TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 03/13 pp579–588
DOI: 10.26599/TST.2019.9010066
Volume 25, Number 5, October 2020

An Attribute-Based Encryption Scheme Based on Unrecognizable Trapdoors

Ruizhong Du*, Ailun Tan, and Junfeng Tian

Abstract: Attribute-Based Encryption (ABE) has been widely used for ciphertext retrieval in the cloud environment. However, bi-flexible attribute control and privacy keywords are difficult problems that have yet to be solved. In this paper, we introduce the denial of access policy and the mutual matching algorithm of a dataset used to realize bidirectional control of attributes in the cloud server. To solve the problem of keyword privacy, we construct a security trapdoor by adding random numbers that effectively resist keyword guessing attacks from cloud servers and external attackers. System security is reduced to the Deterministic Bilinear Diffie-Hellman (DBDH) hypothesis problem. We validate our scheme through theoretical security analysis and experimental verification. Experiments are conducted on a real dataset, and results show that the scheme has higher security and retrieval efficiency than previous methods.

Key words: Attribute-Based Encryption (ABE); unrecognizable trapdoor; two-way access strategy; ciphertext search

1 Introduction

Cloud storage is a mainstream online storage method that removes the hardware and management overhead of users' local storage and takes physical data out of users' control. In this environment, data security is greatly threatened. Data encryption is generally adopted to solve the data security problem in cloud storage, but the availability of encrypted data is limited, especially because the widely used keyword retrieval technology of plaintext information cannot be directly applied to encrypted data. Searchable encryption is a good solution for ciphertext retrieval and has three main research directions: security, accuracy, and efficiency.

1.1 Related work

To enhance the security of data on a server, Song et al.[1] first proposed a searchable encryption scheme with a one-to-one mechanism. However, the multi-user environment is challenging for searchable encryption. Boneh et al.[2] proposed a searchable encryption scheme based on public key cryptography and proved that the public key searchable encryption system satisfies semantic security, but the scheme cannot resist keyword guessing attacks. A keyword guessing attack is an attack mode in searchable encryption that mainly involves statistical analysis of the uploaded trapdoors. The keywords generate fixed trapdoors and can thus be detected by the server or the attacker during mass uploads. Fang et al.[3] proposed a public key encryption scheme that resists keyword guessing attacks without random prediction, but server internal attacks are the weakness of this scheme. Shao and Yang[4] also proposed a scheme that can resist server keyword guessing attack. The best defense against keyword guessing attacks is to make the trapdoors unrecognizable, which means that the same keyword produces a different trapdoor each time. Privacy protection of keywords in searchable encryption mainly ensures the privacy protection of trapdoors. How to construct secure trapdoors is a difficult problem in searchable encryption.

To make ciphertext retrieval more flexible and

• Ruizhong Du and Ailun Tan are with the Cyberspace Security and Computer College, Hebei University, Baoding 071002, China. Ruizhong Du is also with the Key Laboratory on High Trusted Information System in Hebei Province, Baoding 071002, China. E-mail: drzh@hbu.edu.cn; 502317581@qq.com.
• Junfeng Tian are with the Key Laboratory on High Trusted Information System in Hebei Province, Baoding 071002, China. E-mail: tjf@hbu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2019-10-24; accepted: 2019-11-04

efficient, Attribute-Based Encryption (ABE)[5] is proposed. Among them, the Ciphertext-Policy Attribute-Based Encryption (CP-ABE)[6] can embed the access control strategy in ciphertext and control the access of user attributes flexibly, but introduce the problem of attribute cancellation. Data in the cloud environment are dynamic and massive. Hence, the attribute-based encryption scheme needs to modify attributes and policies efficiently. Pirretti et al.[7] proposed a secure attribute-based system scheme that could revoke attributes. By setting an expiration date for attributes, the authorizing agency periodically updates the attribute version and revokes user attributes by revoking the latest version of an attribute. However, attribute-based encryption schemes have security, accuracy, and efficiency problems. To solve the security problem, Hur and Noh[8] proposed attribute-based encryption scheme with attributes and user revocation capabilities, which enhances the forward and backward security of user access control and has attribute revocation ability. Li et al.[9] proposed a searchable ciphertext policy attribute-based encryption scheme, which has the ability to revoke attributes in the Cloud Server (CS). To protect the privacy of users, Ma et al.[10] proposed a privacy-preserving multi-authority CP-ABE with a revocation scheme based on privacy protection, which not only revokes attributes but also protects users' privacy effectively.

To improve the accuracy of attribute revocation in searchable encryption schemes, Yang et al.[11] proposed attribute-based fine-grained access control with an efficient revocation scheme for cloud storage systems. This scheme does not require any cooperative access control by the server, and the Data Owner (DO) does not need to be online in real time. However, the program proves its security under random prediction only. Zu et al.[12] proposed a new CP-ABE with an efficient revocation scheme. The access structure of the scheme adopts the Linear Secret Sharing Scheme (LSSS) mode and has strong performance capability. For more granular access, Sun et al.[13] used CP-ABE and proxy re-encryption to implement file-level access authorization and support data User (U) attribute revocation. Cui et al.[14] proposed ABE with an expressive and authorized keyword search scheme that is more accurate and achieves fine-grained access control of encrypted data in the cloud.

In terms of revocation efficiency, Xue et al.[15] proposed a ciphertext comparable attribute-based encryption scheme based on 0-1 encoding and an efficient construction method based on the generation and management of subattributes with 0 and 1 encoding concepts to reduce communication and computational overhead. Outsourcing computing can greatly reduce user online computing cost. Hence, Chen et al.[16] proposed an online-offline ciphertext policy attribute-based searchable encryption scheme that uses offline preprocessing outsourcing decryption and reduces user online computing cost while improving efficiency. To reduce the computational cost of outsourcing decryption, Zhao et al.[17] proposed a constant cipher-sized attribute-based encryption scheme. The scheme's ciphertext size is constant, which not only improves the outsourcing computational efficiency but also makes the system efficient.

