# CSCAC: one constant-size CPABE access control scheme in trusted execution environment

Yongkai Fan and Shengle Liu*

Beijing Key Lab of Petroleum Data Mining,
China University of Petroleum (Beijing),
No. 18, Fuxue Road, Beijing, 102249, China
Email: fanyongkai@gmail.com
Email: lilian_liu1993@163.com
*Corresponding author

Gang Tan

Department of Computer Science and Engineering,
Penn State University,
201 Old Main, University Park, PA, USA
Email: gtan@cse.psu.edu

Xiaodong Lin

Department of Computer Science and Technology,
China University of Petroleum (Beijing),
No. 18, Fuxue Road, Beijing, 102249, China
Email: upclinxiaodong@163.com

**Abstract:** The popularity of versatile mobile devices has been increasing concerns about their security. How to protect the sensitive data is an urgent issue to be solved. Ciphertext-policy attribute-based encryption (CPABE) is a practical method for encrypting data and can utilise user's attributes to encrypt the sensitive data. In this paper, we propose a constant-size CPABE access control (CSCAC) model by using the Trusted Execution Environment to manage the dynamic key generated by attributes. The original data is encrypted by a symmetric storage key, then the storage key is encrypted under an AND-gate access policy. Only the user who possesses a set of attributes that satisfy the access policy can recover the storage key. The security analysis shows the design of this access control scheme reduces the burden and risk in the case of one single authority.

**Keywords:** constant-size ciphertext; access control; trusted execution environment; TEE; attribute-based encryption; ABE; security.

**Reference** to this paper should be made as follows: Fan, Y., Liu, S., Tan, G. and Lin, X. (2019) 'CSCAC: one constant-size CPABE access control scheme in trusted execution environment', *Int. J. Computational Science and Engineering*, Vol. 19, No. 2, pp.162–168.

**Biographical notes:** Yongkai Fan received his BE, MS and PhD degrees in Computer Science from Jilin University, Changchun, China, in 2001, 2003, 2006, respectively. From 2006 to 2009, he was an Assistant Researcher in Tsinghua University, Beijing. His current appointment is an Assistant Professor in China University of Petroleum (Beijing) since 2010. His current research interests include theories of software engineering and software security.

Shengle Liu received her MS degree in China University of Petroleum (Beijing), in 2015, and BE degree from Ocean University of China, Qingdao, China, in 2011, both in Computing Science. Her current research interests include theories of software engineering and software security.

Gang Tan received his BE degree in Computer Science from Tsinghua University in 1999, and his PhD in Computer Science from Princeton University in 2005. He is an Associate Professor in Penn State University, University Park, USA. He was a recipient of an NSF Career award and won James F. Will Career Development Professorship. He leads the Security of Software (SOS) lab at Penn State. He is interested in methodologies that help create reliable and secure software systems.

Xiaodong Lin received his MS degree in China University of Petroleum (Beijing), in 2016, and the BE degree from China University of Petroleum (East China), Qingdao, China, in 2012, both in Computing Science. His current research interests include theories of software engineering and software security.

## 1 Introduction

Mobile device have become the primary computing device for many individuals, and it becomes increasingly susceptible to various apps (Do et al., 2015). For the implementation of strict privacy security, the inappropriate access of mobile devices by malicious apps is the major concern for mobile users, and access control comes to be more challenging. To solve above problems, attribute-based encryption (ABE) (Goyal et al., 2006) schemes have been applied. Presently, ABE mainly includes two categories called Ciphertext-policy attribute-based encryption (CPABE) (Bethencourt et al., 2007) and key-policy attribute-based encryption (KPABE) (Rahulamathavan et al., 2016). Researchers have proposed various improved ABE (Krishna and Balusamy, 2017) schemes, and use it to achieve access control ends. However, in those schemes, the length of the ciphertext increases linearly with the number of attributes contained in the ciphertext. In a large-scale application environment with a large number of users and attribute sets, the overhead of ciphertext length cannot be ignored for both the storage server and the users. Moreover, the series of critical processes are usually carried out in an untrustworthy environment, it will increase the risk level to some extent.

