Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Secure attribute-based data sharing for resource-limited users in cloud computing

Jin Li [a], *Yinghui Zhang* [b,c,d], *Xiaofeng Chen* [e], *Yang Xiang* [e,f,*]

[a] *School of Computer Science, Guangzhou University, Guangzhou, PR China*
[b] *State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, PR China*
[c] *National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, PR China*
[d] *Westone Cryptologic Research Center, Beijing 100070, PR China*
[e] *State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an 710071, PR China*
[f] *School of Software and Electrical Engineering, Swinburne University of Technology, Australia*

ARTICLE INFO

Article history:
Received 9 January 2017
Received in revised form 25 July 2017
Accepted 14 August 2017
Available online 30 August 2017

Keywords:
Cloud computing
Access control
Attribute-based encryption
Online/offline encryption
Chosen ciphertext security

ABSTRACT

Data sharing becomes an exceptionally attractive service supplied by cloud computing platforms because of its convenience and economy. As a potential technique for realizing fine-grained data sharing, attribute-based encryption (ABE) has drawn wide attentions. However, most of the existing ABE solutions suffer from the disadvantages of high computation overhead and weak data security, which has severely impeded resource-constrained mobile devices to customize the service. The problem of simultaneously achieving fine-grainedness, high-efficiency on the data owner's side, and standard data confidentiality of cloud data sharing actually still remains unresolved. This paper addresses this challenging issue by proposing a new attribute-based data sharing scheme suitable for resource-limited mobile users in cloud computing. The proposed scheme eliminates a majority of the computation task by adding system public parameters besides moving partial encryption computation offline. In addition, a public ciphertext test phase is performed before the decryption phase, which eliminates most of computation overhead due to illegitimate ciphertexts. For the sake of data security, a Chameleon hash function is used to generate an immediate ciphertext, which will be blinded by the offline ciphertexts to obtain the final online ciphertexts. The proposed scheme is proven secure against adaptively chosen-ciphertext attacks, which is widely recognized as a standard security notion. Extensive performance analysis indicates that the proposed scheme is secure and efficient.

1. Introduction

With the advent of Cloud Computing, more and more data are outsourced to cloud servers from individual users and enterprise. Usually, the cloud service can be divided into three types, that is, the public cloud, private cloud and hybrid cloud, where the public cloud is usually untrusted while the private cloud is assumed to be semi-trusted or fully trusted, and hybrid cloud is the combination of public cloud and private

[*] *Corresponding author.*
E-mail addresses: lijin@gzhu.edu.cn (J. Li), yhzhaang@163.com (Y. Zhang), xfchen@xidian.edu.cn (X. Chen), yxiang@swin.edu.au (Y. Xiang).

https://doi.org/10.1016/j.cose.2017.08.007
0167-4048/© 2017 Elsevier Ltd. All rights reserved.

cloud. Thus, when users want to outsource their sensitive data to public cloud, including their personal files, health records, emails etc., they have to implement access control on the data besides preserving privacy and data deduplication (Huang et al., 2017; Li et al., 2014, 2015, 2017). In traditional access control mechanisms, it is usually assumed that the data owner and the storage servers are in the same trusted domain, where the storage servers are fully trusted and are responsible for defining and enforcing access control policies. This assumption however no longer holds in cloud computing in that the data owner and cloud servers are very likely to be in different domains. Some users may encrypt their data and upload corresponding ciphertexts for sharing to protect their privacy. However, the encryption form of the data makes the data sharing difficult, especially for the case of fine-grained data sharing.

Attribute-based encryption (ABE) is one of useful cryptographic primitives to realize fine-grained access control, which has been widely adopted in cloud computing. In ABE, each user obtains a private key related to his attribute set or access policy. More specifically, two kinds of ABE have been defined for access control system, that is, key-policy ABE and ciphertext-policy ABE. In key-policy ABE, the policy for users are bounded in the private keys during the key issuing phase. In ciphertext-policy ABE, such policy is inserted and bounded in the ciphertext instead. Both kinds of ABE have found important application scenarios.

However, the security and efficiency challenges have arisen when typical ABE schemes are directly utilized to design access control systems. For one thing, most of the existing ABE schemes are secure against chosen-plaintext attacks (CPA) which is a notion less desirable than security against adaptive chosen-ciphertext attacks (CCA2). For another, both the encryption and decryption algorithms are bounded with the number of attributes or the size of access formula. The computation overhead is very high especially for the ABE schemes with CCA2 security. Such a drawback becomes more serious for resource-constrained users such as mobile devices and sensors. As a result, these computations cannot be independently completed by such users. For the purpose of reducing the computational overhead, the technique of outsourcing computation was introduced, in which the computation tasks can be outsourced to public cloud servers. In this way, the computational overhead at user side can be reduced greatly. There are many research works for secure outsourcing ABE, such as (Green et al., 2011; Zhou and Huang, 2012). However, all these works require that the users need to blind and upload the computational tasks to the cloud server. After the cloud server returns the results, the users unblind and get its final results. There are three main drawbacks when utilizing such a technique in the computation for resource-constrained users. First, the blind and unblind algorithms require some computational cost, which also has impact on the response time. Second, the users have to interact with the cloud server for computation outsourcing. Finally, the result returned from the cloud servers cannot be fully trusted. As far as the authors' knowledge, the problem of simultaneously achieving fine-grainedness, high-efficiency on the data owner's side, and standard data confidentiality of cloud data sharing still remains unresolved.

1.1. *Our contribution*

Research contributions of this paper can be summarized as follows:

- In order to realize secure attribute-based data sharing (ABDS) suitable for resource-constrained mobile users, we introduce a new online/offline ABE scheme that eliminates a majority of the computation task by adding system public parameters besides moving the encryption computation overhead on the data owner's side to the offline phase.
- A public ciphertext test phase is performed before the decryption phase, which eliminates most of the computational cost resulted from illegitimate ciphertexts. In other words, the public ciphertext test allows a user to check at a low cost whether a potential equation holds for components of a given ciphertext before performing the expensive decryption phase.
- The technique of Chameleon hash function is used to generate an immediate ciphertext, which will be blinded by the offline ciphertexts to obtain the final online ciphertexts. In this way, the proposed scheme is proven CCA2 secure, which is widely recognized as a standard security notion. Theoretical analysis and experimental results indicate that the proposed ABDS system is extremely suitable for resource-limited mobile users in cloud computing.

1.2. *Related work*

In this section, we summarize the related works on ABE, online/offline cryptography and outsourcing computation.

1.2.1. *Attribute-based encryption*

The notion of ABE, known as fuzzy identity-based encryption in Sahai and Waters (2005), was proposed and applied in biometrics encryption by Goyal et al. (2006). In biometrics encryption application, the key extracted from the biometrics such as fingerprint will always be different each time because of the biometric measurement noise during the extraction algorithm. With the technology of fuzzy identity-based encryption, such problem can be solved by introducing error-tolerance in fuzzy identity-based encryption. It allows the private key with slight difference from the original one to decrypt the ciphertext for the original biometric identity. The notion is extended into ABE by defining the identity as a set of attributes. In Goyal et al. (2006), it introduced two different and complementary notions of ABE called KP-ABE and CP-ABE, to deal with the error tolerance in key generation phase or ciphertext generation phase. A secure construction of KP-ABE was given in Goyal et al. (2006) by dividing the private key according to the access policy. A provably secure CP-APE construction supporting tree-based access structure in generic group model was presented by Bethencourt et al. (2007), where a random number for generation of ciphertext is divided according to the access policy specified in the ciphertext.

In the last decade, there are a lot of works on ABE constructions and applications proposed. They range from constructing stronger security schemes to proposing more efficient schemes. For example, to reduce the trust of attribute authority, the notion of ABE with multi-authorities has been proposed (Chase, 2007). Because of the anonymity of attribute private key, to prevent the key-abuse attack, the notion of accountable ABE was given and constructed in (Li et al., 2009). Aiming at improving the decryption efficiency in anonymous ABE, the match-then-decrypt technique was proposed by Zhang et al. (2013, 2017). Unbounded ABE was proposed by Lewko and Waters (2011), which has a large attribute universe and imposes no bound on the size of attribute sets used for encryption. More efficient large universe ABE schemes were presented by Rouselakis and Waters (2013). For the sake of revocable ABE, a direct attribute and user revocation mechanism is proposed in Zhang et al. (2014), where an auxiliary function is introduced to specify the ciphertexts involved in revocation events and then only these involved ciphertexts are updated. To enhance and provide better security, the fully-secure ABE was proposed in Waters (2009). ABE suitable for mobile cloud computing was proposed by Zhang et al. (2014), which features constant computation cost and constant-size ciphertexts. The scheme is used to realize attribute-based data sharing in mobile computing in Zhang et al. (2016). A KP-ABE scheme supporting public ciphertext test was presented by Liu et al. (2014), which realizes CCA2 with the help of Chameleon hashing. The notion of Chameleon hashing was first introduced by Krawczyk and Rabin (2000), further refined respectively by Ateniese and de Medeiros (2005) and Chen et al. (2007). Considering that computational cost in ABE encryption often scales with the complexity of the access structure or number of attributes, Hohenberger and Waters (2014) proposed online/offline ABE. However, the proposed schemes are CPA secure and fail to realize public ciphertext test.