Research on ciphertext retrieval has become increasingly diversified. Qian et al.[18] proposed a ciphertext policy attribute-based searchable encryption scheme for multi-authorization centers in the cloud environment. The scheme uses re-encryption technology to update the ciphertext during the process of attribute revocation, and a multi-authorization center effectively improves the overall efficiency of the program. Broadcast encryption is a typical one-to-many mode. Canard et al.[19] combined broadcast encryption with attribute-based encryption to form a new secret sharing method for one-to-many searchable encryption modes. Xue et al.[20] combined deterministic deletion with attribute-based encryption to propose an attribute-based ciphertext retrieval scheme that supports revocation.

1.2 Our contribution

To solve the problem of trapdoor security and attribute fine-grained access in existing schemes, we propose an Attribute-Based Encryption scheme based on Unrecognizable trapdoors (U-ABE). Our scheme implements one-time trapdoor construction and introduces a denial of access policy that makes attribute control more flexible. The main contributions of this work are as follows.

(1) To solve the trapdoor safety problem, we use the bilinearity of bilinear mapping to construct a one-time trapdoor and prove that the trapdoor is unidentifiable.

(2) To improve the flexible control of access policy, we introduce a denial of access policy that is controlled in both directions.

(3) Through theoretical security analysis, our scheme satisfies the indistinguishable Keywords Guessing Attack (called IND-KGA) and the statistically

indistinguishability under Chosen Ciphertex Attack (called IND-CCA) secure in the Decisional Bilinear Diffie-Hellman (DBDH).

1.3 Organization of this article

The main parts of this paper are organized as follows: Section 2 presents relevant basic knowledge. Section 3 briefly introduces the U-ABE scheme. Section 4 introduces the U-ABE scheme in detail. Section 5 presents an analysis of the accuracy and security of the scheme. Section 6 provides the theoretical and experimental analyses of the scheme. Section 7 summarizes this article.

2 Preliminary

2.1 Bilinear pairing

It is assumed that groups G and G_T are cyclic groups whose order is prime p.g is the generator of group G, and there exists a bilinear map $\hat{e} : G \times G \to G_T$, that satisfies the following properties:

(1) **Bilinearity:** For any $x, y \in G$ and $a, b \in \mathbf{Z}_p$, we have $\hat{e}(x^a, y^b) = \hat{e}(x^b, y^a) = \hat{e}(x, y)^{ab}$, where $\mathbf{Z}_p$ is the set of nonnegative integers less than p.

(2) **Non-degeneracy:** $\hat{e}(g, g) \neq 1$.

(3) **Computability:** There is an efficient algorithm to compute $\hat{e}(x, y)$ for any $x, y \in G$.

2.2 DBDH

Let groups G_1 and G_2 and map $\hat{e} : G_1 \times G_1 \to G_2$, g is the generator of group G_1, randomly generate $(a, b, c, z) \leftarrow \mathbf{Z}_p$, and generate two quintuples $T_0 = (g, A = g^a, B = g^b, C = g^c, Z = \hat{e}(g, g)^z)$ and $T_1 = (g, A = g^a, B = g^b, C = g^c, Z = \hat{e}(g, g)^{abc})$. Write the two five-tuples as

$$P_{\mathrm{BDH}} = (g, g^a, g^b, g^c, \hat{e}(g, g)^{abc}),$$
$$R_{\mathrm{BDH}} = (g, g^a, g^b, g^c, \hat{e}(g, g)^z).$$

DBDH assumption: There is an adversary without polynomial time, who can at least distinguish the quintuples P_{BDH} and R_{BDH} with a non-negligible advantage ϵ.

2.3 List of symbols in the text

The meanings of the symbols in the system are as follows:

B: Encryption parameter
B_i: Decryption parameter
t_1: Allow access policy
t_2: Denial of access policy
U_1: Encrypted allowed access policy
U_2: Encrypted denial of access policy
CT: Ciphertext, including C_m and C_k
V: Version information of table
φ: Keyword index
w: Data keyword collection
w_i: Retrieve keyword collection
Sk_a: Attribute private key set
T_w: Keyword trap
p: Matching information

2.4 Security model

The security model includes the statistically IND-KGA secure and IND-CCA secure.

Definition 1 The statistically IND-KGA secure allows Adversary A to execute the keyword guessing attack to distinguish the trapdoors corresponding to w_0 and w_1, Adversary A and Challenger C perform the game as follows.

Setup: Given the security parameter λ, Challenger C runs the initialization algorithm to generate the public parameter *par*.

Phase 1: Adversary A runs the Trap(w, par) algorithm multiple times.

Challenge: Adversary A randomly selects two keywords w_0 and w_1 from the keyword space, then sends them to Challenger C. Challenger C flips a random coin $\mu \in (0, 1)$, runs the algorithm Trap(w, par), and finally sends the trapdoor T_{w_μ} to Adversary A.

Phase 2: Same as Phase 1.

Guess: Adversary A outputs a keyword μ', and if $\mu = \mu'$, Adversary A wins the security game.

Definition 2 The security model includeds the statistically IND-KGA secure and IND-CCA secure.

Setup: Given security parameter λ, Challenger C executes the setup algorithm Init(l^λ) to generate the public parameter *par*. Given groups G_1 and G_2 and map $\hat{e} : G_1 \times G_1 \to G_2$. Challenger C randomly generates $(a, b, c, z) \to \mathbf{Z}_p$ to generate a quintuple T_0, $T_0 = (g, A = g^a, B = g^b, C = g^c, Z = \hat{e}(g, g)^z)$.

