MANCHESTER 1824
The University of Manchester

The University of Manchester Research

Lightweight attribute based encryption scheme for mobile cloud assisted cyber-physical systems

DOI:
10.1016/j.comnet.2018.01.038

Document Version
Accepted author manuscript

Link to publication record in Manchester Research Explorer

Citation for published version (APA):
He, Q., Zhang, N., Wei, Y., & Zhang, Y. (2018). Lightweight attribute based encryption scheme for mobile cloud assisted cyber-physical systems. *Computer Networks*. https://doi.org/10.1016/j.comnet.2018.01.038

Published in:
Computer Networks

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [http://man.ac.uk/04Y6Bo] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.

OPEN ACCESS

Download date:28. Apr. 2024

Lightweight Attribute Based Encryption Scheme for Mobile Cloud assisted Cyber-Physical Systems

Qian He[a,b,*], Ning Zhang[b], Yongzhuang Wei[c], Yan Zhang[d]

[a] *Key Lab of Cognitive Radio and Information Processing of Ministry of Education, Guilin University of Electronic Technology, Guilin 541004, China*
[b] *School of Computer Science, University of Manchester, Manchester M139PL, UK*
[c] *Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China*
[d] *Department of Informatics, University of Oslo, Oslo, Norway*

Abstract

Protecting data in Cyber-Physical Systems (CPS) has been a longstanding challenge. The tradition Attribute Based Encryption (ABE) is known for its high computation load which creates a significant challenge for resource constrained mobile devices. In this paper, we propose a Lightweight Attribute Based Encryption Scheme (LABE) for mobile cloud-assisted CPS based on a proxy service architecture and a new ciphertext policy ABE. In particular, mobile devices will solely perform symmetric encryption by joining authentication and encryption proxy services encapsulated in RESTful. Encryption shall not need pairing and then ciphertext can be decrypted with one pairing. Security analysis shows that the proposed LABE is secure with fine grained access control and users revocation capability. The computation complexity shows that the proposed scheme imposes low overhead on mobile devices and performs very well in mobile cloud-assisted CPS.

Keywords: Cyber-Physical Systems, Mobile cloud assisted Cyber-Physical Systems, Ciphertext policy attribute-based encryption, Outsourcing data security

*Corresponding author
Email address: heqian@guet.edu.cn (Qian He)

1. Introduction

A Cyber-Physical System (CPS) is tightly integrated with the control system, internet and its users. Cloud computing provides a paradigm for enabling on demand network access to a shared computing resources pool [1][2]. The Cloud-assisted CPS has become increasing popular and is already implemented as PaaS[1]. Resource constrained mobile devices are widely used in CPS for their convenient deployment and the cloud computing with super power can help to break the resource limitation. After the mobile devices in CPS and the cloud computing are combined together, the cloud assisted CPS is changed into mobile cloud assisted CPS.

In the mobile cloud assisted CPS, users' data are outsourced and the outsourced data are managed by a third party which is not fully trustworthy. The issue of preserving data security and data owners privacy is among the most challenging issues and then has raised great concerns among the cloud users, particularly for those with sensitive data[3]. How to preserve the confidentiality of data and privacy are essential requirements for CPS users.

To allow data owners to enjoy fine-grained access control for their data stored on these semi-honest cloud servers, an access control mechanism with data confidentiality protection against unauthorized external as well as internal entities is needed. Such symmetric encryption algorithms as Advanced Encryption Standard (AES) can be used but they are difficult to distribute keys to the intended data users. Identity-based public key algorithms also have their limitations in the mobile cloud for the multicasting requirement[3][4]. These limitations are, firstly, prior to encrypting any data the identities of every data user to whom the message will be delivered must known. Secondly, the data owner must know the public keys of his/her data users. Once data are encrypted, the data cannot be accessed by any new users.

Attribute Based Encryption (ABE) is intended for one-to-many encryption in which cipher texts are encrypted for those who satisfy certain conditions [4][5][6]. Recently, ABE has been imported to protect outsourced data in

cloud computing [3][7][8][9][10].ABE schemes are classified into two categories, key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). Yu et al. firstly achieved secure data access control with provable security in cloud computing using KP-ABE [10]. The authors of [3] remarked that KP-ABE can be combined with proxy encryption for achieving secure, scalable, and fine-grained data access control in cloud computing. With CP-ABE, a data encryptor does not need prior knowledge of who will be the receiver of the data. Hierarchical Attribute-based Encryption (HABE) gave an efficient CP-ABE system to share secure data in the cloud [11]. However, the size of the ciphertext and the decryption time are proportional to the number of attributes in the traditional ABE, and the computing costs affect that it is used in the cloud widely.Comparing with traditional personal computers, mobile devices are typically more resource constrained which have lower computing capability and battery power. How to decrease encryption and decryption algorithms costs of ABE to make it suitable for the mobile device efficiently is a big challenge in CPS.

In this paper, we mainly focus on our proposed Lightweight Attribute Based Encryption Scheme (LABE) for mobile cloud assisted CPS. Based on LABE, the mobile device can exploit the benefits of ciphertext polity provided by ABE. The main contributions of this paper can be summarized as follows:

(1) LABE is encapsulated in RESTful web service [12]. RESTful web services are developed following the REST principles without complex prototype standards like traditional SOAP based web services. LABE can be invoked by the resource constrained mobile device fast and easily and then the confidentiality and personal privacy of CPS users are greatly enhanced using ABE based access policy.

(2) A new lightweight CP-ABE is proposed under a proxy service architecture. Encryption does not need pairing and the decryption needs only one pairing. The high computation load in CP-ABE is offloaded partly to the cloud servers and then decreased. In addition, the communication overheads of transmitting key parameters and ciphertext are all reduced.

(3) LABE has integrating security, fine grained access control, and user

revocation. Numerical results show that the lightweigh CP-ABE algorihtm and proxy service architecture are able to greatly improve the speed of encrypting and decrypting outsourced data for the mobile cloud assisted CPS.

The remainder of this paper is organized as follows. Section 2 and 3 introduce related works and definitions about ABE. Section 4 provides design ideas, system architecture and adversary model. A lightweight and fast CP-ABE algorithm are constructed for LABE in Section 5. Such details as access policy construction, system initialization, data uploading and downloading, and restful interface for LABE are given in Section 6. Section 7 analyzes the security of proposed ABE algorithm and the whole system of LABE. Performances are evaluated in Section 8 and Section 9 conclude works finally.

2. Related work

2.1. Attribute Base Encription

Identity-based encryption (IBE) was first introduced by Shamir [13], in which the sender of a message can specify an identity such that only a receiver with matching identity can decrypt the message. In 2005, Fuzzy Identity-Based Encryption was proposed by Sahai and Waters [14], which was also known as Attribute-Based Encryption (ABE). There are two ABE schemes: KP-ABE and CP-ABE. In KP-ABE [4], a ciphertext is associated with a set of attributes, and a private key is associated with a monotonic access structure specified by using AND, OR and other threshold gates. A user can decrypt the ciphertext if and only if the access structure in his private key satisfies the attributes in the ciphertext. In CP-ABE, the ciphertext is created combining the access structure, and the private key is generated according to users' attributes [15].

In a typical ABE implementation, the size of the ciphertext is proportional to the number of attributes in the access policy and the decryption time is proportional to the number of attributes [5], [15], [16]. How to cut down the computational costs of ABE is still an open research issue. The work [16] presented a ABE system where ciphertexts can be decrypted with a constant

number of pairings, but the private key size is increased by a factor of the set of distinct attributes that appear in the private key. In paper [17] the authors proposed a CP-ABE scheme with constant size conjunctive headers and constant number of pairing operations. A no-pairing ABE scheme based on elliptic curve cryptography (ECC) was proposed in [18] to address the security and privacy issues in Internet of Things, which is not based on the bilinear map theory.

