

Efficient and Secure Data Storage Operations for Mobile Cloud Computing

Zhibin Zhou and Dijiang Huang
{zhibin.zhou,dijiang}@asu.edu
Arizona State University

Abstract—

In a mobile cloud computing system, lightweight wireless communication devices extend cloud services into the sensing domain. A common mobile cloud secure data service is to inquire the data from sensing devices. The data can be collected from multiple requesters, which may drain out the power of sensing devices quickly. Thus, an efficient data access control model is desired. To this end, we present a comprehensive security data inquiry framework for mobile cloud computing. Our solution focuses on the following two research directions: First, we present a novel Privacy Preserving Cipher Policy Attribute-Based Encryption (PP-CP-ABE) to protect sensing data. Using PP-CP-ABE, light-weight devices can securely outsource heavy encryption and decryption operations to cloud service providers, without revealing the data content. Second, we propose an Attribute Based Data Storage (ABDS) system as a cryptographic group-based access control mechanism. Our performance assessments demonstrate the security strength and efficiency of the presented solution in terms of computation, communication, and storage.

I. INTRODUCTION

With the fast development of wireless technology, mobile cloud computing has become an emerging cloud service model [19], [9], where mobile devices and sensors are used as the information collecting and processing nodes for the cloud infrastructure. This new trend demands researchers and practitioners to construct a trustworthy architecture for mobile cloud computing that includes a large numbers of lightweight, resource-constrained mobile devices. In such a mobile cloud sensing environment, cloud users may inquire the data from sensing devices. A simple solution to protect the data is to encrypt the sensing data with a group key and broadcast the encrypted data; only legitimated users can reveal the data content with the predistributed group key. However, this approach demands high key management overhead and it is vulnerable to single point failure problems.

Ciphertext Policy Attribute-Based Encryption (CP-ABE) schemes [3], [7], [6], [21] were proposed to facilitate key management and cryptographic access control in an expressive and efficient way. Under the construction of CP-ABE, an attribute is a descriptive string assigned to (or associated with) a user and each user may be tagged with multiple attributes. Multiple users may share common attributes, which allow sensors to specify a data access policy by composing multiple attributes through logical operators such as “AND”, “OR”,

etc. To decrypt the message, the decryptor’s attributes need to satisfy the access policy.

To illustrate the application scenario, let us consider a mobile remote health sensing scenario, where a doctor using a mobile device (e.g., smart phone) to inquire the sensing data collected from a set of body sensors attached on a patient at home. It is convenient to encrypt the data and enforcing data access policies that only eligible users can decrypt it. To this end, The sensed data can be encrypted using the following policy:

{12/08/2011 AND Doctor AND Saint Luke Hospital}.

In this example, doctors who are working in Saint Luke hospital on 12/8/2011 can decrypt the data. Using CP-ABE scheme, the sensor can use the above described policies to encrypt the data and the data inquirers must satisfy the given policies in order to decrypt the data.

To establish the highlighted mobile cloud data inquiry services, we need to address the following research challenges:

- With the CP-ABE enabled mobile cloud data inquiry services, the main challenge is originated from the fact that CP-ABE schemes always require intensive computing resources for sensors or mobile devices to run the encryption and decryption algorithms.
- Given the sensitivity of data and multitenancy nature of public cloud, critical customer secrets should not be exposed to the cloud.
- Another major challenge is how to upload/download and update encrypted data stored in the mobile cloud system. Frequent upload/download operations will cause tremendous overhead for resource constrained wireless devices.

To address the above described research challenges, in this paper, we present a secure data inquiry framework for mobile cloud computing that includes two major components:

- 1) A Privacy Preserving CP-ABE (PP-CP-ABE) scheme;
- 2) An Attribute-Based Data Storage (ABDS) scheme that achieves information theoretical optimality.

Using PP-CP-ABE, users can securely outsource computation intensive CP-ABE encryption and decryption operations to the cloud without revealing data content and secret keys. In this way, lightweight and resource constrained devices can access and manage data stored in the cloud data store. The

ABDS system achieves scalable and fine-grained data access control, using public cloud services. Based on ABDS, users' attributes are organized in a carefully constructed hierarchy so that the cost of membership revocation can be minimized. Moreover, ABDS is suitable for mobile computing to balance communication and storage overhead and thus reduces the cost of data management operations (such as upload, updates, etc.) for both the mobile cloud nodes and storage service providers. Our performance evaluation demonstrate that the proposed solution is computation efficient (i.e., saving 90% for encryption and 99% for decryption) for lightweight mobile devices and it is storage efficient of ABDS scheme, where both data inquirers and sensors only need to store $\log_2(N)$ private keys while N keys are required when using CP-ABE scheme.