Phase 1: Adversary A runs the encryption algorithm multiple times.

Challenge: Adversary A randomly selects two keywords m_0 and m_1 from the plaintext space, then sends them to Challenger C. Challenger C flips a random coin $\mu \in (0, 1)$, runs the encryption algorithm, and finally sends the ciphertext CT to Adversary A.

Phase 2: Same as Phase 1.

Guess: Adversary A outputs a plaintext μ, if $\mu = \mu'$,

Adversary A wins the security game.

3 System Model

The system model of the scheme is shown in Fig. 1. The scheme includes four entities: DO, CS, U, and AA.

(1) AA: Assuming that the attribute authority is trusted, its main tasks are to generate a random table of attributes, encrypt the policy uploaded by the DO, and calculate the encryption parameters. The attributes uploaded by the U are calculated according to the table, and the attribute private key and decryption parameters are obtained.

(2) DO: The main task of the DO is to encrypt the data by using traditional symmetric encryption. The addition of random numbers generates an unrecognizable index, and the data authority interacts with the AA to obtain an access strategy.

(3) U: The main tasks of U are to generate an unrecognizable random trapdoor, perform data interaction with the CS, obtain the version number returned by the server, and upload the version number and its own attributes to the AA. The AA then calculates and returns the private attribute key and decrypts the parameter. Then, the private attribute key is uploaded to the CS and the ciphertext is returned after the server verifies.

(4) CS: The main tasks of the CS are to receive the trapdoor uploaded by the U, deliver the ciphertext version number to the U, receive the attribute private key uploaded by the U, and deliver the ciphertext to the U after the matching operation.

The following is an introduction to the algorithms used in this article. There are a total of seven algorithms.

(1) Setup(1^λ) $\rightarrow$ (Par, T): The trusted AA runs the algorithm, inputs the security parameter λ, and outputs the public parameter par and the random attribute table. The random attribute table is privately owned and regularly updated by the AA.

(2) EncT(t_1, t_2, ss) $\rightarrow$ (U_1, U_2, B): The algorithm inputs t_1, t_2, and ss, and outputs U_1, U_2, and B. t_1 is the set of allowed access policies uploaded by the DO to the AA. t_2 is the set of denial of access policies uploaded by the DO to the AA. ss is the coordinates where attributes in the property authority query t_1 and t_2 set are positioned in the attribute table. U_1 and U_2 are encrypted from t_1 and t_2, respectively. After ss encryption, the encryption parameter B is obtained, and the version information V of the table is embedded into encryption parameter B. The AA returns the ciphertext to the DO.

(3) Enc(m, k, w, B, par) $\rightarrow$ (CT, φ): The DO calculates the algorithm. m, k, w, B, and par are inputted, CT and φ are outputted. m is plaintext data, k is a key symmetrically encrypted for plaintext, w is the set of keywords contained in the data, and B is the encryption parameter. The ciphertext CT includes two ciphertexts: one is C_m, which is obtained by symmetrically encrypting the plaintext m, and the other is C_k, which is obtained by encrypting the key k. φ is an index obtained by encrypting the keyword set w.

(4) Trap(w, par) $\rightarrow$ T_w: The user calculates the algorithm. The w and par are inputted, and the T_w is outputted. w is the keyword when the user queries. After the calculation, the keyword trapdoor T_w is obtained and uploaded to the CS for retrieval.

(5) KeyGen(att, V) $\rightarrow$ SK_a: The AA calculates the algorithm. att and V are inputted, and SK_a is outputted. att is the attribute uploaded by the user, V is the version information, and SK_a is the attribute parameter.

(6) Search($U_1, U_2, \varphi, T_w, SK_a$) $\rightarrow$ 1 or 0: The server runs the algorithm for matching retrieval. The algorithm is divided into two parts in the system. The first stage

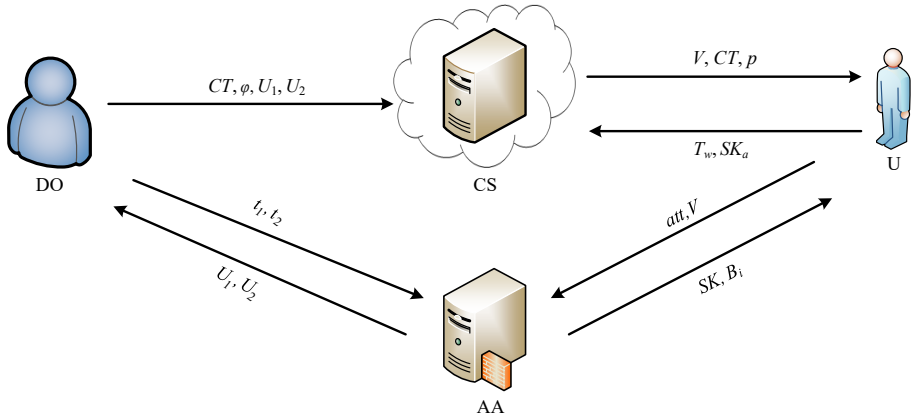

Fig. 1 System model.

is to perform a keyword search to obtain version information V. The second stage is to match the attribute with the access policy. The attribute must not intersect with the attribute set in the denied access policy, and it also contains the set of attributes in the allowed access policy. When both stages meet the requirements, the server sends a ciphertext CT to the user.

(7) Dec$(CT, p, B_i) \rightarrow m$: The user calculates the algorithm, inputs ciphertext CT, matches information p, and decrypts parameter B_i. The plaintext m is obtained by calculation.

