Securely Outsourcing Attribute-Based Encryption with Checkability

Jin Li, Xinyi Huang, Jingwei Li, Xiaofeng Chen, and Yang Xiang, *Senior Member, IEEE*

Abstract—Attribute-Based Encryption (ABE) is a promising cryptographic primitive which significantly enhances the versatility of access control mechanisms. Due to the high expressiveness of ABE policies, the computational complexities of ABE key-issuing and decryption are getting prohibitively high. Despite that the existing Outsourced ABE solutions are able to offload some intensive computing tasks to a third party, the verifiability of results returned from the third party has yet to be addressed. Aiming at tackling the challenge above, we propose a new Secure Outsourced ABE system, which supports both secure outsourced key-issuing and decryption. Our new method offloads all access policy and attribute related operations in the key-issuing process or decryption to a Key Generation Service Provider (KGSP) and a Decryption Service Provider (DSP), respectively, leaving only a constant number of simple operations for the attribute authority and eligible users to perform locally. In addition, for the first time, we propose an outsourced ABE construction which provides checkability of the outsourced computation results in an efficient way. Extensive security and performance analysis show that the proposed schemes are proven secure and practical.

Index Terms—Attribute-based encryption, access control, outsourcing computation, key issuing, checkability

1 Introduction

As a novel public key primitive, attribute-based encryption (ABE) [1] has attracted much attention in the research community. For the first time, ABE enables efficient public key-based fine-grained sharing. In ABE system, users' private keys and ciphertexts are labeled with sets of descriptive attributes and access policies respectively, and a particular key can decrypt a particular ciphertext only if associated attributes and policy are matched. Until now, there are two kinds of ABE having been proposed: key-policy attribute-based encryption (KP-ABE) and ciphertext-policy attribute-based encryption (CP-ABE). In KP-ABE, the access policy is assigned in private key, whereas, in CP-ABE, it is specified in ciphertext.

Recently, as the development of cloud computing [2], users' concerns about data security are the main obstacles that impedes cloud computing from wide adoption. These concerns are originated from the fact that sensitive data resides in public cloud, which is maintained and operated by untrusted cloud service provider (CSP). ABE provides a secure way that allows data owner to share outsourced data on untrusted storage server instead of trusted server with specified group of users. This advantage makes the methodology appealing in cloud storage that requires secure access control for a large number of users belonging to different organizations.

Nevertheless, one of the main efficiency drawbacks of ABE is that the computational cost during decryption phase grows with the complexity of the access formula. Thus, before widely deployed, there is an increasing need to improve the efficiency of ABE. To address this problem, outsourced ABE, which provides a way to outsource intensive computing task during decryption to CSP without revealing data or private keys, was introduced [3], [4]. It has a wide range of applications. For example, in the mobile cloud computing consisting of mobile devices or sensors as information collection nodes, user terminal (e.g., mobile device) has limited computation ability to independently complete basic encryption or decryption to protect sensitive data residing in public cloud. Outsourced ABE allows user to perform heavy decryption through ''borrowing'' the computation resources from CSP. Therefore, in this paradigm, the computation/storage intensive tasks can be performed even by resource-constrained users.

Beyond the heavy decryption outsourced, we observe that the attribute authority has to deal with a lot of heavy computation in a scalable system. More precisely, the attribute authority has to issue private keys to all users, but yet generation of private key typically requires large modular exponentiation computation, which grows linearly with the complexity of the predicate formula. When a large number of users call for their private keys, it may overload the attribute authority. Moreover, key management mechanism, key revocation in particular, is necessary in a secure and scalable ABE system. In most of existing ABE schemes, the revocation of any single private key requires key-update at attribute authority for the remaining unrevoked keys which share common attributes with the one to be re-

- *J. Li is with the School of Computer Science, Guangzhou University, China and State Key Laboratory of Integrated Service Networks (ISN), Xidian University, China. E-mail: lijin@gzhu.edu.cn.*
- *X. Huang is with the Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, China. E-mail: xyhuang81@gmail.com.*
- *J. Li is with the College of Information Technical Science, Nankai University, China. E-mail: lijw@mail.nankai.edu.cn.*
- *X. Chen is with the State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an, China. E-mail: xfchen@xidian.edu.cn.*
- *Y. Xiang is with the School of Information Technology, Deakin University, Australia. E-mail: yang@deakin.edu.au.*

Manuscript received 24 July 2013; revised 30 Sept. 2013; accepted 7 Oct. 2013. Date of publication 20 Oct. 2013; date of current version 16 July 2014. Recommended for acceptance by J. Chen.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2013.271

voked. All of these heavy tasks centralized at authority side would make it an efficiency bottleneck in the access control system.

1.1 Contribution

Aiming at eliminating the most overhead computation at both the attribute authority and the user sides, we propose an outsourced ABE scheme not only supporting outsourced decryption but also enabling delegating key generation. In this construction, we introduce a trivial policy controlled by a default attribute and use an AND gate connecting the trivial policy and user's policy. During key-issuing, attribute authority can outsource computation through delegating the task of generating partial private key for user's policy to a key generation service provider (KGSP) to reduce local overhead. Moreover, the outsourced decryption is realized by utilizing the idea of key blinding. More precisely, user can send the blinded private key to a decryption service provider (DSP) to perform partial decryption and do the complete decryption at local. Following our technique, constant efficiency is achieved at both attribute authority and user sides.

In addition, we observe that when experiencing commercial cloud computing services, the CSPs may be selfish in order to save its computation or bandwidth, which may cause results returned incorrectly. In order to deal with this problem, we consider to realize checkability on results returned from both KGSP and DSP, and provide a security and functionality enhanced construction, which is provable secure under the recent formulized refereed delegation of computation (RDoC) model. Out technique is to make a secret sharing on the outsourcing key for KGSP and let k parallel KGSPs utilize their individual share to generate partial private keys. After that an additional key combination phase is performed at authority side to avoid malicious collaboration between at most $k-1$ KGSPs and users. Moreover, we use the idea of ''ringer'' [5] and appending redundancy to fight against the dishonest actions of KGSPs and DSP. As far as we know, this is the first time considering the checkability of outsourced ABE.

1.2 Related Work

The notion of ABE, which was introduced as fuzzy identity-based encryption in [1], was firstly dealt with by Goyal *et al.* [6]. Two different and complementary notions of ABE were defined in [6]: KP-ABE and CP-ABE. A construction of KP-ABE was provided in the same paper [6], while the first CP-APE construction supporting tree-based structure in generic group model is presented by Bethencourt *et al.* [7]. Accordingly, several constructions supporting for any kinds of access structures were provided [8], [9] for practical applications [10], [11]. Concerning revocation of ABE, a delegatable revocation is proposed in [12] to achieve scalable and fine-grained access control.

To reduce the load at local, it always desires to deliver expensive computational tasks outside. Actually, the problem that how to securely outsource different kinds of expensive computations has drew considerable attention from theoretical computer science community. Atallah *et al.* [13] presented a framework for secure outsourcing of scientific computations such as matrix multiplication and quadrature. Nevertheless, the solution used the disguise technique and thus leaded to leakage of private information. Atallah and Li [14] investigated the problem of computing the edit distance between two sequences and presented an efficient protocol to securely outsource sequence comparison with two servers. Furthermore, Benjamin and Atallah [15] addressed the problem of secure outsourcing for widely applicable linear algebraic computations. Nevertheless, the proposed protocols required the expensive operations of homomorphic encryption. Atallah and Frikken [16] further studied this problem and gave improved protocols based on the so-called weak secret hiding assumption. Recently, Wang *et al.* [17] presented efficient mechanisms for secure outsourcing of linear programming computation.

We note that though several schemes have been introduced to securely outsource kinds of expensive computations, they are not suitable for reliving ABE computational overhead of exponentiation at user side. To achieve this goal, the traditional approach is to utilize server-aided techniques [18], [19], [20]. However, previous work are oriented to accelerating the speed of exponentiation using untrusted servers. Directly utilizing these techniques in ABE will not work efficiently. Another approach might be to leverage recent general outsourcing technique or delegating computation [21], [22], [23], [24], [25] based on fully homomorphic encryption or interactive proof system. However, Gentry [25] has shown that even for weak security parameters on ''bootstrapping'' operation of the homomorphic encryption, it would take at least 30 seconds on a high performance machine. Therefore, even if the privacy of the input and output can be preserved by utilizing these general techniques, the computational overhead is still huge and impractical.

