Faculty of Engineering and Information Sciences - Papers: Part A

Faculty of Engineering and Information Sciences

2015

Searchable atribute-based mechanism with efficiient data sharing for secure cloud storage

Kaitai Liang
City University of Hong Kong, katai.liang@aalto.fi

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Recommended Citation

Liang, Kaitai and Susilo, Willy, "Searchable atribute-based mechanism with efficiient data sharing for secure cloud storage" (2015). *Faculty of Engineering and Information Sciences - Papers: Part A*. 4244.
https://ro.uow.edu.au/eispapers/4244

Searchable atribute-based mechanism with efficiient data sharing for secure cloud storage

Abstract

To date, the growth of electronic personal data leads to a trend that data owners prefer to remotely outsource their data to clouds for the enjoyment of the high-quality retrieval and storage service without worrying the burden of local data management and maintenance. However, secure share and search for the outsourced data is a formidable task, which may easily incur the leakage of sensitive personal information. Efficient data sharing and searching with security is of critical importance. This paper, for the first time, proposes a searchable attribute-based proxy re-encryption system. When compared to existing systems only supporting either searchable attribute-based functionality or attribute-based proxy re-encryption, our new primitive supports both abilities and provides flexible keyword update service. Specifically, the system enables a data owner to efficiently share his data to a specified group of users matching a sharing policy and meanwhile, the data will maintain its searchable property but also the corresponding search keyword(s) can be updated after the data sharing. The new mechanism is applicable to many real-world applications, such as electronic health record systems. It is also proved chosen ciphertext secure in the random oracle model.

Keywords

storage, sharing, secure, cloud, searchable, efficiient, atribute, data, mechanism

Disciplines

Engineering | Science and Technology Studies

Publication Details

Liang, K. & Susilo, W. (2015). Searchable atribute-based mechanism with efficiient data sharing for secure cloud storage. IEEE Transactions on Information Forensics and Security, 10 (9), 1981-1992.

Searchable Attribute-Based Mechanism with Efficient Data Sharing for Secure Cloud Storage

Kaitai Liang and Willy Susilo*, *Senior Member, IEEE*

***Abstract*—To date, the growth of electronic personal data leads to a trend that data owners prefer to remotely outsource their data to clouds for the enjoyment of the high-quality retrieval and storage service without worrying the burden of local data management and maintenance. However, secure share and search for the outsourced data is a formidable task, which may easily incur the leakage of sensitive personal information. Efficient data sharing and searching with security is of critical importance. This paper, for the first time, proposes a searchable attribute-based proxy re-encryption system. When compared to existing systems only supporting either searchable attribute-based functionality or attribute-based proxy re-encryption, our new primitive supports both abilities and provides flexible keyword update service. Specifically, the system enables a data owner to efficiently share his data to a specified group of users matching a sharing policy and meanwhile, the data will maintain its searchable property but also the corresponding search keyword(s) can be updated after the data sharing. The new mechanism is applicable to many real-world applications, such as electronic health record systems. It is also proved chosen ciphertext secure in the random oracle model.**

Keywords: Searchable attribute-based encryption, keyword update, encrypted data sharing.

I. Introduction

By stepping into the era of big data, Internet users usually choose to upload their personal data to remote cloud servers such that they can reduce the cost of local data management and maintenance. In addition to individuals, many industries and research institutions also follow the trend to remotely store commercial and scientific data to clouds to enjoy high-speed data process and retrieval service. Cloud storage service, accordingly, reveals its infinite practical and commercial potential. However, it meanwhile unavoidably encounters with many unpredictable security and privacy challenges.

Motivation. We start with Attribute-Based Encryption (ABE) with a significant reason that it provides fine-grained expressiveness in data share and search. After storing data to a cloud server, the data owner usually needs two necessary operations: one is *data searching*, and the other is *data sharing*.

Leveraging traditional ABE technology to encrypt data that guarantees the confidentiality of the data, but it limits data sharing and searching. Suppose there is a set of genome encryption, $(Enc(g_1, P_1), ..., Enc(g_n, P_n),)$, which are donated by anonymous volunteers for medical research purpose, where a data g_i is encrypted under a policy P_i such that only a group of researchers matching the policy can acquire the data. The ciphertexts are stored in a remote server. To naively search a specific encrypted genomic data, a researcher, say Alice, has to download all the ciphertexts related to her decryption policy P_A from the server, and next to decrypt them to fulfill the search task locally. When sharing one of her accessible data with her colleagues, Alice has to download the encrypted data, decrypt and further re-encrypt it under the decryption policy of the colleagues. Another interesting behavior, which might be done by Alice, is the keyword update for the shared encrypted data. Consider an encrypted genomic data is with a keyword tag ("Materials at Lab A"). After its sharing to scientists in Lab B, Alice may choose to change its tag as ("Shared to Lab B"). Since traditional ABE cannot support keyword update, Alice has to modify the tags for all shared ciphertexts on her own due to protecting the privacy of the keywords.

However, the above naive approaches do not scale well. Because they bring additional decryption/encryption burden to Alice who is required to be on-line all the time. The cost for the data owner will become more cumbersome, when the number of searching and sharing data is increasing. Besides, the size of download data yields a challenge for local data maintenance that definitely downgrades the advantage of remote data storage.

Alternatively, one may allow a (remote) third party to fulfill data search task, the re-encryption of data and keyword update on behalf of Alice. Nevertheless, this requires the party to be fully trusted as it is granted knowledge of search keyword (i.e. what Alice wants to search) and given the secret key of Alice (i.e. knowing the underlying data). The leak of the above information seriously disgraces the privacy of anonymous donators because the genomic data may contain sensitive information, such as illness. Therefore, this approach is also undesirable due to loss of privacy and confidentiality.

From the above discussions, we can see the importance of secure searching and sharing for encrypted data in remote cloud storage scenario. Protecting the privacy of search (including data and keyword) but also supporting efficient encrypted data sharing in the context of ABE that is an interesting and unsolved problem in the literature. This motivates our work. We further show some existing primitives cannot fully solve the open problem.

Attribute-Based Keyword Search (ABKS). To hide search contents as well as search keywords from cloud server, Boneh et al. [6] introduced the notion of Public Key Encryption (PKE) with keyword search, in which a user delivers a special

*Willy Susilo is the corresponding author.

K. Liang is with the Department of Computer Science, Aalto University, Finland E-mail: kaitai.liang@aalto.fi.

W. Susilo is with Centre for Computer and Information Security Research, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia. E-mail: wsusilo@uow.edu.au.

token associated with keyword(s) to the server such that the server can use the token to allocate all encrypted data with the same keyword(s). The server, however, knows nothing about the keyword(s) and the data. To explore the notion into the context of ABE, Zheng, Xu and Ateniese [38] defined ABKS. Although [38] is the most recent work in the literature of PKE with keyword search, it fails to support encrypted data sharing as the only way for a server to convert a given ciphertext to another one is to obtain the corresponding secret key, i.e. accessing the underlying data.

Attribute-Based Proxy Re-Encryption (ABPRE). To efficiently share an encrypted data with others, Mambo and Okamoto [31] introduced PRE whereby a semi-trusted proxy can transform an encryption of a message to another encryption of the same message without knowing the message. To employ the notion into ABE setting, Liang et al. [26] proposed the notion of ABPRE. Recently, Liang et al. [19], [20] introduced new types of ABPRE with stronger security. Nonetheless, these systems cannot achieve our goals as they do not provide privacy-preserving keyword search.

Gaps Between ABE Keyword Search and Data Share. Usually, an ABKS supporting keyword search does not simultaneously provide decryption service, such as [38]. This is due to a technical limitation in the construction method of trapdoor token (used for searching). Specifically, a trapdoor token consists of a user's "re-randomized" secret key. By using this information, the token holder (i.e. a cloud server) can easily recover the data from a ciphertext encrypted under the decryption policy matching the key. Although the server may use the re-randomized secret key to fulfill data sharing, the confidentiality of the data cannot be guaranteed.

On the other hand, an ABPRE system, e.g., [19], is not compatible with secure data search. Specifically, if we regard an attribute as a search keyword, the privacy of the keyword cannot be achieved as the system is built in the attribute publicly known model. One might question that if we can leverage existing anonymous ABE systems, such as [37], to fill the gaps here. Nonetheless, it is unknown that if we can employ anonymous ABE technique to yield both data share and search as well as keyword privacy.

Our paper focuses on tackling the elusive gaps by proposing a novel ABE system supporting keyword private search and encrypted data sharing simultaneously.

A. Our Contributions

Technical Roadmap. We choose an ABE system with fast decryption [16] as a starting point. The reason of employing ABE is that ABE can provide expressiveness for data share and keyword search compared to other encryption systems. To achieve the privacy of keyword search, we first extend the concrete ABE system [16] into the asymmetric pairings group. Under the property of asymmetric pairings, one cannot tell whether a given ciphertext contains a keyword or not even he can make pairing computations from the ciphertext components. This principle is similar with the technique of anonymous Identity-Based Encryption (IBE). We further allow a token related to a keyword to be constructed via an interaction between Private Key Generator (PKG) and a system user (who specifies the keyword). The construction of the token is somewhat similar to that of the secret key of the user. However, the token (related to a keyword) will not enable its holder (i.e. a cloud server) to decrypt the ciphertext associated with the same keyword. This is a necessary requirement for searchable encryption, i.e. a trapdoor token for a keyword cannot deliver decryption ability to cloud server.

To achieve encrypted data sharing, we combine the resulting scheme with the technique of ABPRE. The re-encryption is tricky in the sense that we mask a secret key of a user with two random factors in the re-encryption key generation phase. One random factor is used to mask the partial decryption value of a ciphertext so that a server cannot obtain knowledge of data by using this intermediate value and meanwhile, the random factor is known only by a group of valid delegatess with appropriate decryption rights. The other random factor is used to hide the components of the secret key such that its knowledge will not be leaked to the server.

Our contributions are described as follows.

- We, for the first time, introduce a novel and practical notion, searchable ABPRE. Our notion guarantees that the keyword search ability of a ciphertext can be remained after the sharing of the ciphertext. It is worth mentioning that all existing public key systems with keyword search fail to guarantee this property.
- We design a concrete searchable Key-Policy (KP) ABPRE system satisfying the above notion. We also prove the scheme chosen ciphertext secure in the Random Oracle Model (ROM). The scheme is the first of its type supporting the privacy of keyword search but also encrypted data sharing.
- As of independent interest, our protocol supports keyword update so that a ciphertext's keyword can be further updated before the ciphertext is shared with others. This property brings a convenience to data owner (who can gain access to the data) in the sense that the ciphertext keyword can be freely modified based on data share record.
- Our system has better efficiency regarding to keyword search and decryption phases when compared to existing systems which only support either data sharing or keyword search in the context of ABE.

B. Related Work

Sahai and Waters [33] introduced the notion of ABE. After that, Goyal et al. [15] proposed a KP-ABE system, in which ciphertexts are associated with attributes, and secret keys are associated with access policies (over attributes). Later on, many classic ABE systems and their variants that have been proposed in the literature, e.g., [36], [18].

Song et al. [34] introduced the first (keyword) Searchable Encryption (SE) system, in which full text search over encrypted data is allowable. Following the notion, many SE systems have been proposed. The existing systems can be categorized into two types: searchable symmetric key encryption (e.g. [10], [35]) and Searchable Public Key Encryption (SPKE) (e.g. [6]). This paper deals with the latter case.

Boneh et al. [6] introduced the PKE with single keyword search. Later on, Golle, Staddon and Waters [14] proposed an encryption mechanism supporting conjunctive keyword search. In TCC 2007, Boneh and Waters [7] designed a more expressive keyword search encryption for not only conjunctive but also keyword subset/range queries. In CRYPTO 2007, Bellare, Boldyreva and O'neill [2] proposed an efficient but deterministic searchable encryption. Some variants of SPKE have been proposed in the literature, such as authorized keyword search [17] and verifiable keyword search [3]. Recently, Zheng, Xu and Ateniese [38] introduced ABKS, in which they combine the keyword search with ABE technology. Nevertheless, none of the aforementioned SPKE systems supports encrypted data sharing.

Following the concept of decryption rights delegation [31], Blaze, Bleumer and Strauss [4] defined PRE. PRE is classified as: unidirectional and bidirectional PRE, and single-hop and multi-hop PRE [1]. Our work deals with the single-hop unidirectional case. Since its introduction many works of PRE have been introduced, e.g., [8], [27], [21], [24], [25], [23], [29].