To handle the problem of ciphertext length and execution environment, we present a constant-size CPABE access control scheme, a revised and expanded version of this scheme is proposed in Fan et al. (2018), which will be run in one Trusted Execution Environment (TEE) (Varadarajan et al., 2016). First, the plaintext data is encrypted with a symmetric key, and then the symmetric key is encrypted under an AND-gate access policy to achieve fixed-length ciphertext and reduce the time consumption. Only the user with a set of attributes that satisfy the access policy can recover the symmetric key. Compared with the traditional access control mechanism based on attribute encryption, this scheme limits the length of the ciphertext to the fixed value after encrypting with the symmetric key and reduces the extra storage space while ensuring fine-grained access control. Second, using this scheme can reduce the burden and risk in the case of one single authority. Finally, our prototype is based on ARM TrustZone technology to provide a trusted environment and protect the security of the secret key.

The rest of this paper is organised as follows. Some related works are introduced in Section 2. In Section 3, we give necessary background information and assumptions. Our access control scheme is given in Section 4. We prove and analyse the security of our scheme in Section 5. At last, we conclude this paper in Section 6.

## 2 Related work

Sahai and Waters (2005) first proposed attribute-based encryption, which is extended from IBE (Dan and Franklin, 2001) and has two variants called key-policy attribute-based encryption (KPABE) (Meddah and Toumanari, 2016) and CPABE (Liu et al., 2017). As for KPABE, the ciphertext is associated with an attribute set and the secret key corresponds to an access policy. Decryption of the ciphertext with the secret key is possible if and only if the attribute set of ciphertext satisfies the access policy of that secret key. As for CPABE, the ciphertext is associated with an access structure and the secret key is associated with an attribute set (Ennajjar et al., 2016). Therefore, the ciphertext can be decrypted with the secret key if and only if the attributes of secret key satisfy the ciphertext access structure (Attrapadung et al., 2012). After the introduction of the seminal work of several KPABE schemes (Rahulamathavan et al., 2016; Meddah and Toumanari, 2016) and CPABE schemes (Green et al., 2011; Xu and Lang, 2015; Lewko et al., 2010) are presented in the literature. Note that CPABE enables the data encryptor to control who has access to the ciphertext. It is more appropriate in access control model while comparing with KPABE scheme.

The first CPABE scheme was put forward by Bethencourt et al. (2007). However, the scheme has a restriction that the security proof is only in the generic group model. Lewko et al. (2010) presented a fully secure CPABE scheme by using the dual system encryption techniques. Ling and Newport (2007) proposed the construction of CPABE, which can be proved under the standard model. And their scheme supports AND-gate access policy which deals with negative attributes explicitly and uses wildcards in the ciphertext policies. Emura et al. (2009) introduced a novel CPABE scheme with the constant ciphertext length using AND-gate policy. In addition, there are many CPABE schemes with the constant-size ciphertexts (Herranz and Laguillaumie, 2010; Guo et al., 2014; Zhang et al., 2014). However, there is no credible execution environment to ensure the validity of data and operations.

In most of the previous programs, many researchers focus on how to construct a more complex access control policy and do not pay attention to whether the operating environment is reliable, or the ciphertext storage overhead is huge. In this paper, we propose a provably secure AND-gate access structure CPABE scheme in TEE that supports the constant-size ciphertext.

## 3 Preliminaries

Prior to discussing our access scheme based on CPABE, this section begins with the preliminary introduction of related things used for our scheme. This includes bilinear maps used for encryption and decryption, access structure for access control scheme, DBDH assumption for formalisation analysis of the security of our scheme, and TEE for avoiding attack from the local side.

### *3.1 Bilinear maps*

We first review the concept of bilinear maps. Let $G$ and $G_T$ be two multiplicative cyclic groups of prime order $p$. Let $g$ be a generator of $G$ and e: $G \times G \rightarrow G_T$ be a bilinear map satisfying the following properties (Dan and Waters, 2007):

- *bilinearity:* For all $u, v \in G$ and $a, b \in Z_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$
- *non-degeneracy:* $e(u, v) \neq 1$
- *computability:* $e(u, v)$ can be efficiently computed.

### *3.2 Access structure*

Based on the assumption of ABE, we let $U = \{att_1, ..., att_n\}$ be a set of attributes. For $att_i \in U, V_i = \{v_{i,1}, v_{i,2}, ..., v_{i,n_i}\}$ is a set of possible values, where $n_i$ is the number of possible values for $att_i$. Let $S = \{S_1, S_2, ..., S_n\}, S_i \in V_i$ be an attribute list for a user, and $T = \{T_1, T_2, ..., T_n\}, T_i \in V_i$ be an access structure (Beimel, 1996). The notation $S| = T$ expresses that an attribute list $S$ suits an access structure $T$, especially, $S_i = T_i (i = 1, 2, ..., n)$.