Another several related work similar to us are [4], [26], [3], [27]. In [3], a novel paradigm for outsourcing the decryption of ABE is provided while in [4], [26] the authors presented the ABE schemes which allow to securely outsource both decryption and encryption to third party service providers. Compared with our work, the two lack of the consideration on the eliminating the overhead computation at attribute authority. Additionally, we consider a security and functionality enhanced construction enabling checkability on returned results from CSPs. Recently Lai *et al.* [28] proposed a concrete construction for ABE with verifiable decryption, which achieves both security and verifiability without random oracles. Their work appends a redundancy with ciphertext and uses this redundancy for correctness checking. We emphasize that compared with our scheme their construction does not consider to offload the overhead computation at authority by outsourcing key-issuing.

1.3 Organization

This paper is organized as follows. In Section 2 we describe some preliminaries. In Section 3, we present the system model and security definition. The proposed construction and its security analysis are presented in Section 4. In Section 5, we consider a both security and functionality enhanced construction under RDoC model. The performance

TABLE 1
Notations Used in This Paper

Acronym	Description
AA	attribute authority
KGSP	key generation service provider
DSP	decryption service provider
SSP	storage service provider

analysis for the schemes are given in Section 6. Finally, we draw conclusion in Section 7.

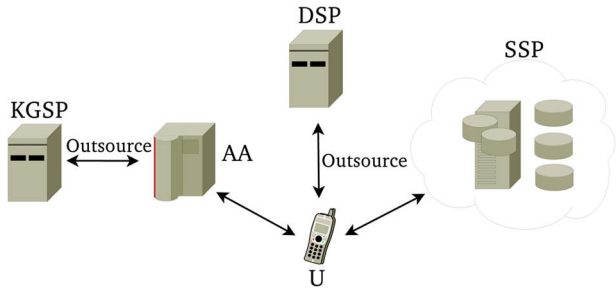


Fig. 1. System model for outsourced ABE scheme.

2 Preliminary

In this section, we define the notations used in this paper and review some cryptographic background.

2.1 Notations

The notations used in this paper are listed in Table 1.

2.2 Cryptographic Background

In this paper, we use the bilinear pairings on elliptic curves. We now give a brief review on the property of pairing and the candidate hard problem that will be used.

Definition 1 (Bilinear Map). *Let $\mathbf{G}, \mathbf{G}_T$ be cyclic groups of prime order q, writing the group action multiplicatively. g is a generator of $\mathbf{G}$. Let $e : \mathbf{G} \times \mathbf{G} \rightarrow \mathbf{G}_T$ be a map with the following properties:*

- *Bilinearity*: $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ *for all* $g_1, g_2 \in \mathbf{G}$, *and* $a, b \in_R \mathbf{Z}_q$;
- *Non-degeneracy*: *There exists* $g_1, g_2 \in \mathbf{G}$ *such that* $e(g_1, g_2) \neq 1$, *in other words, the map does not send all pairs in* $\mathbf{G} \times \mathbf{G}$ *to the identity in* $\mathbf{G}_T$;
- *Computability*: *There is an efficient algorithm to compute* $e(g_1, g_2)$ *for all* $g_1, g_2 \in \mathbf{G}$.

Definition 2 (DBDH Problem). *The decision Bilinear Diffie-Hellman (DBDH) problem is that, given g, g^x, g^y, $g^z \in \mathbf{G}$ for unknown values $x, y, z \in_R \mathbf{Z}_q$, and $T \in_R \mathbf{G}_T$, to decide if $T = e(g, g)^{xyz}$.*

We say that the (t, ϵ)-DBDH assumption holds in $\mathbf{G}$ if no t-time algorithm has probability at least $\frac{1}{2} + \epsilon$ in solving the DBDH problem for non-negligible ϵ.

3 System Model and Security Definition

3.1 System Model

We present the system model for outsourced ABE scheme in Fig. 1. Compared with the model for typical ABE, a KGSP and a DSP are additionally involved.

- KGSP is to perform aided key-issuing computation to relieve AA load in a scale system when a large number of users make requests on private key generation and key-update.
- DSP is to complete delegated expensive operations to overcome the disadvantage that the decryption phase in typical ABE requires a large number of overload operations at U.

Following the custom in [3], we denote $(I_{\text{enc}}, I_{\text{key}})$ as the input to encryption and key generation. In CP-ABE scheme, $(I_{\text{enc}}, I_{\text{key}}) = (\mathbf{A}, \omega)$ while that is $(\omega, \mathbf{A})$ in KP-ABE, where ω and $\mathbf{A}$ are attribute set and access structure, respectively. Then, based on the proposed system model, we provide algorithm definitions as follows.

- $\mathsf{Setup}(\lambda)$: The setup algorithm takes as input—a security parameter λ. It outputs a public key PK and a master key MK.
- $\mathsf{KeyGen}_{\text{init}}(I_{\text{key}}, MK)$: For each user's private key request, the initialization algorithm for delegated key generation takes as input—an access policy (or attribute set) I_{key} and the master key MK. It outputs the key pair $(OK_{\text{KGSP}}, OK_{\text{AA}})$.
- $\mathsf{KeyGen}_{\text{out}}(I_{\text{key}}, OK_{\text{KGSP}})$: The delegated key generation algorithm takes as input—the access structure (or attribute set) I_{key} and the key OK_{KGSP} for KGSP. It outputs a partial transformation key TK_{KGSP}.
- $\mathsf{KeyGen}_{\text{in}}(I_{\text{key}}, OK_{\text{AA}})$: The inside key generation algorithm takes as input—the access structure (or attribute set) I_{key} and the key OK_{AA} for attribute authority. It outputs another partial transformation key TK_{AA}.
- $\mathsf{KeyBlind}(TK)$: The transformation key blinding algorithm takes as input—the transformation key $TK = (TK_{\text{KGSP}}, TK_{\text{AA}})$. It outputs a private key SK and a blinded transformation key $\widetilde{TK}$.
- $\mathsf{Encrypt}(M, I_{\text{enc}})$: The encryption algorithm takes as input—a message M and an attribute set (or access structure) I_{enc} to be encrypted with. It outputs the ciphertext CT.
- $\mathsf{Decrypt}_{\text{out}}(CT, \widetilde{TK})$: The delegated decryption algorithm takes as input—a ciphertext CT which was assumed to be encrypted under the attribute set (or access structure) I_{enc} and the blinded transformation key $\widetilde{TK}$ for access structure (or attribute set) I_{key}. It outputs the partially decrypted ciphertext CT_{part} if $\gamma(I_{\text{key}}, I_{\text{enc}}) = 1$, otherwise outputs $\perp$, where $\gamma(\cdot, \cdot)$ is a predicate predefined.
- $\mathsf{Decrypt}(CT_{\text{part}}, SK)$: The decryption algorithm takes as input—the partially decrypted ciphertext CT_{part} and the private key SK. It outputs the original message M.

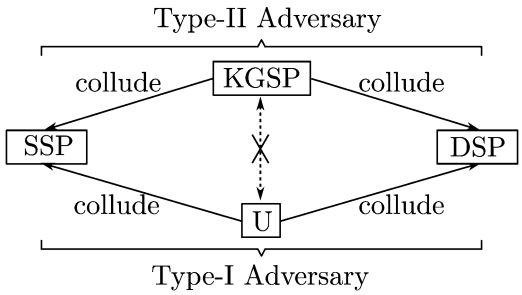


Fig. 2. Adversary model for outsourced ABE scheme.

3.2 Security Definition

In this work, we assume that all the entities except AA are ''honest-but-curious''. More precisely, they will follow our proposed protocol but try to find out as much private information as possible based on their possessions. The adversary model described in Fig. 2 is considered. More precisely, since KGSP and U respectively owns the knowledge of OK_{KGSP} for KGSP and user's private key, they are considered as active attackers which are allowed to collude with DSP and SSP to launch harmful attack separately. Following this consideration, two types of adversaries are categorized.

