crescenzo , R. Ostrovsky , et al. "Public Key Encryption with Keyword Search", *EUROCRYPT-2004*, Springer, LNCS 3027, pp. 506-522, 2004.

[29] B. Dan, F. Matt, "Identity-Based Encryption from the Weil Pairing", *Society for Industrial and Applied Mathematics,* vol. 32, no. 3, pp. 213-229, 2001.

[30] B. Waters, "Efficient Identity-Based Encryption Without Random Oracles," *Lecture Notes in Computer Science,* Springer, LNCS 3494, pp. 114-127, 2005.

[31] M. Abdalla, M. Bellare et al., "Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions", *Journal of Cryptology,* vol. 21, no. 3, pp. 350-391, 2008.

[32] N. Cao, C. Wang, M. Li, K. Ren, W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data", *IEEE Transaction Parallel Distribute System (TPDS),* vol. 25, no. 1, pp. 222-233, 2014.

[33] Q. Tang, "Towards public key encryption scheme supporting equality test with fine-grained authorization", *Australasian Conference on Information Security and Privacy (ACISP),* Springer, LNCS 6812, pp. 389-406, 2011.

[34] S. Ma, "Identity-based encryption with outsourced equality test in cloud computing", *Information Sciences,* vol. 328, pp. 389-402, 2016.

[35] L. Wu, Y. Zhang, D. He, "Dual Server Identity-Based Encryption with Equality Test for Cloud Computing", *Journal of Computer Research and Development,* vol. 54, pp. 2232-2243, 2017.

[36] H. Lee, S. Ling, J. Seo, et al., "Semi-generic construction of public key encryption and identity-based encryption with equality test", *Information Sciences,* vol. 373, pp. 419-440, 2016.

[37] L. Wu, Y. Zhang, K. Choo, et al., "Efficient and secure identity-based encryption scheme with equality test in cloud computing", *Future Generation Computer Systems (FGCS),* vol. 73, pp. 22-31, 2017.
[38] F. Guo, W. Susilo, Y. Mu, "Distance-Based Encryption: How to Embed Fuzziness in Biometric-Based Encryption", *IEEE Transactions on Information Forensics and Security (TIFS),* vol. 11, no. 2, pp. 247-257, 2016.
[39] B. Lynn,"The stanford pairing based crypto library", http://crypto.stanford.edu/pbc/.
[40] Y. Chen, L. Chen, "The twin bilinear diffie-Hellman inversion problem and applications", *International Conference on Information Security and Cryptology,* Springer, LNCS 6219, pp. 113-132, 2010.

Jianfei Sun is currently pursing his PhD degree in the School of Information and Software Engineering, UESTC. His research interests include public key cryptography and network security.

Yangyang Bao is currently pursing his Master degree in the School of Information and Software Engineering, UESTC. His research interests include information and network security.

Xuyun Nie received the Ph.D. degree in Information security from Graduate university of Chinese Academy of Sciences, Beijing, China, in 2007. He is presently an Associate Professor at University of Electronic Science and Technology of China (UESTC). His research interests include cryptography and information security.

Hu Xiong received his PhD degree in the School of Computer Science and Engineering from the University of Electronic Science and Technology of China (UESTC) in 2009. He is currently an associate professor in the School of Information and Software Engineering, UESTC. His research interests include cryptographic protocols and network security.