The rest of this paper is organized as follows. Section II presents system models used in this paper. We present detailed PP-CP-ABE construction and ABDS design in Section III and IV, respectively. In Section V, we analyzed the security and discuss the performance of proposed schemes with comparison to several related works. We describe related works in Section VI. Finally, we conclude our work in Section VII.

II. SYSTEM AND MODELS

A. Notations

The notations used in this paper is listed in the following table:

Acronym	Descriptions
DO	Data Owner
DR	Data Requester/Receiver
ESP	Encryption Service Provider
DSP	Decryption Service Provider
SSP	Storage Service Provider
TA	Trust Authority
T	Access Policy Tree

B. Overview

In our proposed system, we denote the Data Owner as DO. A DO can be a mobile wireless device such as a smart phone or an environmental sensor that can request and/or store encrypted information from/in the Cloud storage. The data are encrypted using the proposed PP-CP-ABE scheme. Other than DO, there are many DRs (Data Requesters or Receivers) who can inquiry the information from the storage services of the mobile cloud. For example, a user may want to inquiry current pollution map of a particular city area. Since the data provided by DOs can be proprietary, it should be encrypted and only pollution map service subscribers can retrieve the data. In this case, the mobile cloud system only provides a service platform and it should not be able to access the data content from the DOs. In this paper, our focus is on the encryption and decryption model to support the described application scenario; thus, due to the space limit, we do not describe how exactly the application is established in details. The presented system model should provide the following properties:

- 1) The data must be encrypted before sending to storage service provider (SSP);

- 2) The encryption service provider (ESP) provides encryption service to the data owner without knowing the actual data encryption key (DEK);
- 3) The decryption service provider (DSP) provides decryption service to data inquirers without knowing the data content;
- 4) Even ESP, DSP and SSP collude, the data content cannot be revealed;

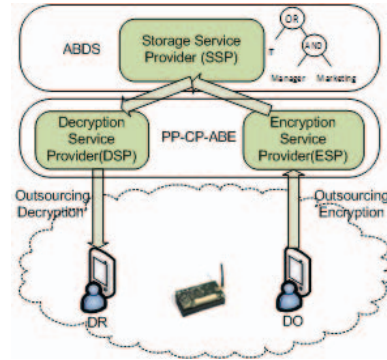


Fig. 1. System Architecture of Our Proposed Framework.

As shown in Figure 1, the SSP, ESP, and DSP form the core components of the proposed system. A DR inquires the data provided by a DO. ESP and DSP provide PP-CP-ABE services and SSP, e.g., Amazon S3, provides storage services. The cloud is semi-trusted, in which the cloud only provides computing and storage services with the assistance on data security; however, the data is blinded to the cloud. In particular, more powerful PCs and Mobile Phones can work as communication proxy for sensors that collect information.

Essentially, the basic idea of PP-CP-ABE to outsource intensive but non-critical part of the encryption and decryption algorithm to the service providers while retain critical secrets. As we can prove later in this paper, the outsourcing of computation does not reduce the security level compared with original CP-ABE schemes, where all computations are performed locally.

The encryption complexity of CP-ABE grows linearly on the size of access policy. During the encryption, a master secret is embedded into ciphertext according to the access policy tree in a recursive procedure, where, at each level of the access policy, the secret is split to all the subtree of the current root. However, the security level is independent on the access policy tree. In other words, even if the ESP possesses secrets of most but not all parts of the access policy tree, the master secret is still information theoretically secure given there at least one secret that is unknown to ESP. Thus, we can safely outsource most part of encryption complexity to ESP by just retaining a small amount of secret information, which is processed locally.

As for the decryption, the CP-ABE decryption algorithm is computationally expensive since bilinear pairing operations over ciphertext and private key is a computational intensive operation. PP-CP-ABE addresses this computation issue by securely blinding the private key and outsourcing the expensive Pairing operations to the DSP. Again, the outsourcing will not expose the data content of the ciphertext to the DSP. This

is because the final step of decryption is performed by the decryptors.

C. Attacking Models

The malicious attackers' goal is to reveal data in the cloud without authorization from DOs. Service providers (ESP, DSP and SSP) and the attacks can combine their information to perform collusion attacks, in which they can try to decrypt the ciphertext and compromise the decryption keys that they are not authorized to access. One particular example of this attack is that they gather enough information to compromise the decryption keys SK from many blind private keys \widetilde{SK} as DSP has the ability to get a lot of \widetilde{SK} . In addition, the attacks may compromise the encrypted data by use of the advantage which ESP provides the encryption service to gain from the DO.

In particular, attackers want to break the *Forward Secrecy*, which is defined as follows: After a user is revoked from accessing a file, he/she may have a local copy of the file; however, the revoked user must not get any future updates on this file.