1.2.2. *Online/offline cryptography*

The notion of online/offline was initiated by Even et al. (1996) for signatures. Later, Shamir and Tauman (2001) developed a paradigm called hash-sign-switch based on Chameleon hashing functions for designing efficient online/offline signature schemes. An online/offline signature scheme consists of two phases and it can efficiently enable handover authentication in wireless networks (Zhang et al., 2014). Before the message to be signed is known, the first offline phase is performed. The second online phase is performed once the message is known, and it is supposed to be very fast. In the online/offline signature schemes based on the hash-sign-switch paradigm (Shamir and Tauman, 2001), one security flaw is the key exposure problem of Chameleon hashing. To solve this problem, a special double-trapdoor hash family based on the discrete logarithm assumption was proposed by Chen et al. (2008), and they applied the hash-sign-switch paradigm to propose a much more efficient generic online/offline signature scheme.

The technique of online/offline encryption was introduced by Guo et al. (2008), where they proposed an identity-based online/offline encryption (IBOOE) scheme. In the proposed scheme, the encryption process is split into two phases: the offline phase and the online phase. The offline phase does the vast majority of the work to encrypt a message, and it requires neither the knowledge of the message to be encrypted nor the receiver's identity. This division of computational tasks makes encryption affordable by mobile devices with limited computation power in that most of the works can be executed offline. A more efficient IBOOE scheme was proposed by Liu and Zhou (2009). Very recently, an improved IBOOE scheme has been proposed by Lai et al. (2015). They proposed an efficient transformation to obtain an online/offline encryption scheme from a traditional identity-based encryption scheme. A new notion called identity-based online/offline key encapsulation mechanism was proposed by Chow et al. (2011), which allows the key encapsulation process to be split into offline and online stages. Especially, Hohenberger and Waters (2014) applied the idea to attribute-based encryption and proposed online/offline ABE. The first fully secure online/offline predicate encryption and attribute-based encryption schemes have recently been presented by Datta et al. (2015). These schemes are CPA secure and the computation efficiency on the data owner's side still needs to be improved. Besides, these schemes fail to realize public ciphertext test before performing expensive decryption.

1.2.3. *Outsourcing computation*

Cloud Computing is an emerging technology enabling the delivery of computing and storage resources as a service (Armbrust et al., 2010). The users can rent and pay the utility computation or storage service provided by the cloud service provider on-demand, which has more flexibility and elasticity comparing with traditional hosting services. Such on-demand scalability is realized because of the recent advancements in virtualization and network management. In cloud computing, users do not need to manage or know details of the underlying cloud infrastructure. However, the users are able to have the control of the specified operating systems, storage space etc. Currently, there are many cloud application examples like Amazon EC2 and S3 and Dropbox.

The issue of secure outsourcing computation, including the scientific computation problem and other computational problem, has drawn much attention for decades (Atallah and Frikken, 2010; Atallah and Li, 2005; Atallah et al., 2002; Benjamin and Atallah, 2008). Especially, with the fast development of mobile device and other resource-limited computational devices, such an outsourcing technique becomes more and more important. However, the previous techniques cannot be applied to ABE directly because of the problems are totally different. Some works also proposed to accelerate the computation of exponentiations such as Bicakci and Baykal (2004; Hohenberger and Lysyanskaya (2005; Jakobsson and Wetzel (2001). Recently, the technique of fully homomorphic encryption has also been utilized to construct and solve the problem of outsourcing computation (Gentry, 2009; Gentry and Halevi, 2011; Goldwasser et al., 2008). These papers are proposed for general problem, instead of the concrete computation problem. However, Gentry and Halevi (2011) pointed out that even for weak security parameters on "bootstrapping" operation of the homomorphic encryption, it takes around 30 seconds on a high performance machine. Therefore, the computational cost is too high to be applied in practical applications.

To avoid the inefficiency of the fully homomorphic encryption, some solutions for practical and concrete problems have been proposed recently. To reduce the computational overhead of access control using ABE, some works have addressed this problem by proposing outsourcing ABE techniques (Green et al., 2011; Li et al., 2012, 2014; Zhou and Huang, 2012). To reduce the computational overhead of decryption algorithm at the user side, in Green et al. (2011), it showed how to outsource the decryption of ABE to the cloud computing while keeping the security of the ABE as its original ABE scheme. In this way, the user only needs to compute constant computation, instead of the computational cost growing with the number of attributes. The encryption computational overhead also grows with the size of access policy or the number of attributes in literature. To further reduce the computational overhead during encryption, an outsourced ABE supporting outsourced encryption and decryption was presented in Li et al. (2012); Zhou and Huang (2012). The main techniques in these works (Green et al., 2011; Li et al., 2014; Zhou and Huang, 2012) are to blind user's attribute private key. Recently, Li et al. (2013) and Sahai et al. (2012). utilized a splitting method to keep the secret of the private key. They also considered to outsource the key generation computation, as well as the encryption and decryption computation simultaneously. To verify the correctness of decryption, Lai et al. (2013) proposed a solution for ABE with verifiable decryption. There are also some works (Li et al., 2017; Wang et al., 2011) proposing solutions of outsourcing linear programming computation, classification and aided revocation in identity-based encryption (Li et al., 2015).

1.3. *Organization*

The rest of this paper is organized as follows. Some preliminaries are reviewed in Section 2. We then present the system architecture, design goals and security model in Section 3. The proposed attribute-based data sharing scheme together with its security results are presented in Section 4. Performance related issues are described in Section 5. Finally, we conclude this paper in Section 6.

2. Preliminaries

In this section, we give a brief review on some cryptographic background, access structures and Chameleon hash functions.

2.1. *Notations*

Throughout this paper, we use $[k_1,k_2]$ to denote the set $\{k_1, k_1+1, \cdots, k_2\}$ containing consecutive integers. For $k \in \mathbb{N}$, we denote by $[k]$ the set $\{1, 2, \cdots, k\}$. For a set S, |S| represents its cardinality, and $s \in_R S$ means the variable s is chosen uniformly at random from S.

2.2. *Cryptographic background*

Definition 1. (Bilinear Pairings). *Let $\mathbb{G}$, $\mathbb{G}_T$ be cyclic multiplicative groups of prime order p. Let $g \in_R \mathbb{G}$ be a generator. We call $\hat{e}$ a bilinear pairing if $\hat{e}: \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map with the following properties:*

1. *Bilinear: $\hat{e}(g^a, g^b) = \hat{e}(g,g)^{ab}$ for all $a, b \in \mathbb{Z}_p$.*
2. *Non-degenerate: There exists $g_1, g_2 \in \mathbb{G}$ such that $\hat{e}(g_1, g_2) \neq 1$.*
3. *Computable: There is an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}$.*

Definition 2. (q-wDBDH Problem). *The q-weak Decisional Bilinear Diffie-Hellman (q-wDBDH) Problem* (Rouselakis and Waters, 2013) *in $\mathbb{G}$ is that, given the following terms:*

$$g, g^x, g^y, g^z, g^{(xz)^2},$$

$$g^{b_i}, g^{xzb_i}, g^{xz/b_i}, g^{x^2zb_i}, g^{y/b_i^2}, g^{y^2/b_i^2}, \forall i \in [q]$$

$$g^{xzb_i/b_j}, g^{yb_i/b_j^2}, g^{xyzb_i/b_j}, g^{(xz)^2 b_i/b_j}, \forall i, j \in [q], i \neq j,$$

where $x, y, z, b_1, b_2, \cdots, b_q$ are randomly chosen from $\mathbb{Z}_p$, and $T \in \mathbb{G}_T$, to decide if $T = \hat{e}(g,g)^{xyz}$.

We say that the (ϵ, q)-wDBDH Assumption holds in $\mathbb{G}$ if no Probabilistic Polynomial Time (PPT) algorithm has probability at least $\frac{1}{2} + \epsilon$ in solving the q-wDBDH problem in $\mathbb{G}$ for non-negligible ϵ.

2.3. *Access structures and linear secret sharing schemes*

Definition 3. (Access Structures (Beimel, 1996)). *Let $\mathcal{U}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\mathcal{U}}$ is monotone if $\forall B \in \mathbb{A}$ and $C \in 2^{\mathcal{U}}$: if $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (monotone access structure) on $\mathcal{U}$ is a collection (monotone collection) $\mathbb{A}$ of non-empty subsets of $\mathcal{U}$, i.e., $\mathbb{A} \subseteq 2^{\mathcal{U}} \backslash \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, otherwise, the sets are called the unauthorized sets.*

In attribute-based encryption systems, the roles of the parties are determined by the attributes in the attribute universe $\mathcal{U}$. Therefore, the access structure $\mathbb{A}$ will contain the authorized sets of attributes.