To combine PRE with ABE, Liang et al. [26] introduced Ciphertext-Policy (CP) ABPRE, and construed a system on top of [11]. Luo et al. [30] proposed another system providing policy with AND gates on multi-valued and negative attributes. Mizuno and Doi [32] proposed a CP-ABPRE scheme which is a bridge for ABE and IBE. Later on, Chandran, Chase and Vaikuntanathan [9] proposed an obfuscation for functional re-encryption with collusion resistant property. Recently, a new CP-ABPRE was proposed in [22], in which the scheme is proven in the ROM. In [20], a CP-ABPRE system was built and proven secure against Chosen Ciphertext Attack (CCA) in the standard model. In [19], a more expressive CP-ABPRE with adaptively CCA security was constructed based on deterministic finite automata. However, the previously introduced systems cannot provide privacy-preserving keyword search.

We compare this work with the most recent SPKE scheme and ABPRE system in terms of functionality and security. We leave the efficiency comparison to Section V. To the best of our knowledge, our scheme is the first to achieve (privacy-preserving) keyword search and encrypted data sharing as well as keyword update. Our system is based on asymmetric decisional l-BDHE assumption (which will be introduced later), while [38], [19] rely on decisional linear assumption and some composite order group assumptions, respectively.

TABLE I
COMPARISON WITH [38], [19]

Sch.	Keyword Search	Encrypted Data Sharing	Security	ROM	Pairings
[38]	✓	✗	CPA	✗	symmetric
[19]	✗	✓	CCA	✗	composite order
Ours	✓	✓	CCA	✓	asymmetric

II. DEFINITION AND ADVERSARY MODELS

A. System Definition

Definition 1: A Searchable Attribute-Based Proxy Re-Encryption with Keyword Update (S-ABPRE-KU) scheme consists of the following algorithms:

1) $(mpk, msk) \leftarrow Setup(1^{\lambda}, U)$: on input a security parameter λ and a universe of description U, output a master public key mpk and a master secret key msk. We will omit mpk in the expression of the following algorithms.
2) $sk_{\mu} \leftarrow KeyGen(msk, \mu)$: on input msk, and a description $\mu \in \{0,1\}^*$, output a secret key sk_{μ}.
3) $CT \leftarrow Enc(m, \nu, KW)$: on input a message $m \in \{0,1\}^{\lambda}$, a description $\nu \in \{0,1\}^*$ for the message and a keyword KW, output an original ciphertext CT which can be further converted.
4) $\tau_{KW} \leftarrow Trapdoor(msk, sk_{\mu}, KW)$: on input msk, sk_{μ} and KW, output a trapdoor token τ_{KW}, which is used to search encrypted data associated with KW.
5) $1/0 \leftarrow Test(CT, \tau_{KW})$: on input a ciphertext CT under a keyword KW', and a trapdoor token τ_{KW}, output 1 if $KW' = KW$, and 0 otherwise.
6) $rk \leftarrow RKGen(sk_{\mu}, \nu, KW)$: on input a sk_{μ}, a new description ν and a new keyword KW, output a re-encryption key rk, where μ does not satisfy ν. The rk is used to convert an original ciphertext under ν' and KW' to a re-encrypted ciphertext of the same message under ν and KW, where μ satisfies ν'.
7) $CT/ \perp \leftarrow ReEnc(CT, rk)$: on input an original ciphertext CT, and a re-encryption key rk, output a re-encrypted ciphertext CT or $\perp$.
8) $m/ \perp \leftarrow Dec(sk_{\mu}, CT)$: on input sk_{μ}, and a ciphertext CT under description ν, output a message m if μ satisfies ν or $\perp$.

B. Threat Models

We define three adversary models below including the selective chosen ciphertext security for original ciphertext and re-encrypted ciphertext, and the keyword privacy models.

Definition 2: An S-ABPRE-KU scheme is selective CCA secure at original ciphertext if no PPT adversary $\mathcal{A}$ can win the game below with non-negligible advantage. In the game, $\mathcal{B}$ is the game challenger.

1) **Init.** $\mathcal{A}$ outputs a challenge description $\nu^* \in \{0,1\}^*$.
2) **Setup.** $\mathcal{B}$ runs $Setup(1^{\lambda}, U)$ and returns mpk to $\mathcal{A}$.
3) **Phase 1.** $\mathcal{A}$ is given access to the following oracles.
    a) $\mathcal{O}_{sk}(\mu)$: given a description μ, output $sk_{\mu} \leftarrow KeyGen(msk, \mu)$.
    b) $\mathcal{O}_{token}(\mu, KW)$: given μ and a keyword KW, output $\tau_{KW} \leftarrow Trapdoor(msk, sk_{\mu}, KW)$, where $sk_{\mu} \leftarrow KeyGen(msk, \mu)$.
    c) $\mathcal{O}_{test}(CT, KW)$: given CT and KW, output $1/0 \leftarrow Test(CT,\ \tau_{KW})$, where $\tau_{KW} \leftarrow Trapdoor(msk,$ $sk_{\mu},\ KW)$, $sk_{\mu} \leftarrow KeyGen(msk,\mu)$.
    d) $\mathcal{O}_{rk}(\mu,\ \nu',\ KW')$: on input μ, a new description for ciphertext ν', and a new keyword KW', output $rk \leftarrow RKGen(sk_{\mu},\ \nu',\ KW')$, where $sk_{\mu} \leftarrow KeyGen(msk, \mu)$.

   e) $\mathcal{O}_{re}(CT, \mu, \nu', KW')$: on input an original ciphertext CT, μ, ν' and KW', output $CT \leftarrow ReEnc(CT, rk)$, where $rk \leftarrow RKGen(sk_\mu, \nu', KW')$ and $sk_\mu \leftarrow KeyGen(msk, \mu)$.
   f) $\mathcal{O}_{dec}(\mu, CT)$: on input μ and a ciphertext CT, output $m \leftarrow Dec(sk_\mu, CT)$, where $sk_\mu \leftarrow KeyGen(msk, \mu)$.

   In Phase 1, the followings are forbidden:

   - $\mathcal{O}_{sk}(\mu)$ for any μ satisfying ν^*;
   - $\mathcal{O}_{rk}(\mu, \nu', KW')$ for any μ satisfying ν^* and meanwhile, $\mathcal{O}_{sk}(\mu')$ for any μ' satisfying ν'.

4) **Challenge.** When $\mathcal{A}$ decides the phase 1 is over, it outputs two equal length messages m_0^*, m_1^*, and a challenge keyword KW^*. $\mathcal{B}$ outputs a challenge original ciphertext as $CT^* = Enc(m_b^*, \nu^*, KW^*)$, where $b \in_R \{0, 1\}$.
5) **Phase 2.** Same as in **Phase 1** except the followings:
   a) $\mathcal{O}_{re}(CT, \mu, \nu'', KW')$: $CT = CT^*$, μ satisfies ν^*, and $\mathcal{O}_{sk}(\mu'')$ for any μ'' satisfying ν''.
   b) $\mathcal{O}_{dec}(\mu, CT)$: if (ν, CT) is a derivative of (ν^*, CT^*). As of [8], a derivative of (ν^*, CT^*) is defined as
      i. (ν^*, CT^*) is a derivative of itself.
      ii. If $\mathcal{A}$ has obtained an rk from $\mathcal{O}_{rk}$ on (μ, ν', KW'), and achieved $CT \leftarrow ReEnc(rk, CT^*)$, (ν', CT) is a derivative of (ν^*, CT^*), where μ satisfies ν^*.
      iii. If $\mathcal{A}$ has issued a re-encryption query on (CT^*, μ, ν', KW') and obtained CT, (ν', CT) is a derivative of (ν^*, CT^*).
6) **Guess.** $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. If $b' = b$, $\mathcal{A}$ wins.

$\mathcal{A}$'s advantage is $Adv_{\mathcal{A}}^{sCCA\text{-}Or}(1^\lambda, U) = |Pr[b' = b] - \frac{1}{2}|$.

Definition 3: An S-ABPRE-KU scheme is selective CCA secure at re-encrypted ciphertext if the advantage $Adv_{\mathcal{A}}^{sCCA\text{-}Re}(1^\lambda, U)$ is negligible for any PPT adversary $\mathcal{A}$ in the following experiment. Set an oracle set $O_1 = \{\mathcal{O}_{sk}, \mathcal{O}_{rk}, \mathcal{O}_{test}, \mathcal{O}_{token}, \mathcal{O}_{dec}\}$ and the advantage as

$$\begin{aligned}
&|Pr[b = b' : (\nu^*, state_1) \leftarrow \mathcal{A}(1^\lambda, U);\\
&(mpk, msk) \leftarrow Setup(1^\lambda, U);\\
&(m_0^*, m_1^*, KW^*, state_2) \leftarrow A^{O_1}(mpk, state_1);\\
&b \in_R \{0, 1\}; CT^* \leftarrow ReEnc(Enc(m_b^*, \nu, KW), rk);\\
&b' \leftarrow A^{O_1}(CT^*, state_2)] - \frac{1}{2}|,
\end{aligned}$$

where $state_1, state_2$ are the state information, ν, KW are chosen by $\mathcal{A}$, $Enc(m_b^*, \nu, KW)$ is generated by $\mathcal{B}$ as well as $rk \leftarrow RKGen(sk_\mu, \nu^*, KW^*)$, μ matches ν. For $\mathcal{O}_{sk}$, if $\mathcal{A}$ issues μ satisfying ν^*, $\mathcal{B}$ outputs $\perp$. There is no restriction for re-encryption key queries, namely, $\mathcal{A}$ can obtain any re-encryption key. This is the reason why we ignore re-encryption oracle here. For decryption query, if the issued ciphertext is the challenge one, output $\perp$.

Remarks. This paper will employ a weaker notion for re-encrypted ciphertext security. Here, O_1 contains $\mathcal{O}_{re}$, and furthermore, all oracles work as in Definition 2 except that there is no restriction on $\mathcal{O}_{re}$, and for $\mathcal{O}_{dec}$ output $\perp$ if CT^* is the challenge re-encrypted ciphertext.

Definition 4: An S-ABPRE-KU scheme guarantees keyword privacy if the advantage is negligible for any PPT adversary $\mathcal{A}$ in the following experiment.

$$\begin{aligned}
&Adv_{\mathcal{A}}^{KP}(1^\lambda, U) = |Pr[b = b' : (\nu^*, state_1) \leftarrow \mathcal{A}(1^\lambda, U);\\
&(mpk, msk) \leftarrow Setup(1^\lambda, U);\\
&(m^*, KW_0^*, KW_1^*, state_2) \leftarrow A^{O_1}(mpk, state_1);\\
&b \in_R \{0, 1\}; CT^* \leftarrow Enc(m^*, \nu^*, KW_b^*);\\
&b' \leftarrow A^{O_1}(CT^*, state_2)] - \frac{1}{2}|,
\end{aligned}$$

where $state_1$, $state_2$ are the state information, KW_0^*, KW_1^* are two distinct keywords. The oracles in O_1 work as in Definition 2 except the followings: for $\mathcal{O}_{test}(CT, KW)$ and $\mathcal{O}_{token}(\mu, KW)$, if either ν is ν^* or μ satisfies ν^*, $\mathcal{B}$ outputs $\perp$. CT^* can be a re-encrypted ciphertext as $CT^* \leftarrow ReEnc(Enc(m^*, \nu^*, KW), rk)$, where $rk \leftarrow RKGen(sk_\mu, \nu, KW_b^*)$, and μ matches ν^*.

Remarks. We make a restriction to the adversary: if he is not granted a decryption right to a ciphertext, he cannot make any further keyword search for the ciphertext. To obtain a keyword search ability, the adversary can query the corresponding secret key, i.e. the decryption rights to the ciphertext, except for the challenge one. This is a weaker notion compared to some existing keyword search definitions. Nonetheless, it is practical enough for real-world applications. In practice, if one cannot have any right to gain access to a data, he/she should not know what the corresponding keyword (associated with the data) is as the keyword may be related to some information of the data. Our definition is able to protect the confidentiality of the data and keyword simultaneously.