4 System Specification

Setup$(1^\lambda) \rightarrow (Par, T)$: Given the security parameter λ, the trusted AA running the algorithm outputs the bilinear mapping par, random numbers $(a, b) \in \mathbf{Z}_q$ are generated, and a hash function $H : (0,1)^* \rightarrow G_1$ is defined. Finally, the public parameter par is outputted,

$$par = (a, b, g, g^a, g^b, G_1, G_2, \hat{e}, q, H).$$

The AA generates an attribute table, which is private to the AA, and various attributes are placed in the table, as shown in Table 1. Its coordinates represent this attribute in the calculation. The coordinates are composed of random numbers, and the coordinate data are periodically replaced. When the coordinate data in Table 1 are replaced, the AA calculates the new version information V.

EncT$(t_1, t_2, ss) \rightarrow (U_1, U_2, B)$: The AA receives the allowed access policy t_1 and the denial of access policy t_2 uploaded by the DO, and then the AA searches in Table 1 to obtain the coordinate data ss,

$$ss = [(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i),$$
$$(x_1, y_1), (x_2, y_2), \ldots, (x_u, y_u)],\ 1 \leqslant i, u \leqslant n.$$

The AA calculates U_1, U_2, and encryption parameter B using the coordinate data in ss,

$$U_1 = [(g^{x_1}, g^{y_1}), (g^{x_2}, g^{y_2}), \ldots, (g^{x_i}, g^{y_i})],$$

Table 1 Property list instance.

y	x						
	x_1	x_2	x_3	x_4	x_5	$\cdots$	x_n
y_1	A_{11}	A_{21}	A_{31}	A_{41}	A_{51}	$\cdots$	A_{n1}
y_2	A_{12}	A_{22}	A_{32}	A_{42}	A_{52}	$\cdots$	A_{n2}
y_3	A_{13}	A_{23}	A_{33}	A_{43}	A_{53}	$\cdots$	A_{n3}
y_4	A_{14}	A_{24}	A_{34}	A_{44}	A_{54}	$\cdots$	A_{n4}
y_5	A_{15}	A_{25}	A_{35}	A_{45}	A_{55}	$\cdots$	A_{n5}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
y_n	A_{1n}	A_{2n}	A_{3n}	A_{4n}	A_{5n}	$\cdots$	A_{nn}

$$U_2 = [(g^{x_1}, g^{y_1}), (g^{x_2}, g^{y_2}), \ldots, (g^{x_u}, g^{y_u})],$$

$$B = \prod_1^n H(x_i^{y_i}).$$

Enc$(m, k, w, B, par) \rightarrow (CT, \varphi)$: The ciphertext CT contains ciphertext C_k and C_m, $CT = (C_k, C_m)$, and the DO uses symmetric encryption to encrypt plaintext m to obtain ciphertext C_m. The encryption key is k, key k is encrypted, a random number $t \leftarrow \mathbf{Z}_p$ is generated, and then calculate

$$C_k = [\hat{e}(g^a, g^b)^t \times k, g^t, B^t].$$

The DO needs to establish an index in the data. Where the set of keywords is w. Then, the random number set $S \leftarrow \mathbf{Z}_p$ is generated, and then calculate

$$\varphi = [(g^{s_1}, w_1^{s_1}), (g^{s_2}, w_2^{s_2}), \ldots, (g^{s_n}, w_n^{s_n})].$$

Trap$(w, par) \rightarrow T_w$: The algorithm inputs the keyword set w that needs to be inquired, generates the random number set $L \leftarrow \mathbf{Z}_p$, and calculates:

$$T_w = [(g^{l_1}, w_{i_1}^{l_1}), (g^{l_2}, w_{i_2}^{l_2}), \ldots, (g^{l_n}, w_{i_n}^{l_n})].$$

KeyGen$(att, V) \rightarrow SK_a$: After data interaction between the user and server, the user receives the version information V sent by the CS, and the user sends the att together with the version information V to the AA. The AA searches the corresponding attribute data in Table 1 of the corresponding version V of the user attribute att, and then calculates the attribute data SK_a from the coordinate data,

$$SK_a = [(g^{\alpha_1}, g^{\beta_1}), (g^{\alpha_2}, g^{\beta_2}), \ldots, (g^{\alpha_n}, g^{\beta_n})].$$

Search$(U_1, U_2, T_w, SK_a) \rightarrow 1$ or 0, the retrieval is divided into two phases.

Phase 1: The user uploads trapdoor T_w to the CS, and the server matches the trapdoor with the index. The calculation process is as follows:

$$\hat{e}(g^{l_1}, w_1^{s_1}) = \hat{e}(g^{s_1}, w_{i_1}^{l_1}).$$

If the equation is true, the proof is the same keyword. If the equation is false, the next keyword is evaluated until the last keyword in the index. The server sends the version information V of the data to the user.

Phase 2: The user uploads the attribute private key SK_a to the server, and the CS matches the attribute with the access policy. Given that two access policies exist in this solution, the attributes contained in the denial of access policy must not be included by the user attribute private key. The user attribute private key SK_a needs to be matched with the denial of access policy U_2 first, and the following is calculated:

$$U_2 \cap SK_a = [(g^{x_1}, g^{y_1}), (g^{x_2}, g^{y_2}), \ldots, (g^{x_u}, g^{y_u}) \cap (g^{\alpha_1}, g^{\beta_1}), (g^{\alpha_2}, g^{\beta_2}), \ldots, (g^{\alpha_n}, g^{\beta_n})].$$

If the matching result is not an empty set, then 0 is output. If the matching result is an empty set, then the user attribute private key SK_a matches the allowed access policy U_1. The calculation is as follows:

$$U_1 \cap SK_a = [(g^{x_1}, g^{y_1}), (g^{x_2}, g^{y_2}), \ldots, (g^{x_i}, g^{y_i}) \cap (g^{\alpha_1}, g^{\beta_1}), (g^{\alpha_2}, g^{\beta_2}), \ldots, (g^{\alpha_n}, g^{\beta_n})].$$

When the result is U_1, the server outputs 1 and sends the ciphertext CT. The server packages the successfully matched attributes to generate matching information p.