### *3.3 DBDH assumption*

The computational assumption for the security of our scheme is the Decisional Bilinear Diffie-Hellman problem defined by (Han et al., 2016). Given a DBDH challenge set ($g$, $g^a$, $g^b$, $g^c$, $Q$) by challenger for random $a$, $b$, $c \in Z_p$ to decide whether $Q = e(g, g)^{abc}$. An algorithm B has the advantage $\varepsilon$ in solving the DBDH problem in $G$. We say that the DBDH assumption holds if no polytime algorithm has a non-negligible advantage of at least $\varepsilon$ in solving the DBDH problem in $G$.

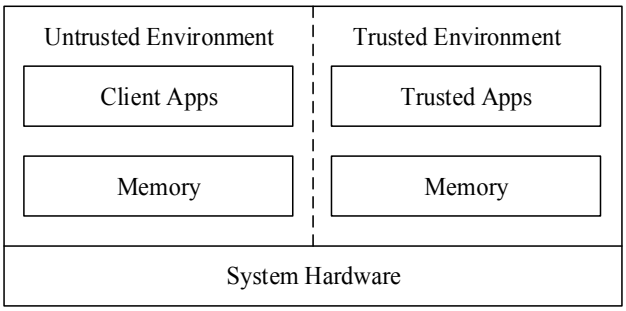
### *3.4 TEE*

The concept of TEE was launched by the Global Platform in 2011. The system is split into two execution environments that are physically separated: the first one is untrusted and hosts the main OS and applications, and the second one hosts trusted applications. These two environments coexist in the system. Each one features its own software stack. The trusted environment is separated from the untrusted environment, and the isolation is guaranteed by hardware.

In general, TEE is a secure environment with its own processing, memory, and storage capabilities, its integrity is protected by hardware. The secure environment provided by TEE gives the possibility to build applications and services with better security and privacy. Sensitive operations are limited in TEE, and sensitive data should leave TEE without explicit declassification.

We use the term TEE to refer to the execution environment in the trusted area, and the term rich execution environment (REE) to refer to the environment in the untrusted area (Jang et al., 2015). A TEE architecture can be abstracted (see Figure 1). In this paper, the CPABE related operations and the construction a novel access control model will be placed in TEE as trusted applications. In our scheme, the breach of TEE is out of our scope and we assume the TEE is completely trusted, TEE can protect the CPABE related operations from malware, trusted applications run in TEE will be protected from the normal world.

**Figure 1** TEE architecture




## 4 Access control model

In this section, we present our proposed CPABE scheme with constant-size ciphertext which consists of six algorithms, namely, system initial settings, encrypt, KeyGen, revocation, decrypt and file deletion. The whole scheme contains four entities: data owner, client application, TEE, storage servers. The interaction of our prototype is shown in Figure 2. The data owner first encrypts the data with the symmetric key and then uses the CPABE algorithm to encrypt the symmetric key under an AND-gate access policy. Only the client application with a set of attributes (e.g., version number, unique identifier, factory date) that satisfy the access policy can recover the symmetric key. Finally, the encrypted data and the encrypted symmetric key will be outsourced to the storage servers. The app that wishes to access the data sends an access request with its attribute set to TEE. After authenticating the requesting application, TEE generates the private key by the attribute set and then sends it back to the application. By doing so, the security model can be used to protect sensitive data which may be compromised by malicious applications.

**Figure 2** Access control scheme based on TEE and CPABE (see online version for colours)

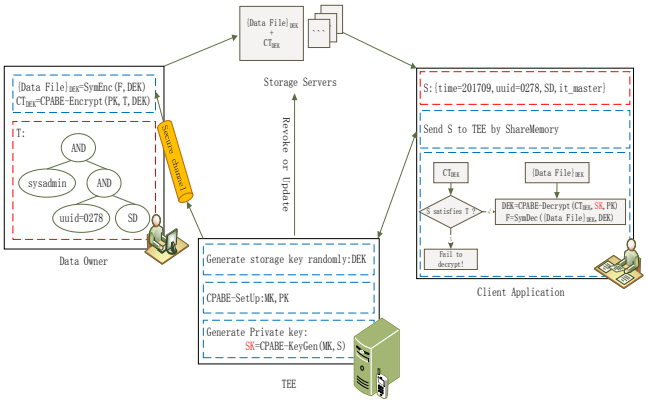



Table 1 outlines the notations used in the remainder of this paper, and the detailed description of the proposed scheme is given in the following subsections.