- Type-I adversary defined as a group of curious users colluding with SSP and DSP, is able to potentially access private keys for all the corrupted users, all the ciphertext stored at SSP, all the blinded transformation keys stored at DSP, etc, and aims to decrypt ciphertext intended for users not in the group.
- Type-II adversary defined as KGSP colluding with SSP and DSP, is able to potentially access all the keys for KGSP, all the ciphertexts stored at SSP, all the blinded transformation keys stored at DSP, etc, and aims to decrypt any ciphertext.

Having this intuition, we follow the RCCA (replayable chosen ciphertext attack) security in [29], [3] to define RCCA security. For saving space, we just show the definition of RCCA security here, and the detailed game can be refered to Appendix A, which is available in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.271.

Definition 3 (RCCA Security). *An outsourced CP-ABE or KP-ABE scheme with delegated key generation and decryption is secure against replayable chosen-ciphertext attack if all polynomial time adversaries have at most a negligible advantage in the RCCA security game for both type-I and type-II adversaries.*

4 Proposed Construction

4.1 Access Structure

Definition 4 (Access Structure). *Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbf{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbf{A}$ and $B \subseteq C$ then $C \in \mathbf{A}$. An access structure (or monotone access structure) is a collection (or monotone collection) $\mathbf{A}$ of non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$. The sets in $\mathbf{A}$ are called authorized sets.*

Furthermore, we could define the predicate $\gamma(\cdot, \cdot)$ as follows:

$$\gamma(\omega, \mathbf{A}) = \begin{cases} 1 & \text{if } \omega \in \mathbf{A} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

In this paper, the role of the party is taken by the attributes. Thus, the access structure $\mathbf{A}$ will contain the authorized sets of attributes. Specifically, our construction supports for access structure described as $\mathbf{A} = \{\omega \subseteq \mathcal{U} : |\omega \cap \omega^*| \geq d\}$ where $\mathcal{U}$ is the attribute universe, ω and ω^* are attribute sets and d is a predefined threshold value.

For simplicity, we will take user's attribute set to input to key generation instead of his access structure which is different from our definition in Section 3.1. We note that such substitution is trivial since user is easy to compute his access structure with the individual attribute set. Furthermore, we deliver the decision for access control to $\gamma(\cdot, \cdot)$ and redefine such predicate as follows:

$$\gamma_d(\omega, \omega^*) = \begin{cases} 1 & \text{if } |\omega \cap \omega^*| \geq d \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

4.2 Intuition for Proposed Construction

The challenge for constructing outsourced ABE scheme is the realization of delegated key generation and decryption.

- To outsource private key generation, we utilize a hybrid key policy $\mathsf{Policy} = \mathsf{Policy}_{\mathrm{KGSP}} \wedge \mathsf{Policy}_{\mathrm{AA}}$ in proposed construction, where $\wedge$ is an AND gate connecting two sub-policies $\mathsf{Policy}_{\mathrm{KGSP}}$ and $\mathsf{Policy}_{\mathrm{AA}}$. $\mathsf{Policy}_{\mathrm{KGSP}}$ is for the request attribute set which will be performed at KGSP while $\mathsf{Policy}_{\mathrm{AA}}$ is a trivial policy controlled by AA. The reason that we say it is trivial is that a single default attribute θ is appended with each request attribute set, which has no effect on the global access control policy. Using this trick, we are allowed to randomly generate an outsourcing key (which is OK_{KGSP} in our construction) to delegate partial key generation operation to KGSP without master or private key leakage.
- To outsource decryption, we make use of the idea in [4] by choosing a random ''blinding factor'' (which is t in our construction) to produce blinded transformation key which is able to be sent to DSP to perform decryption partially instead of private key itself. This skill allows us to delegate partial decryption operation to DSP without private key or original message leakage.

4.3 Construction

Before providing our construction, we define the Lagrange coefficient $\Delta_{i,S}$ for $i \in \mathbf{Z}_q$ and a set S of elements in $\mathbf{Z}_q$: $\Delta_{i,S} = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$. Our scheme is based on ABE in [1] which shares the same access formula. The message space for our construction is $\mathbf{G}_T$. Actually, using the hybrid encryption technique, we can easily extend it to support for message space consisting of $\{0,1\}^*$. The construction in detail is shown as follows.

- $\mathsf{Setup}(\lambda)$: First, define the attributes in universe $\mathcal{U}$ as elements in $\mathbf{Z}_q$. For simplicity, let $n = |\mathcal{U}|$ and we can

take the first n elements in $\mathbf{Z}_q$ (i.e. $1, 2, \ldots, n \bmod q$) to be the universe. Next, select a generator $g \in_R \mathbf{G}$ and an integer $x \in_R \mathbf{Z}_q$, and set $g_1 = g^x$. Then, pick elements $g_2, h, h_1, \ldots, h_n \in_R \mathbf{G}$. Finally, output the public key $PK = (g, g_1, g_2, h, h_1, \ldots, h_n)$ and the master key $MK = x$.

- $\mathsf{KeyGen}_{\text{init}}(\omega, MK)$: For each user's private key request on ω, select $x_1 \in_R \mathbf{Z}_q$ and set $x_2 = x - x_1 \bmod q$. Finally output $OK_{\text{KGSP}} = x_1$ as the outsourcing key for KGSP and $OK_{\text{AA}} = x_2$ for attribute authority itself.
- $\mathsf{KeyGen}_{\text{out}}(\omega, OK_{\text{KGSP}})$: Randomly select a $d-1$ degree polynomial $q(\cdot)$ such that $q(0) = x_1$. Then, for each $i \in \omega$, choose $r_i \in_R \mathbf{Z}_q$, and compute $d_{i0} = g_2^{q(i)} \cdot (g_1 h_i)^{r_i}$ and $d_{i1} = g^{r_i}$. Finally, output $TK_{\text{KGSP}} = (\{d_{i0}, d_{i1}\}_{i \in \omega})$.
- $\mathsf{KeyGen}_{\text{in}}(\omega, OK_{\text{AA}})$: Select $r_\theta \in_R \mathbf{Z}_q$ and compute $d_{\theta 0} = g_2^{x_2} \cdot (g_1 h)^{r_\theta}$ and $d_{\theta 1} = g^{r_\theta}$. Finally, output $TK_{\text{AA}} = (d_{\theta 0}, d_{\theta 1})$.
- $\mathsf{KeyBlind}(TK = (TK_{\text{KGSP}}, TK_{\text{AA}}))$: Select $t \in_R \mathbf{Z}_q$, and compute $\widetilde{TK} = (\{d_{i0}^t, d_{i1}^t\}_{i \in \omega \cup \{\theta\}})$. Finally, output $SK = (t, TK)$ and $\widetilde{TK}$.
- $\mathsf{Encrypt}(M, \omega')$: Firstly, select a random number $s \in_R \mathbf{Z}_q$. Then, compute $C_0 = M \cdot e(g_1, g_2)^s$, $C_1 = g^s$, $E_\theta = (g_1 h)^s$ and $E_i = (g_1 h_i)^s$ for $i \in \omega'$. Finally, publish the ciphertext as $CT = (\omega' \cup \{\theta\}, C_0, C_1, \{E_i\}_{i \in \omega' \cup \{\theta\}})$.
- $\mathsf{Decrypt}_{\text{out}}(CT, \widetilde{TK})$: Suppose that a ciphertext CT is encrypted under an attribute set ω' and we have a blinded transformation key $\widetilde{TK}$ for attribute set ω, which satisfies the restriction that $\gamma_d(\omega, \omega') = 1$. Then, outsourced decryption proceeds as follows. Firstly, an arbitrary d-element subset set $S \subseteq \omega \cap \omega'$ is selected. Then, the partially decrypted ciphertext is computed as follows:

$$\begin{aligned} CT_{\text{part}} &= \frac{e(C_1, d_{\theta 0}^t) \prod_{i \in S} e(C_1, d_{i0}^t)^{\Delta_{i,S}(0)}}{e(d_{\theta 1}^t, E_\theta) \prod_{i \in S} e(d_{i1}^t, E_i)^{\Delta_{i,S}(0)}} \\ &= e(g, g_2)^{stx_2} e(g, g_2)^{st \sum_{i \in S} q(i) \Delta_{i,S}(0)} \\ &= e(g, g_2)^{stx_2} e(g, g_2)^{stx_1} \\ &= e(g_1, g_2)^{st}. \end{aligned} \tag{3}$$