While data integrity and retrievability in the cloud are also important security requirements, they are not the focuses of this paper. Readers can refer to research works in the provable data possession (PDP) [1], [12].

D. Access Policy Tree

In this section, we briefly describe the model of an access policy tree used in PP-CP-ABE as illustrated in Figure 2. The data access policy tree of PP-CP-ABE is composed by leaf nodes and internal nodes. Each leaf node represents an attribute, and each internal node is a logical gate, such as "AND", "OR", "n-of-m". Several functions and terms are defined as follows to facilitate the presentation of our solutions:

- $\text{parent}(x)$: return the parent node of node x ;
- $\text{att}(x)$ denotes the attribute associated with the leaf node x in the data access tree;
- The access tree \mathcal{T} composed by a set of leaf nodes (i.e., attributes) and internal nodes (i.e., logical gates) defines the data access policies, i.e., if a user *owns* a set of attributes that satisfy the logic operations of the tree to reach the root, it can access the secret secured by \mathcal{T} . Here *owns* means that the user has the private keys corresponding to the set of attributes. AND and OR are the most frequently used logical gates.
- num_x is the number of children of a node x . A child y of node x is uniquely identified by an index integer $\text{index}(y)$ from 1 to num_x .
- The threshold value $k_x = \text{num}_x - 1$ when x is an AND, and $k_x = 0$ when x is an OR gate or a leaf node. k_x is used as the polynomial degree for node x using the threshold secret sharing scheme [29].

E. Bilinear Pairing

Our proposed PP-CP-ABE is constructed using bilinear pairing. Pairing is a bilinear map function $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$, where \mathbb{G}_0 and \mathbb{G}_1 are two multiplicative cyclic groups with

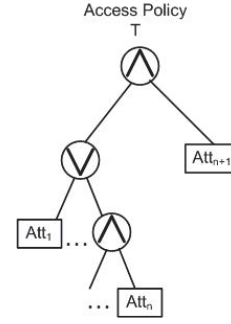


Fig. 2. Illustration of a sample access policy tree.

large prime order p . The Discrete Logarithm Problem on both \mathbb{G}_0 and \mathbb{G}_1 are hard. Pairing has the Bilinearity property:

$$e(P^a, Q^b) = e(P, Q)^{ab}, \quad \forall P, Q \in \mathbb{G}_0, \forall a, b \in \mathbb{Z}_p^*.$$

III. PRIVACY PRESERVING CP-ABE

A. Overview of the Construction

Essentially, the basic idea of PP-CP-ABE to outsource intensive but non-critical part of the encryption and decryption algorithm to the service providers while retain critical secrets. As we can prove later in this paper, the outsourcing of computation does not reduce the security level compared with original CP-ABE schemes, where all computations are performed locally.

The encryption complexity of CP-ABE grows linearly on the size of access policy. During the encryption, a master secret is embedded into ciphertext according to the access policy tree in a recursive procedure, where, at each level of the access policy, the secret is split to all the subtree of the current root. However, the security level is independent on the access policy tree. In other words, even if the ESP possesses secrets of most but not all parts of the access policy tree, the master secret is still information theoretically secure given there at least one secret that is unknown to ESP. Thus, we can safely outsource most part of encryption complexity to ESP by just retaining a small amount of secret information, which is processed locally.

As for the decryption, the CP-ABE decryption algorithm is computationally expensive since bilinear pairing operations over ciphertext and private key is a computational intensive operation. PP-CP-ABE addresses this computation issue by securely blinding the private key and outsourcing the expensive Pairing operations to the DSP. Again, the outsourcing will not expose the data content of the ciphertext to the DSP. This is because the final step of decryption is performed by the decryptors.

B. System Setup and Key Generation

The TA first runs **Setup** to initiate the PP-CP-ABE system by choosing a bilinear map: $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$ of prime order p with the generator g . Then, TA chooses two random $\alpha, \beta \in \mathbb{Z}_p$. The public parameters are published as:

$$PK = \langle \mathbb{G}_0, g, h = g^\beta, e(g, g)^\alpha \rangle. \quad (1)$$

The master key is $MK = (\beta, g^\alpha)$, which is only known by the TA.

Each user needs to register with the TA, who authenticates the user's attributes and generates proper private keys for the user. An attribute can be any descriptive string that defines, classifies, or annotates the user, to which it is assigned. The key generation algorithm takes as input a set of attributes S assigned to the user, and outputs a set of private key components corresponds to each of attributes in S . The **GenKey** algorithm performs the following operations:

- 1) Chooses a random $r \in \mathbb{Z}_p$,
- 2) Chooses a random $r_j \in \mathbb{Z}_p$ for each attribute $j \in S$.
- 3) Computes the private key as:

$$SK = \langle D = g^{(\alpha+r)/\beta}; \\ \forall j \in S : D_j = g^{r_j} \times H(j)^{r_j}; D'_j = g^{r_j} \rangle.$$

- 4) Sends SK to the DO through a secure channel.