**Table 1** Symbol description

| *Symbol* | *Description* |
|---|---|
| PK, MK | Public key and Master key |
| SK | Private key of user |
| M | The plaintext |
| CT | The ciphertext |
| T | AND-gate Access structure |
| S | Set of user attributes |
| DEK | Symmetric storage key |
| csk, cpk, tsk, tpk | Asymmetric key pair for interaction between CA and TA in TEE |

### *4.1 System initial settings*

In the initial settings phase, TEE randomly generates a symmetric storage key *DEK*, then takes the security parameter $\lambda$ as input and outputs the master key *MK* and the public key *PK*. The specific generation process is as follows.

Let $G$ be a bilinear group of prime order $p$, and let $g$ be a generator of $G$. In addition, let e: $G \times G \rightarrow G_T$ denote the bilinear map. It chooses random group elements $g_2$, $h_1$, $h_2 \in G$, and a random exponent $\alpha \in Z_p$. Then we will get $g_1 = g^{\alpha}$ and $Z = e(g_2, g_1)$. The public key is published as $PK = (Z, g, g_1, g_2, h_1, h_2)$ and the master key is $MK = g_2^{\alpha}$.

### *4.2 Encrypt*

In order to ensure the confidentiality and integrity of the data, the data owner will encrypt the data and outsource to the non-volatile memory, which is in REE. Due to the complexity of the CPABE algorithm, it is not suitable to encrypt the large private files. So, we first use *DEK* (symmetric storage key) to encrypt the private file and then encrypt *DEK* with the CPABE algorithm. It outsources the *DEK*ciphertext $CT_{DEK}$ and the data file ciphertext $\{DataFile\}_{DEK}$ as a whole to the storage servers. The storage format of data file is shown in Figure 3.

**Figure 3** The storage format of data file

| ID | $CT_{DEK}$ | $\{Data\ File\}_{DEK}$ |
|---|---|---|

This algorithm encrypts a message $M$ (this article represents *DEK*) with *PK* and $T$, then generates a ciphertext *CT*.

The specific step is: $CT \leftarrow Encrypt(PK, M, T)$. The AND-gate access structure $T$ is excluded in the ciphertext.

First, it chooses a random value $\gamma$ and then gets:

$$C_0 = g^{\gamma}, C_1 = M \cdot Z^{\gamma}, C_2 = \left(h_1 \prod_{i \in T} g_1^{attr_i}\right)^{\gamma}, C_3 = (g_1 h_2)^{\gamma}.$$

So the ciphertext is published as: $CT = (T, C_0, C_1, C_2, C_3)$.

### *4.3 KeyGen*

The client application who wants to access the sensitive file need to send a request with the attribute set $S$ to TEE by safe channel in order to generate *SK*. The key generation algorithm takes *MK*, *PK* and $S = \{att_1, \ldots, att_n\}$ as inputs. It creates the private key *SK*, the specific step is: $SK \leftarrow KeyGen(PK, MK, S)$.

It chooses a random $\mu \in Z_p$. The private key is published as *SK*:

$$D_1 = g^{\mu}, D_2 = g_2^{\alpha} \left(h_1 \prod_{j \in S} g_1^{attr_j}\right)^{\mu}.$$

### *4.4 Decrypt*

If the client application supposes to access the private file, it takes *SK* to decrypt the $CT_{DEK}$, then obtains the private file with *DEK*.

First, the client application checks whether the following two equations are true:

$$\begin{cases} e(g, C_2) = e\left(C_0, h_1 \prod_{i \in T} g_1^{attr_i}\right) \\ e(g, C_3) = e(C_0, g_1 h_2) \end{cases} \quad (1)$$

If either of the two equations does not hold, then return $\perp$; If the equations are established, the ciphertext is valid and it continues to be decrypted.

Second, it determines whether the notation $S| = T$, expressing that an attribute set $S$ satisfies an access structure $T$, namely, $S_i = T_i (i = 1,2,\ldots,n)$. The *CT* can only be decrypted, if and only if the attribute set fulfils the access structure.