2.2. Outsourcing Data Protection in Cloud Computing

Data stored in the cloud storage is managed by a third party, which means that the data owner and keeper are in a different domain.

The work presented in [19] adopts traditional symmetric key cryptographic system to encrypt data to protect the data against the untrusted server (i.e. the data manager). In this solution, the data were classified into a file-group with similar access control lists, and then each file-group is encrypted with a symmetric key. The work proposed in [20] combined a symmetric key with a public key crypto system.

Benaloh et al. proposed to use hierarchical identity-based encryption (HIBE) [21] combined with a searchable encryption method to protect electronic health record over any third party storage device like cloud storage [22]. This solution requires each patient (i.e. data owner) and healthcare provider to create and manage multiple keys. Also the solution does not have an efficient user revocation mechanism.

Recently there have been increasing efforts on devising solutions to protect the confidentiality of data and privacy of data owners[10][23][24][25].In 2010, Yu et al. first used ABE to realize a scalable and fine-grained data access control scheme for cloud computing [10]. In this scheme, a data owner can delegate most of the computation task, such as user revocation, to the cloud server. CP-ABE is more popular than KP-ABE in cloud computing context for its better features in the data access management, as, in CP-ABE, an access policy is associated to a ciphertext, so data sharing is possible without prior knowledge of who will be the receiver preserving the flexibility of the cloud. Narayan et al [23]

designed a secure and privacy preserving electronic health record (EHR) system based on CP-ABE by which users could share their health data discriminately by encrypting data items using respective attributes assigned to different users. Ref. [24] proposed a multi-authority CP-ABE scheme with accountability, which allowed tracing the identity of a misbehaving user who leaked the decryption key to others. The paper [25] focused on providing a dependable and secure cloud data sharing service that allows users dynamic access to their data by utilizing CP-ABE combined with identity-based encryption (IBE) techniques. All these schemes proposed to protect outsourced data in the cloud are based on the basic CP-ABE method given in [5], [15], which involves expensive pairing operations and the number of such operations grows with the complexity of the access policy used, so they are not suitable for the resource constrained mobile.

To reduce ABE computational costs, the work [26] introduced methods for online/offline encryption and key generation. The idea is to shift the computational task of encryption and key generation to an offline phase, thus spreading the the computational cost over a longer period of time. To allow CP-ABE to be used in ARM based mobile devices and speed up the executions of ABE in the devices, the authors modified the original model of ABE with outsourced decryption so that some of the computationally expensive tasks are moved from the mobile device to a proxy [27].

3. Definitions

Definition 1. *(Bilinear Maps [4]): Let G and G_T be two multiplicative cyclic groups of prime order p. Let g be a generator of G and e be a bilinear map,$e : G \times G \rightarrow G_T$. The bilinear map e has the following properties:*

1.Bilinear: for all $u, v \in G$ and $a, b \in Z_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$;

2.Non-degeneracy: $e(g, g) \neq 1$;

3.Computability: there is an efficient algorithm to compute $e(u, v)$ for any $u, v \in G$;

Definition 2. *(Access Structure [26]): Let $p_1, p_2, \ldots, p_n$ be a set of parties.*

A collection $A \subseteq 2^{\{p_1,p_2,\ldots,p_n\}}$ *is monotone if* $\forall B, C$ *: if* $B \in \mathbb{A}$ *and* $B \subseteq C$ *then* $C \in \mathbb{A}$*. An access structure is a collection* $\mathbb{A}$ *of non-empty subsets of* $\{p_1, p_2, \ldots, p_n\}$*, i.e.,*$\mathbb{A} \subseteq 2^{p_1,p_2,\ldots,p_n} \setminus \{\phi\}$ *. The sets in* $\mathbb{A}$ *are called authorized sets, and sets not in* $\mathbb{A}$ *are called unauthorized sets.*

Definition 3. *(Linear Secret-Sharing Schemes (LSSS) [10]): a secret-sharing scheme* $\prod$ *over a set of parties* $\mathcal{P}$ *is called linear over* Z_p *if*

1.The shares for each party form a vector over Z_p *.*

2.There exists a matrix R *with* l *rows and* n *columns called share-generating matrix for* $\prod$*. For all* $i = 1 \ldots l$*, the* i*th row of* R *we let the function* ρ *defined the party labeling row* i *as* $\rho(i)$*. When we consider the column vector* $v = (s, y_2, \ldots, y_n)$*, where* $s \in Z_p$ *is the secret to be shared, and* $s, y_2, \ldots, y_n \in Z_p$ *are randomly chosen, then* $R \cdot v$ *is the vector of* l *shares of the secret* s *according to* $\prod$*. The share* $(R \cdot v)$ *belongs to party* $\rho(i)$*.*

Suppose that $\prod$ is a LSSS for an access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorised set of attributes, and let $I \subset \{1, 2, \ldots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, there exist constants $\{w_i \in Z_p\}_{i \in I}$. Therefore, if $\{y_i\}$ are valid shares of any secret according to $\prod$, then $\sum_{i \in I} w_i y_i = S$ (see [6] for how these constants are computed).

Referring to [4][5], a monotonic access structure can easily be converted into an LSSS representation. When calculating w_i, we use $(1, 0, \ldots, 0)$ as the "target" vector for any linear secret sharing. The target vector should be in the span of I for any satisfying set of rows I in M, and not be in the span of I for any unsatisfied set of rows I. Actually, there will be a vector w such that $w(1, 0, \ldots, 0) = -1$ and $wR_i = 0$ for all $i \in I$.

Assumption 1. *(Discrete Logarithm Assumption): Suppose a group* G *of prime order* p *is chosen according to the security parameter. Given* $h \in G$*, compute* $r \in Z_p$ *such that* $h = g^r$*.* r *is called the discrete logarithm of* h *with respect to base* g*, and written as* $log_g h$*.*

The discrete logarithm assumption holds for G if for all non-uniform PPT algorithms $\mathcal{B}$, $P_r[\mathcal{B}(H) = log_g h]$ is negligible.

Assumption 2. *(decisional Bilinear Diffie-Hellman (BDH) Assumption): Suppose a challenger chooses*$a, b, c, z \in Z_p$ *randomly. The decisional BDH assumption is that there is no polynomial-time adversary who distinguishes the tuple*$(A = g^a, B = g^b, C = g^c, Z = e(g, g)^z)$ *from the tuple with more than a negligible advantage.*

Definition 4. *We say that the decisional BDH assumption holds if no polynomial time algorithm has a non-negligible advantage in solving the decisional BDH problem.*

Besides, the main notations used in this paper are summarised in Table 1.

Table 1: Notation definitions

Notation	*Description*
MD	*Mobile Device*
CS	*Cloud Storage*
AS	*Authentication Service*
EPS	*Encryption Proxy Service*
PK, MK	*System public key and master key*
M, M_0, M_1	*Plaintext message*
CT	*Ciphertext generated from CP-ABE*
A	*LSSS Matrix*
S	*Attributes of a decryptor*
I	*Attributes in access policy*
ENC/DEC	*Symmetric encryption or decryption*
FID	*Identity of the protected data*
SID	*Session identity for encryption/decryption*
SSK	*Session symmetrical key*
SK	*Symmetrical key*

4. Architecture

4.1. Design Ideas

LABE should be a secure and scalable scheme to provide CP-ABE services with fine-grained access control for mobile cloud assisted CPS. The main design goals are as follow:

1) LABE provides a convenient and light weight method for the mobile to invoke CP-ABE operators which support the fine-grained access control to upload and download data in the cloud environment.