III. Preliminaries

A. Asymmetric Pairings

Let $BSetup$ be an algorithm that on input the security parameter λ, outputs the parameters of a bilinear map as $(q, g, \hat{g}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order q, where $q \in \Theta(2^\lambda)$, and g is a random generator of $\mathbb{G}_1$, $\hat{g}$ is a random generator of $\mathbb{G}_2$. The mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ has three properties: (1) *Bilinearity*: for all $a, b \in_R \mathbb{Z}_q^*$, $e(g^a, \hat{g}^b) = e(g, \hat{g})^{ab}$; (2) *Non-degeneracy*: $e(g, \hat{g}) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ is the unit of $\mathbb{G}_T$; (3) *Computability*: e can be efficiently computed. Note that $\mathbb{G}_1$ and $\mathbb{G}_2$ are not the same.

B. Complexity Assumptions

Definition 5: (Symmetric) Decisional Bilinear Diffie-Hellman Exponent (BDHE) Assumption [5]. The decisional l-BDHE assumption is that all PPT algorithms $\mathcal{A}$ given the vector $\vec{y} = (g, g^s, g^a, ..., g^{a^l}, g^{a^{l+2}}, ..., g^{a^{2l}})$, have an advantage negligible in λ of distinguishing $e(g, g)^{a^{l+1}s}$ from a random element R in $\mathbb{G}_T$. The advantage of $\mathcal{A}$ is defined as $|Pr[\mathcal{A}(\vec{y}, e(g, g)^{a^{l+1}s}) = 0] - Pr[\mathcal{A}(\vec{y}, R) = 0]|$, where the probability is taken over the random choice of $a, s \in_R \mathbb{Z}_q^*$, $R \in_R \mathbb{G}_T$, the generator g, the random bits consumed by $\mathcal{A}$.

By leveraging the same technique introduced in [13], we extend the decisional l-BDHE assumption to asymmetric bilinear groups by giving $(g, g^s, g^a, \hat{g}, \hat{g}^a, ..., \hat{g}^{a^l}, \hat{g}^{a^{l+2}}, ..., \hat{g}^{a^{2l}}) \in \mathbb{G}_1^3 \times \hat{\mathbb{G}_2}^{2l-1}$ as input and asking for $e(g, \hat{g})^{a^{l+1}s} \in \mathbb{G}_T$.

Definition 6: (Asymmetric) Decisional l-BDHE Assumption. The asymmetric decisional l-BDHE assumption is that all PPT algorithms $\mathcal{A}$ given the vector $\vec{y} = (g, g^s, g^a, \hat{g}, \hat{g}^a, ..., \hat{g}^{a^l}, \hat{g}^{a^{l+2}}, ..., \hat{g}^{a^{2l}})$, have an advantage negligible in λ of distinguishing $e(g, \hat{g})^{a^{l+1}s}$ from a random element R in $\mathbb{G}_T$. The advantage of $\mathcal{A}$ is defined as $|Pr[\mathcal{A}(\vec{y}, e(g, \hat{g})^{a^{l+1}s}) = 0] - Pr[\mathcal{A}(\vec{y}, R) = 0]|$, where the probability is taken over the choice of $a, s \in_R \mathbb{Z}_q^*$, $R \in_R \mathbb{G}_T$, the generators $g, \hat{g}$, the random bits consumed by $\mathcal{A}$.

C. One-time Symmetric Encryption [12]

We let $\mathcal{K}_D$ be the key space $\{0,1\}^{poly(1^\lambda)}$, and SY be a symmetric encryption scheme, where $poly(1^\lambda)$ is the fixed polynomial size (bound) with respect to the security parameter k. The encryption algorithm $S.Enc$ intakes a key $K \in \mathcal{K}_D$ and a message M, outputs a ciphertext C. The decryption algorithm $S.Dec$ intakes K and C, outputs M or a $\perp$.

IV. An S-ABPRE-KU System

A. A Basic Construction for Small Universe

We propose an S-ABPRE-KU scheme in the key-policy attribute-based setting. Secret key is associated with access policy, and ciphertext is tagged with attribute set. As of [16], we use an LSSS access structure (M, ρ) to represent a policy, U to denote an attribute universe whereby $|U|$ is a polynomial in 1^λ. We use KW to denote either a single keyword or a group of multiple keywords. Since KW is the input of a hash function, if KW represents a group of multiple keywords, it indicates that these keywords are with AND gates. We also note that KW can be arbitrary length.

1) $Setup(1^\lambda, U)$. Run $(q,\ g,\ \hat{g},\ \mathbb{G}_1,\ \mathbb{G}_2,\ \mathbb{G}_T,\ e) \leftarrow BSetup(1^\lambda)$. Choose $\alpha, \hat{\alpha}, \delta, \epsilon_1, \epsilon_2, \epsilon_3, \beta_i, \chi \in_R \mathbb{Z}_q^*$, set $h_i = g^{\beta_i}$, $\hat{h}_i = \hat{g}^{\beta_i}$, $t = g^\delta$, $\hat{t} = \hat{g}^\delta$, $z = g^\chi$, $\hat{z} = \hat{g}^\chi$, $f_1 = g^{\epsilon_1}$, $f_2 = g^{\epsilon_2}$, $f_3 = g^{\epsilon_3}$, $\hat{f}_1 = \hat{g}^{\epsilon_1}$, $\hat{f}_2 = \hat{g}^{\epsilon_2}$, $\hat{f}_3 = \hat{g}^{\epsilon_3}$, $i \in [1, |U|]$. Choose Target Collision Resistant (TCR) hash functions [12]: $H_1 : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \mathbb{Z}_q^*$, $H_2 : \mathbb{G}_T \to \{0,1\}^{2\lambda}$, $H_3 : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_4 : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_5 : \{0,1\}^\lambda \to \mathbb{Z}_q^*$ and $H_6 : \{0,1\}^\lambda \to \{0,1\}^{poly(1^\lambda)}$, a CCA-secure one-time symmetric key encryption $SY = (S.Enc,\ S.Dec)$. The master secret key $msk = (\alpha, \hat{\alpha}, \hat{z}, \hat{t})$, the master public key $mpk = (g$, $\hat{g}$, $\{h_i, \hat{h}_i\}_{i \in [1,|U|]}$, f_1, f_2, f_3, $\hat{f}_1$, $\hat{f}_2$, $\hat{f}_3$, t, z, $e(g, \hat{g})^\alpha$, $e(t, \hat{g})^{\hat{\alpha}}$, $e(z, \hat{g})^{\hat{\alpha}}$, H_1, H_2, H_3, H_4, H_5, H_6, $SY)$.
2) $KeyGen(msk, (M, \rho))$. Let M be an $l \times n$ matrix, and ρ be the function that associates rows of M to attributes. Choose a random vector $(\vec{v}) = (\alpha, y_2, ..., y_n) \in \mathbb{Z}_q^{*n}$ which to be used to share α. For $i = 1$ to l, compute $\phi_i = (\vec{v}) \cdot M_i$, where M_i is the vector related to i-th row of M. Choose $r_1, ..., r_l \in_R \mathbb{Z}_q^*$, and set $sk_{(M,\rho)}$ as
$$D_i = \hat{g}^{\phi_i} \cdot \hat{h}_{\rho(i)}^{r_i}, R_i = \hat{g}^{r_i}, \forall d \in \Gamma/\rho(i), Q_{i,d} = \hat{h}_d^{r_i},$$
where $i \in [1, l]$, Γ is the set of distinct attributes in M (i.e. $\Gamma = \{d : \exists i \in [1, l], \rho(i) = d\}$), and Γ/y denotes the set Γ with the element y removed if present. Note $sk_{(M,\rho)}$ implicitly includes (M, ρ).
3) $Enc(m, S, KW)$. Set the original ciphertext CT as
$$A = (m||\sigma) \oplus H_2(e(g, \hat{g})^{\alpha s}), B = g^s, \{C_x = h_x^s\}_{x \in S},$$
$$D = e(t^{H_3(KW)} z, \hat{g})^{\hat{\alpha} s}, E_1 = f_1^s,$$
$$E_2 = (f_2^{H_4(A, B, \{C_x\}_{x \in S}, D, E_1)} f_3)^s,$$
where $\sigma \in \{0,1\}^\lambda$, $m \in \{0,1\}^\lambda$ and $s = H_1(m, \sigma)$. CT implicitly contains S.
4) $Trapdoor(msk, sk_{(M,\rho)}, KW)$. The trapdoor is generated by the collaboration between a secret key holder and the fully trusted PKG.
   - The PKG chooses a random vector $\vec{\mathbb{V}} = (\hat{\alpha}, \hat{y}_2, ..., \hat{y}_n) \in \mathbb{Z}_q^{*n}$ which to be used to share $\hat{\alpha}$. For $i = 1$ to l, it sets $\hat{\phi}_i = (\vec{\mathbb{V}}) \cdot M_i$. It further sends the following values to the user
   $$\tau_{1,i} = (\hat{t}^{H_3(KW)} \hat{z})^{\hat{\phi}_i} \cdot \hat{h}_{\rho(i)}^{\hat{r}_i}, \tau_{2,i} = \hat{g}^{\hat{r}_i},$$
   $$\forall d \in \Gamma/\rho(i), \tau_{3,i,d} = \hat{h}_d^{\hat{r}_i},$$
   where $\hat{r}_1, ..., \hat{r}_l \in_R \mathbb{Z}_q^*$, $i \in [1, l]$, Γ is the set of distinct attributes in M.
   - The user re-randomizes the values, and sets the trapdoor token τ_{KW} as
   $$\tau_{1,i} = \tau_{1,i} \cdot \hat{h}_{\rho(i)}^{\xi_i}, \tau_{2,i} = \tau_{2,i} \cdot \hat{g}^{\xi_i},$$
   $$\forall d \in \Gamma/\rho(i), \tau_{3,i,d} = \tau_{3,i,d} \cdot \hat{h}_d^{\xi_i},$$
   where $\xi_i \in_R \mathbb{Z}_q^*$. Note (M, ρ) implicitly includes in the token.
5) $Test(CT, \tau_{KW})$. Parse CT as (S, A, B, $\{C_x\}_{x \in S}$, D, E_1, E_2), and τ_{KW} as ($\tau_{1,i}$, $\tau_{2,i}$, $\tau_{3,i,d}$). Suppose S associated with CT satisfies (M, ρ) associated with τ_{KW}, there exists a set of constants $\{w_i\}_{i \in I} \in \mathbb{Z}_q^*$ so that $\forall i \in I$, $\rho(i) \in S$ and $\sum_{i \in I} w_i M_i = (1, 0, ..., 0)$. Given an original ciphertext CT associated with a keyword set KW' and a token τ_{KW}, one can verify that
$$\frac{e(B, \prod_{i \in I} (\tau_{1,i} \prod_{x \in \Delta/\rho(i)} \tau_{3,i,x})^{w_i})}{e(\prod_{x \in \Delta} C_x, \prod_{i \in I} \tau_{2,i}^{w_i})} \overset{?}{=} D.$$
If the equation holds, i.e.
$$\frac{e(B, \prod_{i \in I} (\tau_{1,i} \prod_{x \in \Delta/\rho(i)} \tau_{3,i,x})^{w_i})}{e(\prod_{x \in \Delta} C_x, \prod_{i \in I} \tau_{2,i}^{w_i})}$$
$$= \frac{e(g^s, \prod_{i \in I} ((\hat{t}^{H_3(KW')} \hat{z})^{\hat{\phi}_i} \hat{h}_{\rho(i)}^{\hat{r}_i + \xi_i} \prod_{x \in \Delta/\rho(i)} \hat{h}_x^{\hat{r}_i + \xi_i})^{w_i})}{e(\prod_{x \in \Delta} h_x^s, \prod_{i \in I} (\hat{g}^{\hat{r}_i + \xi_i})^{w_i})}$$
$$= e((t^{H_3(KW)} z)^{\hat{\alpha} s}, \hat{g}),$$
it indicates that $KW = KW'$ so that output 1, and output 0 otherwise. Similarly, if CT is a re-encrypted ciphertext, verify $\frac{e(rk_6, \prod_{i \in I} (\tau_{1,i} \prod_{x \in \Delta/\rho(i)} \tau_{3,i,x})^{w_i})}{e(\prod_{x \in \Delta} rk_{7,x}, \prod_{i \in I} \tau_{2,i}^{w_i})} \overset{?}{=} rk_8$.