- $\mathsf{Decrypt}(CT_{\text{part}}, SK)$: Completely decrypt the ciphertext as follows:

$$\begin{aligned} \frac{C_0}{(CT_{\text{part}})^{\frac{1}{t}}} &= \frac{M \cdot e(g_1, g_2)^s}{\left[e(g_1, g_2)^{st}\right]^{\frac{1}{t}}} \\ &= \frac{M \cdot e(g_1, g_2)^s}{e(g_1, g_2)^s} = M. \end{aligned} \tag{4}$$

4.4 Security Analysis

Theorem 1. *The outsourced ABE scheme is indistinguishable secure against chosen-plaintext attack in selective model under DBDH assumption.*

Proof. Please refer to this proof in Appendix B available online. □

4.5 Practical Consideration

We can consider to utilize our construction in hybrid clouds. More precisely, KGSP is maintained as a private cloud with high trust to deal with sensitive information, but leaving SSP and DSP as public cloud to provide public storage and computation service respectively. Actually, this type of hybrid setting has become more and more attractive as many organizations are moving to the public cloud due to its benefit of highly available and scalable resources but still want to store and process the critical data in the private cloud.

We provide the working process of proposed construction for outsourced key generation and decryption Fig. 3.

In the outsourced key generation, AA and KGSP are allowed to perform computation to produce partial transformation key for customized and default attributes. Specifically, after producing the key pair $(OK_{\text{KGSP}}, OK_{\text{AA}})$ with $\mathsf{KeyGen}_{\text{init}}(\omega, MK)$, two individual phase can be executed simultaneously. 1) At KGSP, OK_{KGSP} is involved to generate partial transformation key for customized attribute set ω. 2) At AA, OK_{AA} is involved to generate the other partial transformation key for default attribute θ. The parallel computation is benefit for improving efficiency in key generation for ABE system.

In the outsourced decryption, user firstly fetches ciphertext from SSP and computes the intersection subset S locally. Therefore, only a partial ciphertext, blinded transformation key and intersection subset need to be delivered to DSP to perform partial decryption. Alternatively, it allows another scenario, in which after key generation user directly sends his attribute set ω and corresponding blinded transformation key $\widetilde{TK}$ to DSP to be stored. In this case, the DSP performs a role as proxy, who can automatically retrieve ciphertexts that user is interested in and forward to him partially decrypted one. The DSP could be the user's mail server, or the same entity along with SSP in cloud environment.

5 Another Construction with Checkability

We observe that in the commercial cloud computing, for saving computation or bandwidththe, CSPs may be selfish to execute only a fraction of delegated operation and return result incorrectly. The dishonest action of CSPs may cause users obtain incorrect keys or messages. We also point out that our first construction provides provable secrecy in the sense that KGSP is maintained as a private cloud with high trust. By this, we mean that an untrusted KGSP is able to collaborate with user to fake private key to enhance his ''power''. More precisely, Suppose a user and the KGSP collude together. They are able to obtain $OK_{\text{KGSP}} = x_1$ and $\{d_{\theta 0}, d_{\theta 1}\}$ corresponding to this user. With this possession, they can generate TK'_{KGSP} for target attribute set ω' and joint it with $\{d_{\theta 0}, d_{\theta 1}\}$ to obtain the faked TK'. Using this one, ciphertext satisfied by ω' can be decrypted.

Therefore, in this section, aiming at providing checkability as well as reducing the trust on KGSP, we propose another construction under the widely used RDoC model.

5.1 Outsourced ABE in Refereed Delegation of Computation Model

RDoC model originates from the model of refereed games in [30], and is later formalized in [20], [31]. In RDoC model, the client is able to interact with multiple servers and it has

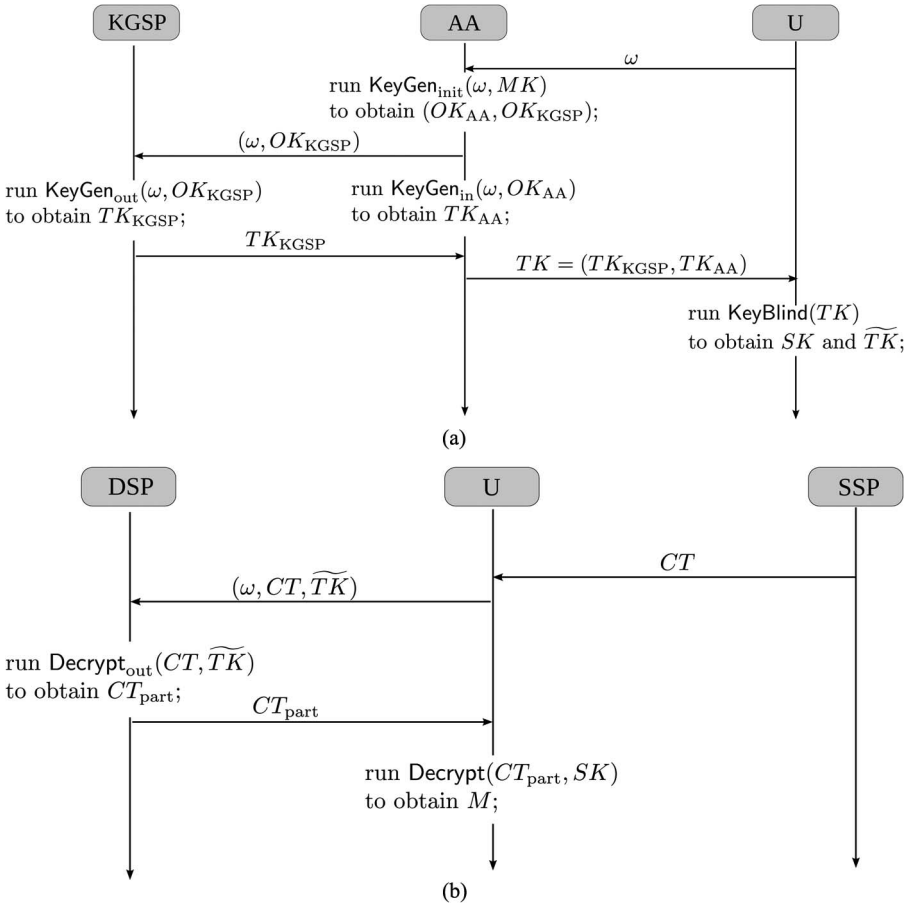


Fig. 3. Outsourced key generation and decryption.

a right output as long as there exists one server that follows the proposed protocol. One of the most advantages of RDoC over traditional model with single server is that the security risk on the single server is reduced to multiple servers involved in. As the result of both the practicality and utility, RDoC model recently has been widely utilized in the literature of outsourced computation [20], [31], [32], [33], [16].

To reduce the trust on KGSP, we will consider the outsourced ABE system in RDoC model, in which k KGSPs cooperatively work together to provide AA with the key generation service ($k-1$ are malicious at most). In this case, an additional collusion between user and at most $k-1$ malicious KGSPs is allowed. Then, the two types adversaries defined in Section 3.2 are semi-merged together and able to obtain $OK_{\mathrm{KGSP}[i]}$ for malicious KGSP$[i]$, private keys for all the corrupted users, all the blinded transformation keys stored at DSP and so on, where $i = 1, 2, \ldots, k-1$.

Having this intuition above, we can redefine the security definition of our setting for RDoC model. The challenger will maintain the table T, set D and integer j to provide the adversary with three type of oracles (if RCCA is considered $\mathcal{O}_M(\cdot)$ should be added).

