Annals of Telecommunications (2019) 74:413–421
https://doi.org/10.1007/s12243-018-0694-8

Secure and flexible keyword search over encrypted data with outsourced decryption in Internet of things

Yinghui Zhang[1,2] · **Axin Wu**[1,2] · **Tiantian Zhang**[1,2] · **Dong Zheng**[1,2]

Received: 12 June 2018 / Accepted: 3 December 2018 / Published online: 14 December 2018

Abstract

The development of Internet of things (IoT) makes data exchange more frequently, and the cloud computing can provide stable storage and efficient computation for data users. To ensure the security and functionality of data, the efficiency of decryption and keyword search should be taken into consideration in resource-constrained IoT scenarios. In order to solve the above problems, a flexible keyword search scheme in IoT is proposed over encrypted data with outsourced decryption. First, the attribute-based encryption technology is applied, by which only users whose attributes meet the access control structure can access the sharing data. Second, the reciprocal mapping of Lagrange polynomials technology is employed to implement keyword search in a large number of ciphertext data. Third, the decryption of ciphertext is outsourced to improve the efficiency of decryption on the client side. The security and performance analysis indicates that the proposed scheme is secure and efficient.

Keywords Internet of things · Attribute-based encryption · Keyword search · Outsourced decryption

1 Introduction

The development of Internet of things (IoT) makes data exchange more frequently. In the IoT scenario, information security is related to enterprises [1] and individuals [2]. In addition, the cloud computing can provide stable storage and efficient computation for data users (smart phones, smart watches, etc). Information stored on cloud servers is exposed to distributed and dynamic network environment [3]. In an open network environment, the security and privacy of data [4] will be greatly threatened in the data storage [5–9], data access [10, 11], data search [12–15], data processing [16], data transmission [17–19] and data sharing [20]. It is a good solution to encrypt data before uploading data. Furthermore, the sharing of data is another very interesting topic. In order to access the data more efficiently, the notion of attribute-base encryption (ABE) [21] is proposed.

In ABE system, the secret key does not involve the identity of users. To prevent the abuse of the secret key, traceable ABE schemes [22–24] are proposed. When a large amount of ciphertext data is stored on the server, data access control [25] and finding the data that users need [26] are problems that have to be taken into consideration. Fortunately, searchable ABE [27, 28] can solve these problems well. ABE, however, requires a lot of computation, especially in the decryption stage. In the ABE system, some work [29, 30] is done to improve the efficiency of ABE. In order to further improve efficiency and make full use of cloud computing technology, a lot of computation can be outsourced to cloud servers to reduce the computational pressure of terminals [31–34]. Therefore,

✉ Yinghui Zhang
yhzhaang@163.com

Axin Wu
waxinsec@163.com

Tiantian Zhang
ttzhng@163.com

Dong Zheng
zhengdong@xupt.edu.cn

[1] National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, People's Republic of China

[2] Westone Cryptologic Research Center, Beijing 100070, China

the outsourcing decryption of ABE [35, 36] can break the bottleneck of the efficiency of ABE.

In order to solve the above problems, we put forward the flexible keyword search scheme over encrypted data with outsourced decryption for IoT. Our scheme has the following three advantages: First, the attribute-based encryption technology is applied, by which only users whose attributes meet the access control structure can access the sharing data. Second, the reciprocal mapping of Lagrange polynomials technology is employed to implement keyword search in a large number of ciphertext data. Third, the decryption of ciphertext is outsourced to improve the efficiency of decryption on the client side.

2 Preliminaries

We give the relevant knowledge that will be used in the construction and proof of the following scheme in this section.

2.1 Bilinear pairing

Let G_1 and G_2 represent the multiplicative cyclic groups with the prime order p. The generator of G_1 is g. A bilinear pair $e : G_1 \times G_1 \rightarrow G_2$ has the following properties [37]:

1. Bilinear: $e(u^a, v^b) = e(u, v)^{ab}$, where $a, b \in Z_p$, $u, v \in G_1$.
2. Non-degenerate: $\exists u, v \in G_1$, make $e(u, v) \neq 1$.
3. Computability: $\forall u, v \in G_1, e(u, v)$ can be calculated in the polynomial time.

2.2 Access control tree

Let x express a node in an access control tree T. For each non-leaf node, there is a relationship between the threshold value k_x and its m child nodes $0 < k_x \leq m$. $k_x = 1$ and $k_x = m$ represent an *or* gate and an *and* gate, respectively. For each leaf node x that is regarded as an attribute, its threshold value is 1. The function $pa(x)$ is defined as the parent of node x. the function $att(x)$ is defined as the attribute value for the leaf node x. The function $index(x)$ returns the number that the node x is numbered from 1 to m.

2.3 Satisfying an access control tree

Let T_x be the subtree of the root node t at node x. Hence, T and T_t are the same. When the attribute set Y satisfies the access control tree T_x, $T_x(Y) = 1$. $T_x(Y)$ is obtained by the following recursive algorithm: For a leaf node, when $att(x) \in Y$, $T_x(Y) = 1$. For a non-leaf node x, if there are at least k_x child nodes returning 1, $T_x(Y) = 1$.