2) LABE should be secure and fast enough, and the resource constrained mobile can do the CP-ABE operations using our scheme.

3) Multiple distributed roles are used to separate security into different parties, and each role is just responsible for part functions to balance loads.It is helpful for the data owner to avoid attacking hazard from the LABE manager. The outsourced data are kept in such general cloud storage as Dropbox, Baidu Cloud Storage, and so on.

4.2. System Architecture Design

LABE has two layers: mobile user layer and cloud service layer. In the mobile user layer, the resource constrained mobile devices (MD) does work in CPS, and the encryption and decryption algorithms of ABE are invoked to protect their outsourced data to be stored in the cloud. In the cloud service layer, $CloudStorage(CS)$, the $AuthenticationService(AS)$, and the Encryption Proxy Service (EPS) work together to provide ABE functions in Restful services. The architecture of LABE is illustrated in Figure 1, where four basic roles (MD, CS, AS, and EPS) are presented in the mobile user layer and the cloud service layers respectably. MD works in the mobile user layer while the other three are belong to the cloud service layer.

MD represents a data owner or a data consumer. As a data owner, MD defines an access control policy for the data to be uploaded onto a CS. To upload data, the owner first encrypted the data by using a symmetric encryption

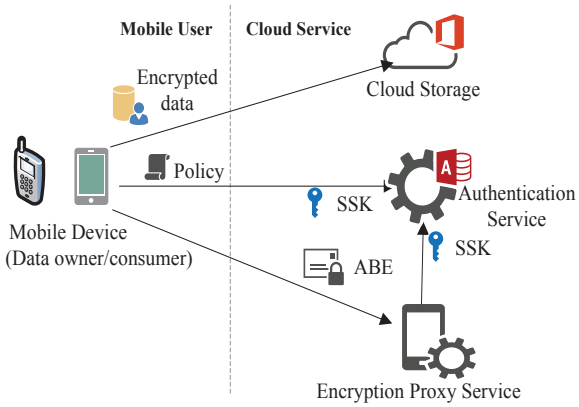


Figure 1: LABE architecture

algorithm as AES with the Symmetric Key (SK). SK is encrypted using CP-ABE, and the resulting ciphertext is uploaded onto CS with the symmetric encrypted data. SK used in the data encryption is generated by the data owner and can be accessed by MD with the satisfied attributes. When MD is a data consumer, it downloads the ciphertext from the corresponding CS, and then decrypt it with the help of AS and EPS.

AS stores the information identifying (MD) and generates Linear Secret-Sharing Schemes matrix based on the attributes of MD. When MD wants to encrypt/decrypt data, AS authenticates the user and then check if the user is allowed to access LABE based on MDs identifying information. If positive, AS generates a Session Symmetric Key (SSK) for MD and then tells EPS to secure the communication channel using (SSK) in the later. Secure Sockets Layer (SSL) is used to distribute SSK.

EPS offers encryption and decryption services to MD. With the cooperation of AS, EPS delegates all CP-ABE algorithms for MD. In this way, CP-ABE encryption and decryption operations do not have to be executed on users' MDs. If MD satisfies the specified access policy attached to the ciphertext downloaded from CS, AS and EPS can jointly perform the decryption of

SK. EPS can be provided by multiple proxy servers with high performance.

AS and EPS are implemented in the form of Restful web services. Actually, AS and EPS may be viewed as encryption/decryption cloud services. In order to separate the working function and enhance the security, the AS and EPS are not deployed in the same security domain as the CS. So, though CS has the data uploaded by MD, CS does't know the decryption way. On the other hand, AS and EPS help to do the encryption and decryption operations, but the outsourced date are not need to be sent to AS and EPS.

4.3. Adversary model

In LABE, adversaries come from external or any of the internal entities, including MD, CS, AS, and EPS. We assume that MD is untrustworthy, and CS, AS and EPS are semi-trusted, in that entities behave honestly according to the functional design, but, in certain situations, these entities may try to acquire private information from the users data for their profits. The semi-trusted adversary model is weaker than the malicious model, but it is commonly used in related work designs [16],[20]. Threats imposed by the four groups of entities in our model are as follows:

1)MD is untrustworthy. MD is the main source of threats normally. MD may want to download and decrypt data for which they do not have access privileges, and multiple MDs may collude with each other or with CS to break the encryptions to gain unauthorized access to data.

2)CS provides storage services for the data owner. CS may sniff the data of MD and try to gain plaintext from MD uploaded data for the business profit. Besides, CS may collude with MD, but it is assumed that CS cannot collude with AS and EPS.

3)AS and EPS collectively provide reliable ABE services to decrypt the ciphertext of SK. After getting SK, the data requester can decrypt the ciphertext in CS. AS and EPS may collude with MDs that have no access privileges for the corresponding CS.

5. CP-ABE Algorithm Construction

5.1. Algorithm Design

A new lightweight CP-ABE is designed specially for LABE. In the special CP-ABE algorithm, MD does not keep the private key under proxy architecture and it becomes more flexible to deploy MD in CPS. *KegGen* is eliminated and the access policies and user attributes description are kept in AS. The LSSS access matrix R is taken as input and a random exponent $s \in Z_p$ is distributed according to R. The algorithms of special CP-ABE are constructed as follows based on the bilinear map [5].

$\boldsymbol{Setup \rightarrow (PK, MK)}$: the setup algorithm chooses a bilinear group G of prime order p, a generator g, and then selects a random number $\alpha \in Z_p$. The algorithm outputs the public parameter PK and the master key MK as:

$PK = (G, p, g, e(g, g)^{\alpha}), MK = \{PK, g^{\alpha}\}$.

$\boldsymbol{Encrypt(PK, M, \mathbb{A}) \rightarrow CT}$: the Encrypt algorithm takes the public parameter PK, a plaintext message M, and an LSSS access structure $\mathbb{A}(\mathbb{R}, \rho)$ on attributes U as input. In $\mathbb{A}(\mathbb{R}, \rho)$, the function ρ associates rows of A to attributes.

Let R be an $l \times n$ matrix. The algorithm first chooses a random vector $\vec{v} = \{s, y_2, \ldots, y_n\} \in Z_p^n$. These values will be used to share the encryption secure exponent s. For, $i = 1$to l it calculates $\lambda = \vec{v} \cdot R_i$, where M_i is the vector corresponding to the ith row of R. The ciphertext CT is generated by:

$CT = (C = Me(g, g)^{\alpha s}, \{C_i = g^{\lambda_i}\}_{i=\rho(i)})$.

$\boldsymbol{Dencrypt(CT, MK, S) \rightarrow \{M, \perp\}}$: The Decrypt algorithm takes a ciphertext CT as input for a LSSS access structure $\mathbb{A}(R, \rho)$ and the decryptors attribute set, S. The algorithm first checks whether or not S satisfies the access structure. It returns $\perp$ if does not satisfy the access structure. Suppose that S satisfies the access structure and lets $I \subseteq 1, 2, \ldots, l$ be defined as $I = i : \rho(i) \in S$. The algorithm next lets $\{w_i \in Z_p\}_{i \in I}$ be a set of constants so that if $\{\lambda_i\}$ are valid shares of any secret a according to W , then $sum_{i \in I} w_i \lambda_i = s$. There could potentially be several different ways to get $\{w_i\}$ in polynomial time [4] [5][10]

and for unauthorized sets, no such constants, $\{w_i\}$ exist.

