Searchable attribute-based signcryption scheme for electronic personal health record

ZHÉNHUA LIU1, YAOHUI LIU1, AND YAYING FAN.1

1School of Mathematics and Statistics, Xidian University, Xi’an 710126, China
Corresponding author: Z.H. Liu (e-mail: zhualiu@hotmail.com).

This work is supported by the National Key Research and Development Program of China under Grants No. 2017YFB0802000, the National Natural Science Foundation of China under Grants No. 61472470 and 61807026, and the Scientific Research Plan Project of Education Department of Shaanxi Province under Grant No.17JK0362.

ABSTRACT Considering the scenario of electronic personal health record system, we put forward a new cryptographic concept called searchable attribute-based signcryption (SABSC), which can support fine-grained access control, data privacy, data authenticity, and data searchability. Then we establish the security models and construct a searchable attribute-based signcryption scheme with hybrid access policy. According to the proposed security model frameworks, our scheme is proven to achieve (i) ciphertext indistinguishability under the Decisional Bilinear Diffie-Hellman Exponent hardness assumption, (ii) existential unforgeability based on the hardness assumption of Computational Diffie-Hellman Exponent problem, (iii) selective security against chosen-keyword attack under the static assumption in the generic group model, (iv) keyword secrecy based on the one-way hardness of hash function. Furthermore, the experimental results show that the proposed scheme is efficient.

INDEX TERMS attribute-based encryption, searchable encryption, signcryption, hybrid access policy, electronic personal health record

I. INTRODUCTION

ELECTRONIC personal health record (ePHR) is a collection of individual’s health related information, which is created and managed by himself. In recent years, there has been enthusiasm in discuss centering the adoption of electronic personal health record systems for patients. The use of electronic personal health record will bring many benefits for human. For example, there are a large number of reliable health information, data and knowledge in electronic personal health record, and then patients can rely on cloud that can easily manage and monitor their health. Electronic personal health record can also offer a constant connection between patients and doctors, which greatly reduces the time to solve problems. Despite of various advantages, the electronic personal health record data includes much sensitive information. It is necessary to design a system that storage and access to electronic personal health record securely.

With the development of cloud computing, many data owners store their data in the cloud server for simplifying local IT management and reducing the cost. As an important application of cloud computing, electronic personal health record system [1] can store, control and share electronic personal health record with some specified users, including doctors, nurses and family members etc. Although the electronic personal health record system can provide many efficient services, outsourcing electronic personal health record to the third-party cloud server will bring security and privacy concerns. Therefore, it is essential to encrypt sensitive electronic personal health record before uploading them to the remote server.

Traditional public key encryption technology can protect the confidentiality of data. But it provides “one-to-one” mapping, which implies only single user can access. Both the efficiency and flexibility are limited. Thus, as a kind of “one-to-many” public key encryption, attribute-based encryption can solve this problem. So, attribute-based encryption is considered as one of the most appropriate encryption technology for the electronic personal health record system in the cloud computing.

Sahai and Waters [2] first introduced the concept of attribute-based encryption, and proposed a concrete attribute-based encryption scheme, which only supports threshold ac-
cess policy. In order to support more flexible access policy, in 2006, Goyal et al. [3] first presented the key-policy attribute-based encryption (KP-ABE). In KP-ABE, the decryption key is associated with the access control policy and ciphertext is associated with attributes. Thus, it is more suitable for inquiry service, such as audit log and pay-TV system etc. On the contrary, if ciphertext is associated with the access control policy and the decryption key is associated with attributes, such an attribute-based encryption is called as ciphertext-policy attribute-based encryption (CP-ABE). So, CP-ABE is more suitable for access control service, such as access of electronic personal health record. Bethencourt et al. [4] proposed the first CP-ABE scheme in 2007. Due to the adaptability of attribute-based encryption to cloud computing, a number of schemes [5]–[7] have been proposed to gain a better expressive, efficiency and security.

Attribute-based encryption provides the data confidentiality and expressive fine-grained access control. However, it is intractable to search electronic personal health record data in the server efficiently. To solve this problem, Song et al. [8] first proposed the concept of searchable encryption, which provides a fundamental approach to search over the encrypted data. Furthermore, Boneh et al. [9] presented the first public key encryption scheme with keyword search scheme, where one can search the encrypted data by a keyword. Since then, a number of searchable public key encryption schemes [10]–[13] have proposed to enrich the search feature of scenarios and improve security and efficiency.

In order to support fine-grained access control, in 2013, Wang et al. [14] constructed a ciphertext-policy attribute-based encryption scheme supporting keyword search. In their scheme, data owners encrypt data with an access policy, generate indexes for the corresponding keyword collection, and then outsource them to the cloud server. It is only when authorized users’ certificate meets the access policy that they could search and decrypt the encrypted data. Later, in 2014, Zheng et al. [15] first proposed a verifiable attribute-based keyword search scheme and constructed two specific schemes based on the access tree: key-policy attribute-based searchable encryption scheme and ciphertext-policy attribute-based searchable encryption scheme. In the recent years, the study of attribute-based searchable encryption has got great development. Taking into account the more realistic scenario, some schemes [16]–[19] were presented.

Nevertheless, the above works can only ensure confidentiality of the data, they cannot provide the authenticity of electronic personal health record in the system. If a malicious attacker gains right to modify the electronic personal health record before an authorized user read it, this immediately rises the safety concern of electronic personal health record. For example, if an attending physician receives a falsified electronic personal health record, he may give a wrong diagnosis or treatment to a patient. For some worse cases, the forged electronic personal health record may facilitate someone to murder the patient. These are very serious threats to the electronic personal health record owner. It is necessary to prove the authenticity of electronic personal health record data.

Signature can provide the authenticity and encryption can guarantee the confidentiality. To achieve this two advantages at the same time, traditional method is “encrypt-then-sign” or “sign-then-encrypt”. For reducing the cumulative cost of encryption and signature, in 1997, Zheng et al. [20] first introduced the concept of signcryption, which is a logical incorporation of signature and encryption. Subsequently, in order to ensure fine-grained access control for the data in the cloud and prove data owners’ genuineness at the cloud servers, Gagne et al. [21] presented the first attribute-based signcryption (ABSC) scheme. In electronic personal health record scenario, Liu et al. [22] proposed an approach for fine-grained access control and secure sharing of signcrypted data. Au et al. [23] proposed a general scheme for secure sharing of electronic personal health record, and their scheme achieved that the doctors can refer the patients’ medical record to specialists for research purposes, whenever they are required. Liu et al. [31] proposed a secure online/offline attribute-based signature scheme for sharing of Mobile Health Records. In order to achieve authentication of electronic personal health record, a number of attribute-based signcryption schemes [24]–[27] have been proposed. So, in electronic personal health record system, it is important to achieve authentication of data. Though these attribute-based signcryption schemes can ensure that data owners share efficiently their encryption data with data users and prove their genuineness at the cloud servers, they cannot take the retrieval of the signcryption data into account.

In practice, it is important to design a scheme that achieve the fine-grained access control for the data, the authenticity of electronic personal health record data at the cloud servers, and the encryption data retrieval. However, to the best of our knowledge, there is few scheme that can realize the three functionalities at the same time.

A. OUR CONTRIBUTION

The main contribution of this paper can be summarized as follows:

- We propose a new concept of searchable attribute-based signcryption (SABSC), which can simultaneously achieve expressive fine-grained access control, efficient data retrieval and authentication, and establish its security model.
- We adapte Rao et al.’s ciphertext-policy attribute-based signcryption scheme [28] to an attribute-based signcryption scheme with hybrid access policy. In the adapted scheme, a signing LSSS access structure is used to generate a signature key, which can achieve the functionality of authentication. Then an encryption LSSS access structure is employed to encrypt message and keyword, which can achieve the access control over the encrypted electronic personal health record.
- Furthermore, we embed a search mechanism into the adapted attribute-based signcryption scheme with hy-
bride access policy, and propose the first searchable attribute-based signcryption scheme, which modifies Zheng et al.’s attribute-based keyword search method [15] (i.e based on an access tree) and supports more expressive access control. In the proposed scheme, an identical secret key is used to generate a trapdoor and decrypt a ciphertext, which can promote the logical combination of the above adapted attribute-based signcryption and attribute-based keyword search schemes. Hence, the cost is significantly reduced than the cumulative cost of signcryption and search.

- The proposed scheme is proven to achieve indistinguishability of data ciphertext, unforgeability of signature, selective encryption access policy secure against chosen-keyword attack and keyword secrecy in the standard model rather than random oracle model [15].

II. PRELIMINARIES

A. BILINEAR PAIRING

Let $G_1$, $G_2$ be two multiplicative (cyclic) groups of prime order $p$, $g$ be a generator of $G_1$, and $e$ be a bilinear map $G_1 \times G_1 \rightarrow G_2$ with following properties:

1) Bilinearity: Given two random elements $x, y \in Z_p$, we have $e(g^x, g^y) = e(g, g)^{xy}$;
2) Non-degeneracy: $e(g, g) \neq 1$;
3) Computability: There exists an efficient algorithm to compute $e(g, g)$.

B. GENERIC BILINEAR GROUP

Let $\psi_0$ and $\psi_1$ be two random encodings of the multiplicative group $Z_p^n$, such that $\psi_0, \psi_1$ are injective maps from $Z_p$ to $\{0, 1\}^n$, where $m > 3 \log(p)$. Let $G_1 = \{\psi_0(x) | x \in Z_p\}$ and $G_2 = \{\psi_1(x) | x \in Z_p\}$. There is an oracle to compute $e : G_1 \times G_1 \rightarrow G_2$ [29]. $G_1$ is known as a generic bilinear group. Let $g$ denote $\psi_1(1)$, $g^*$ denote $\psi_0(x)$, $e(g, g)$ denote $\psi_1(1)$ and $e(g, g)^y$ denote $\psi_1(y)$.