6) $RKGen(sk_{(M,\rho)}, S, KW)$. Choose $\gamma \in_R \mathbb{Z}_q^*$, $\theta_1, \theta_2 \in_R \{0,1\}^\lambda$, and set rk as

$$rk_{1,i} = D_i^{H_5(\theta_1)} \hat{f}_1^\gamma, rk_2 = \hat{g}^\gamma, rk_{3,i} = R_i^{H_5(\theta_1)},$$
$$rk_{4,i} = \forall d \in \Gamma/\rho(i)\ (Q_{i,d})^{H_5(\theta_1)},$$
$$rk_5 = (\theta_1 || \theta_2) \oplus H_2(e(g,\hat{g})^{\alpha\check{s}}), rk_6 = g^{\check{s}},$$
$$rk_{7,x} = (h_x^{\check{s}})_{x\in S}, rk_8 = e(t^{H_3(KW)}, \hat{g})^{\hat{\alpha}\check{s}},$$
$$rk_9 = (f_2^{H_4(rk_5, rk_6, rk_{7,x}, rk_8)} f_3)^{\check{s}},$$

where $sk_{(M,\rho)} = (D_i, R_i, \forall d \in \Gamma/\rho(i)\ Q_{i,d})$, $i \in [1,l]$, $\check{s} = H_1(\theta_1, \theta_2)$. Note rk includes (M,ρ) and S.

7) $ReEnc(CT, rk)$. Parse CT as (S, A, B, $\{C_x\}_{x\in S}$, D, E_1, E_2), rk as ((M,ρ), S', $rk_{1,i}$, rk_2, $rk_{3,i}$, $rk_{4,i}$, rk_5, rk_6, $\{rk_{7,x}\}_{x\in S'}$, rk_8, rk_9), $i \in [1,l]$.

   i. Check the validity of CT as

$$e(B, \hat{f}_1) \overset{?}{=} e(E_1, \hat{g}), \quad (1)$$
$$e(\prod_{x\in S} C_x, \hat{g}) \overset{?}{=} e(B, \prod_{x\in S} \hat{h}_x). \quad (2)$$
$$e(B, \hat{f}_2^{H_4(A,B,\{C_x\}_{x\in S},D,E_1)} \hat{f}_3) \overset{?}{=} e(E_2, \hat{g}). \quad (3)$$

   If one of the equations does not hold, output $\perp$. Else, proceed.

   ii. If S associated with CT satisfies (M,ρ) associated with rk, let $I \subset \{1,2,...,l\}$ be a set of indices and $\{w_i\}_{i\in I} \in \mathbb{Z}_q^*$ be a set of constants so that $\forall i \in I$, $\rho(i) \in S$ and $\sum_{i\in I} w_i M_i = (1,0,...,0)$, and define $\Delta = \{x : \exists i \in I,\ \rho(i) = x\}$. Compute

$$\Omega = \frac{e(B, \prod_{i\in I}(rk_{1,i} \prod_{x\in\Delta/\rho(i)} rk_{4,x})^{w_i})}{e(E_1, rk_2^{\sum_{i\in I} w_i}) e(\prod_{x\in\Delta} C_x, \prod_{i\in I} rk_{3,i}^{w_i})}$$
$$= \frac{e(g^s, \prod_{i\in I} \hat{g}^{\phi_i, H_5(\theta_1) w_i} \prod_{x\in\Delta} \hat{h}_x^{w_i H_5(\theta_1) r_i} \hat{f}_1^{\gamma w_i})}{e(f_1^s, \hat{g}^{\gamma \sum_{i\in I} w_i}) e(\prod_{x\in\Delta} h_x^s, \prod_{i\in I} \hat{g}^{r_i H_5(\theta_1) w_i})}$$
$$= e(g^s, \hat{g}^{\alpha H_5(\theta_1)}).$$

   iii. Compute the re-encrypted ciphertext as $T_1 = S.Enc(CT||\Omega, H_6(S.Key))$, $T_2 = (rk_5,\ rk_6,\ \{rk_{7,x}\}_{x\in S'}, rk_8, rk_9)$, $T_3 = (T_{3,1} = (S.Key||\theta_3) \oplus H_2(e(g,\hat{g})^{\alpha\tilde{s}})$, $T_{3,2} = g^{\tilde{s}}$, $T_{3,3,x} = (h_x^{\tilde{s}})_{x\in S'}$, $T_{3,4} = (f_2^{H_4(T_{3,1}, T_{3,2}, T_{3,3,x})} f_3)^{\tilde{s}})$, where $S.Key, \theta_3 \in_R \{0,1\}^\lambda$, $\tilde{s} = H_1(S.Key, \theta_3)$.

8) $Dec(sk_{(M,\rho)}, CT)$.

   (1) If CT is the original ciphertext,

      i. Verify Eq. (3). If Eq. (3) does not hold, output $\perp$. Otherwise, proceed.

      ii. If S associated with CT satisfies (M,ρ) associated with sk, there exists a set of constants $\{w_i\}_{i\in I} \in \mathbb{Z}_q^*$ so that $\forall i \in I$, $\rho(i) \in S$ and $\sum_{i\in I} w_i M_i = (1,0,...,0)$. Compute

$$e(B, \prod_{i\in I}(D_i \prod_{x\in\Delta/\rho(i)} Q_{i,x})^{w_i}) / e(\prod_{x\in\Delta} C_x, \prod_{i\in I} R_i^{w_i})$$
$$= \frac{e(g^s, \prod_{i\in I} \hat{g}^{\phi_i w_i} (\prod_{x\in\Delta} \hat{h}_x)^{r_i w_i})}{e(\prod_{x\in\Delta} h_x^s, \prod_{i\in I} \hat{g}^{r_i w_i})}$$
$$= e(g,\hat{g})^{\alpha s}.$$

      iii. Output the message by computing $m||\sigma = A \oplus H_2(e(g,\hat{g})^{\alpha s})$ if $B = g^{H_1(m,\sigma)}$, $E_2 = (f_2^{H_4(A,B,\{C_x\}_{x\in S},D,E_1)} f_3)^{H_1(m,\sigma)}$ and $\prod_{x\in S} C_x = \prod_{x\in S} h_x^{H_1(m,\sigma)}$; else, output $\perp$.

   (2) If CT is the re-encrypted ciphertext,

      i. Verify

$$e(T_{3,2}, \hat{f}_2^{H_4(T_{3,1}, T_{3,2}, T_{3,3,x})} \hat{f}_3) \overset{?}{=} e(T_{3,4}, \hat{g}) \quad (4)$$

      If the equation does not hold, output $\perp$. Otherwise, proceed.

      ii. Compute

$$\frac{e(T_{3,2}, \prod_{i\in I}(D_i \prod_{x\in\Delta/\rho(i)} Q_{i,x})^{w_i})}{e(\prod_{x\in\Delta} T_{3,3,x}, \prod_{i\in I} R_i^{w_i})} = e(g,\hat{g})^{\alpha\tilde{s}}.$$

      and recover $S.Key$ by computing $S.Key||\theta_3 = T_{3,1} \oplus H_2(e(g,\hat{g})^{\alpha\tilde{s}})$. Proceed if $T_{3,2} = g^{H_1(S.Key,\theta_3)}$, $T_{3,4} = (f_2^{H_4(T_{3,1},T_{3,2},T_{3,3,x})} f_3)^{H_1(S.Key,\theta_3)}$ and $\prod_{x\in S} T_{3,3,x} = \prod_{x\in S} h_x^{H_1(S.Key,\theta_3)}$; otherwise, output $\perp$.

      iii. Recover θ_1 from rk_5, rk_6, $\{rk_{7,x}\}_{x\in S}$, rk_8 and rk_9 as above.

      iv. Compute $CT||\Omega = S.Dec(T_1, H_6(S.Key))$, and further $\Omega^{H_5(\theta_1)^{-1}} = e(g^s, \hat{g}^\alpha)$.

      v. If Eq. (3) does not hold, output $\perp$. Otherwise, proceed. Compute $m||\sigma = A \oplus H_2(e(g,\hat{g})^{\alpha s})$, and then output the message m if $B = g^{H_1(m,\sigma)}$ and $E_2 = (f_2^{H_4(A,B,\{C_x\}_{x\in S},D,E_1)} f_3)^{H_1(m,\sigma)}$; otherwise, output $\perp$.

B. Discussions

Trapdoor Generation. In our system, the keyword trapdoor is generated via the collaboration between a fully trusted PKG and a secret key holder. When the key holder needs a trapdoor, he first issues the corresponding request (with keyword(s)) to the PKG, the PKG then returns a related intermediate component. Finally, the key holder re-randomizes the component to become a "real" trapdoor. In our current architecture, we assume all master secret keys (the one for secret key generation, and the other one for trapdoor generation) are known by the PKG only. That is why the secret key holder needs a interaction with the PKG when generating a trapdoor. The system can be extended to allow a secret key holder to generate a trapdoor on his own without any help of PKG. This requires the secret key holder to know $\hat{\alpha}$, namely, $\hat{\alpha}$ is chosen by the secret key holder as one of his secret information. We will regard the extension as one of future works.

Large Universe. To support large universe (i.e. $U = \{0,1\}^*$), we need to choose an additional TCR hash function $H_7 : \{0,1\}^* \to \mathbb{Z}_q^*$, which will be operated as a random oracle in the simulation. We further set $\hat{h}_x = \hat{g}^{\beta_i H_7(x)}$ and $h_x = g^{\beta_i H_7(x)}$ for any attribute $x \in \{0,1\}^*$. We state that the challenger of the security game can construct the values $\hat{h}_x$ and h_x as in the proof of Theorem 1 except for additionally raising a random value $\kappa_7 \in_R \mathbb{Z}_q^*$ (which is a response to the query H_7 on an attribute x) as an extra exponent to the values.

C. Proof of the Basic Construction

For simplicity, we suppose $\{H_i\}_{i\in\{1,2,3,4,5,6\}}$ are TCR hash functions (operated as oracles in the simulation), SY is a CCA-secure one time symmetric encryption.

Theorem 1: Our S-ABPRE-KU scheme for small attribute universe U is IND-sCCA-Or secure under the asymmetric decisional $|U|$-BDHE assumption.

Proof: We assume each of the $|U|$ attributes is a unique integer between 1 and $|U|$ (like [16]). If there exists a PPT adversary $\mathcal{A}$ can break the IND-sCCA-Or security of our scheme, we can construct a PPT algorithm $\mathcal{B}$ to break the asymmetric decisional $|U|$-BDHE assumption. $\mathcal{B}$ is given a problem instance of asymmetric decisional $|U|$-BDHE: (g, g^s, g^a, $\hat{g}$, $\hat{g}^a$, ..., $\hat{g}^{a^{|U|}}$, $\hat{g}^{a^{|U|+2}}$, ..., $\hat{g}^{a^{2|U|}}$,T). In the simulation, $\mathcal{B}$ obtains the following lists which are initially empty.

- Secret key list sk^{list}: record $((M,\rho), sk_{(M,\rho)}, *)$, generated by $\mathcal{B}$ in the simulation, where $*$ denotes if the key has been issued to $\mathcal{A}$ (yes: 1; no: 0).
- Re-encryption key list rk^{list}: record ((M,ρ), S, KW, rk, θ_1, $*$), where $*$ is the wildcard denoting the key is valid ("1"), random ("0") or non-generated "$\perp$".