The decryption algorithm outputs either the plaintext M, when the collection of attributes S satisfies the access structure $\mathbb{A}$, or $\perp$ when decryption fails. The decryption algorithm recovers the value $e(g,g)^{\alpha z}$ by computing:

$e(g^{\alpha}, \prod_{i=1}^{I}(C_i)^{w_i}) = e(g^{\alpha}, g^{\sum_{i\in I} w_i \lambda_i}) = e(g,g)^{\alpha s}$.

The decryption algorithm can then divide out this value from C and then obtain the plaintext, i.e. $M = C/e(g,g)^{\alpha s}$.

5.2. Complexity Analysis

In the special CP-ABE algorithm, the encryption algorithm requires one pairing and $|l|$ exponentiation operations, and the decryption algorithm just needs one pairing and $|I|$ exponentiation operations. It is better than the traditional CP-ABE proposed in [5] obviously in the transmitting bandwidth complexity and the computation complexity.

Firstly, the transmitting bandwidth and computation complexity of our CP-ABE is compared with multiple typical CP-ABE in [5][16][21][28][29]. The transmitting bandwidth and computation complexity comparisons are illustrated respectively in Table 2 and 3.

Table 2: Comparison of transmitting bandwidth complexity

Scheme	*Public parameter*	*Ciphertext*
Our scheme	$4l$	$(k+1)l$
[5]	$(n+3)l$	$(2k+2)l$
[16]	$(n+3)l$	$(2k+2)l$
[25]	$(n+1)l$	$(3k+1)l$
[28]	$(2n+5)l$	$(3k+2)l$
[29]	$(6n+2)l$	$4l$

Table 3: Comparison of computation complexity

Scheme	*Scheme*	*Decryption*
Our scheme	$(1+k)\tau_e$	$\tau_p + m\tau_e$
[5]	$(1+2k)\tau_e$	$(m+1)\tau_p + m\tau_e$
[16]	$(1+2k)\tau_e$	$2\tau_p + 2m\tau_e$
[25]	$5k\tau_e$	$2m\tau_p + m\tau_e$
[28]	$(k^2+3k+2)\tau_e$	$(3m+1)\tau_p + m\tau_e$
[29]	$3\tau_e$	$2m\tau_p$

where, l denotes a CP-ABE security level corresponding to bit number of a group element, $k$denotes the size of an access formula, n the size of the attribute space, and m the number of users' attributes. The time costs of computing a bilinear pairing and exponentiation are denoted by τ_p and τ_e respectively. Although τ_p and τ_e are two main cost in CP-ABE algorithms, we note that the cost time of multiple multiplying operations may equals to one τ_e and $\tau_e < \tau_p$.

From Table 4, it can be seen that the complexity of public parameter is a constant $4l$ in our special CP-ABE, whereas, the basic CP-ABE [5] and other CP-ABE [16][25][28][29] have public parameter linear with the attribute numbers. For ciphertext size, our scheme is $(1+k)l$ which is also better than the CP-ABE schemes in [5][16][25][28]. As shown in Table 5, the encryption algorithm needs no pairing and the decryption algorithm only needs one paring in our special CP-ABE. Although the exponentiation of our scheme is linear with the size of an access formula in the encryption stage, it is still better than [5][16][25][28]. Ref [29] has constant encryption complex: $3\tau_e$, but it's decryption complex is $2m\tau_p$, worse than our scheme: $\tau_p + m\tau_e$. Notice that the encryption algorithm is operated once, decryption is invoked by MD in the accessing every time. So, the better decryption performance is more important than encryption in the real application systems. It is most important that our decryption computation complexity is fastest in all CP-ABE algorithms.

6. Realization and Application

6.1. Access Policy

An access policy can be expressed by an access tree, T, with AND and OR gates by using, respectively, 2-of-2 and 1-of-2 threshold gates. A user, u, is permitted to access the data, if and only if the attributes of u satisfies T which is attached to the encrypted data [5]. The access tree used is converted into a boolean formula. We extend the boolean formula to include an attribute element, which is expressed as "name: required value". For example, if we allow a PhD or a Master student at the School of Computer Science, University of Manchester to be an legitimate user in T, the boolean formula can be specified as follows.

A = Student:PhD, B = Student: Master, C=School: Computer Science, D = University: University of Manchester; (A OR B) AND C AND D

In LABE, LSSS structure $\mathbb{A}(R, \rho)$ is derived from Boolean formula using method in [30]. The above Boolean formula can be converted into the LSSS matrix R as follows:

$$R = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

ρ represents the map between the Boolean formula to LSSS matrix. Where, $R_1 = A, R_2 = B, R_3 = C, R_4 = D$, and R_1, R_2, R_3 and R_4 are the first, second, third, and fourth row of R.

MD specifies an access tree when it submits data to a cloud storage before the data are encrypted. As discussed earlier, the policy is presented by using the Boolean formula. The corresponding LSSS is generated by AS. In the LSSS matrix R, the number of rows will be the same as the number of leaf nodes in the access tree. Pieces of R are a vector over the finite field, and the secret will be hidden in the access structure and can be reconstructed using a linear

combination of the pieces. MD having the corresponding attributes can decrypt the ciphertext correctly. In our LABE, the corresponding attributes of MD is generated by AS automatically in the decryption process.

6.2. System Initialization

System initialization is necessary to set up and put system parameter values in AS and EPS, which is run before MDs make any request. New EPS components may be added dynamically after the initialization step is completed. The system initialization processes as follows:

(1) AS chooses a large prime order p and a group generator g, and run the setup algorithm of $Setup \rightarrow (PK, MK)$. The public parameter, PK, and a master key, MK are generated and stored in AS;

(2) AS distributes PK and MK to all EPSs using the secure channel;

(3) EPSs save PK and MK in the configuration file for the future encryption and decryption process.

If a new EPS server should be added, the new EPS actively initiate itself by requesting PK and MK from AS.

6.3. Data Encryption and Uploading

The data encryption service of LABE is invoked when a data owner wants to outsource data to CS. Each MD has a valid user identifier (UID) registered in AS,MD knows the URL of AS, EPS and CS. The interfaces of AS and EPS are encapsulated in Restful Web services and the channel between MD and EPS and the channel between AS and EPS are based on SSL. The data encryption and uploading operation of MD involves four steps as follows:

Step 1: The data owner specifies an access policy (*policy*), described by the monotonic Boolean formula, and sends a data uploading request to AS having UID (the users identifier), FID (a unique identifier for the data) and *policy*. The data with FID will be sent to other MD and used to identify an encrypted data item from CS when an entity wishes to download any data later on. If MD is valid (i.e. if the authentication outcome is positive), AS generates the

Table 4: Encrypted Storage Structure

FID	L_{CT}	CT	$ENC(D, SK)$

monotonic Boolean formula and converts it into an LSSS representation using the standard converting technology given in [30]. A session identity (SID) and a session symmetric key (SSK) are also generated. The session key will be used for MD and EPS to establish a secure communication channel between the two entities instead of SSL in the later. AS then saves the data generated, including SID,FID and LSSS, in its database, and sends a response message containing (SID, SSK), to MD.

Step 2: MD generates a private symmetric key (SK) randomly, and encrypts (SK) using SSK: $ENC(SK, SSK)$. The encrypted data request message containing SID and $ENC(SK, SSK)$ is submitted to EPS. The original data (plaintext) D is encrypt by the symmetric encryption algorithm using SK, and the result $ENC(D, SK)$ is kept in MD to be uploaded in Step 4.

Step 3: EPS, upon the receipt of the request from MD, uses SID to request LSSS and SSK from AS. This is because EPS needs to know the session key SSK to get the plaintext SK that should be encrypted by CP-ABE and the access policy before being able to generate the ciphertext policy, CT. Once EPS has obtained these items from AS. It executes the fast CP-ABE encryption algorithm given in Section 5 to generate ciphertext, i.e.$ENC(PK, M, A) \rightarrow CT$ and then sends CT to MD.