$\mathrm{Dec}(CT, p, B_i) \rightarrow m$: The user uses the algorithm to decrypt the ciphertext CT sent by the server. First, the user obtains the matching attribute information from the matching information p, calculates the decryption parameter, and generates a random number $r \leftarrow \mathbf{Z}_p$. The calculation is as follow:

$$\hat{e}(g^a, g^b)^t \times k \times \frac{\hat{e}(g^r, B^t)}{\hat{e}(g^{ba} B_i^r, g^t)} =$$

$$\hat{e}(g^a, g^b)^t \times k \times \frac{\hat{e}(g^r, B^t)}{\hat{e}(g^{ba}, g^t)\hat{e}(B_i^r, g^t)} =$$

$$\hat{e}(g^a, g^b)^t \times k \times \frac{\hat{e}(g^r, B^t)}{\hat{e}(g^{ba}, g^t)\hat{e}(B_i, g)^{rt}} =$$

$$\hat{e}(g, g)^{abt} \times k \times \frac{\hat{e}(g, B)^{rt}}{\hat{e}(g, g)^{abt}\hat{e}(B_i, g)^{rt}} = k.$$

Then, plaintext m is solved by using the symmetric decryption algorithm.

5 Security Analysis

The scheme can ensure data security. The data are encrypted by a symmetric algorithm, and the key k uses public key encryption to obtain ciphertext C_k. The keyword trapdoor is randomly encrypted; thus the scheme is statistically IND-KGA secure. In addition, the ciphertext C_k is constructed according to the DBDH assumption, and the proposed scheme is statistically IND-CCA secure under the DBDH assumption.

Theorem 1 Indicates that the proposed scheme is statistically IND-KGA secure.

Proof The keyword trapdoor is unrecognizable.

Setup: Challenger C generates a random number $(a, b) \rightarrow \mathbf{Z}_p$, and the public parameters $par = (a, b, g, g^a, g^b, G_1, G_2, \hat{e}, q, H)$.

Phase 1: Adversary A selects the keyword set $(w_1, w_2, \ldots, w_n)$ and sends it to Challenger C. The Challenger outputs the trapdoor set $(T_{w_1}, T_{w_2}, \ldots, T_{w_n})$ generated by the keyword set and sends it to Adversary A.

Challenge: Adversary A selects keywords w_0 and w_1 that are not in the keyword set in Phase 1 from the keyword space. Challenger C flips a random coin $\mu \in (0, 1)$, runs the algorithm $\mathrm{Trap}(w, par)$, and finally, sends the trapdoor T_{w_μ} to Adversary A.

Phase 2: Adversary A sends the keyword set to Challenger C again, the same as in Phase 1.

Guess: Adversary A outputs μ'. If $\mu' = \mu$, Adversary A wins the game. The scheme is statistically IND-KGA secure. The keyword trapdoor introduces random numbers during encryption; thus the trapdoors generated by the same keyword are different, which can effectively resist a statistical keyword guessing attack. ■

Theorem 2 The proposed scheme is statistically IND-CCA secure under the DBDH assumption. Adversary A and Challenger C perform the game as follows.

Proof Adversary A cannot recognize the ciphertext under the DBDH assumption.

Setup: The system is established, the security parameter λ is generated, and then the algorithm $\mathrm{Setup}(1^\lambda)$ is run to obtain the security parameters $par = (a, b, g, g^a, g^b, G_1, G_2, \hat{e}, q, H)$ and the encryption parameter B in the system. Given groups G_1, G_2, and map $\hat{e} : G_1 \times G_1 \rightarrow G_2$, Challenger C randomly generates $(a, b, c, z) \leftarrow \mathbf{Z}_p$ to generate a quintuple T_0. $T_0 = (g, A = g^a, B = g^b, C = g^c, Z = \hat{e}(g, g)^z)$.

Phase 1: Adversary A runs encryption algorithm multiple times.

Challenge: Adversary A randomly selects two keywords m_0 and m_1 from the plaintext space and sends them to Challenger C. Challenger C flips a random coin $\mu \in (0, 1)$, runs the algorithm $\mathrm{Enc}(m, k, w, B, par) \rightarrow (CT, \varphi)$, and finally, sends the ciphertext CT to Adversary A.

Phase 2: Same as Phase 1.

Guess: Adversary A outputs a plaintext μ', if $\mu = \mu'$, Adversary A wins the security game. ■

According to the above proof, we conclude that our proposed scheme is statistically IND-CCA secure under the DBDH assumption. The ciphertext structure in the scheme is similar to the five-tuple structure in the DBDH assumption and has unrecognizable properties.

6 Performance Analysis

6.1 Theoretical analysis

In theory, three main aspects of the U-ABE scheme

are compared with those of several other schemes: functionality, storage cost, and communication cost. The symbols in the comparison process are defined as follows: $|p|$ indicates the length of the data element in $\mathbf{Z}_p$; $|g|$ indicates the length of the data element in G; $|g_T|$ indicates the length of the data element in G_T; n_u indicates the number of attributes associated with the user; n_c indicates the number of attributes associated with the ciphertext; n_k indicates the number of attributes in the user key; and n_a indicates the number of attributes of the entire system.