1) **Init.** $\mathcal{A}$ outputs a challenge attribute set S^* to $\mathcal{B}$.
2) **Setup.** $\mathcal{B}$ chooses random α', $\hat{\alpha}$, β_1, ...,$\beta_{|U|}$, δ, χ, ϵ_1, ϵ_2, $\epsilon_3 \in \mathbb{Z}_q^*$, and sets $t = g^{\delta}$, $\hat{t} = \hat{g}^{\delta}$, $z = g^{\chi}$, $\hat{z} = \hat{g}^{\chi}$, $f_1 = g^{\epsilon_1}$, $f_2 = g^{\epsilon_2}$, $f_3 = g^{\epsilon_3}$, $\hat{f}_1 = \hat{g}^{\epsilon_1}$, $\hat{f}_2 = \hat{g}^{\epsilon_2}$, $\hat{f}_3 = \hat{g}^{\epsilon_3}$, $e(g,\hat{g})^{\alpha} = e(g,\hat{g})^{\alpha'} e(g^a, \hat{g}^{a^{|U|}})$ (i.e. $\alpha = \alpha' + a^{|U|+1}$), $e(g,\hat{g})^{\delta\hat{\alpha}} = e(t,\hat{g})^{\hat{\alpha}}$, $e(g,\hat{g})^{\xi\hat{\alpha}} = e(z,\hat{g})^{\hat{\alpha}}$, for $x \in [1,|U|]$ set $\hat{h}_x = \hat{g}^{\beta_x}$ and $h_x = g^{\beta_x}$ if $x \in S^*$ otherwise set $\hat{h}_x = \hat{g}^{\beta_x}\hat{g}^{a^x}$ and $h_x = g^{\beta_x} g^{a^x}$. $\mathcal{B}$ outputs mpk as (g, $\hat{g}$, $\{h_x, \hat{h}_x\}_{x\in[1,|U|]}$, f_1, f_2, f_3, $\hat{f}_1$, $\hat{f}_2$, $\hat{f}_3$, t, z, $e(g,\hat{g})^{\alpha}$, $e(t,\hat{g})^{\hat{\alpha}}$, $e(z,\hat{g})^{\hat{\alpha}}$, H_1, H_2, H_3, H_4, H_5, H_6, SY), where SY is a CCA-secure one-time symmetric key encryption, and H_i ($i \in \{1,2,3,4,5,6\}$) are the random oracles operated by $\mathcal{B}$. The msk is (α, $\hat{\alpha}$, $\hat{z}$, $\hat{t}$) whereby α is unknown to $\mathcal{B}$.
   a) $H_1(m,\sigma)$: if the query exists on H_1^{list} in a tuple (m,σ,κ_1), return κ_1; else, choose $\kappa_1 \in_R \mathbb{Z}_q^*$, add (m,σ,κ_1) to H_1^{list}, and return $H_1(m,\sigma) = \kappa_1$.
   b) $H_2(R)$: if the query exists on H_2^{list} in a tuple (R,κ_2), return κ_2, where $R \in \mathbb{G}_T$; else, choose $\kappa_2 \in_R \{0,1\}^{2\lambda}$, add (R,κ_2) to H_2^{list}, and return $H_2(R) = \kappa_2$.
   c) $H_3(KW)$: if the query exists on H_3^{list} in a tuple (KW,κ_3), return κ_3; else, choose $\kappa_3 \in_R \mathbb{Z}_q^*$, add (KW,κ_3) to H_3^{list}, and return $H_3(KW) = \kappa_3$.
   d) $H_4(A,B,\{C_x\}_{x\in S},D,E_1)$: if the query exists on H_4^{list} in a tuple $(A,B,\{C_x\}_{x\in S},D,E_1,\kappa_4)$, return κ_4; else, choose $\kappa_4 \in_R \mathbb{Z}_q^*$, add $(A,B,\{C_x\}_{x\in S},D,E_1,\kappa_4)$ to H_4^{list}, and return $H_4(A,B,\{C_x\}_{x\in S},D,E_1) = \kappa_4$.
   e) $H_5(\theta_1)$: if the query exists on H_5^{list} in a tuple (θ_1,κ_5), return κ_5; else, choose $\kappa_5 \in_R \mathbb{Z}_q^*$, add (θ_1,κ_5) to H_5^{list}, and return $H_5(\theta_1) = \kappa_5$.
   f) $H_6(S.Key)$: if the query exists on H_6^{list} in a tuple $(S.Key,\kappa_6)$, return κ_6; else, choose $\kappa_6 \in_R \{0,1\}^{poly(1^{\lambda})}$, add $(S.Key,\kappa_6)$ to H_6^{list}, and return $H_6(S.Key) = \kappa_6$.
3) **Phase 1.** $\mathcal{A}$ issues a series of queries.
   a) $\mathcal{O}_{sk}(M,\rho)$: if there is a tuple $((M,\rho), sk_{(M,\rho)}, *)$ in sk^{list}, $\mathcal{B}$ returns $sk_{(M,\rho)}$. Otherwise,
      - If either (M,ρ) matches S^* or (M,ρ) matches a S where given a rk from (M^*,ρ^*) matching S^* to S, $\mathcal{B}$ outputs $\perp$.
      - Otherwise, let K be the set of rows where the attributes are in S^* (i.e. $i \in K, \rho(i) \in S^*$) and K' be the rows where attributes are not in S^* (i.e. $K' = [1,l]/K$), define a vector $\vec{v} = (v_1 = 1, ..., v_n) \in \mathbb{Z}_q^{*n}$, and for all $i \in K$, $\vec{v}M_i = 0$. For $i \in K$, $\mathcal{B}$ sets $D_i = R_i = \hat{g}^0$; for $i \in K'$, $\mathcal{B}$ computes $c_i = \vec{v}M_i$. Note that we have to let λ_i be $\vec{v}M_i\alpha = c_i\alpha = c_i(\alpha' + a^{|U|+1})$. $\mathcal{B}$ further sets $R_i = \hat{g}^{-c_i a^{|U|+1-\rho(i)}}$, $D_i = \hat{g}^{c_i\alpha'} R_i^{\beta_i}$ (implicitly set $r_i = -c_i a^{|U|+1-\rho(i)}$, where $i \in [1,|U|]$). For all $d \in \Gamma/\rho(i)$, $\mathcal{B}$ sets $Q_{i,d}$ as $\hat{h}_d^{r_i} = \hat{g}^{\beta_d r_i} = \hat{g}^{-\beta_d c_i a^{|U|+1-\rho(i)}}$ if $d \in S^*$ and $\hat{h}_d^{r_i} = \hat{g}^{\beta_d r_i}\hat{g}^{a^d r_i} = \hat{g}^{-\beta_d c_i a^{|U|+1-\rho(i)}} \hat{g}^{-c_i a^{|U|+1-\rho(i)+x}}$ otherwise.

        $\mathcal{B}$ will re-randomize the secret key components. $\mathcal{B}$ chooses $y_2, ..., y_n \in_R \mathbb{Z}_q^{*n-1}$ to form a vector $\vec{v}' = (0, y_2, ..., y_n)$, and sets $\lambda_i' = \vec{v}' M_i$, where $i \in [1,l]$. For all $i \in [1,l]$, $\mathcal{B}$ sets $D_i = D_i \hat{g}^{\lambda_i'} \hat{h}_{\rho(i)}^{r_i'}$, $R_i = R_i \hat{g}^{r_i'}$, $\forall d \in \Gamma/\rho(i)$, $Q_{i,d} = Q_{i,d}\hat{h}_d^{r_i'}$, where $r_i' \in_R \mathbb{Z}_q^*$. Finally, $\mathcal{B}$ adds $((M,\rho), sk_{(M,\rho)}, 1)$ to sk^{list}.
   b) $\mathcal{O}_{token}((M,\rho), KW)$: $\mathcal{B}$ knows knowledge of $\hat{t}$, $\hat{z}$ and $\hat{\alpha}$, and all $\hat{h}_x$ can be constructed from given problem instance. Furthermore, $\mathcal{B}$ chooses a random vector $\vec{\mathbb{V}}$ with the first element $\hat{\alpha}$, and next sets $\hat{\phi}_i = \vec{\mathbb{V}} \cdot M_i$. It chooses $\xi_i, \hat{r}_i \in_R \mathbb{Z}_q^*$, and next constructs the trapdoor token τ_{KW} as in the real scheme.
   c) $\mathcal{O}_{test}(CT, KW)$: $\mathcal{B}$ can always construct a trapdoor τ_{KW} as in $\mathcal{O}_{token}$, it next proceeds to the test easily. If the test holds, $\mathcal{B}$ outputs 1 and 0 otherwise.
   d) $\mathcal{O}_{rk}((M,\rho), S, KW)$: if there is a tuple (($M$, ρ), S, KW, rk, θ_1, 0/1) in rk^{list}, $\mathcal{B}$ returns rk. Otherwise,
      - If (M,ρ) matches S^* and meanwhile, there exists a tuple $((M',\rho'), sk_{(M',\rho')}, 1)$ in sk^{list} so that (M',ρ') matches S, $\mathcal{B}$ outputs $\perp$.
      - If (M,ρ) matches S^* but there is no $sk_{(M',\rho')}$ ((M',ρ') matching S) issued to $\mathcal{A}$, $\mathcal{B}$ chooses D_i, R_i and $Q_{i,d}$ randomly in $\mathbb{G}_2$, $i \in [1,l]$. It further constructs the re-encryption key as in the real scheme. $\mathcal{B}$ finally returns rk and adds the tuple ((M,ρ), S, KW, rk, θ_1, 0) in rk^{list}.
      - Otherwise, $\mathcal{B}$ first constructs the secret key $sk_{(M,\rho)}$ as in $\mathcal{O}_{sk}$, and next generates rk as in the real scheme. $\mathcal{B}$ adds $((M,\rho), sk_{(M,\rho)}, 0)$ and ((M,ρ), S, KW, rk, θ_1, 1) to sk^{list} and rk^{list}, respectively. Note if $sk_{(M,\rho)}$ is already in sk^{list}, $\mathcal{B}$ directly uses it to construct rk.
   e) $\mathcal{O}_{re}(CT, (M,\rho), S, KW)$: if CT is the challenge original ciphertext, and $sk_{(M',\rho')}$ ((M',ρ') satisfying S) is issued to $\mathcal{A}$, output $\perp$. Else if Eq. (1), Eq. (2) and Eq. (3) do not hold, output $\perp$. Otherwise, proceed. If there exists a tuple ((M,ρ), S, KW, rk, θ_1, 0/1) in rk^{list},

$\mathcal{B}$ re-encrypts CT with rk. Otherwise,

- If the first case of step b) does not hold, $\mathcal{B}$ can construct rk as in step b), and next generate the re-encrypted ciphertext by using the re-encryption key. Finally, $\mathcal{B}$ responds the ciphertext to $\mathcal{A}$ and adds $((M,\rho)$, S, KW, rk, θ_1, $0/1)$ to rk^{list}.
- Otherwise, i.e. for the case (M,ρ) satisfies S^* and $sk_{(M',\rho')}$ is in sk^{list} with a symbol tag 1 whereby (M',ρ') matches S. $\mathcal{B}$ checks whether there exist tuples (m,σ,κ_1) and $(e(g,\hat{g})^{\alpha s},\kappa_2)$ in H_1^{list} and H_2^{list}, respectively, such that $A=(m||\sigma)\oplus\kappa_2$ and $B=g^{\kappa_1}$. If not, output $\perp$. Otherwise, $\mathcal{B}$ sets $\Omega=e(g,\hat{g})^{\alpha\kappa_1\kappa_5}$, where κ_5 is the output of issuing H_5 with $\theta_1\in_R\{0,1\}^{\lambda}$. It seals θ_1 in rk_5, rk_6, $\{rk_{7,x}\}_{x\in S}$, rk_8, rk_9 as in the real scheme. It also chooses a $S.Key\in\{0,1\}^{\lambda}$, and sets $T_1=S.Enc(CT||\Omega,\kappa_6)$ where κ_6 is the output of issuing H_6 with $S.Key$. $\mathcal{B}$ constructs T_3 for hiding $S.Key$ as in the real scheme. $\mathcal{B}$ finally returns the re-encrypted ciphertext T_1, T_2, T_3, and adds a tuple $((M,\rho)$, S, KW, $\perp$, θ_1, $\perp)$ to rk^{list}.

f) $\mathcal{O}_{dec}((M,\rho),CT)$: if CT is either a challenge original ciphertext or a derivative of the challenge ciphertext, output $\perp$. Otherwise,

- If CT is an original ciphertext, $\mathcal{B}$ checks Eq. (1), Eq. (2) and Eq. (3). If the equations do not hold, output $\perp$. Else, $\mathcal{B}$ checks whether there exist tuples (m,σ,κ_1) and $(e(g,\hat{g})^{\alpha\kappa_1},\kappa_2)$ in H_1^{list} and H_2^{list}, respectively, such that $A=(m||\sigma)\oplus\kappa_2$ and $B=g^{\kappa_1}$, where $s=\kappa_1$. If not, output $\perp$. Else, output the message m.
- If CT is a re-encrypted ciphertext, $\mathcal{B}$ verifies Eq. (4). If the equation does not hold, output $\perp$. Else, $\mathcal{B}$ checks whether there exist tuples $(S.Key,\theta_3,\tilde{\kappa}_1)$ and $(e(g,\hat{g})^{\alpha\tilde{\kappa}_1},\tilde{\kappa}_2)$ in H_1^{list} and H_2^{list}, respectively, such that $T_{3,1}=(S.Key||\theta_3)\oplus\tilde{\kappa}_2$ and $T_{3,2}=g^{\tilde{\kappa}_1}$. If not, output $\perp$. Else, $\mathcal{B}$ checks $e(\hat{f}_2^{H_4(rk_5,rk_6,rk_{7,x},rk_8)}\hat{f}_3,rk_6)=e(\hat{g},rk_9)$. If the equation does not hold, output $\perp$. Else, $\mathcal{B}$ checks whether there are tuples $(\theta_1,\theta_2,\check{\kappa}_1)$ and $(e(g,\hat{g})^{\alpha\check{\kappa}_1},\check{\kappa}_2)$ in H_1^{list} and H_2^{list}, respectively, such that $rk_5=(\theta_1||\theta_2)\oplus\check{\kappa}_2$ and $rk_6=g^{\check{\kappa}_1}$. If not, output $\perp$. Else, $\mathcal{B}$ computes $CT||\Omega=S.Dec(T_1,H_6(S.Key))$. It checks whether there are tuples (m,σ,κ_1) and $(e(g,\hat{g})^{\alpha\kappa_1},\kappa_2)$ in H_1^{list} and H_2^{list}, respectively, such that $A=(m||\sigma)\oplus\kappa_2$ and $B=g^{\kappa_1}$. If not, output $\perp$. Else, $\mathcal{B}$ checks $\Omega=e(g,\hat{g})^{\alpha\kappa_1H_5(\theta_1)}$. If the equation does not hold, output $\perp$; otherwise, output the message m.