- $\mathcal{O}_{OK_{\mathrm{KGSP}}}(I_{\mathrm{key}}, b)$: Challenger sets $j = j+1$ and runs key generation (including key blinding) completely for I_{key} to obtain $SK, \{OK_{\mathrm{KGSP}[i]}\}_{i=1}^{k}$ and $\widetilde{TK}$. After adding the entry $(j, I_{\mathrm{key}}, SK, \{OK_{\mathrm{KGSP}[i]}\}_{i=1}^{k}, \widetilde{TK})$ into T, return $\{OK_{\mathrm{KGSP}[i]}\}_{i=1, i\neq b}^{k}$.
- $\mathcal{O}_{\widetilde{TK}}(i)$: The challenger checks whether the entry $(i, I_{\mathrm{key}}, SK, \{OK_{\mathrm{KGSP}[j]}\}_{j=1}^{k}, \widetilde{TK})$ exists in T, if so return $\widetilde{TK}$; otherwise return $\perp$.
- $\mathcal{O}_{SK}(i)$: The challenger checks whether the entry $(i, I_{\mathrm{key}}, SK, \{OK_{\mathrm{KGSP}[j]}\}_{j=1}^{k}, \widetilde{TK})$ exists in T, if so set $D = D \cup \{I_{\mathrm{key}}\}$ and return SK, otherwise return $\perp$.

5.2 Intuition for Proposed Construction

For simplicity, we only consider and provide the second construction with two KGSPs. The key challenge for our second construction exists in two folds.

- One is how to prevent from the collusion between the user and the malicious KGSP. Our solution is to intelligently extend the hybrid policy trick in the first construction. Specifically, in addition to building an AND gate between $\mathcal{P}_{\mathrm{AA}}$ and $\mathcal{P}_{\mathrm{KGSP}}$, we introduce a (2, 2)-secret sharing on $\mathcal{P}_{\mathrm{KGSP}}$ and make each KGSP only know its own share $OK_{\mathrm{KGSP}[i]}$ for $i = 1$, 2. In this sense, even if user collude with a KGSP and obtain $\{OK_{\mathrm{KGSP}[i]}\}$ for $i = 1$ or 2, he cannot recover the secret (which is actually x_1 in our construction) to serve the devil.

- The other is how to detect the dishonest action from KGSPs and DSP beyond collusion. To fight against it,
  - we extend the idea of ''ringer'' [5] to our setting to convince that KGSPs do indeed perform all the computations that were outsourced to them. More precisely, AA generates a random value $(d-1)$-degree polynomial $q_{\mathrm{RG}}(\cdot)$ and sends it along with $q_{\mathrm{KGSP}[i]}(\cdot)$ in a random order to $\mathrm{KGSP}[i]$. Each KGSP generates partial transformation key using both $q_{\mathrm{RG}}(\cdot)$ and $q_{\mathrm{KGSP}[i]}(\cdot)$, and AA detects the dishonest action by checking all the partial transformation key computed from $q_{\mathrm{RG}}(\cdot)$ (to make sure that all the honest KGSPs will obtain the same result from OK_{RG} in a honest computation, the random values $\{r_i\}$ for $q_{\mathrm{RG}}(\cdot)$ should be selected by AA in advance).
  - In addition, we detect the dishonest action of a malicious DSP by adding redundancy. Specifically, we can require that all the users in the system agree on a redundancy 0^k (i.e., a k-length 0 bit string) and append it with original message in each encryption. Then, after performing complete decryption to obtain the plaintext, the user can detect the dishonest action of DSP by checking the redundency.

5.3 The Construction under RDoC Model

We provide our second construction with two KGSPs as follows.

- $\mathsf{Setup}(\lambda)$: It is similar to the same algorithm in our previous construction but an integer k should be agreed in public key. Specifically, the $PK = \{g, g_1, g_2, h, h_1, \ldots, h_n, k\}$ and $MK = x$ are output.
- $\mathsf{KeyGen}_{\mathrm{init}}(\omega, MK)$: For each user's private key request on ω, AA picks $x_{11}, x_{12} \in_R \mathbf{Z}_q$ and sets $OK_{\mathrm{KGSP}[1]} = x_{11}$, $OK_{\mathrm{KGSP}[2]} = x_{12}$ and $OK_{\mathrm{AA}} = x_2 = x - x_{11} - x_{12} \bmod q$. Next, select $(d-1)$-degree random polynomials $q_{\mathrm{KGSP}[1]}(\cdot)$ and $q_{\mathrm{KGSP}[2]}(\cdot)$ with the restrictions: 1) let ω' be any $(d-1)$-element subset of ω, $q_{\mathrm{KGSP}[1]}(i) = q_{\mathrm{KGSP}[2]}(i)$ for each $i \in \omega'$; 2) $q_{\mathrm{KGSP}[1]}(0) = x_{11}$; 3) $q_{\mathrm{KGSP}[2]}(0) = x_{12}$. Thirdly, to enable convincing the dishonest action of KGSPs later, select another random polynomial $q_{\mathrm{RG}}(\cdot)$. Furthermore, for each $i \in \omega$, pick $r_{\mathrm{KGSP}[1],i}, r_{\mathrm{KGSP}[2],i}, r_{\mathrm{RG},i} \in_R \mathbf{Z}_q$ with the restriction that $r_{\mathrm{KGSP}[1],j} = r_{\mathrm{KGSP}[2],j}$ where $j \in \omega'$. Finally, AA sends $(S[1]_{\mathrm{REAL}}, S_{\mathrm{RG}})$ and $(S[2]_{\mathrm{REAL}}, S_{\mathrm{RG}})$ to KGSP[1] and KGSP[2] respectively, where the pair $S[j]_{\mathrm{REAL}} = (q_{\mathrm{KGSP}[j]}(\cdot), \{r_{\mathrm{KGSP}[j],i}\}_{i\in\omega})$ and $S_{\mathrm{RG}} = (q_{\mathrm{RG}}(\cdot), \{r_{\mathrm{RG},i}\}_{i\in\omega})$ for $j = 1, 2$. We emphasize that in the both communications $S[\cdot]_{\mathrm{REAL}}$ and S_{RG} should be sent in random orders to avoid KGSPs knowing which one is really to be computed for partial transformation key.
- $\mathsf{KeyGen}_{\mathrm{out}}(S[j]_{\mathrm{REAL}}, S_{\mathrm{RG}})$: $\mathrm{KGSP}[j]$ generates partial transformation key for both $q_{\mathrm{KGSP}[j]}(\cdot)$ and $q_{\mathrm{RG}}(\cdot)$. More precisely, $\mathrm{KGSP}[j]$ computes

$$TK_{\mathrm{KGSP}[j]} = \left(\left\{d[j]_{i0}, d[j]_{i1}\right\}_{i\in\omega}\right)$$

where $d[j]_{i0} = g_2^{q_{\mathrm{KGSP}[j]}(i)}(g_1 h_i)^{r_{\mathrm{KGSP}[j],i}}$, $d[j]_{i1} = g^{r_{\mathrm{KGSP}[j],i}}$ and

$$TK_{\mathrm{RG}_j} = \left(\left\{d[\mathrm{RG}_j]_{i0}, d[\mathrm{RG}_j]_{i1}\right\}\right)$$