Step 4: Upon the receipt of ciphertext CT from EPS, MD appends CT with the encrypted data $ENC(D, SK)$ and sends both to CS. Of course, how to upload data to CS is private secret for MD itself.

After CS receives the uploaded data, the encrypted data are stored in CS as Table 4. Where, L_{CT} (4-bytes long) indicates the length of CT.

6.4. Data Downloading and Decryption

Assuming that MD knows URL of an accessed file about FID in CS, the data downloading and decryption for MD consists of four steps as follows:

Step 1: MD sends a request for the file by sending the file identity, FID, to CS, and the server responds to the requestor by sending a response message. The response message is: $L_{CT} + CT + ENC(D, SK)$ where L_{CT} is the length of CT, CT is the encrypted SK by CP-ABE and $ENC(D, SK)$ is the encrypted data.

Step 2: MD authenticates itself in AS, and the step may be performed by using an existing protocol such as SSL. If MD is a legitimate user, AS will generate (SID) and (SSK). After checking requesters information of SID, UID, SSK stored in AS's database, AS sends a response message, (SID, SSK), to MD.

Step 3: MD requests EPS to acquire SK and decrypts the encrypted data received in step 1. The request message contains SID and $ENC(FID + CT, SSK)$. In order to calculate SK, EPS, LSSS and the user attributes, S, should be known. EPS makes the second request to AS. $LSSS$ is queried according to FID, and then S is generated based on the corresponding MDs. UID is linked to SID. Obtaining LSSS and S, EPS decrypts CT using $Decrypt(CT, MK, S) \rightarrow \{M, \perp\}$.

Step 4: The outcome of this decryption is SK, and then SK is encrypted using SSK to be send to MD. After decrypting $ENC(SK, SSK)$, MD gets SK to be used to decrypt $ENC(D, SK)$ using symmetric algorithm, obtaining the plaintext D.

6.5. Restful web services interfaces of CP-ABE

To make the encryption and decryption services of LABE easily accessible by MD, all the interfaces of AS and EPS are encapsulated in Restful web services. Using Restful web services, CP-ABE algorithms become network resources, and each resource can be identified via a Uniform Resource Locator (URL). Table 5 summarizes the LABE CP-ABE service resources of AS and EPS.If MDs want to use any of the LABE services, they can simply access the services via their corresponding URLs using HTTP protocol directly.

Table 5: CP-ABE Service Resources of LABE

Provider	*Resources*	*URL: http://.../*	*HTTP action*
AS	Generate *LSSS,SSK*	/auth/encrypt/generate_lsss	POST:*UID,FID,Policy*; GET:*SID,SSK*
	Query *LSSS,SSK*	/auth/encrypt/query_lsss	POST:*SID*; GET:*LSSS,SSK*
	Generate *SSK*	/auth/decrypt/generate_ssk	POST:*UID*; GET:*SID,SSK*
	Query *SSK*	/auth/decrypt/query_ssk	POST: *SID*; GET:*SSK*
	Query *LSSS,S*	/auth/decrypt/query_lsss	POST:*FID*; GET:*LSSS,S*
EPS	Initialize *EPS*	/proxy/initialization	POST:*EPSID*; GET:*MK*
	Encryption	/proxy/enc	POST:*SID,ENC(SK,SSK)*; GET:*CT*
	Decryption	/proxy/dec	POST:SID, ENC((FID+CT),SSK); GET:ENC(SK,SSK)

7. Security Analysis

7.1. CP-ABE algorithm Security

Theorem 1. *Supposing the decisional BDH assumption holds, no polynomial adversary can selectively break LABE.*

Proof. Assuming there exists an adversary $\mathcal{A}$ with non-negligible advantage $\epsilon = Adv_A$ that causes the selective chosen-plaintext attack security game against our construction shown in Section 5. A simulator $\mathcal{B}$ that plays the decisional BDH problem with $\mathcal{A}$ as follows:

Initiation: The simulator $\mathcal{B}$ takes in the decisional BDH challenge g, g^a, g^b, g^s, $T = e(g,g)^{abs}$. The adversary gives the algorithm and the challenge access structure $\mathbb{A}(R^*, \rho^*)$, where R^* has $|R^*| = n_{max}$, where n_{max} is the maximum number of columns in the system and it is specified by the attacker.

Setup: The simulator chooses a random number $\alpha^{'} \in Z_p$ and implicitly sets $\alpha = ab + \alpha^{'}$ by letting $e(g,g)^{\alpha} = e(g^a, g^b)e(g,g)^{\alpha^{'}}$.

Phase 1: In this phase the simulator does not answer private key queries as in the case in [5], as, in our special CP-ABE scheme, $KeyGeneration$ algorithm does not exist, so the adversary just chooses an attribute set, S, where S does not satisfy M^*.

Challenge: The challenge ciphertext is built for the adversary. The adversary gives two messages M_0, M_1 to the simulator firstly. The simulator flips a coin β. It creates $C = M_\beta Te(g^s, g^{\alpha'})$. The simulator intuitively chooses random $y_1, \ldots, y_{n^*}$ in Z_p and the share the secret using the vector $\bar{v} = \{s, y_1, \ldots, y_n\} \in Z_p^{n^*}$

For $i = 1, \cdots, n^*$, C_i in the challenge ciphertext is generated as

$C_i = g^{A^*_{i,1}s + A^*_{i,2}y_2 + \cdots + A^*_{i,n^*}y_{n^*}}$.

Phase 2: The same as **Phase 1**.

Guess the adversary will eventually output a guess β' of β . The $\mathcal{B}$ simulator outputs 0 to guesses $T = e(g,g)^{abs}$ if $\beta' = \beta$; otherwise, it output 1 to indicate that it believes T is a random group element in G_T.

When T is a tuple the simulator $\mathcal{B}$ gives a perfect simulation so we have that $Pr[\mathcal{B}(\vec{y}, T = e(g,g)^{abs}) = 0] = \frac{1}{2} + Adv_A$.

When T is a random group element, the message M_β is completely hidden from the adversary and we have $Pr[\mathcal{B}(\vec{y}, T = e(g,g)^{abs}) = 0] = \frac{1}{2}$. Therefore, the simulation algorithm $\mathcal{B}$ can play the decisional BDH game with non-negligible advantage.

□

Theorem 2. *Algorithms of LABE has forward security property.*

Proof. The special CP-ABE system is secure even though some valid pairings of plaintext and ciphertext are obtained by the adversary. Since the security index s is random each time and then the adversary cannot decrypt a new plaintext through the decrypting algorithm: $M = C/e(g,g)^{\alpha s}$. Besides of the valid plaintext and ciphertext pairing is leak, we discuss two cases where the master key or the random security s is leak as follows.

(1) Assume the master key $MK = \{PK, g^\alpha\}$ is leak. Since s is random, the adversary cannot decrypt a ciphertext without a valid key obviously.

(2) Assume a current random security s is leak. Using the leak plaintext and ciphertext pairing (M_1, C_1), the adversary just could calculate $e(g,g)^\alpha$

from $[e(g,g)^a]^s = C_1/M_1$. However, she still cannot know α for the Discrete Logarithm Assumption complex and cannot decrypt an another ciphertext.

Therefore, the special CP-ABE system has forward security property.

□

7.2. System Integrated Security

The security level of LABE is dependent on the security levels of the building blocks used in the service design. These building blocks include a symmetric-key encryption algorithm, e.g. AES, channel security protocol, SSL, and our proposed CP-ABE.