C. ACCESS STRUCTURE

Let $\{P_1, \cdots, P_n\}$ be a set of attributes. A collection $A \subseteq \{P_1, \cdots, P_n\}$ is said to be monotone if $\forall B \in A$ with $B \subseteq C$, then $C \in A$. (A monotone) collection $A$ of non-empty subsets of $\{P_1, \cdots, P_n\}$ can be used to represent an (monotone) access structure [6], where $A \subseteq \{P_1, \cdots, P_n\} \setminus \{\emptyset\}$. The sets in $A$ are called the authorized sets. Otherwise, the sets are called the unauthorized sets with respect to $A$. So, an attribute set $S$ satisfies an access structure $A$ if and only if $S$ is an authorized set in $A$, denoted as $S \in A$.

D. LINEAR SECRET SHARING SCHEME

In this paper, we denote a linear secret sharing scheme (LSSS) [6] as an access control policy $(M, \rho)$, where $M$ is considered as an $I \times n$ share-generating matrix and $\rho$ maps a row of $M$ into an attribute. A LSSS consists of the following two polynomial time algorithms:

- **Distribute** $(M, \rho, \alpha)$ This algorithm is used to share secret value $\alpha$. Firstly, take as input $(M, \rho)$ and $\alpha$.

  Secondly, select $v_2, \cdots, v_n \in Z_p$ at random and set $\bar{V} = (\alpha, v_2, \cdots, v_n) \in Z_p^n$. Finally, output a set $\{\lambda_i | \lambda_i = M_i \cdot \bar{V}\}_{i \in [I]}$, where $M_i \in Z_p^n$ is the $i$-th row of the matrix $M$. The share $\lambda_i$ belongs to the attribute $\rho(i)$.

- **Reconstruct** $(M, \rho, S)$ We can recover $\alpha$ from some secret shares by this algorithm. Suppose $S \in A$ is an authorized set and $I = \{i | \rho(i) \in S\} \subseteq \{1, 2, \cdots, I\}$. Taking as input $(M, \rho)$ and $S$, the algorithm outputs a reconstruction coefficient set $\{w_i\}_{i \in I} \subseteq Z_p$, satisfying $\sum_i w_i \cdot M_i = (1, 0, \cdots, 0)$. Hence, $\sum_i w_i \cdot \lambda_i = \alpha$.

E. COMPUTATIONAL DIFFIE-HELLMAN EXPONENT ASSUMPTION

Computational Diffie-Hellman Exponent Problem. Given a tuple $\bar{y}_z = (g, g^z, g^{z^2}, g^{z^4}, \cdots, g^{z^{2^\ell}})$, where $z \in Z_p^*$, the computational $w$-Diffie-Hellman Exponent (w-CDHE) problem is to output $g^{z^w}$. We say that a probabilistic polynomial time (PPT) algorithm $A$ has at least advantage $\varepsilon$ in solving this problem if

$$\Pr(g^{z^w} \leftarrow A(\bar{y}_z)) \geq \varepsilon.$$ 

Computational Diffie-Hellman Exponent Assumption. We say that $w$-CDHE assumption [28] holds if for any PPT algorithm $A$, the advantage of $A$ in solving the $w$-CDHE problem is at most $\varepsilon$.

F. DECISIONAL BILINEAR DIFFIE-HELLMAN EXPONENT ASSUMPTION

Decisional Bilinear Diffie-Hellman Exponent Problem. Given an instance $(\bar{y}_{z,s}, Z)$, where $\bar{y}_{z,s} = (g, g^z, g^{z^2}, g^{z^4}, \cdots, g^{z^{2^\ell}}, g^s)$, $z, s \in Z_p^*$, $Z \in G_2$, the decisional $w$-Bilinear Diffie-Hellman Exponent (w-BDBHDE) Problem is to determine whether $Z = e(g^{z^w}, g^s)$ or a random element $R \in G_2$. We say that an algorithm $A$ has at least advantage $\varepsilon$ in solving this problem if

$$\Pr\left(e(g^{z^w}, g^s) \leftarrow A(\bar{y}_{z,s})\right) - \Pr\left(R \leftarrow A(\bar{y}_{z,s})\right) \geq \varepsilon.$$ 

Decisional Bilinear Diffie-Hellman Exponent Assumption. We say that the $w$-BDBHDE assumption [28] holds if for any PPT algorithm $A$, the advantage of $A$ in solving the $w$-BDBHDE problem is at most $\varepsilon$.

III. GENERIC SCHEME AND ITS MODELS

A. SYSTEM FRAMEWORK

We consider an electronic personal health record system supporting fine-grained access control over encrypted data, data retrieval and data authentication.

As shown in Fig.1, the system framework consists of four entities: trusted authority (TA), personal health record owners (PHRO), data users (DU), and cloud server (CS).

- TA: Trusted authority is a global trusted authority. It is responsible for generating the public parameters needed...
The system framework.

| PHRO | Personal health record owner takes charge of signing her/his personal health data. She/he firstly obtains a signing key from TA according to her/his attributes. Then, She/he selects an encryption access structure to sign encrypt her/his personal data and generate corresponding index. Finally, data owner uploads the ciphertext and index to the cloud server. |
| DU | Data users are some entities who need to access encrypted personal health data in the cloud server. When they intend to search some data of their interested, they first need to generate a token and send it to the cloud server. Then, with the help of the cloud server, data users complete the data retrieval. Finally, data users check the validity of the returned results and decrypt the valid ciphertext. |
| CS | Cloud server is honest-but-curious. It is responsible for storing data owners’ ciphertext and providing the data retrieval service for data users. |

B. GENERIC SCHEME

Let $\mathcal{M}$ be a message space, $\mathcal{G}$ be an access structure space, $\mathcal{U}_s$ be a space of signing attributes, and $\mathcal{U}_e$ be a space of decryption attributes. A generic searchable attribute-based signcryption scheme consists of the following eight algorithms:

- **Setup**: $k \rightarrow (pp, msk)$: Given a security parameter $k$, TA creates the public parameters $pp$ and master secret key $msk$, where $msk$ is owned by TA.
- **sExtract**: $(msk, \langle S, \rho \rangle) \rightarrow sk_{\langle S, \rho \rangle}$: Taking as input the master secret key $msk$ and a signing access structure $\langle S, \rho \rangle$ over a set $A_s \subset \mathcal{U}_s$, TA outputs a signing key $sk_{\langle S, \rho \rangle}$ to a legitimate data owner over a secret channel.
- **dExtract**: $(msk, A_d) \rightarrow sk_{A_d}$: On input a decryption attribute set $A_d$, TA computes a decryption key $sk_{A_d}$ with $msk$ and sends it to the corresponding data user over a secret channel.
- **Signcrypt**: $(pp, m, sk_{\langle S, \rho \rangle}, \langle D, \phi \rangle) \rightarrow CT$: PHRO performs this algorithm to sign encrypt a message $m \in \mathcal{M}$ with the input of a signing key $sk_{\langle S, \rho \rangle}$, an encryption access structure $\langle D, \phi \rangle$ and public parameters $pp$.
- **GenIndex**: $(\langle D, \phi \rangle, pp, kw) \rightarrow I$: PHRO chooses a keyword $kw$ for the message $m$, encrypts the keyword $kw$ with the input $pp$, $\langle D, \phi \rangle$, and then sends a keyword index or ciphertext $I$ to CS.
- **GenToken**: $(sk_{A_d}, pp, kw') \rightarrow T$: Given a keyword $kw'$ of interested, DU executes this algorithm to generate a corresponding token $T$.
- **Search**: $(I, T, CT) \rightarrow CT'$: After gaining the token $T$, CS firstly matches it with the index $I$, and then returns the relevant search results $CT'$ to DU.
- **Unsigncrypt**: $(pp, CT', sk_{A_d}) \rightarrow m'$: DU first checks the validity of ciphertext $CT'$. If $CT'$ can pass the verification, DU can recover the message $m'$ corresponding to $CT'$.

C. SECURITY MODEL

The security of generic scheme is considered from two aspects. On the one hand, the security of the indistinguishability of data ciphertext against selective encryption access policy and chosen-ciphertext attacks [28] and the unforgeability of signature against selective signature attribute set and chosen-message attacks [30]. On the other hand, the security of searchable encryption is composed of the indistinguishability of keyword ciphertext or index against selective encryption access policy and chosen-keyword attacks, and the secrecy of keyword against chosen-token attacks [15].

1) Indistinguishability of data ciphertext

We formalize the indistinguishability of data ciphertext against selective encryption access policy and adaptive chosen-ciphertext attacks by the following game between a challenger $C$ and an adversary $A$:

- **Init**: $C$ gives the space of signature attributes $\mathcal{U}_s$ and the space of encryption attributes $\mathcal{U}_e$. Next, $A$ chooses a challenge encryption access structure $\langle D^*, \phi^* \rangle$ and sends it to $C$.

- **Setup**: Given the security parameter $k$, $C$ runs the algorithm $Setup(k)$ to output the public parameters $pp$, while keeps the master secret key $msk$ to himself.

- **Phase 1**: $A$ queries the following oracles for polynomially times:
  - $O_{\mathcal{S}_E}(S, \rho)$: On input a signature access structure $\langle S, \rho \rangle$, $C$ runs $sExtract(msk, \langle S, \rho \rangle)$ to generate a signing key $sk_{\langle S, \rho \rangle}$ and sends it to $A$.
  - $O_{\mathcal{D}_E}(A_d)$: On input a decryption attribute set $A_d$ such that $A_d \notin \langle D^*, \phi^* \rangle$, $C$ runs $dExtract(msk, A_d)$ to generate a decryption key $sk_{A_d}$ for $A$.
  - $O_{\mathcal{S}_C}(m, A_s, \langle D, \phi \rangle)$: Given a message $m$, an encryption access structure $\langle D, \phi \rangle$, and a signature attribute set $A_s$, $C$ selects a signature access structure $\langle S, \rho \rangle$ with the restriction $A_s \notin \mathcal{S}_E$. $C$ runs $sExtract(msk, \langle S, \rho \rangle)$ to generate a signing key $sk_{\langle S, \rho \rangle}$ and sends it to $A$.