where $d[\mathrm{RG}_j]_{i0} = g_2^{q_{\mathrm{RG}}(i)}(g_1 h_i)^{r_{\mathrm{RG},i}}$, $d[\mathrm{RG}_j]_{i1} = g^{r_{\mathrm{RG},i}}$, and sends $(TK_{\mathrm{KGSP}[j]}, TK_{\mathrm{RG}_j})$ to AA in its receiving order.
- $\mathsf{KeyGen}_{\mathrm{in}}(\omega, OK_{\mathrm{AA}})$: Similar to the same algorithm described in our basic construction, AA selects $r_\theta \in_R \mathbf{Z}_q$ and computes $d_{\theta 0} = g_2^{x_2} \cdot (g_1 h)^{r_\theta}$ and $d_{\theta 1} = g^{r_\theta}$. Finally output $TK_{\mathrm{AA}} = (\{d_{\theta 0}, d_{\theta 1}\})$.
- $\mathsf{KeyCheck}(TK_{\mathrm{KGSP}[1]}, TK_{\mathrm{RG}_1}, TK_{\mathrm{KGSP}[2]}, TK_{\mathrm{RG}_2})$: AA checks that both KGSPs produce the correct outputs, i.e., $d[1]_{j0} = d[2]_{j0}$, $d[1]_{j1} = d[2]_{j1}$ for all $j \in \omega'$ and $d[\mathrm{RG}_1]_{i0} = d[\mathrm{RG}_2]_{i0}$, $d[\mathrm{RG}_1]_{i1} = d[\mathrm{RG}_2]_{i1}$ for all $i \in \omega$. After that, continue to combine the partial transformation key together by computing $d_{i0} = d[1]_{i0} \cdot d[2]_{i0}$ and $d_{i1} = d[1]_{i1} \cdot d[2]_{i1}$ for all $i \in \omega$. Finally output the complete transformation key $TK = (\{d_{i0}, d_{i1}\}_{i\in\omega\cup\{\theta\}})$.
- $\mathsf{KeyBlind}(TK)$: It is identical to the same algorithm in our previous construction and outputs $SK = (t, TK)$ and $\widetilde{TK}$.
- $\mathsf{Encrypt}(M, \omega')$: User firstly appends the message M to be encrypted with a redundancy 0^k to obtain $M_T = M \| 0^k$ where $\|$ is the concatenation of string. Then, the rest is identical to the same algorithm in the first construction but to encrypt M_T. Finally, output $CT = (\omega' \cup \{\theta\}, \underline{M_T} e(g_1, g_2)^s, g^s, \{g_1 h_i\}_{i\in\omega'}, g_1 h)$.
- $\mathsf{Decrypt}_{\mathrm{out}}(CT, \widetilde{TK})$: It is identical to the same algorithm in previous construction and outputs $CT_{\mathrm{part}} = e(g_1, g_2)^{st}$.
- $\mathsf{Decrypt}(CT_{\mathrm{part}}, SK)$: It is identical to the same algorithm in previous construction except that the dishonest action of DSP should be detected through checking redundancy. Specifically, by executing the decryption algorithm in previous construction, M_T is obtained. The user continues to check whether a redundancy 0^k is appended with M_T. If so (i.e., $M_T = M \| 0^k$), M is obtained through truncation; otherwise, a dishonest action of DSP is detected.

5.4 Analysis

Our second construction has almost the same efficiency with the first one. Specifically, in key-issuing, though another key combination operation is required at attribute authority side, it costs mutiplications for $|\omega|$ times, which is negligible using the modern devices.

Then, we provide the security analysis below.

Theorem 2. *The second construction is secure against chosen-plaintext attack in the sense of the security definition modified in Section 5.1 under DBDH assumption.*

Proof. Please refer to the proof in Appendix C available online. □

5.5 Checkability

Beyond outsourced key generation and decryption, the checkability on KGSP is supported in our second construction. Specifically, since KGSP[1] (or KGSP[2]) cannot distinguish the outsourced private key generation from the two

TABLE 2
Efficiency Comparison

Schemes	Key generation (AA)	Key generation (KGSP)	Decryption (U)	Decryption (DSP)
original ABE [1]	$2\|\omega\|$EXP	–	$2d$P+$2d$EXP	–
outsourced ABE in [3]	$2\|\omega\|$EXP	–	EXP	$2d$P+$2d$EXP
outsourced ABE in [4]	$2\|\omega\|$EXP	–	EXP	$2d$P+$2d$EXP
outsourced ABE in this paper	2EXP	$2\|\omega\|$EXP	EXP	$2d$P+$2d$EXP

The symbol '–' denotes that this property is not considered in the corresponding scheme.

outsourced tasks. If KGSP[1] (KGSP[2]) fails during any execution of $\mathsf{KeyGen}_{\mathrm{out}}(\cdot)$, it will be detected with probability $\frac{d-1+|\omega|}{2|\omega|}$ which is not less than $\frac{1}{2}$. In addition, through appending redundancy, the dishonest action of DSP can be easily detected in our construction.

6 Performance Analysis

In this Section, we provide the performance analysis from both theoretical calculation and empirical evaluation of our main construction in Section 4.3.

6.1 Efficiency Analysis

We compare our scheme with the original ABE [1] and the state-of-the-art [3], [4] in Table 2. We use EXP to denote a multi-based exponentiation operation in $\mathbf{G}$ and P the pairing operation. We assume one multi-based exponentiation multiplies up to 2 single-based exponentiations and takes roughly the same time as single-based exponentiations. ω and d denotes the attribute set and threshold value respectively.

To the best of our knowledge, the outsourced key generation in ABE has not been considered before and our scheme is the first construction achieving this property. Following our terminology, the number of exponentiations in the group $\mathbf{G}$ for AA is reduced to two, while in other ABE schemes [1], [3], [4], it is linear with the number of attributes in the request set (i.e., $2|\omega|$). Actually in our construction, the exponentiation computation is delivered to KGSP and requester. More precisely, after obtaining the transformation key from AA, the requester must spend $|\omega| + 1$ exponentiations on generating private key and blinded transformation key.

In decryption, a trick similar to [3], [4] is used in our scheme and the three schemes achieve the identical efficiency: all the pairing operations are delivered to DSP and the computational cost of decryption for user is constant, only one exponentiation operation. Whereas the original ABE scheme [1] requires $2d$ pairing as well as $2d$