C. PP-CP-ABE Encryption

To outsource the computation of Encryption and preserve the data privacy, a DO needs to specify a policy tree $\mathcal{T} = \mathcal{T}_{ESP} \wedge \mathcal{T}_{DO}$, where \wedge is an AND logic operator connecting two subtrees \mathcal{T}_{ESP} and \mathcal{T}_{DO} . \mathcal{T}_{ESP} is the data access policy that will be performed by the ESP and \mathcal{T}_{DO} is a DO controlled data access policy. \mathcal{T}_{DO} usually has a small number of attributes to reduce the computation overhead at the DO, in which it can be a sub-tree with just one attribute (see the example shown in Figure 3).

In practice, if \mathcal{T}_{DO} has one attribute, DO can randomly specify an 1-degree polynomial $q_R(x)$ and sets $s = q_R(0)$, $s_1 = q_R(1)$, and $s_2 = q_R(2)$. Then DO sends $\{s_1, \mathcal{T}_{ESP}\}$ to ESP, which is noted as:

$$DO \xrightarrow{\{s_1, \mathcal{T}_{ESP}\}} ESP.$$

Here, we must note that sending s_1 and \mathcal{T}_{ESP} will not expose any secret of our solution. We will prove this in Section V-A.

ESP then runs the **Encrypt**(s_1, \mathcal{T}_{ESP}) algorithm, which is described below:

- 1) $\forall x \in \mathcal{T}_{ESP}$, randomly chooses a polynomial q_x with degree $d_x = k_x - 1$, where k_x is the secret sharing threshold value:
 - a) For the root node of \mathcal{T}_{ESP} , i.e., R_{ESP} , Chooses a $d_{R_{ESP}}$ -degree polynomial with $q_{R_{ESP}}(0) = s_1$.
 - b) $\forall x \in \mathcal{T}_{ESP} \setminus R_{ESP}$ sets d_x -degree polynomial with $q_x(0) = q_{\text{parent}(x)}(\text{index}(x))$.
- 2) Generates a temporal ciphertext:

$$CT_{ESP} = \{\forall y \in Y_{ESP} : C_y = g^{q_y(0)}, C'_y = H(\text{att}(y))^{q_y(0)}\},$$

where Y_{ESP} is the set of leaf nodes in \mathcal{T}_{ESP} .

At the meantime, the DO performs the following operations:

- 1) Performs **Encrypt**(s_2, \mathcal{T}_{DO}) and derives:

$$CT_{DO} = \{\forall y \in Y_{DO} : C_y = g^{q_y(0)}, C'_y = H(\text{att}(y))^{q_y(0)}\}.$$

- 2) Computes $\tilde{C} = Me(g, g)^{\alpha s}$ and $C = h^s$, where M is the message.

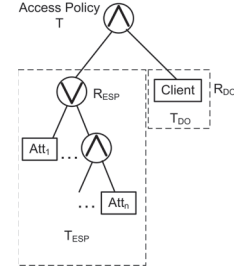


Fig. 3. Illustration of access policy $\mathcal{T} = \mathcal{T}_{ESP} \wedge \mathcal{T}_{DO}$.

- 3) Sends CT_{DO}, \tilde{C}, C to the ESP:

$$DO \xrightarrow{\{CT_{DO}, \tilde{C}, C\}} ESP.$$

On receiving the message from the DO, ESP generates the following ciphertext:

$$CT = \langle \mathcal{T} = \mathcal{T}_{ESP} \wedge \mathcal{T}_{DO}; \tilde{C} = Me(g, g)^{\alpha s}; C = h^s; \\ \forall y \in Y_{ESP} \cup Y_{DO} : C_y = g^{q_y(0)}; C'_y = H(\text{att}(y))^{q_y(0)} \rangle.$$

$$\forall y \in Y_{ESP} \cup Y_{DO} : C_y = g^{q_y(0)}; C'_y = H(\text{att}(y))^{q_y(0)}.$$

Finally, the ESP sends CT to the SSP.

D. Outsourcing Decryption

CP-ABE decryption algorithm is computationally expensive since bilinear pairing is an expensive operation. PP-CP-ABE addresses this computation issue by outsourcing the expensive Pairing operations to the DSP. Again, the outsourcing will not expose the data content of the ciphertext to the DSP.