6.1.1 Functional analysis

Table 2 compares the functional differences between the U-ABE scheme and the other three schemes. The revocation mechanism of each scheme is immediately revoked. The difference between U-ABE and the other three schemes lies in the revocation direction. The U-ABE scheme has a denial of access policy and can thus be revoked in both directions. The U-ABE scheme adopts the AND mode for access strategy, whereas the other schemes adopt the Tree or the LSSS mode, which consumes less resources and is more efficient.

6.1.2 Storage cost

In Table 3, the U-ABE scheme is compared with schemes in Refs. [13, 18, 20] for storage cost. It is mainly divided into four components for comparison: AA, DO, CS, and U. In the U-ABE scheme, the main role of the AA is to generate attribute tables, encrypt two-way access policy, and assign private keys. The AA mainly stores random numbers and tables. Thus the storage cost of the AA in the U-ABE scheme is calculated as $(2n_a + 1)|p|$. In terms of DO, the main job of the DO in the scheme is to receive the policy ciphertext, generate the index, encrypt the data, and then upload it to the CS. The storage cost of the DO is calculated as $2|p| + |g|$, which is smaller than the storage cost of schemes in Refs. [13, 18, 20]. On the CS side, the main task of the CS in the U-ABE scheme is to receive the ciphertext data uploaded by the DO and the search information and attribute private key uploaded by the U, and then matching calculation is performed between the two. Therefore, the storage cost is calculated as $(n_c + n_k)|g| + 2|g_T|$. Unlike schemes in Refs. [13,20], the U-ABE scheme reduces the storage cost of the CS. Finally, in the U aspect, the U's job in the scheme is to receive the attribute private key SK_a from the AA, calculate the attribute private key SK_a, generate the trapdoor, and decrypt the ciphertext. Therefore, the U storage cost is calculated as $2n_k + |p|$, which is lower than the schemes in Refs. [13, 18].

6.1.3 Communication cost

Table 4 shows a theoretical analysis of communication cost, which is mainly divided into the data transmission cost of four lines. The first is AA & U. In U-ABE, the AA and U mainly transmit attributes and attribute parameters; therefore, the calculated communication cost is $2n_k + n_k|p|$. Followed by AA and O, in the U-ABE scheme, the plaintext, ciphertext, and encryption parameters of the two-way access policy are mainly transmitted between the AA and the DO, so the

Table 2 Function comparison.

Program	Access structure	Revocation method	Safe question	Revoke direction
Scheme in Ref. [18]	LSSS	Immediate	q-parallel BDHE	Forward
ABKS-UR[13]	Tree	Immediate	DBDH	Forward
AD-KP-ABE[20]	Tree	Immediate	DBDH	Forward
U-ABE	AND	Immediate	DBDH	Two-way

Table 3 Cost comparison.

Program	AA	DO	CS	U
Scheme in Ref. [18]	$(4 + n_a)\|p\|$	$(2 + n_a)\|g\| + \|g_T\|$	$\|g_T\| + (3n_c + 1)\|g\|$	$(2 + n_k)\|g\|$
ABKS-UR[13]	–	$(3 + n_a)\|g\| + \|g_T\|$	$\|g_T\|(n_c + 2) + \|g\|(n_a + 1)$	$2n_u\|g\| + \lg(n_u + 2)$
AD-KP-ABE[20]	$2n_a\|p\|$	$n_c\|g\| + \|p\|$	$(n_c + n_k)\|g\| + \|g_T\|(n_c + 2)$	$n_k\|p\|$
U-ABE	$(2n_a + 1)\|p\|$	$2\|p\| + \|g\|$	$(n_c + n_k)\|g\| + 2\|g_T\|$	$2n_k + \|p\|$

Table 4 Communication cost comparison.

Program	AA&U	AA&DO	CS&U	CS&DO
Scheme in Ref. [18]	$4\|g\| + n_k\|g\|$	$2\|g\| + \|g_T\| + n_a\|g\|$	$\|g_T\| + (3n_c + 1)\|g\|$	$(3n_c + 1)\|g\|$
ABKS-UR[13]	–	–	$2\|p\| + (n_c + 3)\|g_T\| + (n_c + 4)/2$	$(2n_k + 1)\|g\| + (n_c + 1)\|g\|$
AD-KP-ABE[20]	$n_k\|p\|$	$n_c\|g\| + n_k\|g_T\|$	$\|p\| + (n_c + 2)\|g\| + n_c\|g_T\|$	$\|p\|n_k + \|g\|$
U-ABE	$2n_k + n_k\|p\|$	$4n_c + \|p\|$	$\|p\| + (n_k + 2)\|g_T\| + n_c\|g\|$	$(\|g_T\| + 1)n_c + (n_c + 1)\|p\|$

communication cost can be calculated as $4n_c + |p|$. The comparison of the scheme in this paper with schemes in Refs. [18,20] shows that communication costs are greatly reduced. In the CS and U process, the server in the U-ABE scheme has two data interactions with the data user, mainly the transmission of the trapdoor ciphertext and attribute private key, thus the communication cost is $|p| + (n_k + 1)|g_T| + n_c|g|$, which is superior to the other three schemes. Finally, in the CS and DO process, the DO in the U-ABE scheme unilaterally uploads the ciphertext data to the CS. Thus the communication cost is calculated as $(|g_T| + 1)n_c + (n_c + 1)|p|$.

6.2 Experiment analysis

The experimental platform is a 64 bits Windows operating system, on an Intel Core(TM) i5-4570 3.20 GHz CPU with 8.00 GB memory. The experimental code is modified and written based on Pairing-Based Cryptography library (PBC), using a super singular curve in Class A, that is $E(F_q) : y^2 = x^3 + x$. Group G_s is subgroup of $E(F_q)$, the order of the group G_s is 160 bits, and the base field is 58 bits. This experiment is conducted with four considerations: encryption time, private key generation overhead, retrieval time, and decryption time. The relationship among the number of attributes, the number of keywords, and the time cost is tested.