All the communication channels linking any pair of entities in the LABE architecture are authenticity and confidentiality protected using SSL. Since the authentication procedure and the keys used in protecting the channel security are secure, the session symmetric encryption key SSK is distributed securely and then the communication between MD and EPS will also be secure. When MD wants to upload data to AS, MD will symmetrically encrypt the data by using a random private symmetric key, SK. SK is encrypted by CP-ABE provided by AS and EPS. at last MD will get the corresponding CP-ABE ciphertext CT from EPS.

Data are encrypted before being uploaded onto the cloud storage server, CS. The key materials required to compute the decryption key are managed by a third party, AS. Applying cryptographic algorithms to protect the outsourced data uploaded from MD in CPS, we separate duties of storing CS, AS and EPS. The outsourced data are kept in CS, but the data are ciphertext. The entities including AS,EPS may know the part information of decryption keys, but they do not have the access to CS. The plaintext is just decrypted by the MD satisfying the access policy.

In the LABE system, the outsourced data are blinded with $e(g,g)^{\alpha s}$ where s is a random value and split into a vector $\{\lambda_i\}$. The user who wants to obtain the data M must recover $e(g,g)^{\alpha s}$ by computing $e(g^{\alpha}, \prod_{i=1}^{I}(C_i)^{w_i})$, which requires $\{w_i\}$ being calculated by using the attributes set, S and LSSS structure $\mathbb{A}(R,\rho)$

. To achieve this, MD must have the correct set of attributes satisfying the access policy specified by AS. In other words, if MD has a matching set of attributes, a correct value of $e(g,g)^{\alpha s}$ can be calculated, so the data decryption key is recovered and then the data stored on CS are decrypted. Otherwise, the values of $\{w_i\}$ cannot be calculated and a null result is returned from the decryption algorithm. MD is tied with a set of attributes granted by AS, and the set of attributes is transferred from AS to EPS, where SK is recovered. In other words, a users attributes are not handed out to the user, rather they are kept by AS, so different users cannot compromise a data decryption by combining the attributes of different MD in LABE.

Theorem 3. *LABE is indistinguishable under chosen plaintext (IND-CPA) attack.*

Proof. Considering that there is an adversary, *adv*, and a challenger, we discuss the IND-CPA attack for LABE. At first, the challenger generates PK and MSK, and publish PK to *adv*. *adv* submits two distinct plaintexts M_0 and M_1 to the challenger. *adv* tries to distinguish the input message based on the corresponding ciphertext: $M_0 e(g,g)^{\alpha s_0}$, $M_i e(g,g)^{\alpha s_1}$.

Since s_0 and s_1 are random each time in the CP-ABE of LABE, $M_0 e(g,g)^{\alpha s_0}$ and $M_i e(g,g)^{\alpha s_1}$ may have the same value, even M_0 is not equals to M_1. According Theorem 1, the challenger should not predict what will happen to the cipher by changing the plaintext, and we cannot get any information about the input message from the ciphertext.

In addition, αs is difficult to be calculated from $e(g,g)^{\alpha s}$ based on the discrete logarithm assumption. Assuming s is obtained from the CT when a satisfied attribute is known, the security of α is assured as the exponent αs cannot be computed from $e(g,g)^{\alpha s}$.

Therefore, it does not afford any negligible advantage for *adv* to distinguish the ciphertexts of M_0 and M_1. □

7.3. Fine-grained access control

LABE system can provide fine-grained access control services to MD of CPS based on CP-ABE. The fine-grained features is inherited from the features of CP-ABE, where each MD is assigned with a set of attributes by AS and the access policy is defined by the owner of outsourced data.

Supposing an access policy for data has an required attribute $i \in I$, the attribute has a corresponding row A_i in the LSSS structure $\mathbb{A}(A, \rho)$. If attribute i is required in the access policy but a mobile device, MD_j, does not have this attribute, MD_j would not have a correct $\{w_i\}$ to match with $A \cdot w^T = \{1, 0, \ldots, 0\}$, and then the decryption procedure will fail. MD_j cannot calculate $e(g, g)^{\alpha s}$, and MD_j cannot recover the private symmetric key to decrypt the encrypted data in CS. Therefore, the user without the corresponding attribute defined in access policy can not decrypt the data, and the privileges is attached with attribute item by specifying access policy.

7.4. Privilege Revocation

A data owner may revoke a consumer's access privilege granted earlier, and it is essential to withdraw a user's privilege in the outsourced data protection of CPS. The traditional CP-ABE algorithm does not provide a revocation function, and some CP-ABE algorithms revoke privilege by re-encrypting the file. Normally, once a user has obtained a private key for an encrypted file, the key is valid until the file is re-encrypted by a new access policy. In LABE, the revocation of a MDś privilege does not require the re-encryption for the outsouced data, rather the system only need to change the attributes assigned to the MD in AS.

When a MD wishes to decrypt an encrypted symmetric key so as to recover encrypted data, the user needs a matching set of attributes assigned by AS. So, if the data owner wants to revoke the access privilege of MD, s/he can tell AS to change the corresponding $MD's$ attributes. AS is a trusted third party and it is always online providing these services, the re-encryption is not needed any way. Evidently, it is more efficient for AS to modify the attributes in AS. After

the attributes modified, a MD does not have a matching set of attributes, the user cannot compute $\{w_i\}$ and $A \cdot w^T = \{1, 0, \ldots, 0\}$, thus cannot obtain the symmetric key to decrypt the original data again. This AS-based revocation mechanism is light weight, and introduces very little additional load into the system. In addition, if the data owner want to change the access policy about outsourced data, it also can re-encrypt the outsourced data in CS.

8. Experiment Analysis

In this section, we realize LABE system and construct experiments to evaluate the performances including encryption and decryption.

8.1. Experiment Environment

LABE system with MD, AS and EPS is realized using java language. The protected data owned by MD can be uploaded to any CS system such as Dropbox, Zip cloud, Baidu cloud. CS environments will not be discussed in our experiments, since CS may not affect the encryption and decryption performance evaluation on MD.

The experiments are carried out using a local network with 100M bandwidth and 150M WiFi wireless network for MDs. Table 6 gives the main devices configuration including AS, EPS, MD and wireless network router. AS is run on a Sugon W5801-G10 server with two Intel Xeon E5-2630 CPUs (6 cores, 2.3G HZ) and 64G Memory. EPS shares the same machine with AS. The interfaces of AS and EPS are encapsulated in the Spring Restful web services framework. A smart mobile Huawei Y635-CL00 with Snapdragon 410 (MSM8916) and 1G memory is used as MD to invoke CP-ABE services. The operation systems of AS and MD are Windows 7 Ultimate and Android 4.4.4 respectively, and Java JDK version is 1.7.

In the CP-ABE services experiments, we consider three cases for the mobile device: using services with our proposed LABE (LABE), using services with the basic CP-ABE algorithms proposed in [5] (CP-ABE Service), and using the

basic CP-ABE directly (CP-ABE on MD). The performance affections about data sizes and attribute numbers will also be disscussed.