Definition 4. (Linear Secret Sharing Schemes (LSSS) (Beimel, 1996)). *Let $\mathcal{U}$ be the attribute universe and $\mathbb{A}$ an access structure on $\mathcal{U}$. An LSSS can be used to represent an access structure $\mathbb{A} = (M, \rho)$, where M is an $\ell \times n$ matrix which is called the share-generating matrix and ρ maps a row of M into an attribute. An LSSS consists of two algorithms:*

- ***Share***$((M, \rho), s)$: *This algorithm is used to share a secret value s based on attributes. Considering a vector $\vec{\upsilon} = (s, y_2, \ldots, y_n)^T$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $y_2, \ldots, y_n \in_R \mathbb{Z}_p$, then $\lambda_i = \vec{M}_i \cdot \vec{\upsilon}$ is a share of the secret s which belongs to the attribute $\rho(i)$.*
- ***Reconstruction***$(\lambda_1, \ldots, \lambda_\ell, (M, \rho))$: *This algorithm is used to reconstruct s from secret shares. Let $S \in \mathbb{A}$ be any authorized set and $I = \{i | \rho(i) \in S\} \subseteq \{1, 2, \ldots, \ell\}$. Then there exists coefficients $\{c_i\}_{i \in I}$ such that $\sum_{i \in I} c_i \vec{M}_i = (1, 0, \ldots, 0)$, thus we have $\sum_{i \in I} c_i \lambda_i = s$.*

2.4. *Chameleon hash functions*

A chameleon hash function is a trapdoor collision-resistant hash function, which is associated with a trapdoor/hash key pair (sk_{ch}, pk_{ch}). Anyone who knows the public key pk_{ch} can efficiently compute the hash value for each input. However, there exists no efficient algorithm for anyone except the holder of the secret key sk_{ch}, to find collisions for every given input. A chameleon hash function consists of three polynomial time algorithms as below:

- **KeyGen**$_{\mathbf{ch}}(1^{\lambda}) \to (sk_{ch}, pk_{ch})$: It takes as inputs the security parameter λ, and outputs a trapdoor/hash key pair (sk_{ch}, pk_{ch}).
- $\mathbf{H}_{\mathbf{ch}}(pk_{ch}, m_{ch}, r_{ch}) \to v$: It takes as inputs the Chameleon hash public key pk_{ch}, a message m_{ch}, and an auxiliary random parameter $r_{ch} \in_R \mathcal{R}$, where $\mathcal{R}$ is a universe specified by $\mathbf{H}_{\mathbf{ch}}$, and outputs the hashed value v.
- **TrapCollision**$_{\mathbf{ch}}(sk_{ch}, m_{ch}, r_{ch}, m'_{ch}) \to r'_{ch}$: It takes as inputs the Chameleon hash secret key sk_{ch}, a message m_{ch} with its auxiliary random parameter r_{ch}, and another message $m'_{ch} \neq m_{ch}$, and outputs another auxiliary random parameter r'_{ch} such that $v' = \mathbf{H}_{\mathbf{ch}}(pk_{ch}, m'_{ch}, r'_{ch}) = \mathbf{H}_{\mathbf{ch}}(pk_{ch}, m_{ch}, r_{ch}) = v$.

A secure Chameleon hash function satisfies the requirements of *Collision Resistance* and *Uniformity*.

3. System architecture and security model

3.1. System architecture and design goals

As shown in Fig. 1, the system architecture of attribute-based data sharing suitable for resource-constrained users in cloud computing consists of four entities AA (Attribute Authority), CSP (Cloud Service Provider), MO (Mobile Data Owner), and DU (Data User).

- AA is a key entity who generates system public parameters and master keys. Especially, the system public parameters contain immediate ciphertexts, which can be used by MO in the online phase. Also, AA manages users in the system and it is fully trusted by entities in the attribute-based data sharing system.
- MO is a resource-constrained entity who wishes to safely store a file on cloud storage servers maintained by CSP for sharing. Before it specifies the message, MO can generate offline ciphertexts while accessing the power source. When the message becomes known, MO can calculate final ciphertexts online without significantly draining the battery.
- CSP is in charge of saving the ciphertext data of MO and it consists of a lot of cloud storage servers, which are maintained by a data service manager.
- DU is an entity who has a secret key and intends to access a ciphertext hosted in CSP. In order to improve the efficiency of decryption, a public ciphertext test phase is additionally introduced before the decryption phase. To be specific, after downloading the ciphertext from CSP, DU should perform the test that if the ciphertext is legitimate. And, the decryption phase is performed if and only if the ciphertext passes the test.

In this work, it is assumed that all the entities except AA are “honest-but-curious”. More precisely, they will honestly execute the tasks assigned by legitimate parties but try to find out as much private information as possible. The security goal is semantic security of data, which is closely related to the following two requirements:

- *Data Confidentiality*. Unauthorized users should be prevented from recovering the message of ciphertexts. In addition, unauthorized access from CSP to the message of ciphertexts should also be prevented.
- *Collusion-Resistance*. Malicious users colluding with CSP should not succeed in decrypting the ciphertext by combining their attributes if each of them cannot decrypt the ciphertext alone.

Also, the performance-related issue should be taken into consideration.

- *Online/Offline Encryption*. The scheme allows a resource-constrained mobile user to quickly transform a message into an ABE ciphertext. Specifically, a lot of preparation work can

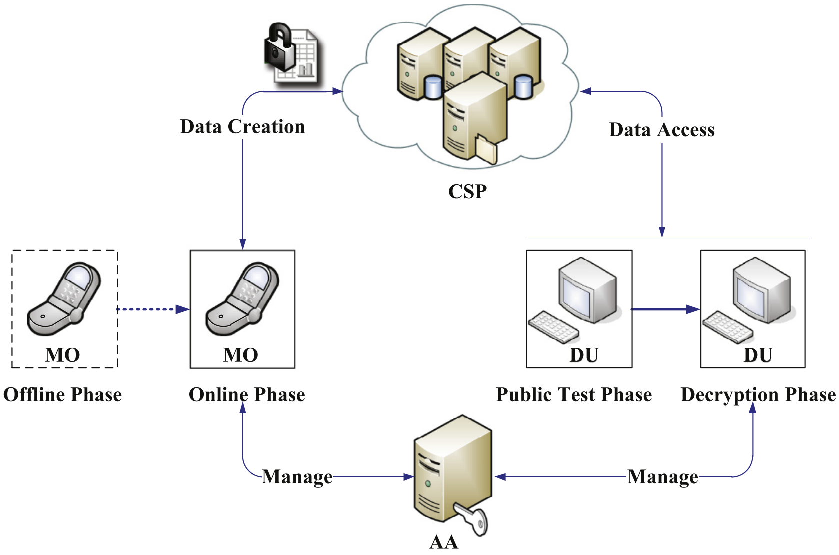


Fig. 1 – System architecture of attribute-based data sharing for resource-constrained users.

be performed by other entities and the mobile user while accessing a power supply.
- *Public Ciphertext Test.* Anyone can verify whether a ciphertext is legitimate without requiring secret keys. Invalid ciphertexts are thrown away without performing the decryption phase.

Based on the proposed system architecture, we define the attribute-based data sharing system suitable for resource-constrained users in cloud computing. The system involves five phases as below.

- **Initialization.** AA generates system public parameters and master keys for the system. All users can obtain the system public parameters, where immediate ciphertexts are calculated by AA and used in the subsequent online data creation phase by MO.
- **User Registration.** A user can join the attribute-based data sharing system by committing an access structure to AA, who issues a secret key to the user based on the access structure.
- **Offline Data Creation.** MO generates offline ciphertexts, which are used in the subsequent online data creation phase by MO.
- **Online Data Creation.** MO encrypts a file based on an attribute set and outsources the final ciphertext to CSP for sharing.
- **Data Access.** DU downloads a ciphertext from CSP. If the ciphertext is legitimate, then MU decrypts it based on his/her secret keys.

3.2. *Security model*

Before giving the formalized security model, we first lay out the definition of an online/offline KP-ABE scheme supporting public ciphertext test, which is the basic tool of the proposed attribute-based data sharing system. An online/offline KP-ABE scheme supporting public ciphertext test consists of the following five algorithms:

- **Setup**$(1^\lambda) \to (PK, MK)$: The setup algorithm is run by AA. It takes as input a security parameter λ, and outputs the system public key PK and the master key MK.
- **KeyGen**$(PK, MK, \mathbb{A}) \to SK_{\mathbb{A}}$: The key generation algorithm is run by AA. It takes as input the system public key PK, the master key MK and an access structure $\mathbb{A}$, and outputs $SK_{\mathbb{A}}$ as the secret key associated with $\mathbb{A}$.
- **Encrypt$_{\text{off}}$**$(PK) \to CT_{\text{off}}$: The offline encryption algorithm is run by MO. On input the system public key PK, it generates an offline ciphertext CT_{off}.
- **Encrypt$_{\text{on}}$**$(PK, m, S, CT_{\text{off}}) \to CT_S$: The online encryption algorithm is run by MO. On input the system public key PK, a message m, an attribute set S and an offline ciphertext CT_{off}, it generates the final ciphertext CT_S.
- **Decrypt**$(PK, CT_S, SK_{\mathbb{A}}) \to m$ or $\perp$: The decryption algorithm is run by DU. It involves a public ciphertext test phase and a decryption phase. Note that the secret key $SK_{\mathbb{A}}$ is only used in the decryption phase. On input the system public key PK, a ciphertext CT_S of a message m under S, and a secret key $SK_{\mathbb{A}}$ associated with $\mathbb{A}$, the ciphertext CT_S is tested and decrypted by DU as follows:
  1. *Public Test Phase*: It returns $\perp$ to terminate decryption if CT_S is illegitimate. Otherwise, the *Public Test Phase* ends by initiating the *Decryption Phase*.
  2. *Decryption Phase*: It outputs the message m if S is an authorized set of $\mathbb{A}$.