To protect the data content, the DO first blinds its private key by choosing a random $t \in \mathbb{Z}_p$ and then calculates $\tilde{D} = D^t = g^{t(\alpha+r)/\beta}$. We denote the blinded private key as \widetilde{SK} :

$$\widetilde{SK} = \langle \tilde{D} = g^{t(\alpha+r)/\beta}; \\ \forall j \in S : D_j = g^{r_j} \cdot H(j)^{r_j}; D'_j = g^{r_j} \rangle. \quad (2)$$

Before invoking the DSP, the DO first checks whether its owned attributes will satisfy the access policy \mathcal{T} . If so, the DO sends $\{\widetilde{SK}\}$ to the DSP, and requests the SSP to send the ciphertext to the DSP. On receiving the request, the SSP sends $CT' = \{\mathcal{T}; C = h^s; \forall y \in Y_1 \cup Y_2 : C_y = g^{q_y(0)}; C'_y = H(\text{att}(y))^{q_y(0)}\}$ and $CT' \subset CT$ to the DSP:

$$SSP \xrightarrow{\{CT'\}} DSP. \quad (3)$$

Once the DSP receives both $\{\widetilde{SK}\}$ and CT' , it then runs the **Decrypt**(\widetilde{SK}, CT') algorithm as follows:

- 1) $\forall y \in Y = Y_{ESP} \cup Y_{DO}$ the DSP runs a recursive function $\text{DecryptNode}(CT', \widetilde{SK}, R)$, where R is the root of \mathcal{T} . The recursion function is the same as defined in [3] and $\text{DecryptNode}(CT', \widetilde{SK}, y)$ is proceeded as follows:

$$\begin{aligned} \text{DecryptNode}(CT', \widetilde{SK}, y) &= \frac{e(D_i, C_y)}{e(D'_i, C'_y)} \\ &= \frac{e(g^r \cdot H(i)^{r_i}, g^{q_y(0)})}{e(g^{r_i}, H(i)^{q_y(0)})} \\ &= e(g, g)^{r q_y(0)} \\ &= F_y. \end{aligned}$$

The recursion is processed as follows: $\forall y$ is the child of x , it calls $DecryptNode(CT'; S\bar{K}; y)$ and stores the output as F_y . Let S_x be an arbitrary k_x -sized set of children nodes y , the DSP computes:

$$\begin{aligned}
F_x &= \prod_{y \in S_x} F_y^{\Delta_{i, S'_x}(0)} \\
&= \prod_{y \in S_x} (e(g; g)^{r \cdot q_y(0)})^{\Delta_{i, S'_x}(0)} \\
&= \prod_{y \in S_x} (e(g; g)^{r \cdot q_{\text{parent}(y)(\text{index}(y))}})^{\Delta_{i, S'_x}(0)} \\
&= \prod_{y \in S_x} (e(g; g)^{r \cdot q_x(i) \cdot \Delta_{i, S'_x}(0)}) \\
&= e(g, g)^{r q_x(0)}, \tag{4}
\end{aligned}$$

where $i = \text{index}(z)$ and $S'_x = \{\text{index}(z) : z \in S_x\}$, $\Delta_{i, S'_x}(0)$ is the Lagrange coefficient. Finally, the recursive algorithm returns $A = e(g, g)^{rs}$.

2) Then, computes

$$e(C, \tilde{D}) = e(h^s, g^{t(\alpha+r)/\beta}) = e(g, g)^{trs} \cdot e(g, g)^{t\alpha s}.$$

3) Sends $\{A = e(g, g)^{rs}, B = e(C, \tilde{D}) = e(g, g)^{trs} \cdot e(g, g)^{t\alpha s}\}$ to the DO:

$$DSP \xrightarrow{\{A, B\}} DO.$$

On receiving $\{A, B\}$, DO calculates $B' = B^{1/t} = e(g, g)^{rs} \cdot e(g, g)^{\alpha s}$ and then it recovers the message:

$$M = \frac{\tilde{C}}{(B'/A)} = \frac{Me(g, g)^{\alpha s}}{(e(g, g)^{rs} \cdot e(g, g)^{\alpha s})/e(g, g)^{rs}}.$$

IV. ATTRIBUTE BASED DATA STORAGE

In this section, we present an Attribute Based Data Storage (ABDS) scheme that is based on PP-CP-ABE to enable efficient, scalable data management and sharing.

A. Data Management Overview

The frequent data updates will cause additional expense for file managements. For example, to update existing files, e.g., changing certain data fields of an encrypted database, in which the encrypted data need to be downloaded from SSP to DSP for decryption. Upon finishing the updates, the ESP needs to be re-encrypted and upload the data to the SSP. Thus, the re-encrypted process requires downloading and uploading the data, which may incur high communication and computation overhead, and as a result, will cost more for DOs.