- **Phase 2**: $A$ provokes the following oracles for polynomially times:
  - $O_{\mathcal{D}_C}(A_d)$: On input a decryption attribute set $A_d$, $A$ queries $dExtract(msk, A_d)$ to get a decryption key $sk_{A_d}$.
  - $O_{\mathcal{S}_C}(m, A_s, \langle D, \phi \rangle)$: Given a message $m$, an encryption access structure $\langle D, \phi \rangle$, and a signature attribute set $A_s$, $C$ selects a signature access structure $\langle S, \rho \rangle$ with the restriction $A_s \notin \mathcal{S}_E$. $C$ runs $sExtract(msk, \langle S, \rho \rangle)$ to generate a signing key $sk_{\langle S, \rho \rangle}$ and sends it to $A$.

- **Challenge**: $A$ sends the space of signature attributes $\mathcal{U}_s$ and the space of encryption attributes $\mathcal{U}_e$. Next, $A$ chooses a challenge encryption access structure $\langle D^*, \phi^* \rangle$ and sends it to $C$.

- **Phase 3**: $A$ queries the following oracles for polynomially times:
  - $O_{\mathcal{S}_E}(S, \rho)$: On input a signature access structure $\langle S, \rho \rangle$, $C$ runs $sExtract(msk, \langle S, \rho \rangle)$ to generate a signing key $sk_{\langle S, \rho \rangle}$ and sends it to $A$.
  - $O_{\mathcal{D}_E}(A_d)$: On input a decryption attribute set $A_d$ such that $A_d \notin \langle D^*, \phi^* \rangle$, $C$ runs $dExtract(msk, A_d)$ to generate a decryption key $sk_{A_d}$ for $A$.
  - $O_{\mathcal{S}_C}(m, A_s, \langle D, \phi \rangle)$: Given a message $m$, an encryption access structure $\langle D, \phi \rangle$, and a signature attribute set $A_s$, $C$ selects a signature access structure $\langle S, \rho \rangle$ with the restriction $A_s \notin \mathcal{S}_E$. $C$ runs $sExtract(msk, \langle S, \rho \rangle)$ to generate a signing key $sk_{\langle S, \rho \rangle}$ and sends it to $A$.

- **Decryption**: $A$ sends the ciphertext $CT$ and challenge ciphertext $CT'$ to $C$. Then $C$ first queries $sExtract(msk, \langle S, \rho \rangle)$ to generate a signing key $sk_{\langle S, \rho \rangle}$ and sends it to $A$.

- **Verification**: $A$ checks the validity of ciphertext $CT'$ and challenge ciphertext $CT'$. If $CT'$ can pass the verification, $C$ can recover the message $m'$ corresponding to $CT'$.
A searchable attribute-based signcryption scheme

Definition 1: A searchable attribute-based signcryption scheme can achieve the ciphertext indistinguishability under selective encryption access policy and chosen-ciphertext attacks adaptively if an adversary $\mathcal{A}$ wins the above game with a negligible advantage.

2) Unforgeability of signature

We formalize the unforgeability of signature against selective signature attribute set and adaptive chosen-message attacks by the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

- Init: $\mathcal{C}$ gives the space of signature attributes $\mathcal{U}_s$ and the space of encryption attributes $\mathcal{U}_e$. Then, $\mathcal{A}$ chooses a signature attribute set $A_s^*$ and send it to $\mathcal{C}$.
- Setup: Given security parameter $k$, $\mathcal{C}$ runs the Setup($k$) algorithm to output the public parameter $pp$ and keeps the master secret key $msk$ to himself.
- Phase: $\mathcal{A}$ can query the following oracles for many times:
  - $\mathcal{O}_{sE}(\mathcal{S}, \rho)$: Given a signature access structure $(\mathcal{S}, \rho)$ such that $A_s^* \notin (\mathcal{S}, \rho)$, $\mathcal{C}$ computes $sk(\mathcal{S}, \rho)$ by running the algorithm $\text{sExtract}(msk, (\mathcal{S}, \rho))$ and sends it to $\mathcal{A}$.
  - $\mathcal{O}_{dE}(A_d)$: On input a decryption attribute set $A_d$, $\mathcal{C}$ runs the algorithm $\text{dExtract}(msk, A_d)$ to output a decryption key $sk_{A_d}^*$ and sends it to $\mathcal{A}$.
  - $\mathcal{O}_{SC}(m, A_s, (\mathcal{D}, \phi))$: Given a message $m$, an encryption access structure $(\mathcal{D}, \phi)$ and a signature attribute set $A_s$, $\mathcal{C}$ selects a signature access structure $(\mathcal{S}, \rho)$ with the restriction $A_s \in (\mathcal{S}, \rho)$, then obtains $sk(\mathcal{S}, \rho)$ by the oracle $\mathcal{O}_{sE}(\mathcal{S}, \rho)$, and runs the algorithm $\text{Signcrypt}(m, pp, sk(\mathcal{S}, \rho), (\mathcal{D}, \phi))$ to generate $CT$ for $A$.
  - $\mathcal{O}_{DS}(CT, A_d)$: On input a ciphertext $CT$ and a decryption attribute set $A_d$, $\mathcal{C}$ runs the algorithm $\text{dExtract}(msk, A_d)$ to obtain a decryption key $sk_{A_d}$ and returns the message $m$ through executing the algorithm $\text{DeSigncrypt}(pp, CT, sk_{A_d})$.
- Forgery: $\mathcal{A}$ sends a forgery $CT_{A_d^*}(m^*, (\mathcal{D}, \phi^*))$ of message $m^*$ for the challenge signature attributes $A_d^*$.

If the ciphertext $CT_{A_d^*}(m^*, (\mathcal{D}, \phi^*))$ is valid and cannot be gained from $\mathcal{O}_{SC}(m, (\mathcal{D}, A_s, \phi))$, $\mathcal{A}$ wins the game. The advantage of $\mathcal{A}$ in the above game is defined as

$$Adv_{\mathcal{A}} = Pr [A \text{ wins}]$$

Definition 2: A searchable attribute-based signcryption scheme can achieve the unforgeability of signature against selective signature attribute set and adaptive chosen-message attacks if an adversary $\mathcal{A}$ wins the above game with a negligible advantage.

3) Indistinguishability of keyword ciphertext

We formalize the indistinguishability of keyword ciphertext or index against selective encryption access policy and adaptive chosen-keyword attacks by the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

- Init: $\mathcal{C}$ gives the space of encryption attribute set $\mathcal{U}_e$ and adversary $\mathcal{A}$ selects a challenged encryption access structure $(\mathcal{D}, \phi^*)$ and sends it to the challenger $\mathcal{C}$.
- Setup: Given the security parameter $k$, $\mathcal{C}$ runs the algorithm Setup($k$) to output the public parameters $pp$, while the master secret key $msk$ is owned by himself.
- Phase 1: $\mathcal{A}$ queries the following oracles for polynomial times. Meanwhile, $\mathcal{C}$ maintains a keyword list $L_{kw}$, which is empty initially.
  - $\mathcal{O}_{dE}(A_d)$: On input a decryption attribute set $A_d$ such that $A_d \notin (\mathcal{D}, \phi^*)$, $\mathcal{C}$ runs the algorithm $\text{dExtract}(msk, A_d)$ to output a decryption key $sk_{A_d}$. Otherwise, $\mathcal{C}$ outputs a $\bot$.
  - $\mathcal{O}_{CT}(A_d, kw)$: $\mathcal{C}$ first runs the algorithm $\text{dExtract}(sk_{A_d}, pp, kw)$ to return $\mathcal{A}$ a token $T$. If the attribute set $A_d$ satisfies the access structure $(\mathcal{D}, \phi^*)$, $\mathcal{C}$ adds $kw$ to $L_{kw}$.
- Challenge: $\mathcal{A}$ sends two equal length keywords $kw_0$ and $kw_1$ such that $kw_0, kw_1 \notin L_{kw}$ to $\mathcal{C}$. Then, $\mathcal{C}$ randomly picks a bit $b \in \{0, 1\}$ and runs the algorithm $\text{GenIndex}(pp, (\mathcal{D}, \phi^*), kw_0)$ to generate a keyword index $I$ for $A$.
- Phase 2: $\mathcal{A}$ issues a series of queries as in Phase 1. The restriction is that if $A_d \in (\mathcal{D}, \phi^*)$, $\mathcal{A}$ cannot query $\mathcal{O}_{CT}$ with $(A_d, kw_0)$ or $(A_d, kw_1)$.
- Guess: $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. If $b' = b$, $\mathcal{A}$ wins the game.

The advantage of $\mathcal{A}$ in breaking the above game is defined as

$$Adv_{\mathcal{A}} = Pr [b = b'] - \frac{1}{2}$$
Definition 3: A searchable attribute-based signcryption scheme can achieve the index indistinguishability under selective encryption access policy and chosen-keyword attacks if an adversary $A$ wins the above game with negligible advantage.

4) Keyword secrecy

Finally, we formalize the secrecy of keyword against adaptive chosen-token attacks by the following game between a challenger $C$ and an adversary $A$.

- **Setup**: Given the security parameter $k$, $C$ runs the algorithm Setup$(k)$ to generate the public parameters $pp$ and master secret key $msk$.

- **Phase**: $A$ issues the following queries for polynomial times. Meanwhile, $C$ maintains a keyword list $L_{Ad}$, which is empty initially.
  - $O_{deA}(A_d)$: On input a decryption attribute set $A_d$, $C$ returns a corresponding secret key $sk_{Ad}$ to $A$ and adds $A_d$ to the list $L_{Ad}$, which is initially empty.
  - $O_{GT}(A_d, kw)$: Given a decryption attribute set $A_d$, $C$ generates a secret key $sk_{Ad}$ by running $O_{deA}(A_d)$ and returns $A$ the corresponding token $T$ by the algorithm GenToken($sk_{Ad}, kw$).

- **Challenge**: $A$ submits a challenge encryption access structure $(\mathbb{D}^*, \phi^*)$ to $C$. Then, $C$ selects $kw'$ and $A_d^*$ such that $A_d^* \in (\mathbb{D}^*, \phi^*)$. $C$ runs the algorithm GenIndex($pp, (\mathbb{D}^*, \phi^*), kw'$) and the algorithm GenToken($sk_{A_d}, pp, kw'$) to return $A$ a keyword index $I^*$ and token $T^*$. Note that $\forall A_d \in L_{Ad}, A_d \notin (\mathbb{D}^*, \phi^*)$.