In the following, based on the system architecture, we formalize the security model by specifying the ability of adversaries. We define the indistinguishability against selective chosen attribute set and chosen ciphertext attacks in KP-ABE systems. The security model is defined through a game played by an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$, as shown in Fig. 2.

Definition 5. *An online/offline KP-ABE scheme supporting public ciphertext test is said to be selective chosen attribute set and chosen ciphertext secure if no PPT adversary can break the above security game with a non-negligible advantage.*

Init: The adversary $\mathcal{A}$ commits to a challenge attribute set S^* and sends it to the challenger $\mathcal{B}$.
Setup: The challenger $\mathcal{B}$ chooses a sufficiently large security parameter λ, and runs the **Setup** algorithm to get a master key MK and the corresponding system public key PK. It retains MK and gives PK to $\mathcal{A}$.
Phase 1: The adversary $\mathcal{A}$ issues a polynomially bounded number of queries to the following key generation oracle and decryption oracle:

- **KeyGen Oracle** $\mathcal{O}_{KeyGen}$: The adversary $\mathcal{A}$ submits an access structure $\mathbb{A}$ that is not satisfied by S^*. The challenger $\mathcal{B}$ gives $\mathcal{A}$ the secret key for $\mathbb{A}$.
- **Decrypt Oracle** $\mathcal{O}_{Dec}$: The adversary $\mathcal{A}$ submits a ciphertext CT_S of a message m with respect to an attribute set S. The challenger $\mathcal{B}$ returns the message m if CT_S is legitimate.

Challenge: Once $\mathcal{A}$ decides that **Phase 1** is over, it outputs two equal-length messages m_0 and m_1 on which it wishes to be challenged with respect to S^*. The challenger $\mathcal{B}$ flips a random coin $b \in \{0,1\}$, computes $CT_{S^*} = \textbf{Encrypt}_{\textbf{on}}(PK, m_b, S^*, CT_{\text{off}})$ and sends CT_{S^*} to $\mathcal{A}$, where $CT_{\text{off}} = \textbf{Encrypt}_{\textbf{off}}(PK)$.
Phase 2: The same as **Phase 1**, except that CT_{S^*} may not be submitted for oracle $\mathcal{O}_{Dec}$.
Guess: The adversary $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$ and wins the game if $b' = b$.
The advantage of an adversary $\mathcal{A}$ in attacking the KP-ABE system with security parameter λ is defined as follows:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{KP\text{-}ABE}}(\lambda) = \left|\Pr[b' = b] - 1/2\right|.$$

Fig. 2 – Formalized security model.

4. ABDS for resource-limited users

4.1. *Proposed ABDS system*

Initialization Phase. In the initialization phase, AA generates system public parameters and master keys by performing the following procedures.

- AA performs the algorithm **Setup**(1^λ): Let $\mathbb{G}$, $\mathbb{G}_T$ be two cyclic multiplicative groups of large prime order p, $\hat{e}:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ be a bilinear map, and g be a generator of $\mathbb{G}$. Let $\mathcal{U}=\left[0,\frac{p-1}{2}\right]$ be the regular attribute universe, and $\mathcal{V}=\left[\frac{p+1}{2},p-1\right]$ the verification universe. A secure Chameleon hash function $\mathbf{H_{ch}}:\{0,1\}^*\to\mathcal{V}$ is adopted in the scheme. AA selects $h,u,\omega\in_R\mathbb{G}$, an exponent $\alpha\in_R\mathbb{Z}_p$ and computes $Y=\hat{e}(g,g)^\alpha$.
- AA runs the $\mathbf{KeyGen_{ch}}(1^\lambda)$ algorithm of $\mathbf{H_{ch}}$ to obtain a Chameleon hash key pair (sk_{ch},pk_{ch}).
- **IT Pool Construction**: AA picks $r_j,x_j\in_R\mathbb{Z}_p$, computes $C'_{j,1}=g^{r_j}$, $C'_{j,2}=\left(u^{x_j}h\right)^{r_j}$, and sets $IT_j=(r_j,x_j,C'_{j,1},C'_{j,2})$ for $j\in[N]$, where N is an integer used by AA to determine the size of the immediate ciphertext pool. Then, AA picks $r_0\in_R\mathbb{Z}_p$, $r_{ch}\in_R\mathcal{R}$, computes $C'_{0,1}=g^{r_0}$, $\upsilon=\mathbf{H_{ch}}(pk_{ch},C'_{0,1}\,||\,C'_{1,1}\,||\,C'_{2,1}\,||\cdots||\,C'_{N,1},r_{ch})$, $C'_{0,2}=(u^\upsilon h)^{r_0}$ and sets $IT_0=(C'_{0,1},C'_{0,2})$, where $\mathcal{R}$ is the auxiliary parameter universe of $\mathbf{H_{ch}}$. Finally, AA sets $IT=\{IT_j\}_{j\in[0,N]}$. Note that the IT pool can be updated by AA if necessary. To be specific, it is necessary for the attribute authority to update the IT pool whenever new attributes are introduced to the system for improving the expressiveness and the number of attributes exceeds the size of the IT pool. Suppose a new attribute $A_{\bar{j}}$ is introduced to the system. In order to update the IT pool, AA picks $r_{\bar{j}},x_{\bar{j}}\in_R\mathbb{Z}_p$, computes $C'_{\bar{j},1}=g^{r_{\bar{j}}}$, $C'_{\bar{j},2}=\left(u^{x_{\bar{j}}}h\right)^{r_{\bar{j}}}$, sets $IT_{\bar{j}}=(r_{\bar{j}},x_{\bar{j}},C'_{\bar{j},1},C'_{\bar{j},2})$ and adds $IT_{\bar{j}}$ to the IT pool. Then, AA computes $r'_{ch}=\mathbf{TrapCollision_{ch}}(sk_{ch},m_{ch},r_{ch},m'_{ch})$, where $m_{ch}=C'_{0,1}\,||\,C'_{1,1}\,||\,C'_{2,1}\,||\cdots||\,C'_{N,1}$ and $m'_{ch}=m_{ch}\,||\,C'_{\bar{j},1}$. In the system public key, AA only needs to update the components r_{ch} and IT with new values.
- The system public key is published as $PK=\langle\mathbf{H_{ch}},r_{ch},pk_{ch},g,h,u,\omega,Y,IT\rangle$. The master key is $MK=\langle\alpha\rangle$.

User Registration Phase. Upon receiving an access structure $\mathbb{A}=(M,\rho)$ from a mobile user, where $M\in\mathbb{Z}_p^{\ell\times n}$ and $\rho:[\ell]\to\mathcal{U}$, AA generates a secret key $SK_{(M,\rho)}$ for the user based on the algorithm **KeyGen**.

- **KeyGen**$(PK,MK,(M,\rho))$: AA first sets $\vec{y}=(\alpha,y_2,\cdots,y_n)^T$ with $y_2,\cdots,y_n\in_R\mathbb{Z}_p$, and calculates the vector of shares $\vec{\lambda}=(\lambda_1,\lambda_2,\cdots,\lambda_\ell)^T=M\vec{y}$. Then AA picks $t_1,t_2,\cdots,t_\ell\in_R\mathbb{Z}_p$, and for $i\in[\ell]$, computes $K_{i,0}=g^{\lambda_i}\omega^{t_i}$, $K_{i,1}=\left(u^{\rho(i)}\cdot h\right)^{-t_i}$, $K_{i,2}=g^{t_i}$. Finally, the secret key is $SK_{(M,\rho)}=\left\langle(M,\rho),\{K_{i,0},K_{i,1},K_{i,2}\}_{i\in[\ell]}\right\rangle$.

Offline Data Creation Phase. MO generates offline ciphertexts CT_{off} based on the algorithm $\mathbf{Encrypt_{off}}$.

- $\mathbf{Encrypt_{off}}(PK)$: MO picks $s\in_R\mathbb{Z}_p$ and computes $K=Y^s$, $C_0=g^s$, $C_\omega=\omega^{-s}$. Then MO sets $CT_{\text{off}}=\langle K,C_0,C_\omega\rangle$.

Note that MO can construct a pool of offline ciphertexts, which can be used in different online phase.

Online Data Creation Phase. In the online phase, MO chooses any one offline module $CT_{\text{off}}=\langle K,C_0,C_\omega\rangle$ from the pool. Then, MO encrypts a file $m\in\mathbb{G}_T$ with respect to an attribute set S by performing the following $\mathbf{Encrypt_{on}}$ algorithm, and outsources the final ciphertext CT_S to CSP for sharing.