2.4 Lagrange interpolation polynomials

In the following, we can get k polynomials of degree $k-1$ from k different numbers x_i, where $i = 1, 2, ..., k$ [38].

$$f_i(x) = \prod_{1 \leq j \neq i \leq k} \frac{x - x_j}{x_i - x_j} = \sum_{j=1}^{k} a_{i,j} x^{j-1}, \tag{1}$$

$$f_i(x_j) = \begin{cases} 1, & j = i \\ 0, & j \neq i. \end{cases}$$

Then, $F_t(x_i) = x_i^{t-1}$, $1 \leq i \leq k$ can be obtained from the constant $t (1 \leq t \leq k)$ and polynomial $F_t(x) = \sum_{i=1}^{k} x_i^{t-1} f_i(x)$. Obviously, for the two functions $F_t(x)$ and x^{t-1} with the same k points, their maximum degree is $k-1$, and we have

$$F_t(x) = \left(\sum_{i=1}^{k} x_i^{t-1} a_{i,1}\right) + \left(\sum_{i=1}^{k} x_i^{t-1} a_{i,2}\right) x + \cdots + \left(\sum_{i=1}^{k} x_i^{t-1} a_{i,k}\right) x^{k-1} = x^{t-1}, \tag{2}$$

then

$$\sum_{i=1}^{k} x_i^{t-1} a_{i,j} = \begin{cases} 1, & j = t \\ 0, & j \neq t. \end{cases}$$

The two mappings $\hat{r}$ and $\hat{R}$ on G_1^k can be constructed from the definition of $f_i(x)$ as follows.

$\hat{r}$: $G_1^k \rightarrow G_1^k$ is defined as follows: $(r_1, \cdots, r_k) \rightarrow (R_1, \cdots, R_k)$, where $R_j = \prod_{i=1}^{k} r_i^{a_{i,j}}$, $j = 1, 2, \cdots, k$.

$\hat{R}$: $G_1^k \rightarrow G_1^k$ is defined as follows: $(R_1, \cdots, R_k) \rightarrow (r_1, \cdots, r_k)$, where $r_j = \prod_{j=1}^{k} R_j^{x_i^{j-1}}$, $i = 1, 2, \cdots, k$.

$\hat{R}$ and $\hat{r}$ are reciprocal: If G_1, $f_i(x)$, $\hat{R}$ and $\hat{r}$ are as defined above, then $\hat{R}(\hat{r}) = 1$, $\hat{r}(\hat{R}) = 1$. Let $\hat{R}_i$ denote the i-th component of $\hat{R}$, then $\hat{R}_i(R_1, \cdots, R_k) = r_i$.

2.5 The truncated (t, q, ε)-ABDHE assumption

Given $(g, g^a, g^{a^2}, \cdots, g^{a^q}, g', g'^{a^{q+2}}, T)$, where $T \in G_2$, $g' = g^z$, $z \in Z_p$. If $T = e(g, g')^{a^{q+1}}$, output 1, otherwise output 0 [39]. Define an arbitrary polynomial-time adversary B's advantage function $Adv_B^{q-ABDHE}$ as

$$|P_r[B(g, g^a, g^{a^2} \cdots, g^{a^q}, g', g'^{a^{q+2}}, e(g, g')^{a^{q+1}})] - P_r[B(g, g^a, g^{a^2} \cdots, g^{a^q}, g', g'^{a^{q+2}}, T)]|.$$

If in time t, $Adv_B^{q-ABDHE} < \varepsilon$, then the (t, q, ε)-ABDHE assumption is said to hold on (G_1, G_2).

2.6 Decisional BDH assumption

Randomly choose $a, b, c, z \in Z_p$. Suppose g is a generator of G_1. The advantage Adv_B^{DBHD} [40] of an arbitrary polynomial time adversary B is defined as

$$|Pr[B(g^a, g^b, g^c, e(g,g)^{abc}) = 0] - Pr[B(g^a, g^b, g^c, e(g,g)^z) = 0]|.$$

If Adv_B^{DBHD} is negligible, the DBDH assumption holds.

3 Model and definition

We mainly introduce the system architecture, the system scheme and the security model in this part.

3.1 System architecture

The system architecture is shown in Fig. 1. CT is the ciphertext, I is the keyword index, T is the trapdoor, PT is the partial ciphertext, TK is the transformation key, and SK is the secret key. The following entities are involved in this system.

- Data owner (DO): DO provides shared data and specifies the attribute set when encrypting. DO is a terminal device of IoT, such as a smart phone and a smart watch, etc.
- Data user (DU): DU is a data consumer in the system. When the attributes of DU satisfy the access control structure, he can access data. DU is a terminal device of IoT, such as a smart phone and a smart watch, etc.
- Attribute Authority (AA): AA manages DO and DU. At the same time, AA is responsible for the distribution of the secret key and the transformation key. AA is a credible entity.
- Cloud server (CS): CS is responsible for storing the shared ciphertext.
- Computing server (OS): OS is a computing server that can decrypt the ciphertext partially.

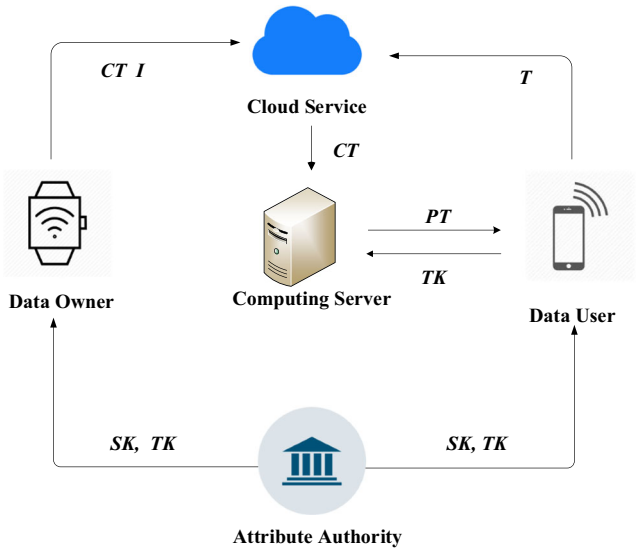

Fig. 1 System architecture

3.2 System scheme

Our scheme is composed of the following algorithms:

- $Setup(U) \rightarrow PK, MK$: The algorithm takes the global attributes U as input with public parameters PK and the master key MK as output.
- $KeyGen(MK, T) \rightarrow TK, SK$: The algorithm takes the master key MK and the access control structure T as input with the transformation key TK and the secret key SK as output.
- $Enc(Y, m, W_1, W_2, \cdots, W_k) \rightarrow CT, I$: The algorithm takes the attribute Y and the message m containing the keywords $W_1, W_2, \cdots, W_k$ as input with the ciphertext CT and the keyword index I as output.
- $Trapdoor(SK, W) \rightarrow T_W$: The algorithm takes the secret key SK and the keyword W as input with the keyword trapdoor T_W as output.
- $Test(I, T_W, x) \rightarrow 0, 1$: The algorithm takes the keyword index I, the keyword trapdoor T_W and the node of user x as input with 0 or 1 as output.
- $Dec(CT, SK, x) \rightarrow m$: The algorithm takes the ciphertext CT, the secret key SK and the node of user x as input with the plaintext m as output.

3.3 Security model

The security model of the scheme is divided into the security model of keywords and the security model of the scheme.

The security model of keywords: A secure model interactive game between an adversary A and a challenger C is as follows:

- Setup: C calls the algorithm $Setup(U) \rightarrow PK, MK$ and returns the PK to A.
- Trapdoor queries: A adaptively asks the C for the trapdoor T_W. C calls the algorithm $Trapdoor(SK, W) \rightarrow T_W$, and returns T_W to A.
- Challenge: A sends a message m containing $k+1$ keywords $W_0, W_1, \cdots, W_k$ to C. C selects $b \in \{0, 1\}$ randomly, and then calls the algorithm $Enc(Y, m, W_b, W_2, \cdots, W_k) \rightarrow CT, I$. C then returns the I to A. In this process, A cannot ask the key trapdoors of W_0 and W_1.
- Phase 2: A does the same inquiry at this stage as the Phase 1.
- Guess: The result guessed $b' \in \{0, 1\}$ is outputted.

The advantage of the adversary A in the game is defined as follows:

$$Pr_A = \left| Pr[b' = b] - \frac{1}{2} \right|.$$

Definition 1 Suppose that an arbitrary polynomial time adversary A in the time t, does inquiries at least q_w times. And A can win the above game with the at most ϵ advantage. Then our scheme is the (t, q_w, ϵ) semantic security.

The security model of the scheme: A secure model interactive game between an adversary A and a challenger C is as follows:

- Init: A declares attributes Y to challenge
- Setup: C calls the algorithm $Setup(U) \rightarrow PK, MK$ and returns the PK to A.
- Phase 1: A queries to C the secret key SK related to the access structure T. C calls the algorithm $KeyGen(MK, T) \rightarrow TK, SK$ and returns SK to A, where $Y \notin T$.
- Challenge: A sends two messages m_1, m_2 containing k keywords $W_1, W_2, \cdots, W_k$ to C. C selects $b \in \{0, 1\}$ randomly, and then calls the algorithm $Enc(Y, m_b, W_1, W_2, \cdots, W_k) \rightarrow CT, I$. C then returns the CT to A.
- Phase 2: A does the same inquiry at this stage as the Phase 1.
- Guess: The result guessed $b' \in \{0, 1\}$ is outputted.

The advantage of the adversary A in the game is defined as follows:

$$Pr_A = |Pr[b' = b] - \frac{1}{2}|.$$

Definition 2 Suppose that an arbitrary polynomial time adversary A can win the above game with at most a negligible advantage. Then our scheme is secure in the Selective-Set model.

4 Secure and flexible keyword search over encrypted data in IoT

Our scheme consists of the following six algorithms: *System Initialization*, *Secret Key Distribution*, *Data Uploading*, *Trapdoor Generation*, *Keyword Search,* and *Data Decryption*. In the system initialization phase, AA calls the algorithm $Setup(U)$, then announces the system parameters PK, and saves the system master secret key MK. When users join in the system, the user sends an access control tree to AA, and then AA calls algorithm $KeyGen(MK, T)$ to generate the transformation key TK and the secret key SK. When an user collects data or has data that he wants to share, he calls the algorithm $Enc(Y, m, W_1, W_2, \cdots, W_k)$, then uploads the ciphertext CT and the keyword index I to CS. When DU wants to retrieve data, he calls the algorithm $Trapdoor(SK, W)$ to generate a keyword trapdoor T_W and sends it to CS. When CS receives the keyword trapdoor T_W, it will call the algorithm $Test(I, T_W, x)$ to check whether the file that the user wants to search exists. When the searched file exists, CS sends CT to OS, and DU sends TK to OS, then OS carries out the partial decryption, returns the partial ciphertext to DU. Finally, DU decrypts the partial ciphertext again.

4.1 System initialization

AA calls the algorithm $Setup(U) \rightarrow PK, MK$, and the concrete process is as follows: Randomly choose $\alpha, y \in \mathbb{Z}_p$, $g_2 \in G_1$ and $t_1, t_2, \cdots, t_{n+1} \in G_1$. Compute $g_0 = g^{\alpha}$, $g_1 = g^{y}$, and g_2 is randomly chosen from $\mathbb{G}_1$. Let $N = \{1, 2, \cdots, n+1\}$. A function F is defined as follows:

$$F(X) = g_2^{X^n} \prod_{i=1}^{n+1} t_i^{\triangle_{i,N}(X)}.$$

The system's public parameters PK and the master secret key MK are as follows:

$$PK = (g_0, g_1, g_2, t_1, t_2, \cdots, t_{n+1}).$$

$$MK = (\alpha, y).$$

4.2 Secret key distribution

In the secret key generation stage, the interaction between AA and users is shown in Fig. 2. When AA receives the access control tree T from the user, AA calls the algorithm $KeyGen(MK, T) \rightarrow TK, SK$. Then, TK and SK are returned to users. The concrete algorithm is as follows:

For every node of the access control tree T, select a polynomial $p(x)$. Here's the equation $d_x + 1 = k_x$, where d_x is the degree of $p(x)$ and k_x is the threshold value of nodes. For the root node, $p_t(0) = y$. The other points are chosed randomly. For other nodes, $p_x(0) = p_{pa(x)}(index(x))$. The other points are chosed randomly. In

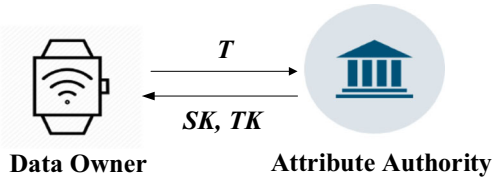

Fig. 2 Secret key distribution

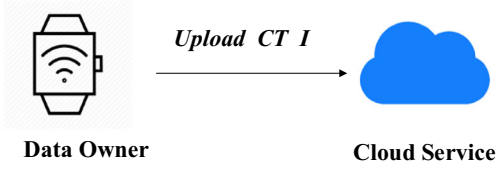

Fig. 3 Data uploading

this way, the polynomials of each node are determined. The transformation key TK is as follows:

$$TK = \begin{cases} D_x = g_2^{\frac{p_x(0)}{\beta}} \cdot F(i)^{\frac{r_x}{\beta}}, \ i = att(x) \\ R_x = g^{\frac{r_x}{\beta}}, \end{cases} \tag{3}$$

where r_x is randomly selected from $\mathbb{Z}_p$. The user's private key is

$$SK = (z, \beta).$$

4.3 Data uploading

In the encryption stage, the interaction between CS and DO is shown in Fig. 3. DO calls the algorithm $Enc(Y, m, W_1, W_2, \cdots, W_k) \rightarrow CT, I$, and then uploads the ciphertext CT and the keyword trapdoor I to CS. The concrete process of the algorithm is as follows:

First compute $H(W_i) = \omega_i$, where $i = 1, 2 \cdots, k$. Next, the mappings $\hat{r}_1$ and $\hat{R}_1$ from G_1^k to G_1^k and the mappings $\hat{r}_2$ and $\hat{R}_2$ from G_2^k to G_2^k will be constructed. Randomly choose $s_j, c \in Z_p$, compute $u_j = g_0^{s_j} g^{-s_j \omega_j}$, $v_j = e(g, g)^{s_j}$, $m_j = e(g, g_2)^{s_j}$, and set $(U_1, \cdots, U_k) = \hat{r}_1(u_1, \cdots, u_k)$, $(V_1, \cdots, V_k) = \hat{r}_2(v_1, \cdots, v_k)$, $(M_1, \cdots, M_k) = \hat{r}_2(m_1, \cdots, m_k)$.

Finally, the ciphertext and keyword index are set as follows:

$$CT = (C' = e(g_1, g_2)^c \cdot m, C'' = g^c, \{C_i = F(i)^c\}_{i \in Y}).$$

$$I_W = (C_u = \{U_j\}, C_v = \{V_j\}, C_m = \{M_j\}).$$

4.4 Trapdoor generation

In the trapdoor generation phase, the interaction between CS and DU is shown in Fig. 4. DU calls the algorithm $Trapdoor(SK, W) \rightarrow T_W$, then uploads the keyword trapdoor T_W to CS. The keyword Trapdoor T_W is generated as follows:

$$T_W = [\omega, r_W, h_W],$$

where $\omega = H(W)$, $r_w \in Z_p$, and $h_W = (g_2, g^{-r_W})^{\frac{1}{z-\omega}}$.

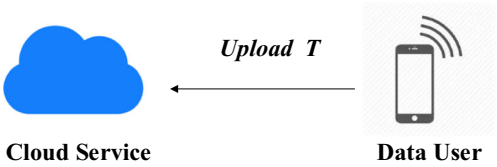

Fig. 4 Trapdoor generation

4.5 Keyword search

When CS receives the keyword trapdoor T_W from DU x, the CS calls algorithm $Test(I, T_W, x) \rightarrow 0, 1$. The concrete process is as follows: First compute the value of U, V, M, where $j = 1, 2, \cdots, k$:

$$U = \prod_{j=1}^{k} U^{\omega^{j-1}}, V = \prod_{j=1}^{k} V^{\omega^{j-1}}, M = \prod_{j=1}^{k} M^{\omega^{j-1}}. \tag{4}$$

If the equation

$$e(U, h_W) V^{r_W} = M$$

holds, the searched file containing the keyword exists; otherwise, it does not exist.

The correctness of the test phase is shown as follows.

Correctness If $W_i \in W_1, W_2, \cdots, W_k$, then according to the mappings $\hat{r}$ and $\hat{R}$, we have

$$M = \prod_{j=1}^{k} M^{\omega^{j-1}} = \hat{R}_2(M_1, M_2, \cdots, M_k) = m_i = e(g, g_2)^{s_i}. \tag{5}$$

Similarly, $U = g_0^{s_i} g^{-s_i \omega_i}$, $V = e(g, g)^{s_i}$. Therefore,

$$\begin{aligned} e(U, h_W) V^{r_W} &= e(g_0^{s_i} g^{-s_i \omega_i}, g_2^{\frac{1}{\alpha - \omega_i}} \cdot g^{\frac{-r_W}{\alpha - \omega_i}}) \cdot e(g, g)^{s_i r_W} \\ &= e(g^{s_i(\alpha - \omega_i)}, g_2^{\frac{1}{(\alpha - \omega_i)}} \cdot g^{\frac{-r_W}{\alpha - \omega_i}}) \cdot e(g, g)^{s_i r_W} \\ &= e(g^{s_i}, g_2 g^{-r_W}) \cdot e(g, g)^{s_i r_W} \\ &= e(g, g_2)^{s_i} \cdot e(g, g)^{-s_i r_W} \cdot e(g, g)^{s_i r_W} \\ &= e(g, g_2)^{s_i} \\ &= M. \end{aligned}$$