- **Guess**: After issuing $q$ distinct keywords, $A$ outputs a keyword $kw'$ and wins the keyword secrecy game if $kw' = kw^*$.

Definition 4: A searchable attribute-based signcryption scheme can achieve the secrecy of keyword against adaptive chosen-token attacks if the advantage of $A$ in breaking the above keyword secrecy game is at most $\frac{1}{|M| - |q|} + \varepsilon$, where $|q|$ denotes the number of queried keywords, $\varepsilon$ is a negligible probability in security parameter $k$, and $M$ is the message space.

IV. OUR CONSTRUCTION

By using Rao et al.'s key-policy attribute-based signcryption scheme [30] and Rao’s ciphertext-policy attribute-based signcryption scheme [28], and modifying Zheng et al.'s attribute-based keyword search method [15], we construct a concrete searchable attribute-based signcryption scheme for electronic personal health record system.

A. THE PROPOSED SCHEME

- **Setup**: Given security parameter $k$, $TA$ outputs two cyclic groups $G_1, G_2$ of a $l$-bit prime order $p$, a generator $g \in G_1$, and a map $e: G_1 \times G_2 \rightarrow \mathbb{G}_2$. Let $U_e = \{att_x\}$ be the space of decryption attributes, $U_t = \{att'_x\}$ be the space of signature attributes, and $M = \{0, 1\}^{l_m}$ be the message space, where $l_m$ denotes the length of message. $TA$ chooses four one-way, collision-resistant hash functions $H_1 : \mathbb{G}_2 \times G_1 \rightarrow \{0, 1\}^m$, $H_2 : \{0, 1\}^r \rightarrow \{0, 1\}^t$, $H_3 : G_1 \rightarrow \mathbb{G}_p$, and $H_4 : \{0, 1\}^r \rightarrow \mathbb{Z}_p$. $TA$ randomly chooses $\alpha, \beta, \gamma, \delta \in \mathbb{Z}_p$, $h_0, T_0, p, y_1, y_2, \cdots, y_t \in G_1$, and $h_x \in G_2$ for each attribute $att_x \in U_e$, $T_x \in G_1$ for each attribute $att'_x \in U_t$, and sets the public parameters $pp$ and master secret key $msk$ as follows:

$$pp = \left\{ G_1, G_2, p, g, e, g^{\alpha}, v, h_0, \langle h_x \rangle_{att_x \in U_e}, g^{\beta} \cdot g^{\gamma}, T_0, \{T_x\}_{att'_x \in U_t} \right\}$$

$$msk = \{ \alpha, \beta, \gamma, \delta \}$$

- **sExtract**: Data owner submits a signature predicate $(S, \rho)$, where $S$ is an $I_s \times n_s$ matrix and row $i$ (i.e. $S_i$) is associated with an attribute $att'_i(\rho)$. Then, $TA$ chooses a random vector $\overline{v}_s = (v_2, v_3, \cdots, v_{n_s}) \in \mathbb{Z}_p^{n_s}$ and sets $\{\lambda_i(\rho) = \overline{S}_i \cdot \overline{v}_s \}_{i \in [I_s]}$. For each row $i \in [I_s]$, $TA$ selects $r_i \in \mathbb{Z}_p$ at random and calculates

$$D_{s,i} = g^{\lambda_i(\rho)(T_0 T_\rho(i))}$$

$$D'_{s,i} = g^{r_i}$$

$$D''_{s,i} = \left\{ D''_{s,i,x} = T'_{x,i}, \forall att'_x \in U_e \setminus \{att'_i(\rho)\} \right\}$$

Finally, the signing key is set as:

$$sk_{(S, \rho)} = \left\{ (S, \rho), \{D_{s,i}, D'_{s,i}, D''_{s,i}\}_{i \in [I_s]} \right\}$$

- **dExtract**: Data user sends a decryption attribute set $A_d$, $TA$ selects $\tilde{r} \in \mathbb{Z}_p$ and calculates

$$A = g^{(\alpha - \tilde{r})/b},$$

$$K_a = g^{\alpha \cdot v_{\tilde{r}}},$$

$$K_d = g^{v_{\tilde{r}}},$$

$$K_{d,e} = h_{\tilde{r}}, \forall att_x \in A_d$$

where $A$ is used to generate a search token. So, $TA$ sets the decryption key as follows:

$$sk_{A_d} = \left\{ A_d, A, K_d, K_{d,e}, \{K_{d,e}\}_{att_x \in A_d} \right\}$$

- **Signcrypt**: To signcrypt a message $m \in \{0, 1\}^{l_m}$, the signcryption selects an authorized signature attribute set $A_s$, which satisfies the signature predicate $(S, \rho)$. Thus, data owner can find a coefficient set $\{w_i\}_{i \in I_s}$ such that $\sum_{i \in I_s} w_i \lambda_i(\rho) = \alpha$, where $I_s = \left\{ i \in [I_s] \right\}$ and a random vector $\overline{v}_d = (v_2, v_3, \cdots, v_{n_s}) \in \mathbb{Z}_p^{n_s}$, which implies $\zeta$ is a secret shared value. Data owner sets $\{\lambda_i(\rho) = D'_i \cdot \overline{v}_d \}_{i \in [I_s]}$. According to the encryption access policy $(\mathbb{D}, \phi)$, where $\mathbb{D}$ is an $I_s \times n_s$ matrix and row $i$ (i.e. $D'_i$) is associated with an attribute $\phi(i)$. Then, the data owner computes the following terms:

1) $E_1 = g^{\phi_i}$
In this section, we illustrate the correctness of the above equations.

**Correctness of (1)** $A_d$ satisfies $(\mathbb{D}, \phi)$, so there is $\sum_{i \in I_e} w'_i \cdot \lambda_{\phi(i)} = \zeta$. Then,

$$E = \prod_{i \in I_e} \left( \frac{e(W_{\phi(i)}, \bar{K}_{d,x})}{e(K'_d, D_{\phi(i)})} \right)^{w'_i} = \frac{e(g,g)^{\alpha \cdot \zeta}}{e(K'_d, D_{\phi(i)})^{w'_i}}$$

and checks whether the following (1) holds or not.

$$e(W_1, tk_1) \cdot E \cdot e(W_2, tk_3) = e(W_0, tk_2) \quad (1)$$

If (1) holds, CS returns the relevant results to data user. Otherwise, output ⊥.

**DeSigncrypt:** On input $CT'$, $A_s, sk_{A_d}$, the algorithm first checks whether the decryption attribute set $A_d$ satisfies the encryption access structure $(\mathbb{D}, \phi)$ or not. If not, it returns ⊥. Otherwise, the algorithm computes

$$\mu = H_3(\mathbb{D}, \phi),$$

$$\beta = H_4(S_1||E_2||E_3||E_4||A_s||\mathbb{D}, \phi),$$

and checks the validity of $CT'$ according to the following equation:

$$e(S_3, g) = e(g,g)^{\alpha} \cdot e(T_o \prod_{\text{att} \in A_s} T_x, S_2) \cdot e(y_0 \prod_{i \in [l]} g_i^{\beta}, E_1) \cdot e((S_1^{d'} \cdot S_2^{d'}), S_1) \quad (2)$$

If (2) does not hold, output ⊥. Otherwise, data user decrypts $CT'$ as follows. Since $A_d$ satisfies $(\mathbb{D}, \phi)$, data user can find a reconstruction coefficient set $\{a_i | i \in I_e\}$ such that $\sum_{i \in I_e} \lambda_{\phi(i)} \cdot a_i = \zeta$, and then obtain $e(g,g)^{\alpha \cdot \zeta}$ through (3).

$$e \left( \sum_{i \in I_e} K_d \cdot \prod_{\text{att} \in A_d} K_{d,\phi(i)}^{a_i}, E_1 \right) = e(g, g)^{\alpha \cdot \zeta} \quad (3)$$

Finally, data user can recover $m$ through

$$m = E_2 \oplus H_1(e(g,g)^{\alpha \cdot \zeta}, S_1)$$

**B. CORRECTNESS**

In this section, we illustrate the correctness of the above equations.

**Correctness of (1)** $A_d$ satisfies $(\mathbb{D}, \phi)$, so there is $\sum_{i \in I_e} w'_i \cdot \lambda_{\phi(i)} = \zeta$. Then,

$$E = \prod_{i \in I_e} \left( \frac{e(W_{\phi(i)} \cdot \bar{K}_{d,x})}{e(K'_d, D_{\phi(i)})} \right)^{w'_i} = e(g,g)^{\alpha \cdot \zeta}$$

$$e(W_1, tk_1) = e(g,g)^{a \cdot c \cdot r \cdot s} \cdot e(g,g)^{b \cdot c \cdot r \cdot s} \cdot H_4(kw')$$

$$e(W_2, tk_3) = e(g,g)^{a \cdot c \cdot s} \cdot e(g,g)^{a \cdot c \cdot s}$$

Z.H. Liu et al.: Searchable attribute-based signcryption scheme
Therefore, we have
\[ e(W_1, tk_1) \cdot E \cdot e(W_2, tk_2) = e(W_0, tk_2). \]

**Correctness of (2).** When \( A_s \) satisfies \((S, p)\), there exists \( \sum_{i \in I_s} \lambda_{p(i)} \cdot w_i = \alpha. \) Then,
\[
\begin{align*}
&\prod_{i \in I_s} \left( D_{s,i} \cdot \prod_{\text{att}_t \in A_s, x \neq p(i)} D_{s,i,x}'' \right)^{w_i} \\
&= \prod_{i \in I_s} \left( g^{\lambda_{p(i)}(T_0 T_p(i))^{r_i} \cdot \prod_{\text{att}_t \in A_s, x \neq p(i)} T_x} \right)^{w_i} \\
&= \sum_{i \in I_s} w_i \lambda_{p(i)}(T_0 \prod_{\text{att}_t \in A_s} T_x) \\
&= g^{\sum_{i \in I_s} \lambda_{p(i)}(T_0 \prod_{\text{att}_t \in A_s} T_x)} \\
&= g^{(S_1, S_2)} e^{\zeta \theta}\end{align*}
\]