- $\mathbf{Encrypt_{on}}(PK,m,S,CT_{\text{off}})$: Suppose the attribute set $S=\{A_1,A_2,\cdots,A_\kappa\}$, where $\kappa=|S|$. MO chooses any κ immediate modules $IT_j=(r_j,x_j,C'_{j,1},C'_{j,2})$ and $IT_o=(C'_{0,1},C'_{0,2})$ in the immediate ciphertext pool. Then, for $j\in[\kappa]$, MO computes $C=m\cdot K$, $C_{j,1}=C'_{j,1}=g^{r_j}$, $C_{j,2}=C'_{j,2}\cdot C_\omega=\left(u^{x_j}h\right)^{r_j}\cdot\omega^{-s}$, $C_{j,3}=r_j\cdot(A_j-x_j)$, and $C_{0,1}=C'_{0,1}=g^{r_0}$, $C_{0,2}=C'_{0,2}\cdot C_\omega=(u^\upsilon h)^{r_0}\cdot\omega^{-s}$. Finally, the ciphertext of m with respect to S is $CT_S=\left\langle S,C,C_0,C_{0,1},C_{0,2},\{C_{j,1},C_{j,2},C_{j,3}\}_{j\in[\kappa]}\right\rangle$.

Data Access Phase. DU downloads a ciphertext CT_S from CSP, and performs the following algorithm **Decrypt** based on his/her secret key $SK_{(M,\rho)}$ to recover the corresponding message.

- **Decrypt**$(PK,CT_S,SK_{(M,\rho)})$: The ciphertext CT_S is tested and decrypted by DU as follows:
  - *Public Test Phase*: DU computes $\upsilon=\mathbf{H_{ch}}\left(pk_{ch},\hat{C}_{0,1}\,||\,\hat{C}_{1,1}\,||\cdots||\,\hat{C}_{N,1},r_{ch}\right)$, where $\hat{C}_{j,1}=C_{j,1}$ if $C_{j,1}$ is a component of CT_S, otherwise $\hat{C}_{j,1}=C'_{j,1}$. Then, DU verifies whether the ciphertext is legitimate based on the following equation:

$$\hat{e}\left(g,C_{0,2}\prod_{i=1}^{\kappa}C_{i,2}\cdot u^{C_{i,3}}\right)\hat{e}(C_0,\omega^\kappa)=\hat{e}\left(\prod_{i=0}^{\kappa}C_{i,1},h\right)\hat{e}\left(C_{0,1}^{\upsilon}\prod_{i=1}^{\kappa}C_{i,1}^{A_i},u\right),$$

  where $\kappa=|S|$. If the above equation holds, then CT_S is legitimate and DU proceeds. Note that even if the IT pool has been updated, the computation of υ is correct because the value of υ is not altered based on the trapdoor collision property of Chameleon hash functions. Hence, the proposed scheme is not affected even if different IT pools are used in the encryption and decryption.
  - *Decryption Phase*: If S is not an authorized set of (M,ρ), it returns ⊥. Otherwise, DU finds the set of rows in M that provides shares to attributes in S, i.e., $I=\{i|\rho(i)\in S\}$. Then DU calculates $\{\omega_i\in\mathbb{Z}_p\}_{i\in I}$ such that $\sum_{i\in I}\omega_i\vec{M}_i=(1,0,\cdots,0)$, where $\vec{M}_i$ is the i-th row of the matrix M. Finally, the message m can be recovered by computing

$$B=\prod_{i\in I}\left(\hat{e}(K_{i,0},C_0)\hat{e}(K_{i,1},C_{j,1})\hat{e}\left(K_{i,2},C_{j,2}\cdot u^{C_{j,3}}\right)\right)^{\omega_i},$$

  where j is the index of the attribute $\rho(i)$ in S, and then $m=C/B$.

4.2. *Security results*

Theorem 1. The proposed ABDS system is secure in the proposed selective chosen attribute set and chosen ciphertext security model under the $(q+1)$-wDBDH assumption.

Proof 1. The proposed ABDS system is based on a potential online/offline KP-ABE scheme supporting public ciphertext test. We denote the potential scheme by $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}_{\mathsf{off}}, \mathsf{Encrypt}_{\mathsf{on}}, \mathsf{Decrypt})$, which is an improved version of a typical KP-ABE scheme (Liu et al., 2014), denoted by $\Pi_o = (\mathsf{Setup}_o, \mathsf{KeyGen}_o, \mathsf{Encrypt}_o, \mathsf{Decrypt}_o)$. Because the scheme Π_o is selective chosen attribute set and chosen ciphertext secure under the (q + 1)-wDBDH assumption, if we can reduce the security of Π to that of Π_o, then the proposed ABDS system is secure in the proposed security model under the (q + 1)-wDBDH assumption. In the following, we will show that any PPT attacker $\mathcal{A}$ with a non-negligible advantage ε in the proposed security model against Π can be used to design a PPT simulator $\mathcal{B}$, which can break the security of Π_o with an advantage ε. The simulator $\mathcal{B}$ plays the challenger and interacts with $\mathcal{A}$ in the proposed security model. The simulation proceeds as follows:

Init: *Initially,* $\mathcal{A}$ *gives* $\mathcal{B}$ *a challenge attribute set* $S^* = \{A_1^*, A_2^*, \cdots, A_\kappa^*\}$*, and* $\mathcal{B}$ *forwards it to the challenger of* Π_o*, where* $\kappa = |S^*| \le q$*.*

Setup: *The challenger* $\mathcal{B}$ *receives public parameters* $\langle \mathbf{H}_{\mathbf{ch}}, r_{ch}, pk_{ch}, g, h, u, \omega, Y\rangle$ *from the challenger of* Π_o*. Additionally,* $\mathcal{B}$ *picks* $r_j, x_j \in_R \mathbb{Z}_p$*, computes* $C'_{j,1} = g^{r_j}$*,* $C'_{j,2} = (u^{x_j} h)^{r_j}$*, and sets* $IT_j = (r_j, x_j, C'_{j,1}, C'_{j,2})$ *for* $j \in [N]$*, where N is an integer used by AA to determine the size of the immediate ciphertext pool. Then,* $\mathcal{B}$ *picks* $r_0 \in_R \mathbb{Z}_p$*, computes* $C'_{0,1} = g^{r_0}$*,* $v = \mathbf{H}_{\mathbf{ch}}(pk_{ch}, C'_{0,1} \| C'_{1,1} \| C'_{2,1} \| \cdots \| C'_{N,1}, r_{ch})$*,* $C'_{0,2} = (u^v h)^{r_0}$ *and sets* $IT_0 = (C'_{0,1}, C'_{0,2})$*. Also,* $\mathcal{B}$ *sets* $IT = \{IT_j\}_{j\in[0,N]}$*. Finally,* $\mathcal{B}$ *sends* $PK = \langle \mathbf{H}_{\mathbf{ch}}, r_{ch}, pk_{ch}, g, h, u, \omega, Y, IT\rangle$ *to* $\mathcal{A}$*.*

Phase 1: $\mathcal{A}$ *makes the following queries.*

- `KeyGen Query` $\mathcal{O}_{\text{KeyGen}}(M, \rho)$*: The adversary* $\mathcal{A}$ *submits an access structure* $\mathbb{A} = (M, \rho)$ *that is not satisfied by* S^**, where* $M \in \mathbb{Z}_p^{\ell\times n}$ *and* $\rho : [l] \to \mathcal{U}$*. Because the secret keys in both schemes* Π *and* Π_o *are the same,* $\mathcal{B}$ *just passes (M, ρ) to the challenger of* Π_o *and obtains the secret key* $SK_{(M,\rho)} = \langle (M,\rho), \{K_{i,0}, K_{i,1}, K_{i,2}\}_{i\in[\ell]}\rangle$*. Then,* $\mathcal{B}$ *gives* $\mathcal{A}$ *the secret key* $SK_{(M,\rho)}$*.*
- `Decryption Query` $\mathcal{O}_{\text{Dec}}(CT_S)$*: The adversary* $\mathcal{A}$ *submits a ciphertext*

$$CT_S = \langle S, C, C_0, C_{0,1}, C_{0,2}, \{C_{j,1}, C_{j,2}, C_{j,3}\}_{j\in[|S|]}\rangle.$$

$\mathcal{B}$ *verifies whether the ciphertext is legitimate based on the following equation:*

$$\hat{e}\left(g, C_{0,2}\prod_{i=1}^{|S|} C_{i,2}\cdot u^{C_{i,3}}\right)\hat{e}(C_0, \omega^{|S|}) = \hat{e}\left(\prod_{i=0}^{|S|} C_{i,1}, h\right)\hat{e}\left(C_{0,1}^{v}\prod_{i=1}^{|S|} C_{i,1}^{A_i}, u\right),$$

where $v = \mathbf{H}_{\mathbf{ch}}(pk_{ch}, \hat{C}_{0,1} \| \hat{C}_{1,1} \| \cdots \| \hat{C}_{N,1}, r_{ch})$*, and* $\hat{C}_{j,1} = C_{j,1}$ *if* $C_{j,1}$ *is a component of* CT_S*, otherwise* $\hat{C}_{j,1} = C'_{j,1}$*. If the above equation holds,* CT_S *is legitimate and* $\mathcal{B}$ *proceeds.*