4) **Challenge.** when **Phase 1** is ended, $\mathcal{A}$ outputs two equal length messages m_0,m_1 and a challenge keyword KW^* to $\mathcal{B}$. $\mathcal{B}$ constructs the challenge original ciphertext as

a) Set $B^*=g^s$, $\{C_x^*=(g^s)^{\beta_x}\}_{x\in S^*}$, and $E_1^*=(g^s)^{\epsilon_1}$.

b) Flip a random coin $b\in\{0,1\}$, choose $\sigma^*\in\{0,1\}^{\lambda}$ and $A^*\in\{0,1\}^{2\lambda}$, implicitly define $H_2(T\cdot e(g^s,\hat{g})^{\alpha'})=A^*\oplus(m_b||\sigma^*)$.

c) Issue an H_3 query on (KW^*) to achieve κ_3^*, and define $D^*=e(g^s,\hat{g})^{\hat{\alpha}(\delta\kappa_3^*+\chi)}$.

d) Issue an H_4 query on $(A^*$, B^*, $\{C_x^*\}_{x\in S^*}$, D^*, $E_1^*)$ to achieve κ_4^*, and define $E_2^*=(g^s)^{\epsilon_2\kappa_4^*}(g^s)^{\epsilon_3}$.

e) Output the challenge original ciphertext as $(S^*$, A^*, B^*, $\{C_x^*\}_{x\in S^*}$, D^*, E_1^*, $E_2^*)$.

If $T=e(g,\hat{g})^{a^{|U|+1}s}$, the ciphertext is valid. However, if $T\in_R\mathbb{G}_T$, the challenge ciphertext is independent of the value of the bit b in the view of $\mathcal{A}$.

5) **Phase 2.** Same as in **Phase 1**.

6) **Guess.** $\mathcal{A}$ outputs b'. If $b=b'$, $\mathcal{B}$ outputs 1 (guessing $T=e(g,\hat{g})^{a^{|U|+1}s}$); else, it outputs 0 ($T\in_R\mathbb{G}_T$).

Probability Analysis. The simulations of the oracles are perfect except H_1 and H_2. Let H_1^* and H_2^* be the events that $\mathcal{A}$ has queried (m_b,σ^*) to H_1 and $e(g,\hat{g})^{\alpha\cdot s}$ to H_2 before the challenge phase, respectively. Except for the two cases, the simulations of H_1 and H_2 are perfect. We let $Adv_{H_1^*}$ be the probability of $\mathcal{A}$ in querying (m_b,σ^*) from H_1 before the challenge phase. Similarly, we have $Adv_{H_2^*}$. In the simulations of secret key and test, the responses to $\mathcal{A}$ are perfect. In the simulation of trapdoor token, $\mathcal{B}$ can correctly respond any token query with knowledge of $\hat{t}$ and $\hat{\alpha}$. The response is also perfect. As to the simulation of re-encryption key, the responses to $\mathcal{A}$ are also perfect except for the case where the re-encryption key is randomly generated. It can be seen that $rk_{1,i}$, rk_2, $rk_{3,i}$ and $rk_{4,i}$ (generated by $\mathcal{B}$) can take the form of the corresponding components of the valid re-encryption key, respectively. The indistinguishability between the random re-encryption key and the valid one is equal to the indistinguishability between the encryption (for θ_1) generated by $\mathcal{B}$ and the one constructed in the real scheme. If there exists a $\mathcal{A}_1$ can distinguish the encryptions above, $\mathcal{B}$ can break the asymmetric decisional $|U|$-BDHE problem by using $\mathcal{A}_1$. The simulation given in the challenge phase is also perfect.

In the simulation of re-encryption, the responses to $\mathcal{A}$ are perfect with an exception that $\mathcal{A}$ submits a valid original ciphertext generated without issuing any query to H_1. We denote by $Pr[REErr]$ the probability of the exception. We have $Pr[REErr]\leq\frac{q_{re}}{q}$, where q_{re} is the total number of re-encryption queries.

In the simulation of decryption, it might be possible that $\mathcal{B}$ cannot provide a decryption for a valid ciphertext. Suppose $\mathcal{A}$ can generate a valid ciphertext without querying $e(g,\hat{g})^{\alpha s}$ to H_2, where $s=H_1(m,\sigma)$. Let $valid$ be the event that the original ciphertext or the re-encrypted ciphertext is valid, QH_1 be the event that $\mathcal{A}$ has queried (m,σ) to H_1 and QH_2 be the event that $\mathcal{A}$ has queried $e(g,\hat{g})^{\alpha s}$ to H_2. From the simulation, we have $Pr[valid|\neg QH_2]\leq\frac{q_{H_1}}{2^{2\lambda}}+\frac{1}{q}$, and similarly we have $Pr[valid|\neg QH_1]\leq\frac{q_{H_2}}{2^{2\lambda}}+\frac{1}{q}$, where q_{H_1} and q_{H_2} are the maximum number of random oracle queries to H_1 and H_2. Let $Pr[DErr]$ be the probability that the event $valid|(\neg QH_1\vee\neg QH_2)$ occurs, then we have $Pr[DErr]\leq(\frac{q_{H_1}+q_{H_2}}{2^{2\lambda}}+\frac{2}{q})\cdot q_{dec}$, where q_{dec} denotes the total numbers of decryption queries.

Let Bad denote the event that $(H_1^*|\neg H_2^*)\vee H_2^*\vee REErr\vee DErr$. We have $\bar{\epsilon}=|Pr[b'=b]-\frac{1}{2}|\leq\frac{1}{2}Pr[Bad]\leq\frac{1}{2}(Adv_{H_2^*}+\frac{q_{H_1}+(q_{H_1}+q_{H_2})\cdot q_{dec}}{2^{2\lambda}}+\frac{2q_{dec}+q_{re}}{q})$.

Therefore, we have $Adv_{\mathcal{A}}^{sCCA\text{-}Or} \geq \frac{1}{q_{H_2}}(Adv_{H_2^*}) \geq \frac{1}{q_{H_2}}(2\overline{\epsilon} - \frac{q_{H_1}+(q_{H_1}+q_{H_2})\cdot q_{dec}}{2^{2\lambda}} - \frac{2q_{dec}+q_{re}}{q})$.

From the simulation, the running time of $\mathcal{B}$ is bound by

$$\begin{aligned} t' \leq t + O(1)(&q_{H_1} + q_{H_2} + q_{H_3} + q_{H_4} + q_{H_5} + q_{H_6} \\ &+ q_{sk} + q_{rk} + q_{re} + q_{dec} + q_{test} + q_{token}) \\ &+ t_e(O(l)q_{sk} + O(l + |S|)q_{rk} + O(|S|)q_{re} + O(|S|)q_{dec} \\ &+ O(l)q_{token} + q_{H_1}(q_{re} + q_{dec})O(1)) \\ &+ O(1)t_p(q_{re} + q_{dec} + q_{test}), \end{aligned}$$

where q_{H_i} denotes the total number of random oracle queries to H_i ($i \in \{1,2,3,4,5,6\}$), q_{sk}, q_{rk}, q_{test} and q_{token} denote the total numbers of secret key queries, re-encryption key queries, test queries and trapdoor queries, t_e denotes the running time of an exponentiation in group $\mathbb{G}_1$ (resp. $\mathbb{G}_2$), t_p denotes the running time of a pairing in group $\mathbb{G}_T$, t is the running time of $\mathcal{A}$, l is the number of rows of matrix, $|S|$ is the number of attributes in a set S. ■

Theorem 2: Our S-ABPRE-KU scheme for small attribute universe U is IND-sCCA-Re secure under the asymmetric decisional $|U|$-BDHE assumption.

Proof:

1) **Init.** $\mathcal{A}$ outputs a challenge attribute set S^*.

2) **Setup.** Same as the proof of Theorem 1.

3) **Phase 1.**

a) $\mathcal{O}_{sk}(M,\rho)$: Same as the proof of Theorem 1.

b) $\mathcal{O}_{token}((M,\rho),KW)$: Same as the proof of Theorem 1.

c) $\mathcal{O}_{test}(CT,KW)$: Same as the proof of Theorem 1.

d) $\mathcal{O}_{rk}((M,\ \rho),\ S,\ KW)$: Same as the proof of Theorem 1.

e) $\mathcal{O}_{re}(CT,(M,\rho),S,KW)$: Same as the proof of Theorem 1 except that there is no restriction for query (the issued ciphertext CT has to be in a valid form) here.

f) $\mathcal{O}_{dec}((M,\rho),CT)$: Same as the proof of Theorem 1 except that CT can be any ciphertext but not the challenge re-encrypted ciphertext.

4) **Challenge.** $\mathcal{A}$ outputs m_0, m_1, KW^* and (M,ρ) that does not satisfy S^*. $\mathcal{B}$ sets the challenge re-encrypted ciphertext as follows.

a) Generate a secret key $sk_{(M,\rho)}$ as in $\mathcal{O}_{sk}$, and next construct a re-encryption key $rk^* = (rk_{1,i}^*,\ rk_2^*,\ rk_{3,i}^*,\ rk_{4,i}^*,\ rk_5^*,\ rk_6^*,\ \{rk_{7,x}^*\}_{x\in S^*},\ rk_8^*,\ rk_9^*)$ from (M,ρ) to S^* under KW^* as in the real scheme.

b) Generate an original ciphertext CT as in the real scheme $A^* = (m_b||\sigma) \oplus \kappa_2$, $B^* = g^{s'}$, $\{C_x^* = h_x^{s'}\}_{x\in S}$, $D^* = e(g,\hat{g})^{\hat{\alpha}(\delta\kappa_3+\chi)s'}$, $E_1^* = f_1^{s'}$, $E_2^* = (f_2^{\kappa_4}f_3)^{s'}$, where $b \in \{0,1\}$, $\sigma \in_R \{0,1\}^\lambda$, $s' = H_1(m_b,\sigma)$, S is the attribute set satisfying (M,ρ), κ_2, κ_3, κ_4 are the output of issuing oracles H_2, H_3, H_4 with $e(g,\hat{g})^{\alpha s'}$, KW, $(A^*,B^*,\{C_x\}^*,D^*,E_1^*)$, respectively. Note h_x, t, f_1, f_2 and f_3 can be constructed by $\mathcal{B}$.

c) Re-encrypt CT with rk to obtain Ω^* as in the real scheme. Since CT and rk are correctly constructed, Ω^* is a valid value here. Choose $S.Key \in \{0,1\}^{poly(1^\lambda)}$, compute $T_1^* = S.Enc(CT||\Omega^*,\kappa_6)$, where κ_6 is the output of issuing oracle H_6 with $S.Key$.

d) Set $T_{3,2}^* = g^s$ and $\{T_{3,3,x}^* = (g^s)^{\beta_x}\}_{x\in S^*}$.

e) Choose $\theta_3^* \in \{0,1\}^\lambda$ and $T_{3,1}^* \in \{0,1\}^{2\lambda}$, implicitly define $H_2(T\cdot e(g^s,\hat{g})^{\alpha'}) = T_{3,1}^* \oplus (S.Key||\theta_3^*)$.

f) Issue an H_4 query on $(T_{3,1}^*, T_{3,2}^*, T_{3,3,x}^*)$ to achieve κ_4^*, and define $T_{3,4}^* = (g^s)^{\epsilon_2\kappa_4^*}(g^s)^{\epsilon_3}$.

g) Set $T_2^* = (rk_5^*,\ rk_6^*,\ \{rk_{7,x}^*\}_{x\in S^*},\ rk_8^*,\ rk_9^*)$, and $T_3^* = (T_{3,1}^*,\ T_{3,2}^*,\ T_{3,3,x}^*,\ T_{3,4}^*)$.

h) Output the challenge ciphertext $(S^*,\ T_1^*,\ T_2^*,\ T_3^*)$.