6.2.1 Encryption time

Figure 2 shows a comparison of the encryption time of the U-ABE scheme with schemes in Ref. [18], ABKS-UR[13], and AD-KP-ABE[20]. The analysis shows that the U-ABE is superior to the scheme in Ref. [18] and ABKS-UR[13].

6.2.2 Private key generation time

Figure 3 shows a comparison of the private key generation overhead of the U-ABE, the scheme in Ref. [18], and ABKS-UR[13]. Figure 3 indicates that as the number of attributes submitted by users increases, the private key generation time increases linearly. The private key of U-ABE is generated by hashing and exponential operation, which gives this scheme higher computational efficiency than the other two schemes.

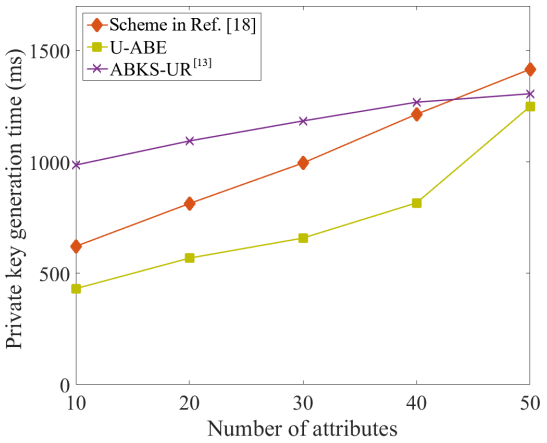

Fig. 3 Private key cost experiment comparison.

6.2.3 Search time

Figure 4 shows the results of a comparison experiment of the retrieval cost for U-ABE, scheme in Ref. [18], and ABKS-UR[13]. The comparison experiment in Fig. 4 is the effect of the number of attributes in the user's private key on the retrieval time. Also, the keyword is set to 10, because the attribute matching of U-ABE is matched in the form of a set, the time overhead is much smaller than that of the other two schemes. Although U-ABE joins the random number calculation that forms an unrecognizable trapdoor, because only one bilinear calculation and one exponential operation are used in the trapdoor construction process, the calculation amount is

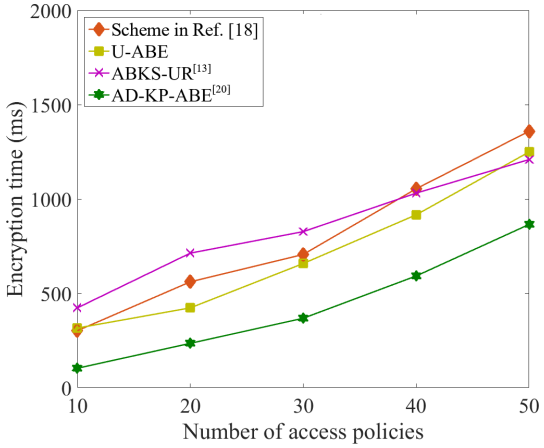

Fig. 2 Encryption time experiment comparison.

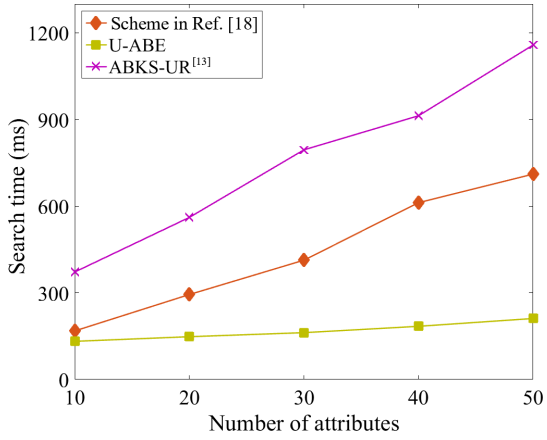

Fig. 4 Search time experiment comparison.

less than that of the other two schemes.

6.2.4 Decryption time

Figure 5 shows a comparison of the decryption time of U-ABE with scheme in Ref. [18] and AD-KP-ABE[20]. As seen from Fig. 5, the decryption time increases as the number of attributes in the user's private key increases. The analysis shows that U-ABE has obvious advantages over the scheme in Ref. [18] in terms of decryption time. When the number of attributes in the user's private key reaches 50, the decryption time of U-ABE is less than 1 s.

The decryption time in U-ABE increases as the number of attributes in the private key increases mainly because of the calculation of the decryption parameters, but the decryption algorithm is only a multiplication operation and the decryption time expansion rate is low.

7 Conclusion and Future Work

In this paper, we propose a ciphertext retrieval scheme called U-ABE based on unrecognizable trapdoors. First, we introduce a denial of access policy that can implement two-way revocation to implement the flexible control of access attributes. Moreover, because the access policy is not embedded in the ciphertext, flexible attribute revocation can be completed by modifying the access policy. Second, we introduce the mechanism of the attribute table. Through the reliable AA, an attribute table with random coordinates is created. The attributes of the users and owners are based on Table 1, which reduces the matching calculation time. We finally use the bilinearity of the bilinear pair to construct a one-time trapdoor that realizes the same keyword with different encryption results each time, ensuring keyword privacy.

Through theoretical safety analysis and experimental verification, we prove that our scheme has high security and retrieval efficiency in real data sets. Future work will be performed to improve the solution in terms of AA credibility, accuracy, and efficiency, and to conduct a profound study of trapdoor security issues with the goal of achieving higher security.