Subsequently, $\mathcal{B}$ *transforms* CT_S *into* CT_o *of a* Π_o *ciphertext.* $\mathcal{B}$ *computes* $\bar{C}_{j,2} = C_{j,2}\cdot u^{C_{j,3}}$ *and* $\bar{r}_{ch} = \mathbf{TrapCollision}_{\mathbf{ch}}(sk_{ch}, m, r_{ch}, \bar{m})$*, where* $\hat{m} = \hat{C}_{0,1} \| \hat{C}_{1,1} \| \cdots \| \hat{C}_{N,1}$*, and* $\bar{m} = C_{0,1} \| C_{1,1} \| \cdots \| C_{|S|,1}$*. Then,* $\mathcal{B}$ *sets* $CT_o = \langle S, \bar{r}_{ch}, C, C_0, C_{0,1}, C_{0,2}, \{C_{j,1}, \bar{C}_{j,2}\}_{j\in[|S|]}\rangle$*. Finally,* $\mathcal{B}$ *sends* CT_o *to the challenger of* Π_o*, and passes the received results to* $\mathcal{A}$ *unchanged.*

Challenge: *The adversary* $\mathcal{A}$ *submits two challenge messages* m_0 *and* m_1*. The challenger* $\mathcal{B}$ *sends them to the* Π_o *challenger and receives a challenge ciphertext* $CT_o^* = \langle S^*, r_{ch}^*, C, C_0, C_{0,1}, C_{0,2}, \{C_{j,1}, C_{j,2}\}_{j\in[\kappa]}\rangle$*, which is the* Π_o *ciphertext of the message* m_b *with* $b \in {}_R\{0, 1\}$ *chosen by the challenger of* Π_o*. It then selects* $z_1, z_2, \cdots, z_\kappa \in_R \mathbb{Z}_p$ *and sets* $CT_{S^*} = \langle S^*, C, C_0, C_{0,1}, C_{0,2}, \{C_{j,1}, C_{j,2}^*, C_{j,3}^*\}_{j\in[\kappa]}\rangle$*, where* $C_{j,2}^* = C_{j,2}\cdot u^{-z_j}$ *and* $C_{j,3}^* = z_j$*. Obviously,* CT_{S^*} *is a challenge ciphertext of* Π*, and* $\mathcal{B}$ *just sends it to* $\mathcal{A}$*.*

Phase 2: *The same as* ***Phase 1****, except that* CT_{S^*} *may not be submitted for oracle* $\mathcal{O}_{\text{Dec}}$*.*

Guess: *Finally, the adversary* $\mathcal{A}$ *outputs a guess bit* $\tau_{\mathcal{A}} \in \{0, 1\}$*. The challenger* $\mathcal{B}$ *just sets its guess bit as* $\tau_{\mathcal{B}} = \tau_{\mathcal{A}}$*. Thus, if* $\mathcal{A}$ *can break the proposed ABDS system with an advantage* ε*, then* $\mathcal{B}$ *breaks the scheme* Π_o *with the same probability. In terms of the security result of* Π_o*, it follows that the theorem is proved.*

5. Performance evaluation

In this section, we compare the proposed scheme with some existing schemes (Hohenberger and Waters, 2014; Lewko and Waters, 2011; Liu et al., 2014; Rouselakis and Waters, 2013) in the security and efficiency respects. The comparison results are summarized in Tables 1 and 2, where **P**, **E** and **M** represent a pairing operation, an exponentiation operation and a multiplication operation in bilinear groups, respectively. We ignore minor cost factors such as arithmetic operations in $\mathbb{Z}_p$. The symbol "×" means that the scheme fails to realize the corresponding property. The symbol N is the size of offline ciphertext pool and it is determined by the size of the attri-

Table 1 – Communication cost and security comparisons of attribute-based data sharing schemes supporting LSSS.

Schemes	Computation cost				Security
	Offline encryption	Online encryption	Public ciphertext test	Decryption	
LW (Lewko and Waters, 2011)	×	$(5\ell_a+2)\mathbf{E}+(2\ell_a+1)\mathbf{M}$	×	$4k\mathbf{P}+k\mathbf{E}+4k\mathbf{M}$	CPA
RW (Rouselakis and Waters, 2013)	×	$(3\ell_a+3)\mathbf{E}+(2\ell_a+1)\mathbf{M}$	×	$3k\mathbf{P}+k\mathbf{E}+2k\mathbf{M}$	CPA
LLW (Liu et al., 2014)	×	$(3\ell_a+6)\mathbf{E}+(2\ell_a+3)\mathbf{M}+1\mathbf{H}$	$(2k+3)\mathbf{P}+(k+1)\mathbf{E}+(2k+2)\mathbf{M}+1\mathbf{H}$	$3k\mathbf{P}+k\mathbf{E}+2k\mathbf{M}$	CCA2
HW (Hohenberger and Waters, 2014)	$(3N+3)\mathbf{E}+N\mathbf{M}$	$(\ell_a+1)\mathbf{M}$	×	$3k\mathbf{P}+2k\mathbf{E}+3k\mathbf{M}$	CPA
Ours	$3\mathbf{E}$	$(\ell_a+2)\mathbf{M}$	$4\mathbf{P}+(2k+2)\mathbf{E}+(3k+2)\mathbf{M}+1\mathbf{H}$	$3k\mathbf{P}+2k\mathbf{E}+3k\mathbf{M}$	CCA2

Table 2 – Parameter size comparison of attribute-based data sharing schemes supporting LSSS.

Schemes	*PK*	*SK*	Offline *CT*	Online *CT*
LW (Lewko and Waters, 2011)	$5\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T}$	$4\ell_a\ell_{\mathbb{G}}$	×	$(3k+1)\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T}$
RW (Rouselakis and Waters, 2013)	$4\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T}$	$3\ell_a\ell_{\mathbb{G}}$	×	$(2k+1)\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T}$
LLW (Liu et al., 2014)	$4\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T} + \ell_{ch} + r_{ch}$	$3\ell_a\ell_{\mathbb{G}}$	×	$(2k+3)\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T} + \lvert q\rvert$
HW (Hohenberger and Waters, 2014)	$(2N+6)\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T} + 2N\lvert q\rvert$	$3\ell_a\ell_{\mathbb{G}}$	$(2P+1)\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T} + 2P\lvert q\rvert$	$(3k+1)\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T}$
Ours	$(2N+6)\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T} + 2N\lvert q\rvert + \ell_{ch} + r_{ch}$	$3\ell_a\ell_{\mathbb{G}}$	$2\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T}$	$(3k+3)\ell_{\mathbb{G}} + \ell_{\mathbb{G}_T}$

bute universe. The symbol P means the bound of attributes associated with a ciphertext. The number of attributes in the attribute list and the complexity of the access structure are denoted by ℓ_a and k, respectively. In Table 2, $\ell_{\mathbb{G}}$, $\ell_{\mathbb{G}_T}$, $|q|$, ℓ_{ch} and ℓ_r represent the size of an element in $\mathbb{G}$, the size of an element in $\mathbb{G}_T$, the size of an element in $\mathbb{Z}_q$, the size of a Chameleon hash value, and the size of a random value used in Chameleon hash, respectively. A Chameleon hash operation is denoted as **H**.