Third, we can obtain the plaintext $M$ (i.e., *DEK*):

$$M = \frac{C_1 \cdot e(C_2, D_1)}{e(D_2, C_0)} = \frac{C_1 \cdot e\left(\left(h_1 \prod_{i \in S} g_1^{attr_i}\right)^{\gamma}, g^{\mu}\right)}{e\left(g_2^{\alpha} \left(h_1 \prod_{j \in T} g_1^{attr_j}\right)^{\mu}, g^{\gamma}\right)}$$

$$= \frac{M \cdot e(g_2, g_1)^{\gamma} \cdot e\left(h_1 \prod_{i \in S} g_1^{attr_i}, g\right)^{\gamma\mu}}{e(g_2^{\alpha}, g^{\gamma}) \cdot e\left(\left(h_1 \prod_{j \in T} g_1^{attr_j}\right)^{\mu}, g^{\gamma}\right)} \tag{2}$$

### 4.5 Revocation

Once a user has been revoked, the system must ensure that the revoked user cannot access the relevant data, while the user who has not been revoked still has the authority to access the data as before. A typical approach is to re-encrypt the data that the revoked user can access.

In this paper, we add an attribute timestamp X to each user's attribute set. At the same time, a time description Z is added to the access policy. The user can decrypt a ciphertext correctly, if and only if X ≥ Z and the user's attribute set satisfies the access structure.

We can also authorise different attributes to different users by TEE in order to complete the revocation function. In this way, we will reduce the number of interactions between the data owner and the trusted authority, reducing communication overhead.

### 4.6 File deletion

File deletion operation can only be initiated by the data owner. The data owner sends the request to TEE and provides the unique *ID* of the deleted file. TEE will verify the signature and decrypt the ciphertext. If there is an associated *ID* file, it will be deleted.

Firstly, the data owner invokes the request function *Request*() to send the deleted file ID for TEE, which can be encrypted with the public key of TEE (tpk) and signed with the private key of date owner (csk), and uses the HMAC function to encapsulate:

$$\begin{aligned} &m\_request \leftarrow Enc_{tpk}\left((f\_ID), Sign_{csk}(f\_ID)\right) \| \\ &HMAC\left(Enc_{tpk}\left((f\_ID), Sign_{csk}(f\_ID)\right)\right) \end{aligned} \tag{3}$$

Secondly, when TEE receives *m_request*, it finds the relevant sensitive file according to the *ID*, and verifies the integrity by the HMAC function. Then, TEE uses its own private key (tsk) to decrypt this message and verifies the signature with the CA's public key (cpk).

## 5 Security proof

In this section, we prove and analyse the security of this scheme.

### 5.1 Proving the security of CSCAC scheme

*Theorem 1.* Assuming that the DBDH assumption holds, the proposed scheme is has a certain degree of security in the challenge of access structure.

*Proof.* Suppose the existence of an adversary $A$ that can attack CSCAC scheme with the advantage $\varepsilon$. We build a simulator $B$ that can play the DBDH game with the advantage $\varepsilon / 2$.

Given a DBDH challenge set($g$, $g^a$, $g^b$, $g^c$, $Q$) by challenger for random $a, b, c \in Z_p$, the simulator $B$ creates the following simulation. When $Q = e(g, g)^{abc}$, $B$ outputs 1, otherwise 0.

*Init.* The adversary $A$ chooses the access policy $T^*$ and the challenge attribute set $S^*$, then sends them to $B$, where $attr^*$ is one of the attribute set $S^*$ to satisfy $T^*$.

*Setup.* $B$ simulates the system public parameter as follows.

1 $B$ randomly picks $\lambda, \delta \in Z_p$.

2 Set the following parameters:

$$\begin{aligned} &g_1 = g^{\alpha}, \ g_2 = g^{b}, \\ &h_1 = g^{\lambda} \cdot \prod_{i \in T^*} g_1^{-attr_i^*}, \ h_2 = g^{\delta} \cdot g_1^{-1}, \\ &Z = e(g_2, g_1) = e(g, g)^{ab} \end{aligned} \tag{4}$$

3 To provide a public key $PK$ to $A$, $B$ generates $PK = (Z, g, g_1, g_2, h_1, h_2)$.

*Phase 1.* $A$ makes the following queries.

*Private key query.* The attribute set must satisfy $S \mid \neq T^*, S^* \neq S$ or else $B$ simply aborts and takes a random guess. $B$ generates $SK_{S^*}$ as follows.