So,
\[ S_3 = g^\alpha \left( T_0 \prod_{\text{att}_t \in A_s} T_x \right)^{\xi} \cdot \left( y_0 \prod_{i \in [l]} y_i^{j_i} \right)^{\zeta} \cdot (S_1, S_2)^{\zeta \theta}, \]
where \( \xi = \xi + \sum_{i \in I_s} r_i w_i \).
\[ S_2 = g^\xi \left( D_{s,i} \right)^{w_i} = g^\xi \prod_{i \in I_s} (g^{r_i})^{w_i} = g^\xi. \]
Hence,
\[
e = e(S_3, g) = e(g^\alpha, g) \cdot e(T_0 \prod_{\text{att}_t \in A_s} T_x, g^\xi) \cdot e(y_0 \prod_{i \in [l]} y_i^{j_i}, g^\xi) \cdot e \left( (S_1, S_2)^{\zeta \theta}, g \right) = e(g, g)^{\alpha \xi} \cdot e(T_0 \prod_{\text{att}_t \in A_s} T_x, S_2) \cdot e(y_0 \prod_{i \in [l]} y_i^{j_i}, E_1) \cdot e \left( (S_1, S_2)^{\zeta \theta}, S_1 \right)
\]

**Correctness of (3)** Since \( A_d \) satisfies \((D, \phi)\), we have \( \sum_{i \in I_d} \lambda_{\phi(i)} a_i = \zeta. \) Thus,
\[
e \left( K_d, \prod_{i \in I_d} K_{d, \phi(i)}, E_1 \right) \\
= e \left( K_d', \prod_{i \in I_d} (E_3, i)^{\alpha_i} \right) \\
= e \left( g^{\alpha_i} \cdot v^{\zeta} \prod_{i \in I_d} h_{\phi(i)}^{s_i} g' \right) \\
= e \left( g^{\zeta} \cdot \prod_{i \in I_d} v^{\lambda_{\phi(i)}} \cdot h_{\phi(i)}^{s_i} \right)^{\alpha_i} \\
= e(g, g)^{\alpha \zeta} \cdot e(g, v)^{\zeta \cdot \alpha} \cdot e \left( \prod_{i \in I_d} h_{\phi(i)}^{\alpha_i}, g \right)^{\zeta \cdot \alpha} \\
= e(g, v)^{\zeta \cdot \alpha} \cdot e \left( \prod_{i \in I_d} h_{\phi(i)}^{\alpha_i}, g \right) \\
= e(g, g)^{\zeta \cdot \alpha}.
\]

Therefore, we prove that the proposed scheme is correct.

**C. SECURITY PROOF**

Based on Rao et al.'s scheme [28] [30] and Zheng et al.'s scheme [15], the security of the proposed scheme can be guaranteed through the following four theorems. The detailed proof will be shown in Appendix A-D.

**Theorem 1:** The proposed scheme can achieve ciphertext indistinguishability under selective encryption access policy and chosen-ciphertext attacks based on the hardness assumption of the \( w \)-DBDH problem without using any random oracle.

**Theorem 2:** The proposed scheme is unforgeable under selective signature attribute set and adaptive chosen-message attacks based on the hardness assumption of the \( w \)-CDHE problem without using the random oracle.

**Theorem 3:** The proposed scheme can achieve keyword indistinguishability under selective encryption access policy and chosen-keyword attacks in the the generic bilinear group model.

**Theorem 4:** The proposed scheme can guarantee keyword secrecy in the case of the given one-way hash function \( H_4 \).

**V. PERFORMANCE EVALUATION**

We evaluate the efficiency of the proposed SABSC scheme in terms of both computation cost and size in different phase. For convenience, we give some notations in TABLE 1:

**TABLE 1.** Notations

<table>
<thead>
<tr>
<th>Notations</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e )</td>
<td>Time cost of an exponentiation</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Time cost of a bilinear pairing</td>
</tr>
<tr>
<td>( B_{G_1} )</td>
<td>Bit length of an element of the group ( G_1 )</td>
</tr>
<tr>
<td>(</td>
<td>A_s</td>
</tr>
<tr>
<td>(</td>
<td>A_d</td>
</tr>
<tr>
<td>( \phi_s )</td>
<td>Number of signature attributes required in the Signcrypt</td>
</tr>
<tr>
<td>( \phi_d )</td>
<td>Number of decryption attributes required in the DeSigncrypt</td>
</tr>
</tbody>
</table>

From TABLE 2, when a data owner joins the system, TA runs \( 4 |A_s| \) exponentiation operations to generate a corresponding signing key, which has \( 3 |A_s| \) group elements in group \( G_1 \). Similarly, a decryption key of a data user needs to be executed \( \phi_d + 2 \) exponentiation operations and contains \( \phi_d + 2 \) group elements in \( G_1 \). Before sending the data, the data owner needs to encrypt the data. At the Signcrypt phase,
data ciphertext contains $|A_d| + 4$ group elements in group $G_1$, which needs to be executed $|A_d| + 7 + 2|A_d|$ exponentiation operations. In addition, in the GenIndex phase, the keyword ciphertext contains $2|A_d| + 4$ group elements in group $G_1$, and needs to be run $2|A_d| + 4$ exponentiation operations. With the decryption key, the data user can run the GenToken algorithm to generate the token, which needs $\phi_d + 5$ exponentiation operations to generate $\phi_d + 4$ group elements in $G_1$. Furthermore, the cloud server searches over the index that the computation cost is $3p$ bilinear pairing operations and 1 exponentiation operations. Finally, in the DeSigncrypt phase, there only exits $2d$ exponentiation operations.

Under the Windows XP environment, we test the efficiency of the proposed scheme based on the PBC Library. The program runs on a laptop computer configured as Genuine, Intel, CPU, T1500@3.40GHZ, and 2GB RAM. In the experiments, the modulus of the elements in the group is chosen to be 512 bits, the number of attributes ranges from 20 to 80. Fig.2 shows that the cost of time and the size of elements change with the increase of attributes. Furthermore, in ePHR system, the number of attributes is often fewer. In the proposed scheme, when the number of attributes is 80, the maximum time-consuming is about 0.6s. Thus, the experiments show that our solution is efficient and practical in ePHR system.

VI. CONCLUSIONS

We have proposed a new concept of searchable attribute-based signcryption and constructed a concrete scheme with hybrid access policy for electronic personal health record system. The proposed scheme is more appealing on account of its supporting for expressive fine-grained access control, data retrieval and authentication. Furthermore, the proposed scheme achieves essential security goals, such as data confidentiality, keyword secrecy, unforgeable and signcryptor privacy. The proof of security is realized in the standard model. In the future, we will do more work to enhance the efficiency of the scheme. Meanwhile, we will do some work that support the size of ciphertext is constant and user revocation, and ciphertext updating.

VII. CONCLUSION

We have proposed a new concept of searchable attribute-based signcryption and constructed a concrete scheme with hybrid access policy for electronic personal health record system. The proposed scheme is more appealing on account of its supporting for expressive fine-grained access control, data retrieval and authentication. Furthermore, the proposed scheme achieves essential security goals, such as data confidentiality, keyword secrecy, unforgeable and signcryptor privacy. The proof of security is realized in the standard model. In the future, we will do more work to enhance the efficiency of the scheme. Meanwhile, we will do some work that support the size of ciphertext is constant and user revocation, and ciphertext updating.

REFERENCES

ZHENHUA LIU received his B.S. degree from Henan Normal University, and Master and Ph.D degrees from Xidian University, China, in 2000, 2003 and 2009, respectively. He is currently a professor with Xidian University. His research interests include cryptography and information security.

YAOHUI LIU received his B.S. degree from Henan Institute of Science and Technology in 2016. Now, he is a master degree student in Mathematics at Xidian University. His research interests include cryptography and cloud security.

YAOQING FAN received her B.S. degree in 2015 from College of Mathematics and Information Sciences, Taiyuan Normal University. Now, she is a master degree student in Mathematics at Xidian University. Her research interests include cryptography and cloud security.
APPENDIX A PROOF OF THEOREM 1

In this phase, the hash functions \{H_i\}_{i=1} are collision resistant. Assume that the simulator \( C \) has a \( w \)-DBDHE instance \((g, \zeta, Z \in G_2)\), where 

\[ y_{\zeta} = g \cdot g_1^{-1} \cdot g_2 \cdot \cdots \cdot g_w \cdot g^\omega, \quad \omega = g^z, \quad z, \zeta \in \mathbb{Z}_p. \]

Then, \( C \) attempts to distinguish \( Z \) is \( \epsilon (g_{w+1}, g_\zeta) \) or a random element of \( G_2 \) through the following game. \( C \) also receives the pairing parameters \( \sum = (p, G_1, G_2, \epsilon) \). In addition, \( C \) plays the role of a challenger in the game of the security model and interacts with an adversary \( A \).

- **Init:** \( C \) gives the space of signature attributes \( \mathcal{U}_s = \{\text{att}_s\} \), the space of encryption attributes \( \mathcal{U}_e = \{\text{att}_e\} \) and the message space \( \mathcal{M} = \{0, 1\}^m \). Then, the adversary \( A \) chooses an encryption access structure \((\mathbb{D}^*, \phi^*)\) and sends it to \( C \), where \( \mathbb{D}^* \) is an \( I_c \times n_c \) matrix with a row labeling function \( \phi^* : [I_c] \rightarrow \mathcal{U}_e \) and \( n_c \leq w \). Let \( D_{i}^* = \{D_{i,1}^*, D_{i,2}^*, \ldots, D_{i,n_c}^*\} \) be the \( i \)-th row of the matrix \( \mathbb{D}^* \).