4.6 Data decryption

In the decryption stage, the interaction between CS, OS, and DU is shown in Fig. 5. The decryption stage is divided into two parts: outsourcing decryption and local decryption. DU sends a keyword trapdoor T to CS. If the file exists, CS sends the ciphertext to OS, and DU uploads the transformation key TK to OS. OS decrypts CT partially, and then returns partial ciphertext PT to DU. The concrete process is as follows:

– $Dec_{out}(CT, TK, x) \rightarrow PT$: The function $Dec(CT, TK, x)$ is defined as follows:

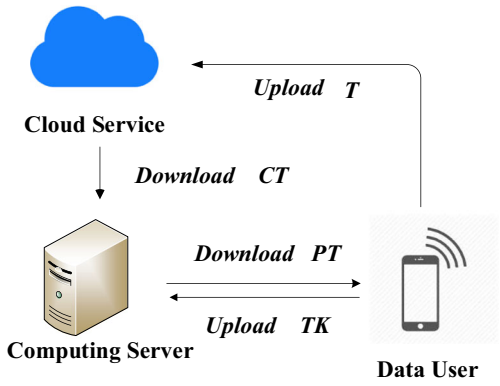

Fig. 5 Data decryption

When x is a leaf node, where $i = att(x)$, then,

$$Dec(CT, TK, x) = \begin{cases} \frac{e(D_x, C'')}{e(R_x, C_i)}, & i \in Y \\ \perp, & \text{otherwise} \end{cases} \quad (6)$$

If $W \in W_1, W_2, \cdots, W_k$, we know

$$\begin{aligned} \frac{e(D_x, C'')}{e(R_x, C_i)} &= \frac{e(g_2^{\frac{p_x(0)}{\beta}} \cdot F(i)^{\frac{r_x}{\beta}}, g^c)}{e(g^{\frac{r_x}{\beta}}, F(i)^c)} \\ &= \frac{e(g_2^{\frac{p_x(0)}{\beta}} \cdot g^c) \cdot e(F(i)^{\frac{r_x}{\beta}}, g^c)}{e(g^{\frac{r_x}{\beta}}, F(i)^c)} \\ &= e(g, g_2)^{\frac{c \cdot p_x(0)}{\beta}}. \end{aligned}$$

When x is a non-leaf node, $Dec(CT, TK, x)$ is performed as follows: For each child node x', let $D_{x'} = Dec(CT, TK, x')$. If there are k such child nodes, the following operation is performed, otherwise the algorithm will be terminated.

$$\begin{aligned} D_x &= \left(\prod_{x' \in S_x} D_{x'}^{\triangle_{i, S'_x(0)}} \right) \\ &= \left(\prod_{x' \in S_x} \left(e(g, g_2)^{\frac{c \cdot p_{x'}(0)}{\beta}} \right)^{\triangle_{i, S'_x(0)}} \right) \\ &= \left(\prod_{x' \in S_x} \left(e(g, g_2)^{\frac{c \cdot p_{pa(x')}(index(x'))}{\beta}} \right)^{\triangle_{i, S'_x(0)}} \right) \\ &= \left(\prod_{x' \in S_x} e(g, g_2)^{\frac{c \cdot p_x(i) \cdot \triangle_{i, S'_x(0)}}{\beta}} \right) \\ &= e(g, g_2)^{\frac{c \cdot p_x(0)}{\beta}}, \end{aligned}$$

where $i = index(x)$, $S'_x = \{index(x') : x' \in S_x\}$.

Then, OS can get the partial ciphertext $PT = Dec(CT, TK, t) = e(g, g_2)^{\frac{c \cdot y}{\beta}} = e(g_1, g_2)^{\frac{c}{\beta}}$ by calling $Dec(CT, TK, t)$, where t is the root node, if and only if the ciphertext satisfies the access control tree.

– $Dec(PT, SK)$: When receiving the partial ciphertext sent by OS, DU decrypts the partial ciphertext as follows:

$$m = \frac{C'}{PT^{SK}}.$$

5 Analysis of our scheme

The security model of the scheme is divided into the security analysis of keywords and the security analysis of the scheme.

5.1 Security analysis of keywords

Theorem 1 *Suppose that the (t, q, ε)-ABDHE assumption holds on (G_1, G_2), then our scheme has semantic security, where $q_W = q - 1$, $\varepsilon' = \varepsilon + 2/p$, $t' = t - O(t_{exp} \cdot q^2)$, t_{exp} is the time of the exponential operation on G_1.*

Proof The security of our scheme can be reduced to that of the scheme [15], which is denoted by $\prod_L = (Setup_L, KeyGen_L, Enc_L, Trap_L, Test_L, Dec_l)$. Note that the security of the scheme in [15] is based on the (t, q, ε)-ABDHE assumption. The game among adversary A, simulator B and challenger C in [15] is as follows:

Init: B calls the C running algorithm $Setup_L$ to get the system public parameters $PK = (g_0, g_1, g_2, t_1, t_2, ..., t_{n+1})$. Then, B returns PK to A.

Trapdoor queries: When A inquires about the trapdoor of $W \in \{0, 1\}^*$, B makes the same inquires to C. C executes the algorithm $Trap_L$. Then, C returns $T_W = (\omega, f_1(\omega), g^{F_W(a)}, D_x, R_x)$ to B. Finally, B returns $T_W = (\omega, f_1(\omega), g^{F_W(a)})$ to A.