If $T = e(g,\hat{g})^{a^{|U|+1}s}$, the ciphertext is valid by implicitly letting $H_1(S.Key,\theta_3^*) = s$. However, if $T \in_R \mathbb{G}_T$, the challenge ciphertext is independent of the value of the bit b in the view of $\mathcal{A}$.

5) **Phase 2.** Same as in **Phase 1**.

6) **Guess.** $\mathcal{A}$ outputs a guess bit b'. If $b = b'$, $\mathcal{B}$ outputs 1 (guessing $T = e(g,\hat{g})^{a^{|U|+1}s}$); else, it outputs 0 (guessing $T \in_R \mathbb{G}_T$).

The probability analysis and running time calculation are identical to that of the proof of Theorem 1. ■

Theorem 3: Our S-ABPRE-KU scheme for small attribute universe U is keyword private under the asymmetric decisional $|U|$-BDHE assumption.

Proof: If there exists a PPT adversary $\mathcal{A}$ can break the keyword privacy of our scheme, we can construct a PPT algorithm $\mathcal{B}$ to break the asymmetric mDBDH assumption. $\mathcal{B}$ is given a problem instance of the problem: (g, g^s, g^a, $\hat{g}$, $\hat{g}^a$, ..., $\hat{g}^{a^{|U|}}$, $\hat{g}^{a^{|U|+2}}$, ..., $\hat{g}^{a^{2|U|}}$,T). In the simulation, $\mathcal{B}$ obtains the following lists which are initially empty.

- List sk^{list}: record $((M,\rho),sk_{(M,\rho)})$.
- List rk^{list}: record $((M,\rho),S,KW,rk)$.

1) **Init.** $\mathcal{A}$ outputs a challenge attribute set S^*.

2) **Setup.** $\mathcal{B}$ chooses random $\hat{\alpha}'$, α, β_1, ...,$\beta_{|U|}$, δ, χ, ϵ_1, ϵ_2, $\epsilon_3 \in \mathbb{Z}_q^*$, and sets $t = g^\delta$, $\hat{t} = \hat{g}^\delta$, $z = g^\chi$, $\hat{z} = \hat{g}^\chi$, $f_1 = g^{\epsilon_1}$, $f_2 = g^{\epsilon_2}$, $f_3 = g^{\epsilon_3}$, $\hat{f}_1 = \hat{g}^{\epsilon_1}$, $\hat{f}_2 = \hat{g}^{\epsilon_2}$, $\hat{f}_3 = \hat{g}^{\epsilon_3}$, $e(g,\hat{g})^\alpha$, $e(g^\delta,\hat{g})^{\hat{\alpha}'}e(g^{a\delta},\hat{g}^{a^{|U|}}) = e(t,\hat{g})^{\hat{\alpha}}$ (i.e. $\hat{\alpha} = \hat{\alpha}' + a^{|U|+1}$), $e(g^\chi,\hat{g})^{\hat{\alpha}'}e(g^{a\chi},\hat{g}^{a^{|U|}}) = e(z,\hat{g})^{\hat{\alpha}}$, for $x \in [1,|U|]$ set $\hat{h}_x = \hat{g}^{\beta_x}$ and $h_x = g^{\beta_x}$ if $x \in S^*$ otherwise set $\hat{h}_x = \hat{g}^{\beta_x}\hat{g}^{a^x}$ and $h_x = g^{\beta_x}g^{a^x}$. $\mathcal{B}$ outputs mpk as (g, $\hat{g}$, $\{h_x,\hat{h}_x\}_{x\in[1,|U|]}$, f_1, f_2, f_3, $\hat{f}_1$, $\hat{f}_2$, $\hat{f}_3$, t, z, $e(g,\hat{g})^\alpha$, $e(t,\hat{g})^{\hat{\alpha}}$, $e(z,\hat{g})^{\hat{\alpha}}$, H_1, H_2, H_3, H_4, H_5, H_6, SY), where SY is a CCA-secure one-time symmetric key encryption, and H_i ($i \in \{1,2,3,4,5,6\}$) are the random oracles operated by $\mathcal{B}$. The msk is (α, $\hat{\alpha}$, $\hat{z}$, $\hat{t}$) whereby $\hat{\alpha}$ is unknown to $\mathcal{B}$.

3) **Phase 1.**

a) $\mathcal{O}_{sk}(M,\rho)$: since $\mathcal{B}$ has knowledge of α and $h_{\rho(i)}$, it can construct secret key corresponding to any (M,ρ), and next adds the tuple to sk^{list}. When (M,ρ) satisfies S^*, output $\perp$.

b) $\mathcal{O}_{token}((M,\rho),KW)$: if (M,ρ) matches S^*, $\mathcal{B}$ outputs $\perp$. Else, $\mathcal{B}$ is able to set $\tau_{2,i}$ and $\tau_{3,i,d}$ as in $\mathcal{O}_{sk}$ of the proof of Theorem 1 with the exception that $r_i = -(\delta H_3(KW)+\chi)c_i a^{|U|+1-\rho(i)}$, and further computes $\tau_{1,i}$ as $\hat{g}^{(\delta H_3(KW)+\chi)c_i\alpha'}R_i^{(\delta H_3(KW)+\chi)\beta_i}$, where δ, χ are known to $\mathcal{B}$. $\mathcal{B}$ next re-randomizes the elements as in the real scheme, and finally outputs τ_{KW}.

c) $\mathcal{O}_{test}(CT, KW)$: if the attribute set S associated with CT is S^*, $\mathcal{B}$ outputs $\perp$. Else, $\mathcal{B}$ can always compute a trapdoor τ_{KW} as in $\mathcal{O}_{token}$, it next proceeds to the test easily. If the test holds, $\mathcal{B}$ outputs 1 and 0 otherwise.

d) $\mathcal{O}_{rk}((M,\ \rho),\ S,\ KW)$: if there is a tuple $((M, \rho),\ S,\ KW,\ rk)$ in rk^{list}, $\mathcal{B}$ returns rk. Otherwise,

- If (M, ρ) matches S^* and meanwhile, there exists a tuple $((M', \rho'), sk_{(M',\rho')})$ in sk^{list} so that (M', ρ') matches S, $\mathcal{B}$ outputs $\perp$.
- Otherwise, $\mathcal{B}$ first constructs the secret key $sk_{(M,\rho)}$ as in $\mathcal{O}_{sk}$, and next generates rk as in the real scheme. $\mathcal{B}$ adds $((M, \rho),\ S,\ KW,\ rk)$ to sk^{list} and rk^{list} finally.

e) $\mathcal{O}_{re}(CT, (M, \rho), S, KW)$: if C is the challenge original ciphertext, and $sk_{(M',\rho')}$ ((M', ρ') satisfying S) is issued to $\mathcal{A}$, output $\perp$. Else if Eq. (1), Eq. (2) and Eq. (3) do not hold, output $\perp$. Otherwise, proceed. If there exists a tuple $((M, \rho),\ S,\ KW,\ rk)$ in rk^{list}, $\mathcal{B}$ re-encrypts CT with rk. Otherwise, $\mathcal{B}$ first generates a re-encryption key by using the secret key of the delegator (with knowledge of α), next constructs the re-encrypted ciphertext as in the real scheme.

f) $\mathcal{O}_{dec}((M, \rho), CT)$: if CT is either a challenge original ciphertext or a derivative of the challenge ciphertext, output $\perp$. Otherwise, $\mathcal{B}$ can uses the knowledge of α to construct the corresponding secret key to recover m as in the real scheme.

4) **Challenge.** $\mathcal{A}$ outputs m^*, KW_1^* and KW_0^*. $\mathcal{B}$ sets the original challenge ciphertext as

a) Set $B^* = g^s$, $\{C_x^* = (g^s)^{\beta_x}\}_{x \in S^*}$, and $E_1^* = (g^s)^{\epsilon_1}$.

b) Choose $\sigma^* \in \{0,1\}^\lambda$ and set $A^* = (m^*||\sigma^*) \oplus H_2(e(g^s, \hat{g})^\alpha)$.

c) Issue an H_3 query on (KW_b^*) to achieve κ_3^*, flip a random coin $b \in \{0,1\}$, and define $D^* = e(g^s, \hat{g})^{\hat{\alpha}'(\delta\kappa_3^* + \chi)} T^{\delta\kappa_3^* + \chi}$.

d) Issue an H_4 query on $(A^*,\ B^*,\ \{C_x^*\}_{x \in S^*},\ D^*,\ E_1^*)$ to achieve κ_4^*, and define $E_2^* = (g^s)^{\epsilon_2 \kappa_4^*}(g^s)^{\epsilon_3}$.

e) Output the challenge original ciphertext as $(S^*,\ A^*,\ B^*,\ \{C_x^*\}_{x \in S^*},\ D^*,\ E_1^*,\ E_2^*)$.

If $T = e(g, \hat{g})^{a^{|U|+1} s}$, the ciphertext is valid by implicitly setting $H_1(m^*, \sigma^*) = s$. However, if $T \in_R \mathbb{G}_T$, the challenge ciphertext is independent of the value of the bit b in the view of $\mathcal{A}$.

We note that the keyword privacy also keeps in the re-encrypted ciphertext. The construction approach of the challenge re-encrypted ciphertext is somewhat identical to the above ciphertext with the exception: $\mathcal{B}$ can construct the original ciphertext CT as in the real scheme as well as Ω, T_1 and T_3, it further embeds g^s and T into rk_5, rk_6, $rk_{7,x}$, rk_8 and rk_9.

5) **Phase 2.** Same as in **Phase 1**.

6) **Guess.** $\mathcal{A}$ outputs a guess bit b'. If $b = b'$, $\mathcal{B}$ outputs 1 (guessing $T = e(g, \hat{g})^{a^{|U|+1} s}$); else, it outputs 0 (guessing $T \in_R \mathbb{G}_T$).

The probability analysis is identical to the previous one. ■

V. Efficiency Comparison

A. Theoretical Analysis

We compare our scheme with an ABKS (the concrete key-policy ABKS scheme) [38] and an ABPRE [19] in terms of computation and communication cost. Table II shows the comparison of computational cost, and Table III shows the communication comparison.

We now define the notations used in the Tables. Let $|\mathbb{G}|$ and $|\mathbb{G}_T|$ denote the bit-length of an element in groups $\mathbb{G}_1$ (resp. $\mathbb{G}_2$) and $\mathbb{G}_T$, $|St|$ denote the number of state in a deterministic finite automaton, l is the number of row of a matrix, $|W|$ denote the length of a string, $|TL|$ denote the number of leaf in an access tree, $|S|$ denote the number of attribute in an attribute set, λ denote the security parameter and its length $|\lambda|$, $p, e^{(1)}, e^{(2)}$ denote the computation cost of a bilinear pairing, an exponentiation in $\mathbb{G}_1$ (resp. $\mathbb{G}_2$), and an exponentiation in $\mathbb{G}_T$, respectively. For comparison convenience, we suppose our scheme is also in symmetric pairings group, i.e. $\mathbb{G}_1 = \mathbb{G}_2$.

Table II shows that our scheme is the most efficient one in decryption and search (enjoying constant pairings cost), while [38] and [19] suffer from linearly cost, i.e. $O(|S|)$ and $O(|St|)$ pairings, respectively. Compared with [19], our system only requires constant pairings cost in re-encryption phase. Although our scheme needs $O(l^2)$ exponentiations in $\mathbb{G}$ in the generation of trapdoor token, it enjoys better performance in keyword search, re-encryption and decryption phases.

TABLE III
Communication Comparison with [38], [19]

Sch.	Size/Length		
	Ciphertext (Or)	Token	Secret Key
[38]	$O(\|S\|)\|\mathbb{G}\|$	$O(\|TL\|)\|\mathbb{G}\|$	$O(\|TL\|)\|\mathbb{G}\|$
[19]	$O(1)\|\mathbb{G}_T\| + O(\|W\|)\|\mathbb{G}\|$	$\perp$	$O(\|St\|)\|\mathbb{G}\|$
Ours	$O(\|S\|)\|\mathbb{G}\| + O(1)\|\lambda\|$	$O(l^2)\|\mathbb{G}\|$	$O(l^2)\|\mathbb{G}\|$

Table III shows that our scheme has similar complexity with [38] in the size of ciphertext though it needs $O(l^2)$ elements in $\mathbb{G}$ for the token and secret key.