- **Setup:** \( C \) generates public parameters for \( A \) as follows:
  1. Select \( \alpha' \in \mathbb{Z}_p \) and set 
     \[ e(g, g)^{\alpha} = e(g, g)^{\alpha'} \cdot e(g_1, g_w) \] 
     implicitly by setting \( \alpha = \alpha' + z^{w+1} \).
  2. For each \( \text{att}^*_s(i) \in \mathcal{U}_s \), select \( f_x \in \mathbb{Z}_p^w \) and set 
     \[ h_x = \begin{cases} 
     g^{f_x} \prod_{j \in [n_c]} g_\phi^{-D_{i,j}}, & \text{if } \phi^*(i) = x, \\
     g^{f_x}, & \text{otherwise},
     \end{cases} \]
  3. For each \( \text{att}^*_e \in \mathcal{U}_e \), choose \( t_0, t_x \in \mathbb{Z}_p \), and compute \( T_0 = g_1 \cdot g^{t_0} \) and \( T_x = g^{t_x} \).
  4. Set \( v = g_1, E_1^* = g^{\alpha'}, \mu^* = H_3(\alpha^*) \), and compute 
     \[ S_1 = g_w^d, S_2 = g_w^{-1} \cdot g^d, \]
     where \( d \in \mathbb{Z}_p^w \).
  5. Pick \( a, b, c, x_0, \ldots, x_l \in \mathbb{Z}_p^w \) and compute 
     \[ g^a, g^b, g^c, y_0 = g^a, \ldots, y_l = g^{x_l}. \]

- **Phase 1:** \( A \) queries the following oracles for polynomial times.

  - \( \mathcal{O}_{SE}(\mathcal{S}, \rho) \): Given a signature LSSS access structure \((\mathcal{S}, \rho)\), \( C \) generates a signing key \( sk(\mathcal{S}, \rho) \) for \( A \) as follows. Note that \( C \) does not know \( z^{w+1} \) and hence the master key \( \alpha = \alpha' + z^{w+1} \), but has the knowledge of \( \alpha' \).
    1. Let \( \mathcal{S} = \{s_{i,j}\}_{i,j} \), where \( S_{i,j} = \{s_{i,1}, \ldots, s_{i,n_c}\} \) is the \( i \)-th row of \( \mathcal{S} \).
    2. Pick \( v_2, \ldots, v_{n_c} \in \mathbb{Z}_p^w \) and set \( v_x = (\alpha' + z^{w+1}, v_2, \ldots, v_{n_c}) \), which implies that the secret value is \( \alpha' + z^{w+1} \).

  - \( \mathcal{O}_{SC}(m, \mathcal{D}, \phi, \mathcal{A}_s) \): On input a message \( m \in \mathcal{M} \), an encryption access structure \((\mathcal{D}, \phi)\) and a signature attribute set \( A_s \in \mathcal{U}_s \), \( C \) chooses a signature access structure \((\mathcal{S}, \rho)\) such that \( A_s \in (\mathcal{S}, \rho) \), runs \( \mathcal{O}_{SE}(\mathcal{S}, \rho) \) to generate a signing key \( sk(\mathcal{S}, \rho) \) and runs the Signcrypt \((pp, m, sk(\mathcal{S}, \rho), (\mathcal{D}, \phi)) \) to return \( A \) the ciphertext \( CT \).

  - \( \mathcal{O}_{DS}(CT, (\mathcal{D}, \phi)) \): Given \( CT = \{A_s, (\mathcal{D}, \phi), E_1, E_2, E_3, E_4, S_1, S_2, S_3\} \), \( C \) first checks whether \( E_1 = E_1^* \) or not. If yes, \( C \) outputs \( \bot \). Otherwise, \( C \) continues the following process.

  1. Check whether the decryption attribute set \( A_d \) satisfies the encryption access structure \((\mathcal{D}, \phi)\) or not. If not, \( C \) returns \( \bot \). Otherwise, \( C \) continues to execute the algorithm.
2) If $A_d \not\in (\mathbb{D}, \phi)$, $C$ obtains a decryption key $sk_{(\mathbb{D}, \phi)}$ according to the oracle $\mathcal{O}_{\text{DS}}(CT, \mathbb{D}, \phi)$, then runs the DeSgincrypt($CT, sk_{(\mathbb{D}, \phi)}$) algorithm to return $A$ the message $m$.

3) Otherwise, $C$ computes $\mu = H_3 (E_1)$ and sets

$$e(g, g)^{\alpha \cdot \epsilon} = e \left( \frac{E_1}{E_1}, g_1 \right)^{\frac{\mu}{p - 1}} \cdot e \left( E_1, g^{\alpha} \right).$$

4) Finally, the message

$$m = E_2 \oplus H_1 \left( e(g, g)^{\alpha \cdot \epsilon}, S_1 \right)$$

is send to $A$.

Note that $E_1 = g^\epsilon$ is random for $A$, so the probability of $E_1 = E_1^*$ is at most $\frac{1}{p}$.

**Challenge**: $A$ sends two messages $m_0, m_1 \in \mathcal{M}$ and a signature attribute set $A^*_s$ to $C$. $C$ picks a random bit $b^* \in \{0, 1\}$ and signcrypt the message $m_{b^*}$ with the input $(\mathbb{D}^*, \phi^*)$ and $A^*_s$ as follows:

1) Set $E_1^* = g^{b^*}$ and $\mu^* = H_3 (E_1^*)$.

2) Select $\theta \in \mathbb{Z}_p^*$, compute

$$S_1^* = (g^{b^*})^\theta, E_2^* = H_1 \left( Z \cdot e \left( g^{b^*}, g^\alpha \right)^\epsilon, S_1^* \right) \oplus m_{b^*}$$

3) Pick $o_1, \ldots, o_{n_z} \in \mathbb{Z}_p$ and set

$$\vec{v}_d = (\zeta \cdot z + o_2, \zeta \cdot z^2 + o_3, \ldots, \zeta \cdot z^{n_z - 1} + o_{n_z})$$

for each $\phi^*(i) \in A^*_s$, then

$$E_3^* = \left\{ E^*_3,i = \sum_{i=2}^{n_z} o_i \cdot D^*_i \cdot (g^{b^*})^{f_s} \right\}_{i \in [l_z]}.$$

4) Select $\xi \in \mathbb{Z}_p^*$, set $S_2^* = g^{\xi} \cdot g^{-1}$, which implies $\xi = \xi - z^{w}$.

5) Set $E_4^* = (g^{b^*})^d$.

6) Let

$$\left( j_1^*, \ldots, j_l^* \right) = H_2 \left( S_2^* \| A^*_s \| (\mathbb{D}^*, \phi^*) \right),$$

$$\beta^* = H_4 \left( S_2^* \| E_2^* \| E_3^* \| E_4^* \| A^*_s \| (\mathbb{D}^*, \phi^*) \right),$$

and

$$S_3^* = g^\alpha \left( T_0 \prod_{\text{att}^*_t \in A^*_s} T_x \right)^\xi \left( \prod_{\text{att}^*_t \in A^*_s} g_{w-t_0}^{-t_0} \prod_{\text{att}^*_t \in A^*_s} g_{w-t_x}^{-t_x} \right)^{d \cdot \beta^* \cdot \theta + \sum_{i \in [l]} j_i x_i}.$$

where $j_i^* \in \{0, 1\}$, for all $i \in [l]$.

**Phase 2**: $A$ continues to query the oracles as in Phase 1. The restriction is that $A$ cannot query the $\mathcal{O}_{\text{DS}}(CT, (\mathbb{D}^*, \phi^*))$ for any $A_d$ with $A_d \in (\mathbb{D}^*, \phi^*)$.

**Guess**: $A$ returns a guess $b^* \in \{0, 1\}$. If $b^* = b$, then $C$ can guess that $Z = e(g_{w+1}, g^{\epsilon})$ in the $w$-BDHDE instance.

**Probability analysis**: The event is which $C$ aborts the game in when $A$ queries the $\mathcal{O}_{\text{DS}}$ with the ciphertext satisfying $E_1 = E_1^*$. The probability of this event happened is at most $q_{\text{DS}}/p$, where $q_{\text{DS}}$ is the maximum number of unsigncrypt queries made by adversary. If $C$ dose not abort and $Z = e(g_{w+1}, g^{\epsilon})$, then $C$ provides perfect simulation and hence $Pr[1 \leftarrow C(\vec{y}, \zeta, c, e(g_{w+1}, g^{\epsilon}))] \geq \frac{1}{2} + \epsilon - \frac{q_{\text{DS}}}{p}$. If $Z$ is random element $X \in \mathbb{G}_2$, then $A$ cannot obtain any information about $m_{b^*}$ and hence $Pr[1 \leftarrow C(\vec{y}, \zeta, X)] = \frac{1}{2}$.

Therefore, the challenger $C$ can solve the $w$-BDHE problem with advantage at least $\epsilon - (\frac{q_{\text{DS}}}{p})$. The $w$-BDHDE problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is not hard, so the adversary cannot attack the scheme successfully.

**APPENDIX B PROOF OF THEOREM 2**

Given a $w$-CDHDE problem instance

$$(g, g^z, \ldots, g^{z^w}, g^{-z^{w+2}}, \ldots, g^{z^{2w}}) \in \mathbb{G}_2^{2w},$$

where $z \in \{2, \ldots, p - 1\}$ and $g$ is a generator of $\mathbb{G}_1$. The simulator $C$ attempts to compute $g^{z^{w+1}}$ though the following game. $C$ plays the role of a challenger and interacts with an adversary $A$. Here, $\{H_i\}_{i=1}^q$ are one-way, collision resistant hash functions.

**Init**: $C$ specifies the encryption attribute space $\mathcal{U}_e = \{\text{att}_x\}$ and the signature attribute space $\mathcal{U}_s = \{\text{att}_s\}$. Then, $A$ sends a challenge signature set $\mathcal{A}^*_s \subseteq \mathcal{U}_s$ to $C$.

**Setup**: Given the security parameter $k$, $C$ generates the public parameters as follows:

1) Sample $\alpha^* \in \mathbb{Z}_p^*$, and set $e(g, g)^\alpha = e(g, g)^\alpha$.

2) For all $\text{att}_s \in \mathcal{U}_s$, pick $t_s \in \mathbb{Z}_p^*$ and set $T_s = g^{t_s} \cdot g_{w+1}^{-x_s}$.