1 $B$ randomly picks the number $\bar{\mu} \in Z_p$ and computes

$$D_1^* = g^{\bar{\mu}} \cdot g_2^{\frac{-1}{S-S^*}}, D_2^* = \left(h_1 \prod_{j \in S} g_1^{attr_j}\right)^{\bar{\mu}} \cdot g_2^{\frac{-\lambda}{S-S^*}}.$$

2 Send the $SK_{S^*} = (D_1^*, D_2^*)$ to $A$.

If $\mu = \bar{\mu} - \frac{b}{S - S^*}$, we can deduce that the above-mentioned private key is indeed a valid private key $SK_S$:

$$D_1^* = g^{\bar{\mu}} \cdot g_2^{\frac{-1}{S-S^*}} = g^{\bar{\mu} - \frac{b}{S-S^*}} = g^{\mu} \tag{5}$$

$$\begin{aligned} D_2^* &= \left(h_1 \prod_{j \in S} g_1^{attr_j}\right)^{\bar{\mu}} \cdot g_2^{\frac{-\lambda}{S-S^*}} \\ &= \left(g^{ab} \cdot g^{-ab}\right)\left[\left(h_1 \prod_{j \in S} g_1^{attr_j}\right)^{\bar{\mu}} \cdot g_2^{\frac{-\lambda}{S-S^*}}\right] \\ &= \left(g_2^{a} \cdot g_1^{-b}\right)\left[\left(h_1 \prod_{j \in S} g_1^{attr_j}\right)^{\bar{\mu}} \cdot g^{\frac{-\lambda b}{S-S^*}}\right] \\ &= g_2^{a} \cdot \left(h_1 \prod_{j \in S} g_1^{attr_j}\right)^{\bar{\mu}} \cdot \left(g^{\lambda} \cdot g_1^{S-S^*}\right)^{\frac{-b}{S-S^*}} \end{aligned}$$

$$= g_2^a \cdot \left( h_1 \prod_{j \in S} g_1^{attr_j} \right)^{\overline{\mu}} \cdot \left( g^{\lambda} \prod_{i \in T^*} g_1^{-attr_i} \cdot \prod_{j \in S} g_1^{attr_j} \right)^{\frac{-b}{S-S^*}}$$
$$= g_2^a \cdot \left( h_1 \prod_{j \in S} g_1^{attr_j} \right)^{\overline{\mu}} \cdot \left( h_1 \prod_{j \in S} g_1^{attr_j} \right)^{\frac{-b}{S-S^*}} \quad (6)$$
$$= g_2^a \cdot \left( h_1 \prod_{j \in S} g_1^{attr_j} \right)^{\mu}$$

*Challenge.* $B$ flips a fair binary coin $\beta \in \{0,1\}$. It creates $CT^* = (g^c, M_\beta \cdot Q, (g^c)^{\lambda}, (g^c)^{\delta})$ and sends it to $A$.

If $Q = e(g, g)^{abc}$, we can deduce that the above-mentioned $CT^*$ is indeed a valid challenge ciphertext:

$$\begin{aligned} M_\beta \cdot Q &= M_\beta \cdot e(g,g)^{abc} \\ &= M_\beta \cdot e\left(g_2, g_1\right)^c = M_\beta \cdot Z^c \end{aligned} \quad (7)$$

$$\begin{aligned} \left(g^c\right)^{\lambda} &= \left(g_1^{T^*-T^*} \cdot g^{\lambda}\right)^c \\ &= \left( g^{\lambda} \prod_{i \in T^*} g_1^{-attr_i^*} \cdot \prod_{i \in T^*} g_1^{attr_i^*} \right) \\ &= \left( h_1 \cdot \prod_{i \in T^*} g_1^{attr_i^*} \right)^c \end{aligned} \quad (8)$$

$$\begin{aligned} \left(g^c\right)^{\delta} &= \left(g^{\delta}\right)^c \\ &= \left(g_1 \cdot g_1^{-1} \cdot g^{\delta}\right)^c = \left(g_1 \cdot h_2\right)^c \end{aligned} \quad (9)$$

If $Q$ is a random element in the group $G_T$, the adversary $A$ considers that the challenge ciphertext $CT^*$ and the binary coin $\beta$ are independent.

*Phase 2.* The simulator acts exactly as it did in Phase 1.