To address the described cost issue, it is reasonable to divide a file into independent blocks that are encrypted independently. To update files, the DO can simply download the particular blocks to be updated. In this way, we can avoid re-encrypting the entire data. Moreover, data access control can be enforced on individual blocks using “lazy” re-encryption strategy. For example, when the data access memberships to a particular file are changed (i.e., the access tree is changed), this event can be recorded but no file changes are invoked. Until the

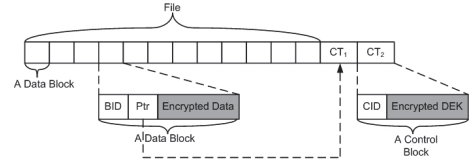


Fig. 4. Illustration of a file organized into blocks with multiple control blocks.

data content needs to be updated, the re-encryption is then performed using the proposed PP-CP-ABE scheme.

Partitioning the data into multiple small blocks also introduces addition overhead. This is because the extra control information needs to be attached for each data block for data management. For example, the control message should include a block ID and a pointer to its corresponding data access tree \mathcal{T} . In Figure 4, we depicted a sample file stored in SSP. As shown in Figure 4, each file is divided into blocks. A block is a tuple $\{\text{BID}, \text{Ptr}, \text{Encrypted Data}\}$, where BID is the unique identification of the block; Ptr is the pointer to the control block CT; and data is encrypted with a DEK. A control block $\{\text{CID}, \text{Encrypted DEK}\}$ has a control block ID, i.e., CID and DEK encrypted by using PP-CP-ABE scheme.

The ABDS system should determine what is the appropriate data block size to be partitioned with a known file size. In this work, our goal is to minimize the storage and communication overhead with the considerations of the following simple assumptions:

- 1) Every data update should only affect a small amount of data, e.g., updating certain data fields in the Database;
- 2) In each unit time period, the number of blocks to be updated is known;
- 3) Each data block has the same probability to be updated;

Based on the above discussions, we can model the total cost C in a unit time period as follows:

$$C = 2nS_bC_c + \frac{F}{S_b}S_cC_s, \tag{5}$$

where n is the number of updated blocks in a unit time period and $2n$ stands for an update includes one encryption and one decryption that require two transmissions; S_b is the size of block; C_c is the cost rate of data transmission that is charged by both cloud storage providers and wireless communication service providers; F is the size of file; S_c is the size of control data for each data block, and C_s is the charging rate of storage. To minimize cost C , DO can minimize (5) and derive the optimal block size:

$$S_b \geq 2\sqrt{2nC_cFS_cC_s}.$$

B. Setup

PP-CP-ABE enables expressive policy with descriptive attributes to enforce data access control on the stored data. For example, if Alice wants to share a file to all CS students, she can specify the policy “CS AND Student”. All the users whose attributes satisfy this policy can decrypt the data.

Besides the set of descriptive attributes enabled in the system, each user is assigned a unique binary ID:

$b_0b_1\dots b_{n-2}b_{n-1}$. We can define the term “bit-assignment attribute” that is represented as “ B_i ” or “ \overline{B}_i ” to indicate the binary value at position i in the ID. B_i indicates that the i 'th bit of an ID is 1; \overline{B}_i indicates that the i 'th bit of an ID is 0. If the length of an ID is n , then the total number of bit-assignment attributes is $2n$. This means that two binary values are mapped to one bit position (one for value 0 and one for value 1). Thus, a DO with ID u is uniquely identified by the set of bit-assignments S_u . Also, multiple DOs may have a common subset of bit-assignments. For example, a DO u_1 's ID is 000 and a DO u_2 's ID is 001, $S_{u_1} = \{\overline{B}_0, \overline{B}_1, \overline{B}_2\}$ and $S_{u_2} = \{\overline{B}_0, \overline{B}_1, B_2\}$ and $S_{u_1} \cap S_{u_2} = \{\overline{B}_0, \overline{B}_1\}$. Bit-assignment attributes can be used when the DO wants to share data to any arbitrary set of DOs. In this case, it may be hard to describe the set of DOs efficiently using descriptive attributes.

C. Upload New Files

Before uploading new files to the SSP, both ESP and DO are required to determine the encryption parameters such as the block size. DO then invokes ESP with an access policy \mathcal{T}_{ESP} , which is the access policy to be enforced on the uploaded files. Here, we define some terms used in the following presentations:

- *Literal*: A variable or its complement, e.g., $b_1, \overline{b_1}$, etc.
- *Product Term*: Literals connected by AND, e.g., $\overline{b_2}b_1\overline{b_0}$.
- *Sum-of-Product Expression (SOPE)*: Product terms connected by OR, e.g., $\overline{b_2}b_1b_0 + b_2$.