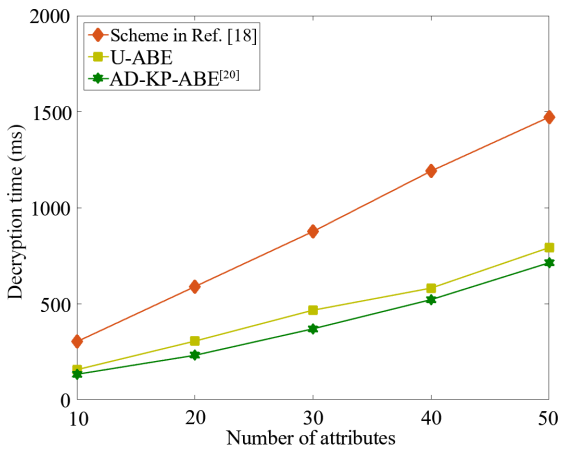

Fig. 5 Decryption time comparison experiment.

References

[1] D. Song, D. Wagner, and A. Perrig, Practical techniques for searches on encrypted data, in *Proc. of IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, 2000, pp. 44–55.

[2] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, Public key encryption with keyword search, in *Proc. of Advances in Cryptology-Eurocrypt*, Interlaken, Switzerland, 2004, pp. 506–522.

[3] L. M. Fang, W. Susilo, and C. Ge, Public key encryption with keyword search secure against keyword guessing attacks without random oracle, *Information Sciences*, vol. 238, no. 7, pp. 221–241, 2017.

[4] Z. Y. Shao and B. Yang, On security against the server in designated tester public key encryption with keyword search, *Information Processing Letters*, vol. 115, no. 8, pp. 1757–1761, 2015.

[5] A. Sahai and B. Waters, Fuzzy identity-based encryption, in *Proc. of International Conference on Theory and Applications of Cryptographic Techniques*, Berlin, Germany, 2005, pp. 457–473.

[6] J. Bethencourt, A. Sahai, and B. Waters, Ciphertext-policy attribute-based encryption, in *Proc. of IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, 2007, pp. 321–334.

[7] M. Pirretti, P. Traynor, P. Mcdaniel, and B. Waters, Secure attribute-based systems, *Journal of Computer Security*, vol. 18, no. 5, pp. 799–837, 2010.

[8] J. Hur and K. D. Noh, Attribute-based access control with efficient revocation in data outsourcing systems, *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 814–821, 2011.

[9] J. Li ,Y. Shi, and Y. Zhang, Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage, *International Journal of Communication Systems*, vol. 30, no. 1, pp. 811–820, 2015.

[10] H. Ma, E. Dong, and Z. Liu, Privacy-preserving multi-authority ciphertext-policy attribute-based encryption with revocation, in *Proc. of International Conference on Broadband and Wireless Computing, Communication and Applications*, Taiwan, China, 2018, pp. 811–820.

[11] K. Yang, X. Jia, and K. Ren, Attribute-based fine-grained access control with efficient revocation in cloud storage systems, in *Proc. of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, Hangzhou, China, 2013, pp. 523–528.

[12] L. Zu, Z. Liu, and J. Li, New ciphertext-policy attribute-based encryption with efficient revocation, in *Proc. of IEEE International Conference on Computer and Information Technology*, Xi'an, China, 2014, pp. 281–287.

[13] W. Sun, S. Yu, and W. Lou, Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud, *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1187–1198, 2016.
[14] H. Cui, H. R. Deng, and K. J. Liu, Attribute-based encryption with expressive and authorized keyword search, in *Proc. of Australasian Conference on Information Security and Privacy*, Auckland, New Zealand, 2017, pp. 106–126.
[15] K. Xue, J. Hong, and Y. Xue, CABE: A new comparable attribute-based encryption construction with 0-encoding and 1-encoding, *IEEE Transactions on Computers*, vol. 66, no. 9, pp. 1491–1503, 2017.
[16] D. D. Chen, Z. Cao, and X. L. Dong, Online/offline ciphertext-policy attribute-based searchable encryption, *Journal of Computer Research and Development*, vol. 53, no. 10, pp. 2365–2375, 2016.
[17] Y. Zhao, M. Ren, and S. Jiang, An efficient and revocable storage CP-ABE scheme in the cloud computing, *Computing*, vol. 101, no. 4, pp. 1–25, 2018.
[18] H. Qian, J. Li, and Y. Zhang, Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation, *International Journal of Information Security*, vol. 14, no. 6, pp. 487–477, 2015.
[19] S. Canard, H. D. Phan, and D. Pointcheval, A new technique for compacting ciphertext in multi-channel broadcast encryption and attribute-based encryption, *Theoretical Computer Science*, vol. 723, no. 5, pp. 51–72, 2018.
[20] L. Xue, Y. Yu, and Y. Li, Efficient attribute-based encryption with attribute revocation for assured data deletion, *Information Sciences*, vol. 479, no. 4, pp. 640–650, 2018.

Ruizhong Du received the MS degree from Hebei University in 2004 and the PhD degree from Wuhan University in 2012. He is a professor and the deputy dean of the School of Cyberspace Security and Computer, Hebei University, a member of the Chinese Computer Society Fault Tolerant Computing Committee and the secretary-general of Hebei Cyberspace Security Society. His research interests include network security, trusted computing, and cloud security.

Junfeng Tian received the BS degree and the MS degree from Hebei University in 1986 and 1995, respectively, and the PhD degree from the University of Science and Technology of China in 2004. He is now the dean of the School of Computer Science, Hebei University. His research interests include distributed computing, network security, and trusted computing.

Ailun Tan received the BS degree from Shenyang University of Technology in 2017. He is now a master student in the School of Network Security and Computer Science, Hebei University. His main research focuses on information security.