*Guess.* $A$ outputs a guess $\beta'$, if and only if $\beta' = \beta$, B outputs 1. Therefore, the advantage of breaking the DBDH assumption is:

$$\begin{aligned} \Pr\left[\beta = \beta'\right] - \frac{1}{2} &= \Pr\left[\beta = \beta' \mid \beta = 0\right] \Pr[\beta = 0] \\ &\quad + \Pr\left[\beta = \beta' \mid \beta = 1\right] \Pr[\beta = 1] - \frac{1}{2} \\ &= \left(\frac{1}{2} + \varepsilon\right) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} = \frac{\varepsilon}{2} \end{aligned} \quad (10)$$

### 5.2 Analysing the security of CSCAC scheme

In the scheme, we consider three types of adversaries including access to security management, data confidentiality, and user information security. Based on the scheme proposed in this paper, we analyse the security of our scheme.

- *Access to security management.* Considering one scenario: one malicious application tries to break the access control and wants to access the protected data as it wishes. The first thing it tries to do is to bypass the access control by knowing the knowledge about operations related keys which are critical in our scheme. To handle this situation, we take TEE as the trusted execution environment to do the generation and management of keys operation, which will eliminate the possibility of losing keys in the whole process. Furthermore, the whitelist of applications reported by anti-virus company will help the data owner to decide which application has privileges to access the sensitive file. Untrusted or suspicious applications have no opportunity to access the sensitive file because the data owner will never add the untrusted application into the sensitive file's access list as an attribute in the attributes set. The data can be decrypted only if the application has a set of attributes that satisfy the access policy.
- *Data confidentiality.* Considering the scenario: if one skillful attacker can access the protected data by sophisticated means, he/she can read the data protected intentionally, or he/she can even know the *DEK* (symmetric storage key) somehow. Traditionally, he/she can use the *DEK* to recover the encrypted data to plaintext directly. Is it a challenge in our proposed scheme? We use a hybrid encryption system to ensure that the sensitive data are not obtained by illegal users. First the data are encrypted with a symmetric storage key, then the CPABE algorithm is used to encrypt the *DEK*, and finally, the data are outsourced to non-volatile memory in the form of Figure 3. Only users who meet the access control policy can recover *DEK* and then decrypt the data. Therefore, the scenario is not a challenge in our scheme, the data confidentiality can be kept well.
- *Information security.* Considering the scenario: the protected sensitive data is the user related personal information (e.g., contact lists, personal medical records, or private pictures) which will be used commonly. If one hacker or information eavesdropper tries to steal them purposely, how our scheme can respond to this situation. To prevent the illegal user from breaking the information of the legitimate user in the storage system, the data cannot be decrypted without a private key, which is associated with an attribute set. If the illegal user wants to manipulate a private file, without the signature key (csk) which is protected by TEE, the illegal operation on the date can be checked out by trusted applications run in TEE, then can be ignored and aborted, therefore, the protected information is in the safe status.

## 6 Conclusions

The security of sensitive data is a major concern nowadays. In this paper, we propose a CSCAC (constant-size CPABE access control model) by using TEE to manage the dynamic key generated by attribute. The scheme based on CPABE and TEE can ensure the confidentiality of data, and achieve the access control ends. Moreover, it can mitigate the risk of

sensitive information leakage. Based upon our formulisation analysis, we proved that proposed scheme can meet the security requirements of protecting sensitive files from malicious access. In the next step, we will pay more attention to enhance our scheme, aiming to improve the efficiency of access control and do some overhead experiments to verify the feasibility and effectiveness of our proposed scheme.

## Acknowledgements

This work was partially supported by Beijing Higher Education Teacher Project (No. 00001149), and by Science Foundation of China University of Petroleum (Beijing) (No. 01JB0423).

## References

Attrapadung, N., Herranz, J., Laguillaumie, F., Libert, B., Panafieu, E. and Rafols, C. (2012) 'Attribute-based encryption schemes with constant-size ciphertexts', *Theoretical Computer Science*, Vol. 422, No. 3, pp.15–38.

Beimel, A. (1996) 'Secure schemes for secret sharing and key distribution', *International Journal of Pure & Applied Mathematics*.

Bethencourt, J., Sahai, A. and Waters, B. (2007) 'Ciphertext-policy attribute-based encryption', *IEEE Symposium on Security and Privacy*, Vol. 2008, pp.321–334.

Dan, B. and Franklin, M. (2001) 'Identity-based encryption from the weil pairing', *International Cryptology Conference*, pp.213–229.