3) Select $t_0 \in \mathbb{Z}_p^*$, and set $T_0 = g_{t_0}^0 \prod_{\text{att}^*_t \in A^*_s} T_x^{-1}$.

4) For all $\text{att}_s \in \mathcal{U}_s$, select $f_s \in \mathbb{Z}_p^*$ and set $h_s = g^{f_s}$.

5) Set $v = g_1$. Next, pick $d, d^* \in \mathbb{Z}_p$ and set $S_1 = g^d, S_2 = g^{d^*}$.

6) Let $\psi = k$, where $\psi (l + 1) < p$ and $k$ is the security parameter. Select an integer $\varpi$ with the restriction $0 \leq \varpi \leq l$. Pick $(d_0, \ldots, d_l) \in \mathbb{Z}_p^{l+1}$, $(x_0, \ldots, x_l) \in \mathbb{Z}_p^l + 1$ For $\vec{j} = (j_1, \ldots, j_l) \in \{0, 1\}^l$, let

$$F(\vec{j}) = p - \psi \varpi + d_0 + \sum_{i \in [l]} j_i d_i,$$

and

$$J(\vec{j}) = x_0 + \sum_{i \in [l]} j_i x_i.$$

So, $y_0 \prod_{i \in [l]} g_{y_i}^i = g_{w}^F(\vec{j}) \cdot g^J(\vec{j})$. In addition, define a function $K: \{0, 1\}^l \to \{0, 1\}$ by

$$K(\vec{j}) = \begin{cases} 0, & \text{if } d_0 + \sum_{i \in [l]} j_i d_i = 0 \mod \psi \\ 1, & \text{otherwise.} \end{cases}$$
Due to \( \psi(l+1) < p \), when \( K(\vec{y}) = 1 \),
\( F(\vec{a}) \neq 0 \) holds.

**Phase**: \( A \) queries the following oracles for many times:
- \( O_{SE}(S, \rho) \): Given a signature LSSS access structure \( (S, \rho) \) such that \( A^*_\rho \notin (S, \rho) \), where \( S = (S_{i,j})_{i \times n_s} \). C computes a signing key \( sk(S, \rho) \) as follows.
  1) Since \( A^*_\rho \notin (S, \rho) \), there is a vector \( \vec{w} = (-1, w_2, \ldots, w_{n_s}) \) so that \( S\vec{X} \cdot \vec{w} = 0 \) for all \( \forall i \in \{l\} \), where \( \rho(i) \notin A^*_\rho \).
  2) Pick \( v_2, \ldots, v_{n_s} \in \mathbb{Z}_p \) and set
    \[
    \vec{v} = \begin{pmatrix}
    \alpha' + z^{w+1} + 1 \\
    - (\alpha' + z^{w+1}) w_2 + v'_s \\
    \vdots \\
    - (\alpha' + z^{w+1}) w_{n_s} + v'_{n_s} \\
    \end{pmatrix}
    \]
    3) Let \( \vec{v} = - (\alpha' + z^{w+1}) \cdot \vec{w} + \vec{v} \), where \( \vec{v} = (0, v_2, \ldots, v_{n_s}) \).
  4) If \( \rho(i) \in A^*_\rho \), there exists \( \vec{S}_i \cdot \vec{w} = 0 \). Then
    \[
    \lambda_{\rho(i)} = \vec{S}_i \vec{v} = \vec{S}_i(\vec{w} - \vec{v}) = \vec{S}_i \cdot (\vec{v} - \vec{v}) z^{w+1}.
    \]
    Select \( \tau_i \in \mathbb{Z}_p \) and compute
    \[
    D_{s,i} = g^\vec{v} \cdot (T_0 \cdot T_{\rho(i)})^\tau_i,
    \]
    \[
    D_{s,i}' = g^\vec{v},
    \]
    \[
    D_{s,i}'' = \{ D''_{s,i,x} = T_{x}^\tau, \forall att_x \in U_x \}. \]
  5) Otherwise, we have
    \[
    \lambda_{\rho(i)} = \vec{S}_i \vec{v} = \vec{S}_i(\vec{w} - \vec{v}) = \vec{S}_i \cdot (\vec{v} - \vec{v}) z^{w+1}.
    \]
    Let \( g_{s,i} = g_{w+1-x+1+\rho(i)} \) and set \( \tau'_i \in \mathbb{Z}_p \) then,
    \[
    D_{s,i}' = g^{\vec{v}} \cdot g_{\rho(i)},
    \]
    \[
    D_{s,i} = g^{\vec{S}_i(\vec{w} - \vec{v})} \cdot (T_0 \cdot T_{\rho(i)})^\tau'_i \cdot L,
    \]
    \[
    D_{s,i}'' = \{ D''_{s,i,x} = T_{x}^\tau (g^{\vec{S}_i(\rho(i))} g_{s,i}^\tau), \forall att_x \in U_x \setminus \{\rho(i)\} \}
    \]
    where
    \[
    L = \left( g_{\rho(i)}^{-t_{\rho(i)}} \prod_{att_x \in A_x} \left( g_{\rho(i)}^{t_{\rho(i)}} g_{s,i}^\tau \right) \right)^{-1}.
    \]
    - Implicity, \( \tau_i = \tau'_i + (\vec{S}_i \cdot \vec{w}) z^{\rho(i)} \),

- **Phase** \( O_{SE}(A_d) \): Given a decryption attribute set \( A_d \), \( C \) generates a decryption key \( sk_{A_d} \) as follows.
  1) Select \( \tau' \in \mathbb{Z}_p \) and set
    \[
    K_d = g^\tau, \psi^\tau',
    \]
    \[
    K_d' = g^{\tau}, \psi^\tau',
    \]
    \[
    K_d, x = h^\tau \cdot g^{\tau}, att_x \in A_d
    \]
  2) Return the decryption key
    \[
    sk_{A_d} = \left\{ A_d, K_d, K_d', \{ K_d, x \}_{att_x \in A_d} \right\}
    \]
    to \( A \).
- \( O_{SC}(m, (D, \phi), A_x) \): C constructs a signature access structure \( (S, \rho) \) such that \( A \in (S, \rho) \). If \( A^*_\rho \notin (S, \rho) \), C runs \( O_{SE}(S, \rho) \) to obtain the signing key \( sk(S, \rho) \), then outputs a ciphertext \( CT \) by running the algorithm **Signcrypt** \( (pp, m, sk(S, \rho), \phi) \).
  Otherwise, \( C \) outputs a ciphertext as follow.
  1) Pick \( c', e' \in \mathbb{Z}_p \) and set \( S_2 = g^c' \).
  2) Compute
    \[
    j = (j_1, \ldots, j_l) = H_2 (S_2 || A_x || (D, \phi)) \in \{0, 1\}^l
    \]
    If \( K(\vec{y}) = 0 \), abort. Otherwise, set \( E_1 = g^{c'} g_1^{\phi} \), which implies \( \zeta = c' - \frac{1}{F(\vec{y})} \).
  3) Compute
    \[
    E_2 = H_1 \left( \left( g^c g^\alpha \cdot e(g, g) \right)^{-1} \cdot e(g_2, g_1) \right)^{\frac{1}{r(\vec{y})}},
    \]
    \[
    \cdot e(g_2, g_1) \left( g^{c} g_1^{\phi} \right)^{\frac{1}{r(\vec{y})}}, S_1 \right) \in \mathbb{Z}.
    \]
  4) Select \( s_2, \ldots, s_{n_s} \in \mathbb{Z}_p \) and set
    \[
    \vec{v} = \left( c' - \frac{1}{F(\vec{y})}, s_2, \ldots, s_{n_s} \right),
    \]
    thus \( \lambda_{\phi(i)} = \left( c' - \frac{1}{F(\vec{y})} \right) D_{i,1} + \sum_{j=2}^{n_s} s_j D_{i,j} \).
    Set \( \forall i \in \{l\} \),
    \[
    E_3 = \psi \left( \left( c_1 + \sum_{j=2}^{n_s} s_j D_{i,j} \right) - \frac{D_{i,1} l}{F(\vec{y})} g_2^{c_1} h_1^c \right)^{\frac{1}{r(\vec{y})}}.
    \]
  5) \( \mu = H_3 (E_1) \).
  6) \( E_4 = (S_2^\mu S_2) c' \left( g^{\mu d^d + d} \right)^{-\frac{1}{r(\vec{y})}} \).
  7) \( \beta = H_4 (S_1 || E_2 || E_4 || A_x || (D, \phi)) \).
  8) \( S_3 = g^\tau \left( T_{x} \prod_{att_x} (g^{\tau}) \right)^{c'} \left( g^{c_1} g^{\phi} \right)^{\frac{1}{r(\vec{y})}} \).
    \[
    \left( g_1^{-\beta} \right)^{E_4} \cdot \left( \frac{1}{F(\vec{y})} \right).
    \]
- **Phase** \( O_{DS}(CT, A_d) \): Given the ciphertext \( CT \) and the decryption attribute set \( A_d \), \( C \) first runs the oracle \( O_{DS}(A_d) \) to get a decryption key \( sk_{A_d} \) and outputs a message \( m \) by the algorithm **Unsigncrypt** \( (pp, CT, sk_{A_d}) \).