Challenge: A sends $k + 1$ keywords $W_0, W_1, \cdots, W_k$. If $a \in H(W_j)$, where $j = 1, 2, \cdots, k$, then B makes the same inquires to C. C executes the algorithm Enc_L. Then, C returns $I_L = (C_u, C_v, C_m)$ to B. B returns $I = I_L = (C_u, C_v, C_m)$ to A.

Phase 2: A does the same inquiry at this stage as the Phase 1.

Guess: The result guessed $b' \in \{0, 1\}$ is outputted.

Therefore, if the adversary A breaks our scheme with the advantage that can not be ignored, the simulator B will be able to call the challenger C in [15] to break the (t, q, ε)-ABDHE assumption. □

5.2 Security analysis of the scheme

Theorem 2 *Our scheme has semantic security under the Decisional BDH assumption.*

If the adversary A breaks our system with a non-negligible advantage, the simulator B will break the Decisional BDH assumption with a non-negligible advantage.

Proof The interaction between A and B is as follows:

Init: A selects the attribute set Y to challenge and sends Y to B.

Setup: B randomly generates $a, b \in Z_p$, then let $g_0 = g^a$, $g_1 = g^b$, where g is the generator of G, and a polynomial $f_1(x)$ of degree n will be randomly selected. Make the polynomial $f_1(x)$ to satisfy: if $x \in Y$, $u(x) = -x^n$, otherwise $u(x) \neq -x^n$. After that, let $t_i = g_2^{u(i)} g^{f_1(i)}$, where $i = 1, 2, \cdots, n+1$, then $F(i) = g_2^{i^n+u(i)} g^{f_1(i)}$. Then, the public parameter is $PK = (g_0, g_1, t_1, t_2, \cdots, t_{n+1})$. Finally B returns PK to A.

Phase 1: A inquires the secret key related to an access structure tree T, where $T(Y) = 0$. For each node in T, an polynomial Q_x of degree d_x is determined by B. We use $Poly(T_x, Y, g^{\lambda_x})$ to determine the process of accessing the polynomial of node x in T, where T_x expresses the access subtree of the root node, Y is the attribute set and $\lambda_x \in \mathbb{Z}_p$. q_x is defined the polynomial of the root node x, where $q_x(0) = \lambda_x$. If the number h_x of children of x satisfies attribute Y, where $h_x < d_x$, then for each satisfied child x' of x, let $q_x(index(x')) = \lambda'_x$. The procedure randomly selects $d_x - h_x$ points which do not satisfy the attribute. Then, we can recursively call $Poly(T_x, Y, g^{\lambda_x})$ to determine all polynomials.

For each node in the tree T, $Poly(T_x, Y, A)$ is called to define the polynomial q_x. For each leaf node x in T, where $T(Y) = 1$, q_x is known. Otherwise $g^{q_x(0)}$ is known. Now B defines $Q_x(\cdot) = q_x(\cdot)$, $Q_r(0) = q_r(0) = y = c$ for each node x in T. The secret key of each leaf node is defined as follows:

If $i \in Y$,

$$\begin{cases} D_x = g_2^{Q_x(0)} \cdot F(i)^{r_x} = g_2^{q_x(0)} \cdot F(i)^{r_x} \\ R_x = g^{r_x}, \end{cases}$$

where $r_x \in \mathbb{Z}_p$

If $i \notin Y$, where $g_3 = g^{Q_x(0)} = g^{q_x(0)}$,

$$\begin{cases} D_x = g_3^{\frac{-f_2(i)}{i^n+u(i)}} \cdot F(i)^{r'_x} \\ R_x = g_3^{\frac{-1}{i^n+u(i)}} \cdot g^{r'_x}, \end{cases}$$

where $r'_x \in \mathbb{Z}_p$

Let $r_x = r'_x - \frac{q(0)}{i+u(i)}$, then

$$\begin{cases} D_x = g_2^{q_x(0)} F(i)^{r_x} \\ R_x = g_3^{\frac{-1}{i^n+u(i)}} g^{r'_x} = g^{r_x}, \end{cases}$$

Table 1 Efficiency comparison

Schemes	Trapdoor generation	Keyword matching
Zhang et al.'s scheme	$(3k+2)t_g$	$(2k+1)p$
Our scheme	$2t_g$	$p + t_t + (k-1)(t_g + 2t_t)$

where

$$\begin{aligned} D_x &= g_3^{\frac{-f_1(i)}{i^n+u(i)}} \cdot F(i)^{r'_x} \\ &= (g^{q_x(0)})^{\frac{-f_1(i)}{i^n+u(i)}} (g_2^{i^n+u(i)} g^{f_1(i)})^{r_x + \frac{q_x(0)}{i^n+u(i)}} \\ &= g_2^{q_x(0)} F(i)^{r_x}. \end{aligned}$$

After that, B randomly selects $z, \beta \in Z_p$, then calculates $D'_x = D_x^{\frac{1}{\beta}}$, $R'_x = R_x^{\frac{1}{\beta}}$, and finally returns $TK = (D'_x, R'_x)$ and $SK = (z, \beta)$ to A.

Challenge: A sends two messages m_0^* and m_1^* with the same length to B. B randomly selects $b \in \{0, 1\}$ to encrypt m_b^*. The ciphertext is as follows:

$$CT = (C' = m_b^* \cdot Z, C'' = g^c), \{C_i = F(i)^c\}_{i \in Y},$$

where $c \in Z_p$ is selected randomly. Finally, the ciphertext CT is sent to A.

Phase 2: A makes the same query as Phase 1.

Challenge: A outputs the guess $b \in \{0, 1\}$.