Dan, B. and Waters, B. (2007) 'Conjunctive, subset, and range queries on encrypted data', *The Theory of Cryptography Conference*, pp.535–554.

Do, Q., Martini, B. and Choo, K.K.R. (2015) 'Exfiltrating data from Android devices', *Elsevier Advanced Technology Publications*, Vol. 48, pp.74–91.

Emura, K., Miyaji, A., Nomura, A., Omote, K. and Soshi, M. (2009) 'A ciphertext-policy attribute-based encryption scheme with constant ciphertext length', *International Conference on Information Security Practice and Experience*, Vol. 5451, pp.13–23.

Ennajjar, I., Tabii, Y. and Benkaddour, A. (2016) 'An enhanced approach for data sharing security in cloud computing', *International Journal of Cloud Computing*, Vol. 5, No. 3, p.198.

Fan, Y., Liu, S., Tan, G. and Qiao, F. (2018) 'Fine-grained access control based on trusted execution environment', *Future Generation Computer Systems*, In press.

Goyal, V., Pandey, O., Sahai, A. and Waters, B. (2006) 'Attribute-based encryption for fine-grained access control of encrypted data', *ACM Conference on Computer and Communications Security*, Vols. 89–98, pp.89–98.

Green, M., Hohenberger, S. and Waters, B. (2011) 'Outsourcing the decryption of ABE ciphertexts', *Proc.20th USENIX Conf. Security*, pp.34–59.

Guo, F., Mu, Y., Susilo, W. and Wong, D.S. (2014) 'CP-ABE with constant-size keys for lightweight devices', *Information Forensics & Security IEEE Transactions on*, Vol. 9, No. 5, pp.763–771.

Han, Z., Qiu, S., Liu, J. and Shi, Y. (2016) 'Deterministic attribute-based encryption', *International Journal of High Performance Computing and Networking*, Vol. 9, Nos. 5/6, p.443.

Herranz, J. and Laguillaumie, F. (2010) 'Constant size ciphertexts in threshold attribute-based encryption', *International Conference on Practice and Theory in Public Key Cryptography*, Vol. 6056, pp.19–34.

Jang, J., Kong, S., Kim, M., Kim, D. and Kang, B.B. (2015) 'SeCReT: secure channel between rich execution environment and trusted execution environment', *Network and Distributed System Security Symposium*.

Krishna, P.V. and Balusamy, B. (2017) 'Simplified and efficient framework for managing roles in cloud-based transaction processing systems using attribute-based encryption', *International Journal of Computational Science and Engineering*, Vol. 14, No. 2, p.135.

Lewko, A., Okamoto, T., Sahai, A., Takashima, K. and Waters, B. (2010) 'Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption', *International Conference on Theory and Applications of Cryptographic Techniques*, Vol. 6110, pp.62–91.

Ling, C. and Newport, C. (2007) 'Provably secure ciphertext policy ABE', *ACM Conference on Computer and Communications Security*, pp.456–465.

Liu, Z., Ma, H., Bai, C. and Zhang, Y. (2017) 'Expressive ciphertext-policy attribute-based encryption with direct user revocation', *International Journal of Embedded Systems*, Vol. 9, No. 6, p.495.

Meddah, N. and Toumanari, A. (2016) 'Reinforce cloud computing access control with key policy attribute-based anonymous proxy reencryption', *International Journal of Cloud Computing*, Vol. 5, No. 3, p.187.

Rahulamathavan, Y., Veluru, S., Han, J., Lu, R., Li, F. and Rajarajan, M. (2016) 'User collusion avoidance scheme for privacy-preserving decentralized key-policy attribute-based encryption', *IEEE Transactions on Computers*, Vol. 65, No. 9, pp.2939–2946.

Sahai, A. and Waters, B. (2005). 'Fuzzy identity-based encryption', *International Conference on Theory and Applications of Cryptographic Techniques*, pp.457–473.

Varadarajan, S., Lal, R. and Zmudzinski, K.C. (2016) *System and Method for Providing Global Platform Compliant Trusted Execution Environment*, US 20160191246 A1, Patent.

Xu, R. and Lang, B. (2015) 'A CP-ABE scheme with hidden policy and its application in cloud computing', *International Journal of Cloud Computing*, Vol. 4, No. 4, pp.279–298..

Zhang, Y., Zheng, D., Chen, X., Li, J. and Li, H. (2014) 'Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts', Vol. 8782, pp.259–273.