Given the set of shared data receivers S , the membership functions $f_S()$, which is in the form of SOPE, specifies the list of receivers:

$$f_S(b_1^u, b_2^u, \dots, b_n^u) = \begin{cases} 0 & \text{iff } u \in S, \\ 1 & \text{iff } u \notin S. \end{cases}$$

For example, if the subgroup $S = \{000, 001, 011, 111\}$, then $f_S = \overline{b_0}\overline{b_1}\overline{b_2} + \overline{b_0}\overline{b_1}b_2 + \overline{b_0}b_1b_2 + b_0b_1b_2$.

Then, the DO runs the Quine-McCluskey algorithm [24] to reduce f_S to minimal SOPE f_S^{min} . The reduction can consider *do not care* values $*$ on those IDs that are not currently assigned to any DO to further reduce number of product terms in the membership function. For example, if $S = \{000, 001, 011, 111\}$, $f_S^{min} = \overline{b_0}\overline{b_1} + b_1b_2$.

Finally, DO uploads the data blocks and the control block to SSP, where each data block is encrypted by the DEK and DEK is protected by the access policy in control block.

D. Data Updates

Now, we investigate into how to efficiently handle the data updates, i.e., how to modify encrypted data with or without changing data access control policy.

1) *Data Updates With Access Policy Change*: In Section IV-A, we described the “lazy” re-encryption strategy adopted by DOs. Using the “lazy” re-encryption scheme, the DO continuously records the revoked data receivers. When there is a need to modify the data, the DO will choose a new data access tree that can revoke all previously recorded data receivers.

When DO updates a data block with access policy change, we need to consider the following cases:

- If there is no control block associated with the latest access policy, i.e., no data updates occurred after the latest access policy change event, the DO encrypt a new random DEK associated with the latest access policy with PP-CP-ABE and attach a new control block to the end of the file, see Figure 4.
- If there exists a control block associated with the latest access policy, i.e., at least one data block was encrypted with the newest access policy, the DO can simply redirect the control block pointer, see Figure 4, to the control block associated with the latest access policy.
- If a control block is not pointed by any data block, this control block should be deleted.

2) *Updates Without Access Policy Change*: If no change is required to the access policy, DO can simply perform the PP-CP-ABE scheme and upload the updated data block in the SSP. The Block ID and the pointer to control the block are not changed.

V. PERFORMANCE EVALUATION

In this section, we first present the security assessments of the presented solution. Then, we present the computation, communication, and storage performance evaluation.

A. Security Assessments

We now briefly analyze the security of PP-CP-ABE scheme. We first describe the hardness assumptions used in this scheme: Given a bilinear map group system $\mathbb{S} = (p, \mathbb{G}, \mathbb{G}_T, e(\cdot, \cdot))$, where two groups \mathbb{G}, \mathbb{G}_T have the prime order p . The security of this scheme is constructed on two basic assumptions: Co-Decision Bilinear Diffie-Hellman (Co-DBDH) assumption and Decision Linear Diffie-Hellman (DLDH) assumption.

The data structure of ciphertext and private key in PP-CP-ABE is the same as the original BSW CP-ABE [3]. Thus PP-CP-ABE can be viewed as a variation of CP-ABE. Particularly, in PP-CP-ABE, the access policy tree is constructed by two sub-trees $\mathcal{T} = \mathcal{T}_{ESP} \wedge \mathcal{T}_{DO}$. In general, \mathcal{T}_{DO} contains a single attribute to reduce the computation and communication overhead. Thus, DO randomly specifies a 1-degree polynomial $q(x)$ and sets $s = q(0)$, $s_1 = q(1)$ and $s_2 = q(2)$. The tuple $\{s_1, \mathcal{T}_{ESP}\}$ is sent to ESP. It is easy to prove that, based on the threshold secret sharing scheme [29], for a given 1-degree polynomial $q(x)$, knowing s_1 , secrets s and s_2 are information theoretically secure. In order to avoid the leakage of encrypted information for cloud service providers (including ESP and DSP), the following theorem proves that this scheme is secure against the adaptive chosen plaintext attacks (IND-CPA) based on Co-DBDH assumption.

Theorem 1: Let \mathcal{E} is a PP-CP-ABE scheme. If Co-DBDH is (t, q, ε) -hard on \mathbb{G} , then the PP-CP-ABE scheme is (t', q', ε') -secure against the adaptive chosen plaintext attacks (IND-CPA), where $\varepsilon' > \varepsilon/4$, $q = q'$ and $t' > t$. Here q is the

TABLE I
AVAILABLE RATIONAL TERMS TO ESP AND DSP

ESP	s_1	$e(g, g)^{\alpha s_1}$	$g^{\beta s_1}$	$g^{s_1/\beta}$
DSP	$e(g, g)^{r' s_1}$	$e(g, g)^{r' s_2}$	$e(g, g)^{r' s}$	$e(g, g)^{t\alpha s + tr' s}$

number of hash function queries made by the adversary, and t is the run time of attacks.