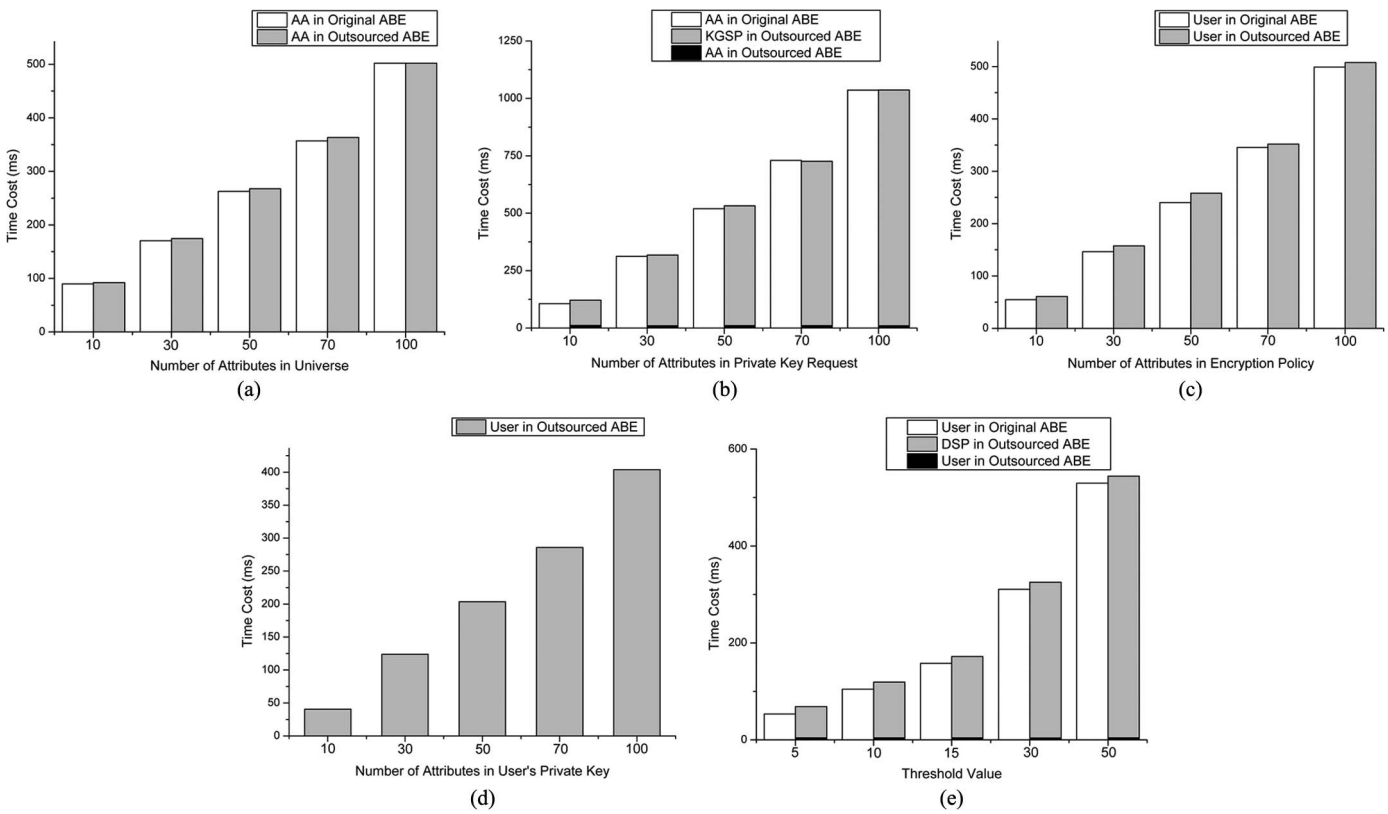


Fig. 4. Evaluation of our main construction. (a) Setup. (b) Key generation. (c) Encryption. (d) Key blinding. (e) Decryption.

exponentiation operations for a single decryption, where d is the threshold value.

Concerning on the communication complexity in our scheme, user has to send a private key request to AA and receive $2|\omega|+2$ elements in $\mathbf{G}$. Furthermore, he is able to send blinded transformation key (as well as $2|\omega|+2$ elements in $\mathbf{G}$) to DSP to perform partially decryption in future. In general, an element in $\mathbf{G}$ is set to be 160-bit long for 2^{80} security. The data transferred among the cloud service providers, AA and user is tens of KBs at most, which can be processed efficiently.

6.2 Experiment

Note that in order to precisely measure the overhead of outsourced and local computation, all the computations involving our construction are performed in an identical environment, that is on a Linux Mint 13 machine with Intel(R) Core(TM)2 Duo CPU clocked at 2.40 GHz and 2 GB of system memory.

Generally, as shown in Fig. 4, it is not surprising to see that our outsourced construction totally takes more time than the original ABE scheme. This is because the outsourcing computation cannot be realized in the manner of ''one plus one equals two'', and some additional cost should be paid for preserving privacy.

Fig. 4a illustrates the efficiency comparison between our outsourced construction and original ABE in setup phase. Compared with the original scheme, our construction requires an additional initialization of the default attribute, leading to its slowness. Similarly, our key generation time in total (i.e., including the time cost at both AA and KGSP) is relatively longer than that of the original scheme (as shown in Fig. 4b). This is because key generation for user's real attributes are delegated to KGSP, while a default attribute is controlled by AA at local. Compared with the original scheme, our outsourced construction involves the computation for an additional one attribute (i.e., the default attribute). Fortunately, owing to outsourced computation, the computation cost at AA side is reduced to constant (nearly three single-based modular exponentiations in $\mathbf{G}$). File encryption in the outsourced construction is also slower than the original scheme because the default attribute is required to be naturally embedded in encryption policy.

Regarding to decryption, our construction requires the key blinding phase which is not demanded in original scheme. As shown in Fig. 4d, the key blinding costs time on ms level. But we point out that the key blinding can be realized in amortized model. Specifically, user is able to run the key blinding process just once, and then enjoy his/her efficient local decryption. Fig. 4e demonstrates the decryption efficiency comparison for a varying threshold value. Though our outsourced scheme takes more time in total, user just needs to pool the shadow in the partial decrypted ciphertext, which involves one modular exponentiation and division in $\mathbf{G}_T$.

To sum up, our outsourced construction achieves efficiency at both AA and user sides during key-issuing and decryption without introducing significant overhead compared to the original approach (our execution time is still within ms).

7 Conclusion

We provide a new outsourced ABE scheme simultaneously supporting outsourced key-issuing and decryption. With the aid of KGSP and DSP, our scheme achieves constant efficiency at both authority and user sides. In addition, we provide a trust-reduced construction with two KGSPs which is secure under recently formulized RDoC model. Unlike the state-of-the-art outsourced ABE, checkability is supported by this construction. The security of proposed schemes have been analyzed and given in this paper. Experimental results demonstrate that our constructions are efficient and practical.

Acknowledgment

This work is supported by National Natural Science Foundation of China (Grants 61100224, 61202450, 61272455), Natural Science Foundation of Guangdong Province (No. S2013010013671), Distinguished Young Scholars Fund of Department of Education, Guangdong Province (No. Yq2013126), PhD Programs Foundation of Ministry of Education of China (Grant 20123503120001), Distinguished Young Scholars Fund of Department of Education, Fujian Province (JA13062), Fok Ying Tung Education Foundation (Grant No. 141065), Doctoral Fund of Ministry of Education of China (No. 20130203110004), ISN Research Fund (ISN 15-02, 15-03), Program for New Century Excellent Talents in University (No. NCET-13-0946), and China 111 Project (No. B08038).

References

[1] A. Sahai and B. Waters, ''Fuzzy Identity-Based Encryption,'' in *Proc. Adv. Cryptol.-EUROCRYPT*, LNCS 3494, R. Cramer, Ed., Berlin, Germany, 2005, pp. 457-473, Springer-Verlag.

[2] D. Zeng, S. Guo, and J. Hu, ''Reliable Bulk-Data Dissemination in Delay Tolerant Networks,'' *IEEE Trans. Parallel Distrib. Syst.* http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.221

[3] M. Green, S. Hohenberger, and B. Waters, ''Outsourcing the Decryption of ABE Ciphertexts,'' in *Proc. 20th USENIX Conf. SEC*, 2011, p. 34.

[4] Z. Zhou and D. Huang, ''Efficient and Secure Data Storage Operations for Mobile Cloud Computing,'' in *Cryptology ePrint Archive*, Report 2011/185, 2011.

[5] P. Golle and I. Mironov, ''Uncheatable Distributed Computations,'' in *Proc. Conf. Topics Cryptol., CT-RSA*, 2001, pp. 425-440.

[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters, ''Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,'' in *Proc. 13th ACM Conf. Comput. Commun. Security*, 2006, pp. 89-98.

[7] J. Bethencourt, A. Sahai, and B. Waters, ''Ciphertext-Policy Attribute-Based Encryption,'' in *Proc. IEEE Symp. Security Privacy*, May 2007, pp. 321-334.

[8] L. Cheung and C. Newport, ''Provably Secure Ciphertext Policy ABE,'' in *Proc. 14th ACM Conf. CCS*, 2007, pp. 456-465.

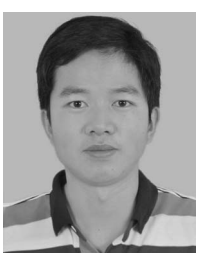
[9] T. Nishide, K. Yoneyama, and K. Ohta, ''Attribute-Based Encryption with Partially Hidden Encryptor-Specified Access Structures,'' in *Proc. Appl. Cryptogr. Netw. Security*, LNCS 5037, S. Bellovin, R. Gennaro, A. Keromytis, and M. Yung, Eds., Berlin, Germany, 2008, pp. 111-129, Springer-Verlag.

[10] F. Han, J. Qin, H. Zhao, and J. Hu, ''A General Transformation from KP-ABE to Searchable Encryption,'' *Future Gen. Comput. Syst.*, vol. 30, pp. 107-115, Jan. 2014.

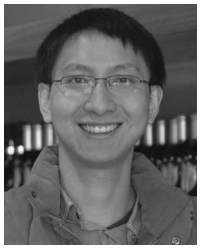
[11] H. Zhao, J. Qin, and J. Hu, ''Energy Efficient Key Management Scheme for Body Sensor Networks,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 11, pp. 2202-2210, Nov. 2013.

[12] S. Yu, C. Wang, K. Ren, and W. Lou, ''Achieving Secure, Scalable, Fine-Grained Data Access Control in Cloud Computing,'' in *Proc. IEEE 29th INFOCOM*, 2010, pp. 534-542.