If $Z = e(g_1, g_2)^c = e(g, g)^{abc}$, the ciphertext is a valid ciphertext. Otherwise, $Z = e(g_1, g_2)^z$ is a random number. Therefore, if A can break through our system with the advantage that can not be ignored, then B can solve the Decisional BDH assumption. □

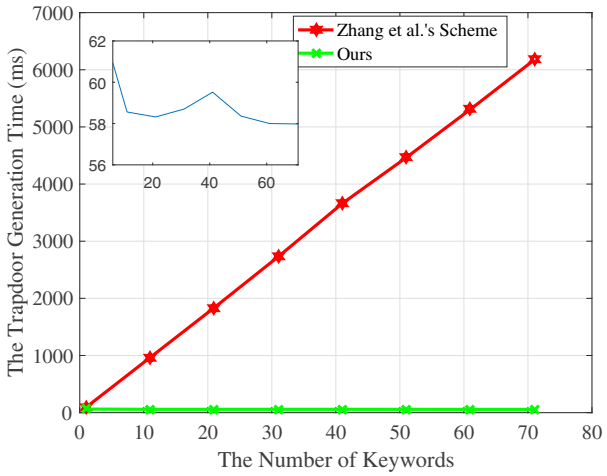

Fig. 6 Comparison of the trapdoor generation efficiency

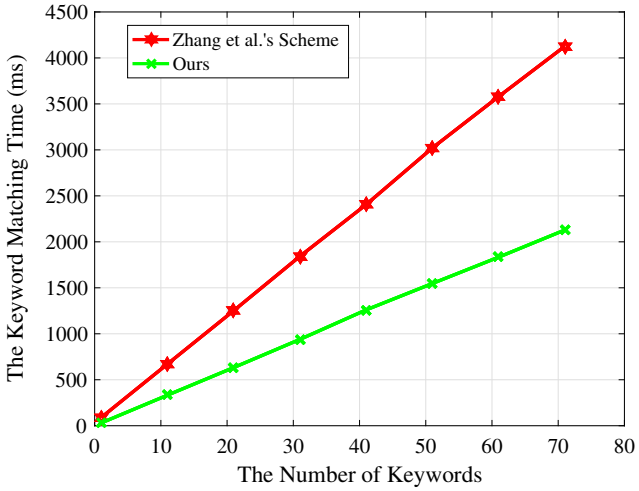

Fig. 7 Comparison of keyword matching efficiency

5.3 Efficiency analysis

The proposed scheme can realize one-to-many encryption and multi-keyword search. We compare our scheme with Zhang et al's scheme [41]. Let t_g represent the time of an exponent operation in G. Let t_t represent the time for an exponent operation in G_T. Let p represent the time of a pairing operation. n is the total number of keywords. k is the number of keywords carried by the ciphertext. The multiplication and hash operation are ignored. The comparison results are displayed in Table 1. For more accurate comparisons, we conducted efficiency tests on the same platform, and the results are shown in Figs. 6 and 7.

As shown in Figs. 6 and 7, the performance of our scheme is better than that of the scheme [41] in the threshold generation and keyword matching phases. In addition, the outsourcing technology is used in our scheme. The efficiency of decryption stage does not increase with the increase of the number of attributes. In general, our scheme is feasible in the IoT scenario.

6 Conclusion

In this paper, in order to solve the ciphertext retrieval and the efficiency bottleneck in the IoT scenario, the flexible keyword search scheme over encrypted data with outsourced decryption for IoT is presented. The scheme has the following three features: First, the attribute-based encryption technology is applied, by which only users whose attributes meet the access control structure can access the sharing data. Second, the reciprocal mapping of Lagrange polynomials technology is employed to implement keyword search in a large number of ciphertext data. Third, the decryption of ciphertext is outsourced to improve the efficiency of decryption on the client side. In general, our scheme is feasible in the IoT scenario. In the future work, in order to deal with dishonest calculations in cloud computing, we will verify the correctness of outsourcing calculations.

Funding information This work is supported by National Key R&D Program of China (No. 2017YFB0802000), National Natural Science Foundation of China (No. 61772418, 61472472, 61402366), Natural Science Basic Research Plan in Shaanxi Province of China (No. 2018JZ6001, 2015JQ6236), and the Youth Innovation Team of Shaanxi Universities. Yinghui Zhang is supported by New Star Team of Xi'an University of Posts and Telecommunications (No. 2016-02).

References

1. Jhaveri RH, Patel NM, Zhong Y, Sangaiah AK (2018) Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks in industrial iot. IEEE Access 6:20085–20103
2. Shen J, Wang C, Li T, Chen X, Huang X, Zhan ZH (2018) Secure data uploading scheme for a smart home system. Inform Sci 453:186–197
3. Wu A, Zheng D, Zhang Y, Yng M (2018) Hidden policy attribute-based data sharing with direct revocation and keyword search in cloud computing. Sensors 18(7):1–17. https://doi.org/10.3390/s18072158
4. Zhang Y, Wu A, Zheng D (2018) Efficient and privacy-aware attribute-based data sharing in mobile cloud computing. J Ambient Intell Humaniz Comput 9(4):1039–1048
5. Chen X, Li J, Weng J, Ma J, Lou W (2016) Verifiable computation over large database with incremental updates. IEEE Trans Comput 65(10):3184–3195
6. Li J, Liu Z, Chen X, Xhafa F, Tan X, Wong DS (2015) L-encdb: a lightweight framework for privacy-preserving data queries in cloud computing. Knowl-Based Syst 79:18–26
7. Zhang Y, Zheng D, Deng RH (2018) Security and privacy in smart health: e policy-hiding attribute-based access control. IEEE Internet Things J 5(3):2130–2145
8. Wang J, Chen X, Huang X, You I, Xiang Y (2015) Verifiable auditing for outsourced database in cloud computing. IEEE Trans Comput 64(11):3293–3303
9. Zhang Y, Yang M, Zheng D, Lang P, Wu A, Chen C (2018) Efficient and secure big data storage system with leakage resilience in cloud computing. Soft Comput 22(23):7763–7772
10. Zhang Y, Zheng D, Guo R, Lan Q (2018) Fine-grained access control systems suitable for resource-constrained users in cloud computing. Comput Inf 37(2):327–348
11. Zhang Y, Deng RH, Han G, Zheng D (2018) Secure smart health with privacy-aware aggregate authentication and access control in Internet of Things. J Netw Comput Appl 123:89–100
12. Li H, Liu D, Dai Y, Luan TH, Shen XS (2015) Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage. IEEE Trans Emerging Topics Comput 3(1):127–138
13. Wang J, Chen X, Li J, Zhao J, Shen J (2017) Towards achieving flexible and verifiable search for outsourced database in cloud computing. Futur Gener Comput Syst 67:266–275
14. Zhang Y, Deng RH, Jiangang S, Kan Y, Dong Z (2018) Tkse: trustworthy keyword search over encrypted data with two-side verifiability via blockchain. IEEE Access 6:31077–31087