In conclusion, our scheme achieves more practical functionalities (including keyword search, data sharing and keyword update) without requiring a great amount of additional computation and communication cost.

B. Practical Analysis

For the system simulation, we leverage the Java Pairing Based Cryptography Library [28] to calculate the system running time shown in Table IV. Our testbed is: Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz, 3 GB RAM, Ubuntu 10.04; pairing type is a with 160-bit group oder (supersingular curve $Y^2 = X^3 + X$). We also assume all systems only make one time operation for the (equal length) keyword update, ciphertext share and search, respectively. Besides, we suppose the schemes must at least achieve a security level comparable to a symmetric key cryptosystem with an 80-bit key. An elliptic curve cryptosystem with 160-bit keys is needed. Therefore, we set $|\lambda| = 160$ bits, $|\mathbb{G}| = 160$ bits and $|\mathbb{G}_T| = 1024$ bits, respectively. We further set

TABLE II
COMPUTATION COMPARISON WITH [38], [19]

Sch.	Computation Cost					
	Enc	Token Gen	Search	ReEnc	Dec (Or)	Dec (Re)
[38]	$O(\|S\|)e^{(1)}$	$O(\|TL\|)e^{(1)}$	$O(\|S\|)p + O(\|S\|)e^{(2)}$	$\perp$	$\perp$	$\perp$
[19]	$O(\|W\|)e^{(1)} + O(1)e^{(2)}$	$\perp$	$\perp$	$O(\|W\|)e^{(1)} + O(1)e^{(2)} + O(\|St\|)p$	$O(\|St\|)p$	$O(\|St\|)p$
Ours	$O(\|S\|)e^{(1)} + O(1)e^{(2)}$	$O(l^2)e^{(1)}$	$O(\|S\|)e^{(1)} + O(1)p$	$O(1)p$	$O(1)p$	$O(1)p$

$|S| = |TL| = l = 50$, $|W| = 50$, $|St| = 51$ and each state in St can be a successful state. This initialization indicates that an attribute set shares the same number of attributes in an access tree, and each row of the Matrix (used to represent access policy) corresponds to a distinct attribute (where the total number of attribute is also 50). Table V is the comparison of concrete communication cost.

TABLE IV
COMPARISON IN SYSTEM RUNNING TIME

Sch.	Algorithms (ms)					
	Enc	Token Gen	Search	ReEnc	Dec (Or)	Dec (Re)
[38]	940.52	1869.92	1494.71	$\perp$	$\perp$	$\perp$
[19]	436.68	$\perp$	$\perp$	2708.05	2242.06	4502.71
Ours	153.60	7364.35	958.71	1161.36	1888.11	3778.40

TABLE V
COMPARISON IN COMMUNICATION COST

	Components Length (bit)		
Sch.	Ciphertext (Or)	Token	Secret Key
[38]	8480	16320	10600
[19]	17984	$\perp$	26818
Ours	1984	408000	408000

In summary, we can see from Table IV and Table V that our system enjoys better computational efficiency in each metric except token generation, and meanwhile, we need a larger space to store our keyword search token compared to others. We state that how to reduce the size of keyword token and its corresponding computational cost is an interesting open problem, which is regarded as one of our future works.

VI. CONCLUSIONS

We defined a new notion searchable attribute-based proxy re-encryption with keyword update, and proposed a concrete construction satisfying the notion. We also proved the new scheme CCA secure in the ROM. The scheme is the first of its type to integrate searchable attribute-based encryption with attribute-based proxy re-encryption, which is applicable to many real-world applications.

Although the new system enjoys its valuable advantages, it motivates some interesting open problems, e.g., how to reduce the size of search token, how to allow a secret key holder to generate search token individually, and how to provide more expressive keyword search.

VII. ACKNOWLEDGEMENTS

K. Liang is supported by Privacy-aware retrieval and modelling of genomic data (No. 13283250), Academy of Finland, Finland. W. Susilo is partially supported by the Australian Research Council Discovery Project ARC DP130101383.

REFERENCES

[1] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM TISSEC*, 9(1):1–30, 2006.

[2] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, volume 4622 of *Lecture Notes in Computer Science*, pages 535–552. Springer, 2007.

[3] S. Benabbas, R. Gennaro, and Y. Vahlis. Verifiable delegation of computation over large datasets. In P. Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 111–131. Springer, 2011.

[4] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *EUROCRYPT '98*, pages 127–144. Springer, 1998.

[5] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT '05*, volume 3494 of *LNCS*, pages 440–456. Springer, 2005.

[6] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer, 2004.

[7] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In S. P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554. Springer, 2007.

[8] R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re-encryption. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*, pages 185–194. ACM, 2007.

[9] N. Chandran, M. Chase, and V. Vaikuntanathan. Functional re-encryption and collusion-resistant obfuscation. In R. Cramer, editor, *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 404–421. Springer, 2012.

[10] M. Chase and S. Kamara. Structured encryption and controlled disclosure. In M. Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 577–594. Springer, 2010.

[11] L. Cheung and C. C. Newport. Provably secure ciphertext policy ABE. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM Conference on Computer and Communications Security*, pages 456–465. ACM, 2007.

[12] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.*, 33(1):167–226, January 2004.

[13] L. Ducas. Anonymity from asymmetry: new constructions for anonymous HIBE. In *CT-RSA'10*, volume 5985 of *LNCS*, pages 148–164. Springer, 2010.

[14] P. Golle, J. Staddon, and B. R. Waters. Secure conjunctive keyword search over encrypted data. In M. Jakobsson, M. Yung, and J. Zhou, editors, *Applied Cryptography and Network Security, Second International Conference, ACNS 2004, Yellow Mountain, China, June 8-11, 2004, Proceedings*, volume 3089 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2004.

[15] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In A. Juels, R. N. Wright, and S. D. C. di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 89–98. ACM, 2006.

[16] S. Hohenberger and B. Waters. Attribute-based encryption with fast decryption. In K. Kurosawa and G. Hanaoka, editors, *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, volume 7778 of *Lecture Notes in Computer Science*, pages 162–179. Springer, 2013.

[17] Y. Hwang and P. Lee. Public key encryption with conjunctive keyword search and its extension to a multi-user system. In T. Takagi, T. Okamoto, E. Okamoto, and T. Okamoto, editors, *Pairing-Based Cryptography Pairing 2007*, volume 4575 of *Lecture Notes in Computer Science*, pages 2–22. Springer Berlin Heidelberg, 2007.

[18] A. B. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO*, volume 7417 of *LNCS*, pages 180–198. Springer, 2012.

[19] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, T. V. X. Phuong, and Q. Xie. A dfa-based functional proxy re-encryption scheme for secure public cloud data sharing. *IEEE Transactions on Information Forensics and Security*, 9(10):1667–1680, 2014.

[20] K. Liang, M. H. Au, W. Susilo, D. S. Wong, G. Yang, and Y. Yu. An adaptively cca-secure ciphertext-policy attribute-based proxy re-encryption for cloud data sharing. In X. Huang and J. Zhou, editors, *Information Security Practice and Experience - 10th International Conference, ISPEC 2014, Fuzhou, China, May 5-8, 2014. Proceedings*, volume 8434 of *Lecture Notes in Computer Science*, pages 448–461. Springer, 2014.

[21] K. Liang, C. Chu, X. Tan, D. S. Wong, C. Tang, and J. Zhou. Chosen-ciphertext secure multi-hop identity-based conditional proxy re-encryption with constant-size ciphertexts. *Theor. Comput. Sci.*, 539:87–105, 2014.

[22] K. Liang, L. Fang, W. Susilo, and D. S. Wong. A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security. In *INCoS*, pages 552–559. IEEE, 2013.

[23] K. Liang, Q. Huang, R. Schlegel, D. S. Wong, and C. Tang. A conditional proxy broadcast re-encryption scheme supporting timed-release. In R. H. Deng and T. Feng, editors, *ISPEC*, volume 7863 of *Lecture Notes in Computer Science*, pages 132–146. Springer, 2013.

[24] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo. An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing. In M. Kutylowski and J. Vaidya, editors, *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part I*, volume 8712 of *Lecture Notes in Computer Science*, pages 257–272. Springer, 2014.

[25] K. Liang, Z. Liu, X. Tan, D. S. Wong, and C. Tang. A CCA-secure identity-based conditional proxy re-encryption without random oracles. In T. Kwon, M.-K. Lee, and D. Kwon, editors, *ICISC*, volume 7839 of *LNCS*, pages 231–246. Springer, 2012.

[26] X. Liang, Z. Cao, H. Lin, and J. Shao. Attribute based proxy re-encryption with delegating capabilities. In W. Li, W. Susilo, U. K. Tupakula, R. Safavi-Naini, and V. Varadharajan, editors, *Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009, Sydney, Australia, March 10-12, 2009*, pages 276–286. ACM, 2009.

[27] B. Libert and D. Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. In *PKC'08*, volume 4939 of *PKC'08*, pages 360–379. Springer, 2008.

[28] J. Library. http://gas.dia.unisa.it/projects/jpbc/benchmark.html#.U5FXwZS1bLd/, 2013. Online; accessed 01-March-2015.

[29] R. Lu, X. Lin, J. Shao, and K. Liang. Rcca-secure multi-use bidirectional proxy re-encryption with master secret security. In S. S. M. Chow, J. K. Liu, L. C. K. Hui, and S. Yiu, editors, *Provable Security - 8th International Conference, ProvSec 2014, Hong Kong, China, October 9-10, 2014. Proceedings*, volume 8782 of *Lecture Notes in Computer Science*, pages 194–205. Springer, 2014.

[30] S. Luo, J. bin Hu, and Z. Chen. Ciphertext policy attribute-based proxy re-encryption. In M. Soriano, S. Qing, and J. López, editors, *ICICS*, volume 6476 of *LNCS*, pages 401–415. Springer, 2010.

[31] M. Mambo and E. Okamoto. Proxy cryptosystems: Delegation of the power to decrypt ciphertexts. *IEICE Transactions*, E80-A(1):54–63, 1997.

[32] T. Mizuno and H. Doi. Hybrid proxy re-encryption scheme for attribute-based encryption. In F. Bao, M. Yung, D. Lin, and J. Jing, editors, *Information Security and Cryptology*, volume 6151 of *LNCS*, pages 288–302. Springer Berlin Heidelberg, 2011.

[33] A. Sahai and B. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *Advances in Cryptology EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer Berlin Heidelberg, 2005.

[34] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *2000 IEEE Symposium on Security and Privacy, Berkeley, California, USA, May 14-17, 2000*, pages 44–55. IEEE Computer Society, 2000.

[35] C. Wang, N. Cao, K. Ren, and W. Lou. Enabling secure and efficient ranked keyword search over outsourced cloud data. *IEEE Trans. Parallel Distrib. Syst.*, 23(8):1467–1479, 2012.

[36] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *Public Key Cryptography*, volume 6571 of *LNCS*, pages 53–70. Springer, 2011.

[37] Y. Zhang, X. Chen, J. Li, D. S. Wong, and H. Li. Anonymous attribute-based encryption supporting efficient decryption test. In K. Chen, Q. Xie, W. Qiu, N. Li, and W. Tzeng, editors, *8th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '13, Hangzhou, China - May 08 - 10, 2013*, pages 511–516. ACM, 2013.

[38] Q. Zheng, S. Xu, and G. Ateniese. VABKS: verifiable attribute-based keyword search over outsourced encrypted data. In *2014 IEEE Conference on Computer Communikations, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*, pages 522–530. IEEE, 2014.

Kaitai Liang received the Ph.D. degree in the Department of Computer Science, City University of Hong Kong (2014). He is currently a post-doctoral researcher at Department of Computer Science, Aalto university in Finland. His research interest is applied cryptography; in particular, cryptographic protocols, encryption/signature, and RFID. He is also interested in cybersecurity, such as network security, big data security, privacy enhanced technology, and security in cloud computing.

Willy Susilo received the Ph.D. degree in computer science from the University of Wollongong, Australia. He is a Professor and the Head of School of Computing and Information Technology at the University of Wollongong. He is also the Director of Centre for Computer and Information Security Research, University of Wollongong. He has been awarded the prestigious ARC Future Fellow by the Australian Research Council. His main research interests include cloud security, cryptography and information security. He has served as a program committee member in major international conferences.