Table 6: Devices' configuration

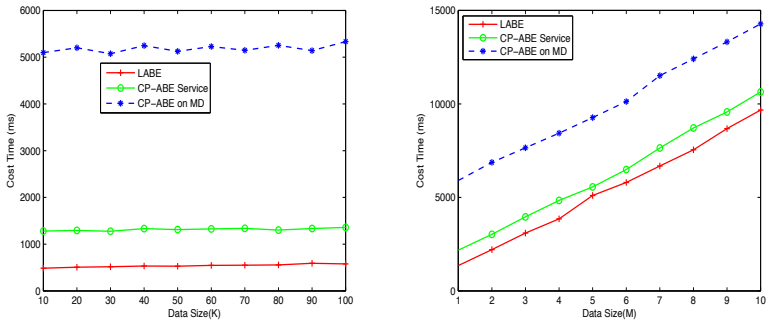
Component	*Device Type*	*Configuration*
AS	Sugon W5801-G10	Xeon E5-2630; 64G Memory
EPS	Sugon W5801-G10	Xeon E5-2630; 64G Memory
MD	Huawei Y635-CL00	Snapdragon 410; 1G Memory
Router	TPLink WR742N	802.11n 150M WiFi

8.2. Encryption

Considering the typical CPS application scenes, outsourced data sizes are randomly between 10K to 10M bytes. The data are separated into two categories, one is small size (10–100 K bytes), and the other is media size (1–10 M bytes). The total time is the time for MD to finish the whole procedures of encrypting data defined Section 6.3, which includes CP-ABE encryption of sk and symmetric encryption of original data using sk. 10 attributes are used in CP-ABE and 8 attributes are specified in the access policy.

Figure 2 shows the total time of encrypting data with various sizes under LABE,"CP-ABE Service", and "CP-ABE on MD". After utilizing the proxy based service architecture, the encryption speed of LABE and "CP-ABE Service" is much faster than that of "CP-ABE on MD". Especially, the total time of LABE is about 25% of that of "CP-ABE on MD" in Figure 2.a, and the total time of "CP-ABE on MD" is about 3700 ms (million second). LABE works always the best since the proxy architecture and the special lightweight CP-ABE are used.

The average decreased total time of LABE is about 59% and 13.8% of that of "CP-ABE Service" in the cases of small size and medium size respectively. Without the proxy service architecture, the basic CP-ABE works on mobile device very slowly, and the encryption time of "CP-ABE on MD" reaches to about 5 seconds. The ratio of CP-ABE encryption time in the total is over



(a) Data with small size (b) Data with medium size

Figure 2: Total time of encrypting data with various sizes

98% when the encrypted data is smaller than 100K in "CP-ABE on MD'. The CP-ABE encryption time of "CP-ABE Servic" decreased 74% of that of "CP-ABE on MD", and LABE is about 40% of "CP-ABE Service". For LABE, the average CP-ABE encryption time is only 0.5 ms which is just 10% of "CP-ABE on MD", and the speed up is 10 times. Obviously, it is too expensive to run the CP-ABE on mobile devices directly and LABE is a good scheme which can help the resource constrained mobile device to do CP-ABE.

The total time of the three cases is stable when the encrypted data are increased from 10K to 100K, whereas, the total times are nearly linear to the data size from 1M to 100M. The reason is that the ratio of symmetric encryption time in the total time in encrypting data with medium size is higher than that in encrypting data with small size. With the increasing of data size, the cost ratio of symmetric encryption become higher and higher. CP-ABE just encrypts the symmetric key for the protected data. We can see that the encryption time of CP-ABE is not affected by the data size, and the cost time depends on the architecture and CP-ABE algorithms.

The encryption time of CP-ABE under various attribute numbers is shown in Fig.3. After fixing the encrypted data to 100 KB, the attribute number is increased from 10 to 50 when 10 is added each time with all the attributes are

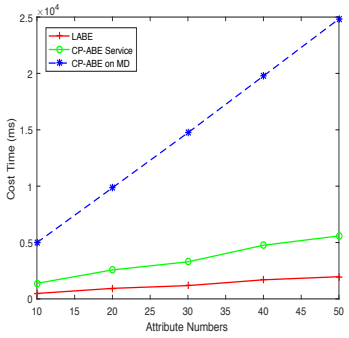


Figure 3: CP-ABE encryption time under various attribute numbers

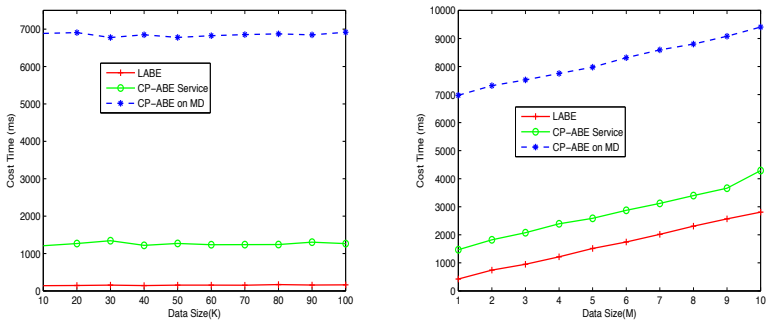
required in the access policy.

In Figure 3, we can find that the encryption times of CP-ABE increase linearly with attribute numbers, but LABE increased slower than "CP-ABE Servic" and "CP-ABE on MD". Although "CP-ABE on MD" and "CP-ABE Service" uses the same basic CP-ABE given in [5], "CP-ABE on MD" increases faster than "CP-ABE Service" with the increasing attributes numbers for the constrained computation resource on MD. While the attribute number reaches 50, the CP-ABE encryption time of "CP-ABE on MD" is 24832 ms, which is 4.45 times that of "CP-ABE Service". The cost time of LABE is only 35.2% of that of "CP-ABE Service" when there are 50 attributes specified in the access policy.

8.3. Decryption

Decryption is called more frequently than encryption and the performance of decryption is more important than that of encryption. We also evaluate the encryption performances under three cases: LABE, "CP-ABE Service", and "CP-ABE on MD". Figure 4 illustrates the total time of decrypting the ciphertext with various sizes including the small size and medium size.

The total decryption time in Figure 4 is stable for the small size, whereas, the total times increase nearly linearly with the data size for the medium. The



(a) Data with small size (b) Data with medium size

Figure 4: Total time of decrypting data with various sizes

total times of decryption are smaller than the corresponding time of encryption, exception for the "CP-ABE on MD" in decrypting data with small size. Our LABE works the best, "CP-ABE Service" works the second, and "CP-ABE on MD" works the worst. If the data size is less than 100 K, "CP-ABE on MD" needs 7 seconds to complete decryption. Without the proxy based service architecture, the mobile device finishes CP-ABE too poorly to be applied fluently in reality.

As shown in Fig.4, the average total time of "CP-ABE Service" is fewer over 5400 ms than that of "CP-ABE on MD", and the average total time of LABE is fewer over 1100 ms than that of "CP-ABE Service". According to the Section 6.4, the whole data decryption is divided two procedures: symmetric decryption and CP-ABE decryption. The symmetric decryption is to obtain the original data, whereas, the CP-ABE decryption is to obtain the key for the symmetric decryption. In the next, we analyze the cost time of CP-ABE decryption's performance under various attributes, and the results are shown in Fig.5.

With the increasing of the attribute numbers of the MD , the CP-ABE decryption time increases for LABE, "CP-ABE Service" , and "CP-ABE on MD". Especially, without any optimization, the original "CP-ABE on MD" increases so quickly that the decryption time reaches to more than 40 seconds

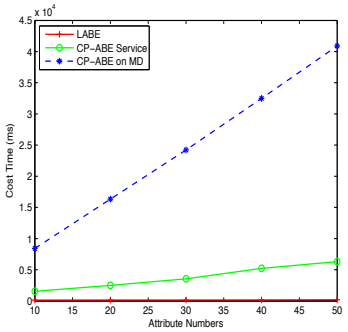


Figure 5: Decryption time of CP-ABE under various attribute numbers

when the attribute number is 50. The CP-ABE decryption time of "CP-ABE on MD" is obviously a big burden for mobile devices with constrained resources. With the proxy service architecture, the CP-ABE decryption time of "CP-ABE Service" decreases to 6.3 seconds on the attribute number equalling to 50 that is about 15.4% of that of "CP-ABE on MD".