15. Li R, Zheng D, Zhang Y, Su H, Yang M, Lang P (2017) Attribute-based encryption with multi-keyword search. In: IEEE 2nd international conference on data science in cyberspace, pp 172–177
16. Li P, Li T, Ye H, Li J, Chen X, Xiang Y (2018) Privacy-preserving machine learning with multiple data providers. Futur Gener Comput Syst 87:341–350
17. Zhang Y, Lang P, Dong Z, Yang M, Guo R (2018) A secure and privacy-aware smart health system with secret key leakage resilience. Secur Commun Netw 2018:1–13. https://doi.org/10.1155/2018/7202598
18. Wang C, Shen J, Liu Q, Ren Y, Li T (2018) A novel security scheme based on instant encrypted transmission for internet of things. Secur Commun Netw 2018(2):1–7
19. Zhang Y, Deng RH, Ximeng L, Dong Z (2018) Blockchain based efficient and robust fair payment for outsourcing services in cloud computing. Inf Sci 462:262–277
20. Zheng D, Wu A, Hui Y, Lang Q (2018) Efficient and privacy-preserving medical data sharing in Internet of Things with limited computing power. IEEE Access 6:28019–28027
21. Sahai A, Waters B (2005) Fuzzy identity-based encryption. In: Annual international conference on the theory and applications of cryptographic techniques. Springer, Berlin, pp 457–473
22. Ning J, Dong X, Gao Z, Wei L, Lin X (2015) White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes. IEEE Trans Inf Forensics Secur 10(6):1274–1288
23. Ning J, Gao Z, Dong X, Wei L (2018) White-box traceable CP-ABE for cloud storage service: how to catch people leaking their access credentials effectively. IEEE Trans Dependable Secure Comput 15(5):883–897
24. Ning J, Gao Z, Dong X, Wei L, Lin X (2014) Large universe ciphertext-policy attribute-based encryption with white-box traceability. European Symposium on Research in Computer Security 15(5):55–72
25. Li J, Chen X, Chow SSM, Huang Q, Wong DS, Liu Z (2018) Multi-authority fine-grained access control with accountability and its application in cloud. J Netw Comput Appl 112:89–96
26. Li H, Liu D, Dai Y, Luan TH, Yu S (2018) Personalized search over encrypted data with efficient and secure updates in mobile clouds. IEEE Transactions on Emerging Topics in Computing 6(1):97–109
27. Sun W, Yu S, Lou W, Hou YT, Li H (2014) Protecting your right: attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. In: 2014 Proceedings IEEE INFOCOM, pp 226–234
28. Zheng Q, Xu S, Ateniese G (2014) Vabks: Verifiable attribute-based keyword search over outsourced encrypted data. In: IEEE INFOCOM, pp 522–530
29. Li J, Zhang Y, Chen X, Xiang Y, Li J, Zhang Y, Chen X, Xiang Y (2018) Secure attribute-based data sharing for resource-limited users in cloud computing. Comput Secur 72:1–12
30. Zhang Y, Zheng D, Li Q, Li J, Li H (2016) Online/offline unbounded multi-authority attribute-based encryption for data sharing in mobile cloud computing. Secur Commun Netw 9(16):3688–3702
31. Li J, Li J, Chen X, Jia C, Lou W (2015) Identity-based encryption with outsourced revocation in cloud computing. IEEE Trans Comput 64(2):425–437
32. Zhang Y, Deng RH, Liu X, Zheng D (2018) Outsourcing service fair payment based on blockchain and its applications in cloud computing, IEEE transactions on services computing. https://doi.org/10.1109/TSC20182864191
33. Li J, Huang X, Li J, Chen X, Xiang Y (2014) Securely outsourcing attribute-based encryption with checkability. IEEE Trans Parallel Distrib Syst 25(8):2201–2210
34. Zhang Y, Chen X, Li J, Wong DS, Li H, You I (2017) Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. Inform Sci 379:42–61
35. Green M, Hohenberger S, Waters B (2014). In: Usenix conference on security, pp 34–34
36. Ning J, Gao Z, Dong X, Ma KLiangH, Wei L (2018) Auditable σ-time outsourced attribute-based encryption for access control in cloud computing. IEEE Trans Inf Forensics Secur 13(1):94–105
37. Menezes A (2009) An introduction to pairing-based cryptography. Recent trends in cryptography 477:47–65
38. Haoxing L, Fenghua L, Chenggen S, Mang S, Xin L (2015) Public key encryption with multi-keywords search. Journal of Xidian University 42(5):20–25
39. Gentry C (2006) Practical identity-based encryption without random oracles. Lect Notes Comput Sci 4004:445–464
40. Dan B, Boyen X (2004) Efficient selective-ID secure identity-based encryption without random oracles. Springer, Berlin, pp 223–238
41. Zhang B, Zhang F (2011) An efficient public key encryption with conjunctive-subset keywords search. J Netw Comput Appl 34(1):262–267