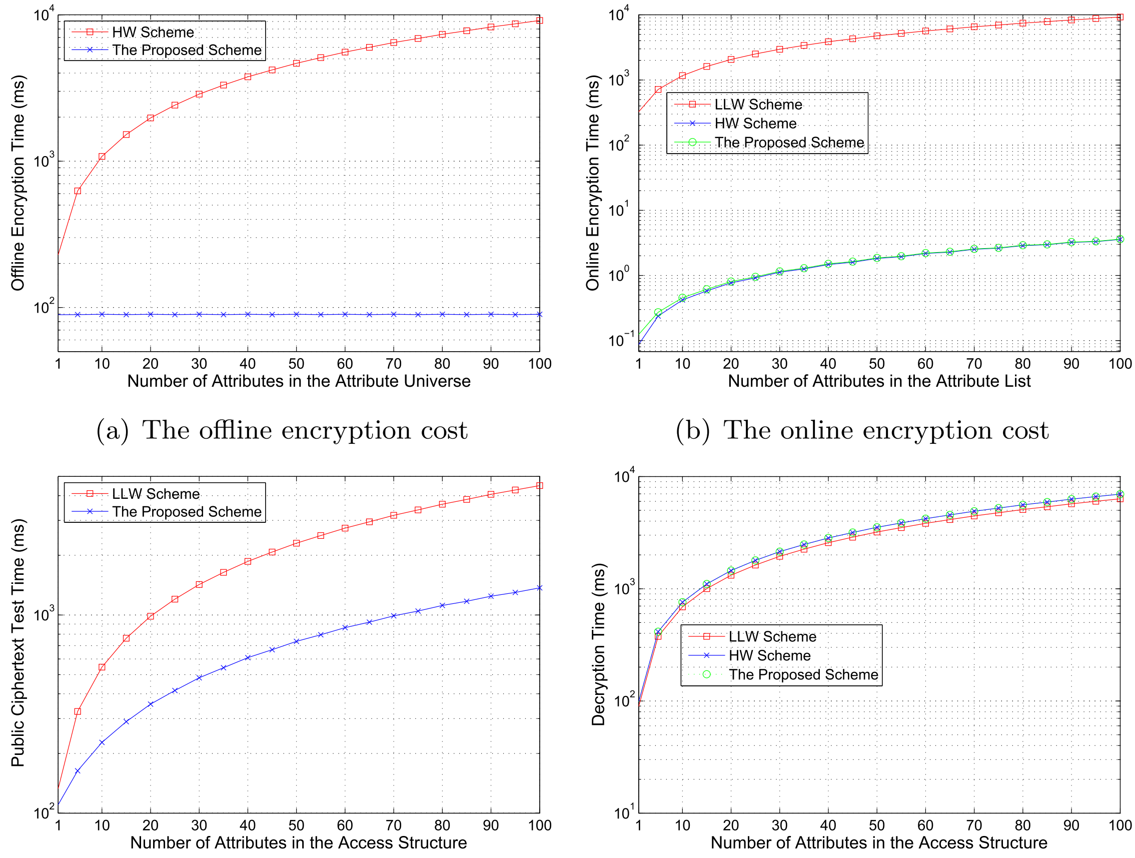
In Table 1, each scheme is compared in terms of the offline encryption cost, the online encryption cost, the public ciphertext test cost, the decryption cost and the security. It is noted that the offline encryption mode can eliminate a majority of the computation task, which is suitable for resource-limited mobile users in cloud computing. All these schemes are expressive and allow LSSS key policies. As shown in Table 1, only the scheme (Hohenberger and Waters, 2014) and ours support offline encryption. In particular, the proposed scheme only needs three **E** operations, while the number of **E** operations in the scheme (Hohenberger and Waters, 2014) is linearly proportional to N. On the other hand, during the decryption phase, the public ciphertext test can be performed by anyone and it eliminates most of energy consumption due to illegitimate ciphertexts. We note that only the scheme (Liu et al., 2014) and ours provide public ciphertext test mechanisms. As for the efficiency of the public ciphertext test mechanism, the proposed scheme is more efficient than the scheme (Liu et al., 2014) in that the number of the most expensive **P** operations is constant in ours and will not linearly increase with the complexity of access structures. In the security respect, only the proposed scheme and the scheme (Liu et al., 2014) realize CCA2 security and others are CPA-secure. We know from Table 2 that, although the system public key size of the proposed scheme is a little larger than that of Hohenberger and Waters (2014), our scheme has smaller offline ciphertexts. Similar to Hohenberger and Waters (2014; Liu et al. (2014); (Rouselakis and Waters (2013), the secret key size of our scheme is $3\ell_a\ell_{\mathbb{G}}$, which is smaller that that of (Lewko and Waters, 2011). In general, the proposed scheme is the first KP-ABE scheme, which can simultaneously support the online/offline encryption mode and public ciphertext test, and it efficiently realizes CCA2 security.

In order to precisely evaluate the performance, we implement and compare the computation cost of Liu et al.'s scheme LLW (Liu et al., 2014), Hohenberger et al.'s scheme HW (Hohenberger and Waters, 2014) with that of ours in Fig. 3. Note that the vertical axis is log scale. In Fig. 3(a) and (b), our simulation experiments are based on the Java Pairing-Based Cryptography Library (JPBC) (Caro and Iovino, 2011) and a lenovo P780 smartphone with Android OS 4.2 operation system. JPBC library provides the support of **E** and **M** operations and it is used to implement offline encryption and online encryption running in mobile smartphones. In Fig. 3(c) and (d), our simulation experiments are based on the Stanford Pairing-Based Cryptography Library (PBC) (Lynn) and a Linux machine with Intel Core 2 processors running at 2.40 GHz and 2G memory. PBC library can provide the support of **P**, **E** and **M** operations and it is used to implement public ciphertext test and decryption running in personal computers (PC). In our experiments, Type A pairings are adopted, which are constructed on the curve $y^2 = x^3 + x$ over the field $\mathbb{F}_q$ for some prime q satisfying $q = 3$ mod 4. Because the simulation does not involve the trapdoor collision of Chameleon hashing, we use the SHA-1, instead. This will not affect the simulation accuracy in that the computation cost of a Chameleon hashing just involves constant and a small amount of **E** operations. We consider the worst case of access structures, which ensures that all the ciphertext components are involved in decryption. Specifically, we generate 100 distinct access structures in the form of $(A_1 \wedge A_2 \wedge \cdots \wedge A_k)$ with k increasing from 1 to 100, where each component A_i is not wildcard. In each case, a corresponding secret key that contains exact k attributes is generated. For each access structure, the experiment is repeated 100 times on the PC and 50 times on the smartphone, and the average values are used as the final experimental results. Obviously, the experiment results indicate that the proposed scheme is very efficient in terms of the offline encryption cost, the online encryption cost, the public ciphertext test cost and the decryption cost. Therefore, we argue that the proposed ABDS system is more suitable for resource-limited users in cloud computing.

6. Conclusions and future work

Aiming at tackling the computation efficiency and weak data security issues in cloud data sharing, we propose an attribute-based data sharing scheme suitable for resource-limited mobile users in cloud computing. The proposed scheme supports online/offline encryption modes and allows anyone to check the validity of ciphertexts before expensive full decryption. Even the computation task in offline phase is significantly reduced by adding system public parameters. The proposed scheme is proven secure in the proposed selective chosen attribute set and chosen ciphertext security model under the wDBDH assumption. Theoretical analysis and experimental results indicate that the proposed data sharing scheme is extremely suitable for resource-limited mobile users.

A possible goal for our future research would be to consider direct attribute revocation in data sharing for resource-limited users in cloud computing.



(a) The offline encryption cost

(b) The online encryption cost

(c) The public ciphertext test cost

(d) The decryption cost

Fig. 3 – The computation overhead comparison.

Acknowledgment

The authors would like to thank the anonymous referees for their invaluable suggestions. This work is supported by the National Key Research and Development Program of China (No. 2017YFB0802002), the National Natural Science Foundation of China (No. 61472091, 61402366 and 61572382), the Distinguished Young Scholars Fund of Department of Education, Guangdong Province (No. Yq2013126), the Natural Science Basic Research Plan in Shaanxi Province (No. 2015JQ6236, 2016JM6033), and the Doctoral Fund of Ministry of Education of China (No. 20130203110004). Yinghui Zhang is supported by New Star Team of Xi'an University of Posts and Telecommunications.

REFERENCES

Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, et al. A view of cloud computing. Commun ACM 2010;53(4):50–8.

Atallah MJ, Frikken KB. Securely outsourcing linear algebra computations. In: Proceedings of the 5th ACM symposium on information, computer and communications security, ASIACCS '10. New York, NY, USA: ACM; 2010. p. 48–59.

Atallah MJ, Li J. Secure outsourcing of sequence comparisons. Int J Inf Secur 2005;4:277–87.

Atallah MJ, Pantazopoulos K, Rice JR, Spafford EE. Secure outsourcing of scientific computations. In: Zelkowitz MV, editor. Trends in software engineering, vol. 54. of Advances in Computers. Elsevier; 2002. p. 215–72.

Ateniese G, de Medeiros B. On the key exposure problem in chameleon hashes. In: Security in communication networks. Springer; 2005. p. 165–79.

Beimel A. Secure schemes for secret sharing and key distribution [Ph.D. thesis]. Technion-Israel Institute of Technology, Faculty of Computer Science; 1996.

Benjamin D, Atallah MJ. Private and cheating-free outsourcing of algebraic computations. In: Proceedings of the 2008 sixth annual conference on privacy, security and trust, PST '08. Washington, DC, USA: IEEE Computer Society; 2008. p. 240–5.

Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: IEEE symposium on security and privacy 2007. 2007. p. 321–34.

Bicakci K, Baykal N. Server assisted signatures revisited. In: Okamoto T, editor. Topics in cryptology – CT-RSA 2004, vol. 2964. Lecture Notes in Computer Science. Springer Berlin / Heidelberg; 2004. p. 1991–2.

Caro AD, Iovino V. Proceedings of the 16th IEEE symposium on computers and communications, ISCC 2011. IEEE; 2011. p. 850–5.

Chase M. Multi-authority attribute based encryption. In: Vadhan S, editor. Theory of Cryptography, vol. 4392. Lecture Notes in Computer Science. Springer Berlin / Heidelberg; 2007. p. 515–34.

Chen X, Zhang F, Susilo W, Mu Y. Efficient generic on-line/off-line signatures without key exposure. In: Applied cryptography and network security. Springer; 2007. p. 18–30.

Chen X, Zhang F, Tian H, Wei B, Susilo W, Mu Y, et al. Efficient generic on-line/off-line (threshold) signatures without key exposure. Inf Sci (Ny) 2008;178(21):4192–203.

Chow SS, Liu JK, Zhou J. Identity-based online/offline key encapsulation and encryption. In: Proceedings of the 6th ACM symposium on information, computer and communications security. ACM; 2011. p. 52–60.

Datta P, Dutta R, Mukhopadhyay S. Fully secure online/offline predicate and attribute-based encryption. In: Information security practice and experience. Springer; 2015. p. 331–45.