[13] M.J. Atallah, K. Pantazopoulos, J.R. Rice, and E.E. Spafford, "Secure Outsourcing of Scientific Computations," in *Trends in Software Engineering*, vol. 54, M.V. Zelkowitz, Ed. Amsterdam, The Netherlands: Elsevier, 2002, pp. 215-272.
[14] M.J. Atallah and J. Li, "Secure Outsourcing of Sequence Comparisons," *Int'l J. Inf. Security*, vol. 4, no. 4, pp. 277-287, Oct. 2005.
[15] D. Benjamin and M.J. Atallah, "Private and Cheating-Free Outsourcing of Algebraic Computations," in *Proc. 6th Annu. Conf. PST*, 2008, pp. 240-245.
[16] M.J. Atallah and K.B. Frikken, "Securely Outsourcing Linear Algebra Computations," in *Proc. 5th ACM Symp. ASIACCS*, 2010, pp. 48-59.
[17] C. Wang, K. Ren, and J. Wang, "Secure and Practical Outsourcing of Linear Programming in Cloud Computing," in *Proc. IEEE INFOCOM*, 2011, pp. 820-828.
[18] K. Bicakci and N. Baykal, "Server Assisted Signatures Revisited," in *Proc. Topics Cryptol.-CT-RSA*, LNCS 2964, T. Okamoto, Ed., Berlin, Germany, 2004, pp. 1991-1992. Springer-Verlag.
[19] M. Jakobsson and S. Wetzel, "Secure Server-Aided Signature Generation," in *Proc. Public Key Cryptogr.*, 2001, pp. 383-401.
[20] S. Hohenberger and A. Lysyanskaya, "How to Securely Outsource Cryptographic Computations," in *Proc. Theory Cryptogr.*, LNCS 3378, J. Kilian, Ed., Berlin, Germany, pp. 264-282, Springer-Verlag.
[21] S. Goldwasser, Y.T. Kalai, and G.N. Rothblum, "Delegating Computation: Interactive Proofs for Muggles," in *Proc. 40th Annu. ACM STOC*, 2008, pp. 113-122.
[22] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *Proc. 41st Annu. ACM STOC*, 2009, pp. 169-178.
[23] R. Gennaro, C. Gentry, and B. Parno, "Non-Interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers," in *Proc. Adv. Cryptol.-CRYPTO*, LNCS 6223, T. Rabin, Ed., Berlin, Germany, 2010, pp. 465-482, Springer-Verlag.
[24] K.-M. Chung, Y. Kalai, F.-H. Liu, and R. Raz, "Memory Delegation," in *Proc. Adv. Cryptol.-CRYPTO*, LNCS 6841, P. Rogaway, Ed., Berlin, 2011, pp. 151-168, Springer-Verlag.
[25] C. Gentry and S. Halevi, "Implementing Gentry's Fully-Homomorphic Encryption Scheme," in *Proc. Adv. Cryptol.-EUROCRYPT*, LNCS 6632, K. Paterson, Ed., Berlin, Germany, 2011, pp. 129-148, Springer-Verlag.
[26] J. Li, C. Jia, J. Li, and X. Chen, "Outsourcing Encryption of Attribute-Based Encryption with Mapreduce," in *Proc. Int'l Conf. Inf. Commun. Security*, 2012, pp. 191-201.
[27] J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, "Fine-Grained Access Control System Based on Outsourced Attribute-Based Encryption," in *Proc. 18th ESORICS*, 2013, pp. 592-609.
[28] J. Lai, R. Deng, C. Guan, and J. Weng, "Attribute-based Encryption with Verifiable Outsourced Decryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 8, pp. 1343-1354, Aug. 2013.
[29] R. Canetti, H. Krawczyk, and J. Nielsen, "Relaxing Chosen-Ciphertext Security," in *Proc. Adv. Cryptol.-CRYPTO*, LNCS 2729, D. Boneh, Ed., Berlin/Heidelberg, 2003, pp. 565-582, Springer-Verlag.
[30] U. Feige and J. Kilian, "Making Games Short (Extended Abstract)," in *Proc. 29th Annu. ACM STOC*, 1997, pp. 506-516.
[31] R. Canetti, B. Riva, and G. Rothblum, "Two Protocols for Delegation of Computation," in *Proc. Inf. Theor. Security*, LNCS 7412, A. Smith, Ed., Berlin, Germany, 2012, pp. 37-61, Springer-Verlag.
[32] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New Algorithms for Secure Outsourcing of Modular Exponentiations," in *Proc. ESORICS*, LNCS 7459, S. Foresti, M. Yung, and F. Martinelli, Eds., Berlin, Germany, 2012, pp. 541-556, Springer-Verlag.
[33] R. Canetti, B. Riva, and G.N. Rothblum, "Practical Delegation of Computation Using Multiple Servers," in *Proc. 18th ACM Conf. CCS*, 2011, pp. 445-454.



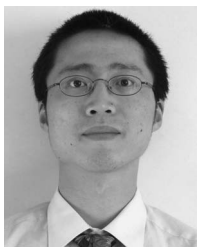
Jin Li received the BS degree in mathematics from Southwest, University, in 2002 and the PhD degree in information security from Sun Yat-sen University, in 2007. Currently, he works at Guangzhou University as a Professor. He has been selected as one of science and technology new star in Guangdong province. His research interests include applied cryptography and security in cloud computing. He has published over 50 research papers in refereed international conferences and journals and has served as the program chair or program committee member in many international conferences.



Xinyi Huang received the PhD degree from the School of Computer Science and Software Engineering, University of Wollongong, Australia, in 2009. He is currently a Professor at the Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, China. His research interests include cryptography and information security. He has published over 60 research papers in refereed international conferences and journals. His work has been cited more than 1000 times at Google Scholar. He is in the Editorial Board of the *International Journal of Information Security* (*IJIS*, Springer) and has served as the program/general chair or program committee member in over 40 international conferences.

Jingwei Li received the BS degree in mathematics from the Hebei University of Technology, China, in 2005 and is currently pursuing the PhD degree in computer technology in Nankai University. He is currently a visiting student (sponsored by The State Scholarship Fund of China) in Penn State University. His research interests include applied cryptography and cloud security.

Xiaofeng Chen received the BS and MS degrees in mathematics from the Northwest University, China and the PhD degree in cryptography from Xidian University, in 2003. Currently, he works at Xidian University as a professor. His research interests include applied cryptography and cloud computing security. He has published over 80 research papers in refereed international conferences and journals. His work has been cited more than 1000 times at Google Scholar. He has served as the program/general chair or program committee member in over 20 international conferences.



Yang Xiang received the PhD degree in computer science from Deakin University, Australia. He is currently a Full Professor at School of Information Technology, Deakin University. He is the Director of the Network Security and Computing Lab (NSCLab). His research interests include network and system security, distributed systems, and networking. In particular, he is currently leading his team developing active defense systems against large-scale distributed network attacks. He is the Chief Investigator of several projects in network and system security, funded by the Australian Research Council (ARC). He has published more than 130 research papers in many international journals and conferences, such as *IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Information Security and Forensics*, and *IEEE Journal on Selected Areas in Communications*. Two of his papers were selected as the featured articles in the April 2009 and the July 2013 issues of *IEEE Transactions on Parallel and Distributed Systems*. He has published two books, Software Similarity and Classification (Springer) and Dynamic and Advanced Data Mining for Progressing Technological Development (IGI-Global). He has served as the Program/General Chair for many international conferences such as *ICA3PP 12/11, IEEE/IFIP EUC 11, IEEE TrustCom 13/11, IEEE HPCC 10/09, IEEE ICPADS 08, NSS 11/10/09/08/07*. He has been the PC member for more than 60 international conferences in distributed systems, networking, and security. He serves as the Associate Editor of *IEEE Transactions on Computers*, *IEEE Transactions on Parallel and Distributed Systems, Security and Communication Networks* (Wiley), and the Editor of the *Journal of Network and Computer Applications*. He is the Coordinator, Asia for IEEE Computer Society Technical Committee on Distributed Processing (TCDP). He is a Senior Member of the IEEE.