- **Forgery**: For message \( m^* \), such that \( (m^*, A^*_\rho, D_\phi, \phi^*) \) has never been queried in \( O_{SC}(m, A_x, (D, \phi)) \), \( A \) sends a forgery \( CT^* \) of \( m^* \) and a decryption attribute set \( A^* \) to \( C \), where **Unsigncrypt** \( (CT^*, sk_{A^*}) \) = \( m^* \). Then, \( C \) computes \( j = (j_1, \ldots, j_l) = H_2 (S_2^\mu || A^*_\rho || (D, \phi^*)) \) and checks whether \( F(j) = 0 \). If not, \( C \) aborts.
  Otherwise, \( C \) verifies the validity of \( CT^* \) through the (2).
If $A$ wins the game, i.e. $CT^*$ can pass the verification, which means that

$$
E_1^* = g^\xi,
S_1^* = g^{\xi \cdot \theta},
S_2^* = g^{\xi'},
\mu^* = H_3(E_1^*),
\beta^* = H_4(S_1^* \| E_3^* \| E_4^* \| A^*_x \| (D, \phi)),
E_4^* = (S_1^* S_2^*)^\xi,
S_3^* = g^\alpha^{z+1} \cdot \left( T_0 \prod_{a \in A^*} T_x \right)^\xi' \cdot \left( g^{F(\tilde{j}_1)} \cdot g^{j \tilde{j}_1} \right)^\zeta \cdot (E_4^*)^{\beta - \theta}.
$$

As $F(\tilde{j}_1) \equiv 0 \pmod{p}$, the challenger $C$ can compute

$$
S_3^* / (S_2^*)^{\lambda_1} (E_3^*)^{\lambda_1^*} (S_1^*)^\theta \beta^* = g^{z+1}
$$

In order to provide a perfect simulation, the game cannot be aborted in Forgery phase. Following [2], we have

$$
\Pr[\text{abort}] = \frac{1}{\psi} \cdot \frac{1}{l+1} = \frac{1}{k(l+1)}.
$$

Suppose that $A$ wins the game with an advantage $\varepsilon$, $C$ can solve the $w$-CDHE problem with the advantage $\varepsilon' = \frac{\varepsilon}{\psi (l+1)}$.

**APPENDIX C PROOF OF THEOREM 3**

In the indistinguishability of keyword ciphertext game, an adversary $A$ attempts to distinguish $X^s \cdot H_k(\nu)$ and $X^r \cdot H_k(\nu)$. Given a random element $\nu \in \mathbb{Z}_p^*$, the probability of distinguishing $X^r \cdot H_k(\nu)$ from $g^\alpha$ is the same as that of distinguishing $X^r \cdot H_k(\nu)$ from $g^\alpha$, where $X = g^\alpha$. Therefore, if $A$ can break the game with an advantage $\varepsilon$, then it can distinguish $X^r \cdot H_k(\nu)$ from $g^\alpha$ with an advantage $\xi$. Thus, we consider the game in which $A$ can distinguish $X^r \cdot H_k(\nu)$ from $g^\alpha$. The game is shown as follows:

- **Init:** $C$ gives the encryption attribute space $U_c = \{att\}$.

  Then, $A$ chooses a challenge encryption access structure $(\mathbb{D}^*, \phi^*)$ and sends it to $C$, where $\mathbb{D}^*$ is an $l^*_x \times n^*_c$ matrix.

- **Setup:** $C$ selects $a, b, \alpha, c \in \mathbb{Z}_p^*$, $\nu \in \mathbb{G}_1$, then, for each $att \in U_c$, selects $f_x \in \mathbb{Z}_p^*$ and sets $h_\alpha = g^{f_x}$. Finally, $C$ outputs the public parameters $(c, g, p, g^\alpha, g^\beta, g^\gamma)$.

- **Phase 1:** $A$ queries $O_{DE}$ and $O_{GT}$ as follows:
  - $O_{DE}(A_d, msk)$: $C$ computes $A = g^{\alpha c gross}$, then $C$ selects random element $\bar{r} \in \mathbb{Z}_p^*$ and computes
    \[ K_d = g^\alpha \bar{r}, K_d' = g^{\bar{r}}, K_{d,x} = g^{\alpha \bar{r}} \forall att_x \in A_d. \]
    Finally, $C$ returns a decryption key
    \[ sk_{A_d} = (A_d, A, K_d, K_d', \{K_{d,x}\}_{att_x \in A_d}). \]
  - $O_{GT}(sk_{A_d}, kw)$: $C$ first issues $O_{DE}(A_d, msk)$ oracle in order to gain a secret key
    \[ sk_{A_d} = \{A_d, A, K_d, K_d', \{K_{d,x}\}_{att_x \in A_d}\} \]
    then $C$ selects a random element $s \in \mathbb{Z}_p^*$ and computes
    \[ t_k = (g^s \cdot g^{b H_k(kw)})^s, \]
    \[ t_k = g^{s'}, \]
    \[ t_k = A^s, \]
    \[ K_{d,x} = (K_d, k_{d,x})^s, \]
    \[ \{K_{d,x} = (K_d, K_{d,x})^s\}_{att_x \in A_d}. \]

Once attribute set $A_d$ matches the encryption access structure $(\mathbb{D}^*, \phi^*)$, $C$ adds $kw$ to the keyword set list $L_{kw}$, which is initially empty.

- **Challenge Phase:** Given keyword sets $kw_0, kw_1$ which do not belong to keyword set list $L_{kw}$, $C$ selects random elements $r, \zeta \in \mathbb{Z}_p^*$, selects $v_\alpha = \{\zeta, v_2, \ldots, v_n\} \in \mathbb{Z}_p^*$, and computes secret shares of $\zeta$ for each $i \in [l^*_x]$. Then $C$ randomly selects a bit $b^* \in \{0, 1\}$. If $b^* = 0$, $C$ outputs
  \[ W_0 = g^\theta, W_1 = g^{cr}, W_2 = g^{bc}, \]
  \[ \{W_{\phi(i)} = g^{\lambda_{\phi(i)}}, D_{\phi(i)} = \left( v g^f_{\phi(i)} \right)^{\lambda_{\phi(i)}} \}_{i \in [l^*_x]}, \]
  where $\theta \in \mathbb{Z}_p^*$. Otherwise, $C$ returns
  \[ W_0 = g^{a(r+c)}, W_1 = g^{cr}, W_2 = g^{bc}, \]
  \[ \{W_{\phi(i)} = g^{\lambda_{\phi(i)}}, D_{\phi(i)} = \left( v g^f_{\phi(i)} \right)^{\lambda_{\phi(i)}} \}_{i \in [l^*_x]}, \]

- **Phase 2:** This phase is performed as in Phase 1. We notice that as long as $A$ can construct $e(g, g)^{\alpha c (r+c)}$ from some $g^\xi$ that he has queried, then, it can distinguish $g^{a(r+c)}$ from $g^\alpha$, where $X = g^\alpha$.

In the generic group model, let $\psi_0, \psi_1$ be two random injective maps from $\mathbb{Z}_p^*$ into a set of $p^3$ elements. $A$ has a negligible probability to guess an element in the image $\psi_0(\nu)$ and $\psi_1$. Recall that $G_1 = \{\psi_0(x) | x \in \mathbb{Z}_p^*\}$ and $G_2 = \{\psi_1(x) | x \in \mathbb{Z}_p^*\}$. Here, we consider the probability of $A$ constructing $e(g, g)^{\xi H_k(\nu)}$ from some $g^\xi$ with an negligible probability. In other words, $A$ do not have an non-negligible advantage in the indistinguishability of keyword ciphertext game.

We notice that $r$ can only be found in the term $cr$, in order to construct $e(g, g)^{\xi H_k(\nu)}$, we assume that $\xi = \xi' c$, $\xi' \in \mathbb{Z}_p^*$ then $A$ just needs to construct $e(g, g)^{\xi c \alpha}$ through using $\zeta_\alpha$ and $\frac{\alpha c}{\beta}$. However, it is difficult. Because $\alpha c$ is reconstructed if and only if the attribute associated with $h_\alpha$ satisfies the encryption access structure $(\mathbb{D}^*, \phi^*)$. Therefore, the advantage of $A$ in breaking the indistinguishability of keyword ciphertext game is negligible.
APPENDIX D PROOF OF THEOREM 4

We utilize a challenger $C$ to conduct the following keyword secrecy game.

- **Setup:** $C$ first chooses random elements $\alpha, a, b, c \in \mathbb{Z}_p$, $f \in \mathcal{G}_1$. For each $att_x \in U\mathcal{e}$, $C$ selects $f_x \in \mathbb{Z}_p^*$ and sets $h_x = g^{f_x}$. Then, public parameters are set as $pp = (e, g, g^a, g^b, g^c, \{h_x | att_x \in U\mathcal{e}\})$ and master key is set as $msk = (\alpha, a, b, c)$.

- **Phase:** The adversary $A$ queries the following two oracles for polynomially times.
  - $O_{dE}(A_d)$: $C$ runs the $dExtract(msk, A_d)$ algorithm to gain secret key $sk_{A_d}$ and sends it to $A$, then adds $A_d$ to the list $L_{dE}$.
  - $O_{GT}(kw, (\mathbb{D}, \phi))$: $C$ first runs the oracle $O_{dE}(\mathbb{D}, \phi)$ to obtain secret key $sk_{(\mathbb{D}, \phi)}$, then calls the algorithm $GenToken(sk_{A_d}, kw)$ algorithm to generate a token $T$ for $A$.

- **Challenge:** $A$ first chooses an encryption access structure $(\mathbb{D}^*, \phi^*)$, then $C$ selects a decryption attribute set $A_d^*$ such that $A_d^* \in (\mathbb{D}^*, \phi^*)$ and computes $sk_{A_d^*}$ according to the oracle $O_{dE}(msk, A_d^*)$. Next, $C$ randomly selects a keyword $kw^*$ and computes $I^*$ and $T^*$, where $\forall A_d \in L_{dE}, A_d \notin (\mathbb{D}^*, \phi^*)$.

- **Guess:** $A$ outputs a keyword set $kw'$ to $C$, then $C$ computes $I'$ by running the algorithm $GenIndex((\mathbb{D}^*, \phi^*), kw')$. If $Search(T^*, I') = 1$, then $A$ wins the game.

Suppose that $A$ has issued $q_{kw}$ different keyword sets before returning $kw'$, and the probability of $A$ winning the keyword secrecy game is at most $\frac{1}{|M| - |q_{kw}|} + \varepsilon$, where $|q_{kw}|$ is denoted as the number of the different keywords. The size of remaining keyword set space is $|M| - |q_{kw}|$, and $H_4$ is denoted as a one-way hash function, which means recovering $kw^*$ from $H_4(kw^*)$ has at most a negligible probability $\varepsilon$. Therefore, given $|q_{kw}|$ distinct keywords $A$ has queried, $A$ wins the keyword secrecy game with the probability at most $\frac{1}{|M| - |q_{kw}|} + \varepsilon$.

***