LABE works fast enough in the evaluation. According to the 10, 20, 30, 40, and 50 attributes, the ratio of LABE on "CP-ABE on MD" is $1.5\%, 0.8\%, 0.6\%, 0.48\%$, and 0.43%, and the ratio on "CP-ABE Service" is $8.2\%, 5.5\%, 4\%, 3\%$, and 2.8% respectively. We can see that the optimization improvement performance becomes more and more obvious with the attribute number increasing. When the attribute number equals to 50, the decryption time of LABE is 177 ms, which still is a good accepted response time for MD.

9. Conclusion

In order to provide a fast, secure, and fine grained protection for the outsourced data in the mobile cloud assisted CPS, we propose a lightweight attribute based encryption scheme and presents its realization. Using proxy based service architecture, the functions of encryption/decryption are separated into different components, and then the working load of the mobile is lowered. Algorithms used in LABE is optimized specially in the transmitting bandwidth and

computation complexity, the encryption does not need pairing and decryption only needs one pairing. LABE is secure with fine grained access control and user revocation capability. The experiment results show that the encryption and decryption performance of LABE can be improved dozens of times for the traditional schemes and can be suitable for mobile cloud assisted CPS.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (61661015,61572148), Ministry of Education Key Lab of Cognitive Radio and Information Processing Found (CRKL160101), Guangxi Collaborative Innovation Center of Cloud Computing and Big Data Found (YD16801), and High Level of Innovation Team of Colleges and Universities in Guangxi Outstanding Scholars Program Funding.

References

[1] Clemens Krainer , Christoph M. Kirsch, Cyber-physical cloud computing implemented as PaaS, Proceedings of the 4th ACM SIGBED International Workshop on Design, Modeling, and Evaluation of Cyber-Physical Systems, April 14-17, 2014, Berlin, Germany [doi¿10.1145/2593458.2593461]

[2] Michael Armbrust , Armando Fox , Rean Griffith and et al."A view of cloud computing," in *Communications of the ACM 53(4).* pp.50-58, 2010.

[3] Y. Shucheng, W. Cong, R. Kui et al.,"Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing." pp. 1-9. 2010.

[4] V. Goyal, O. Pandey, A. Sahai et al. "Attribute-based encryption for fine-grained access control of encrypted data,"in *Proceedings of the 13th ACM conference on Computer and communications security, Alexandria, Virginia, USA*. 2006, pp. 89-98.

[5] B. Waters,"Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization," Public Key Cryptography C PKC 2011, Lecture Notes in Computer Science D. Catalano, N. Fazio, R. Gennaro et al., eds., pp. 53-70: Springer Berlin Heidelberg, 2011.

[6] A. Sahai and B. Waters. "Fuzzy identity based encryption" in *EUROCRYPT.* pp. 457C473, 2005.

[7] M. Horvth, "Attribute-Based Encryption Optimized for Cloud Computing," in *SOFSEM 2015: Theory and Practice of Computer Science, Lecture Notes in Computer Science G. Italiano, T. Margaria-Steffen, J. Pokorny et al., eds.,* . pp. 566-577: Springer Berlin Heidelberg, 2015.

[8] D. Koo, J. Hur, and H. Yoon, "Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage".in *Computers and Electrical Engineering.* vol. 39, no. 1, pp. 34-46, 2013.

[9] N. S. Kumar, G. V. R. Lakshmi, and B. Balamurugan, "Enhanced Attribute Based Encryption for Cloud Computing," in *Communications of the ACM 53(4).* Procedia Computer Science, vol. 46, no. 0, pp. 689-696, 2015.

[10] Yu S, Wang C, Ren K, Lou W. Achieving secure, "scalable, and finegrained data access control in cloud computing. In: International conference on computer communications (INFOCOM). IEEE; 2010a. pp. 1-9.

[11] Wang G, Liu Q, Wu J."Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. " in *Proceedings of the 17th ACM conference on computer and communications security (CCS). ACM.* 2010. pp. 735-737.

[12] F. Belqasmi, R. Glitho, and F. Chunyan, " RESTful web services for service provisioning in next-generation networks: a survey. Communications Magazine, IEEE, 2011. 49(12): " pp. 66-73.

[13] A. Shamir, "Identity-based cryptosystems and signature schemes, in Advances in Cryptology. Berlin, Germany: Springer-Verlag, " 1985, pp. 47C53.

[14] A. Sahai and B. Waters,"Fuzzy identity-based encryption, in Advances in Cryptology . Berlin, Germany: Springer-Verlag," 2005, pp. 457-473.

[15] J. Bethencourt, A. Sahai, and B. Waters,"Ciphertext-policy attribute-based encryption, "in *Proc. IEEE Symp. Security Privacy* . 2007, pp.321-334.

[16] S. Hohenberger and B. Waters," Attribute-Based Encryption with Fast Decryption, in Public-Key Cryptography C PKC 2013, K. Kurosawa and G. Hanaoka, Editors. 2013, Springer Berlin Heidelberg." p. 162-179.

[17] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length, in Proc. 5th Int. Conf. Inf. Security Practice Experi-ence . Springer-Verlag, " 2009, pp. 13-23.

[18] X. Yao, Z. Chen, and Y. Tian,"A lightweight attribute-based encryption scheme for the Internet of Things. Future Generation Computer Systems, " 2014(0).

[19] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, K. "Fu, Plutus: scalable secure file sharing on untrusted storage, in: Proceedings of the USENIX Conference on File and Storage Technologies (FAST)," 2003, pp. 29-42.

[20] E.J. Goh, H. Shacham, N. Modadugu, D."Boneh, Sirius: securing remote untrusted storage, in: Proceedings of Network and Distributed Systems Security Symposium (NDSS), 2003, pp. 131-145.

[21] C. Gentry, A. Silverberg, "Hierarchical id-based cryptography, in: Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT), 2002, pp. 149-155.

[22] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter. "Patient controlled encryption: Ensuring privacy in medical health records. In ACM CCSW 2009, 2009.

[23] S. Narayan, M. Gagne, R."A view of cloud computing," Safavi-Naini, Privacy preserving EHR system using attribute-based infrastructure, in: Proceedings of the ACM workshop on Cloud Computing Security (CCS), 2010, pp. 47-52.

[24] J. Li, Q. Huang, X. Chen, S.S.M. Chow, D.S. Wong, D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability, in:Proceedings of the ACM Symposium on Information, Computer and Communications Security (ASIACCS), 2011, pp. 386-390.

[25] X. Dong, J. Yu, Y. Luo, et al., "Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing. Computers and Security, 2014. 42(0): p. 151-164.

[26] S. Hohenberger and B. Waters, "Online/Offline Attribute-Based Encryption, in Public-Key Cryptography - Pkc 2014, H. Krawczyk, Editor. " 2014. p. 293-310.

[27] B. Qin, R.H. Deng, S. Liu, et al., "Attribute-Based Encryption With Efficient Verifiable Outsourced Decryption. Information Forensics and Security, IEEE Transactions on, " 2015. 10(7): p. 1384-1393.

[28] H. Deng, Q. Wu, B. Qin, et al."Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts. Information Sciences, " 2014. 275(0): p. 370-384.

[29] Z. Zhibin, H. Dijiang, and W. Zhijie, Efficient Privacy-Preserving Ciphertext-Policy Attribute Based-Encryption and Broadcast Encryption. Computers, IEEE Transactions on, 2015. 64(1): p. 126-138.

[30] Liu Z, Cao Z."On efficiently transferring the linear secret-sharing scheme matrix in ciphe rtext-policy attribute-based encryption; 2010 [Technical Re port; ," in *Cryptology ePrint Archiv e,Report 2010/374*].