Even S, Goldreich O, Micali S. On-line/off-line digital signatures. J Cryptol 1996;9(1):35–67.

Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st annual ACM symposium on theory of computing, STOC '09. New York, NY, USA: ACM; 2009. p. 169–78.

Gentry C, Halevi S. Implementing gentry's fully-homomorphic encryption scheme. In: Paterson K, editor. Advances in cryptology – EUROCRYPT 2011, vol. 6632. Lecture Notes in Computer Science. Springer Berlin / Heidelberg; 2011. p. 129–48.

Goldwasser S, Kalai YT, Rothblum GN. Delegating computation: interactive proofs for muggles. In: Proceedings of the 40th annual ACM symposium on theory of computing, STOC '08. New York, NY, USA: ACM; 2008. p. 113–22.

Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on computer and communications security. 2006. p. 89–98.

Green M, Hohenberger S, Waters B. Outsourcing the decryption of abe ciphertexts. In: Proceedings of the 20th USENIX conference on security, SEC'11. Berkeley, CA, USA: USENIX Association; 2011. p. 34.

Guo F, Mu Y, Chen Z. Identity-based online/offline encryption. In: Financial cryptography and data security. Springer; 2008. p. 247–61.

Hohenberger S, Lysyanskaya A. How to securely outsource cryptographic computations. In: Kilian J, editor. Theory of cryptography, vol. 3378. Lecture Notes in Computer Science. Springer Berlin / Heidelberg; 2005. p. 264–82.

Hohenberger S, Waters B. Online/offline attribute-based encryption. In: Public-key cryptography–PKC 2014. Springer; 2014. p. 293–310.

Huang Z, Liu S, Mao X, Chen K, Li J. Insight of the protection for data security under selective opening attacks. Inf Sci (Ny) 2017;412–413:223–41.

Jakobsson M, Wetzel S. Secure server-aided signature generation. In: Public key cryptography. 2001. p. 383–401.

Krawczyk H, Rabin T. Chameleon hashing and signatures. In: Proc. of NDSS. Citeseer; 2000. p. 143–54.

Lai J, Deng R, Guan C, Weng J. Attribute-based encryption with verifiable outsourced decryption. IEEE Trans Inf Forens Secur 2013;8(8):1343–54.

Lai J, Mu Y, Guo F, Susilo W. Improved identity-based online/offline encryption. In: Information security and privacy. Springer; 2015. p. 160–73.

Lewko A, Waters B. Unbounded hibe and attribute-based encryption. In: Advances in cryptology–EUROCRYPT 2011. Springer; 2011. p. 547–67.

Li J, Ren K, Zhu B, Wan Z. Privacy-aware attribute-based encryption with user accountability. In: Samarati P, Yung M, Martinelli F, Ardagna C, editors. Information Security, vol. 5735. Lecture Notes in Computer Science. Springer Berlin / Heidelberg; 2009. p. 347–62.

Li J, Jia C, Li J, Chen X. Outsourcing encryption of attribute-based encryption with mapreduce. In: 14-th international conference on information and communications security (ICICS). 2012.

Li J, Chen X, Li J, Jia C, Ma J, Lou W. Fine-grained access control system based on outsourced attribute-based encryption. In: Crampton J, Jajodia S, Mayes K, editors. Computer security – ESORICS 2013, vol. 8134. Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2013. p. 592–609.

Li J, Huang X, Li J, Chen X, Xiang Y. Securely outsourcing attribute-based encryption with checkability. IEEE Trans Parallel Distrib Syst 2014;25(8):2201–10. doi:10.1109/TPDS.2013.271.

Li J, Chen X, Li M, Li J, Lee PP, Lou W. Secure deduplication with efficient and reliable convergent key management. IEEE Trans Parallel Distrib Syst 2014;25(6):1615–25.

Li J, Li J, Chen X, Jia C, Lou W. Identity-based encryption with outsourced revocation in cloud computing. IEEE Trans Comput 2015;64(2):425–37. doi:10.1109/TC.2013.208.

Li J, Yan H, Liu Z, Chen X, Huang X, Wong DS. Location-sharing systems with enhanced privacy in mobile online social networks. IEEE Syst J 2015;doi:10.1109/JSYST.2015.2415835.

Li P, Li J, Huang Z, Gao C-Z, Chen W-B, Chen K. Privacy-preserving outsourced classification in cloud computing. Cluster Comput 2017;doi:10.1007/s10586-017-0849-9.

Li P, Li J, Huang Z, Li T, Gao C-Z, Yiu S-M, et al. Multi-key privacy-preserving deep learning in cloud computing. Fut Gener Comput Syst 2017;74:76–85.

Liu JK, Zhou J. An efficient identity-based online/offline encryption scheme. In: Applied cryptography and network security. Springer; 2009. p. 156–67.

Liu W, Liu J, Wu Q, Qin B, Zhou Y. Practical direct chosen ciphertext secure key-policy attribute-based encryption with public ciphertext test. In: Computer security-ESORICS 2014. Springer; 2014. p. 91–108.

Lynn B. The pairing-based cryptography library. Available from: http://crypto.stanford.edu/pbc/.

Rouselakis Y, Waters B. Practical constructions and new proof methods for large universe attribute-based encryption. In: Proceedings of the 2013 ACM SIGSAC conference on computer & communications security. ACM; 2013. p. 463–74.

Sahai A, Waters B. Fuzzy identity-based encryption. In: Cramer R, editor. Advances in cryptology – EUROCRYPT 2005, vol. 3494. Lecture Notes in Computer Science. Springer Berlin / Heidelberg; 2005. p. 457–73.

Sahai A, Seyalioglu H, Waters B. Dynamic credentials and ciphertext delegation for attribute-based encryption. In: Safavi-Naini R, Canetti R, editors. Advances in cryptology – CRYPTO 2012, vol. 7417. Lecture Notes in Computer Science. Springer Berlin / Heidelberg; 2012. p. 199–217.

Shamir A, Tauman Y. Improved online/offline signature schemes. In: Advances in CRYPTOLOGYCRYPTO 2001. Springer; 2001. p. 355–67.

Wang C, Ren K, Wang J. Secure and practical outsourcing of linear programming in cloud computing. In: IEEE international conference on computer communications (INFOCOM). 2011. p. 820–8.

Waters B. Dual system encryption: realizing fully secure ibe and hibe under simple assumptions. In: Halevi S, editor. Advances in cryptology – CRYPTO 2009, vol. 5677. Lecture Notes in Computer Science. Springer Berlin / Heidelberg; 2009. p. 619–36.

Zhang Y, Chen X, Li J, Wong DS, Li H. Anonymous attribute-based encryption supporting efficient decryption test. In: Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM; 2013. p. 511–16.

Zhang Y, Chen X, Li J, Li H. Generic construction for secure and efficient handoff authentication schemes in eap-based wireless networks. Comput Netw 2014;75:192–211.

Zhang Y, Chen X, Li J, Li H, Li F. Attribute-based data sharing with flexible and direct revocation in cloud computing. KSII Trans Int Inf Syst 2014;8(11):4028–49.

Zhang Y, Zheng D, Chen X, Li J, Li H. Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts. In: Provable security. Springer; 2014. p. 259–73.

Zhang Y, Zheng D, Chen X, Li J, Li H. Efficient attribute-based data sharing in mobile clouds. Pervasive Mob Comput 2016;28:135–49.

Zhang Y, Chen X, Li J, Wong DS, Li H, You I. Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. Inf Sci (Ny) 2017;379:42–61.

Zhou Z, Huang D. Efficient and secure data storage operations for mobile cloud computing, in: 8th International conference on Network and service management. IEEE 2012;37–45.

Jin Li received the BS degree in mathematics in 2002 from Southwest University. He received the PhD degree in information security from Sun Yat-sen University at 2007. He is currently 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 100 research papers in refereed international conferences and journals and has served as the program chair or program committee member in many international conferences.

Yinghui Zhang received his B.S. (2007) and M.S. (2010) from Nanchang Hangkong University and Xidian University, both in Mathematics. He got his Ph.D degree in Cryptography from Xidian University at 2013. Currently, he works at Xi'an University of Posts and Telecommunications. His research interests are in the areas of wireless network security, cloud security and cryptography.

Xiaofeng Chen received his B.S. and M.S. in Mathematics from Northwest University, China in 1998 and 2000, respectively. He got his Ph.D 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 100 research papers in refereed international conferences and journals. His work has been cited more than 4000 times at Google Scholar. He is in the Editorial Board of IEEE Transactions on Dependable and Secure Computing (IEEE TDSC), Security and Communication Networks (SCN), and Computing and Informatics (CAI) etc. He has served as the program/general chair or program committee member in over 30 international conferences.

Professor **Yang Xiang** received his PhD in Computer Science from Deakin University, Australia. He is the Dean of Digital Research & Innovation Capability Platform, Swinburne University of Technology, Australia. His research interests include cyber security, which covers network and system security, data analytics, distributed systems, and networking. He has published more than 200 research papers in many international journals and conferences. He has published two books, Software Similarity and Classification (Springer) and Dynamic and Advanced Data Mining for Progressing Technological Development (IGI-Global). He served 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 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.