Based on the security assumptions presented in Section II-C, ESP, DSP and SSP are untrusted but honest service providers that will perform proper computation according to PP-CP-ABE protocol and returns correct results. In order to compromise users' secret information, the ESP, DSP and SSP can perform collusion attacks. In this scenario, an authorized user u' who satisfies the access tree \mathcal{T} provides his blinded private key \widetilde{SK} to the DSP for decryption. Then, ESP and DSP can try to utilize the blinded private key of u' to derive M from $Me(g, g)^{\alpha s}$. ESP has s_1 , and thus it can easily derive $e(g, g)^{\alpha s_1}$. This is because $e(g, g)^\alpha$ is available from the public parameters presented in (1). As the user u' satisfies the access policy \mathcal{T}_{DO} , DSP can derive the following values $e(g, g)^{r' s_1}$, $e(g, g)^{r' s_2}$, $e(g, g)^{r' s}$, and $e(g, g)^{t\alpha s + tr' s}$ through the F_x function (see (4)) without knowing α and r' . In the following table, we listed all rational terms that are available to ESP and DSP.

As we can see, ESP has the values s_1 and $e(g, g)^{\alpha s_1}$, but it is unaware of values s_2 or s . DSP possesses more terms as well as the blinded private key \widetilde{SK} of u' (see (2)). We must note that \widetilde{SK} is not a valid CP-ABE private key, since the $\widetilde{D} = g^{t(\alpha+r')/\beta}$ is embedded with tr' and $t\alpha$, and the rest of all private key components $\{\forall j \in S : D_j = g^{r'} \cdot H(j)^{r_j}, D'_j = g^{r_j}\}$ are embedded with r' . Essentially, this blinded private key can be a valid CP-ABE private key when (i) the master key is $MK = \{\beta, g^{t\alpha}\}$; (ii) a colluding user contributes $D = g^{t(\alpha+r')/\beta}$, which is a valid component embedded with tr' ; and (iii) a colluding user contributes $\{\forall j \in S : D_j = g^{r'} \cdot H(j)^{r_j}, D'_j = g^{r_j}\}$, which are binded by a random r' , which is different from tr' in D . Since the t is the exponent of the generator g , deriving it is equivalent to solve the DLP problem, which is considered to be hard. Thus, given the security of secret sharing and hardness of DLP on \mathbb{G}_0 and \mathbb{G}_1 , ESP and DSP cannot derive $e(g, g)^{\alpha s_2}$ or $e(g, g)^{\alpha s}$ even if they collude.

Strictly speaking, the following theorem proves that this scheme holds the collusion key security against the blinded key attacks (KS-BKA) based on the DLDH assumption:

Theorem 2: Let \mathcal{E} is a PP-CP-ABE scheme. If DLDH is (t, q, ε) -hard on \mathbb{G} , then the PP-CP-ABE scheme is (t, q, ε) -secure against the blinded key attacks. Here q is the number of hash function queries made by the adversary, and t is the run time of attacks.

B. Performance Evaluation

To evaluate the performance of the presented PP-CP-ABE scheme, we evaluate the computation overhead of service providers and users based on both theoretical analysis and experimental results. In our experimental analysis, we compared the computing overhead of various cryptographic operations in PC, Pocket PC and mobile sensors. The result showed that,

TABLE II
NUMBER OF CRYPTOGRAPHIC OPERATIONS COMPUTED BY ESP AND USER

	Exp $\mathbb{G}_0/\mathbb{G}_1$	Mul \mathbb{G}_1	Hash to \mathbb{G}_0
ESP	$2a_1/0$	0	a_1
User	3/1	1	1

TABLE III
NUMBER OF CRYPTOGRAPHIC OPERATIONS COMPUTED BY DSP AND USER

	Exp \mathbb{G}_1	Mul \mathbb{G}_1	Inv \mathbb{G}_1	Pairing
DSP	a_1	$2a_1$	a_1	$2a_1 + 1$
User	1	2	1	0

without outsourcing, it is rather infeasible for the resource constrained devices to perform the operations.

Firstly, we analyzed the number of expensive cryptographic operations over \mathbb{G}_0 and \mathbb{G}_1 , i.e., pairing, exponentiation, multiplication, performed by service providers and users' devices. In our analysis we assume that the access policy T_{ESP} has a_1 attributes connected by an AND logical gate and T_{DO} only has 1 attribute. In addition, the root node is an AND gate.

In the Table V-B, we compared the number of exponentiations, multiplications and hash to \mathbb{G}_0 operations incurred on ESP side and user side in the encryption outsourcing, where a_1 is the number of attributes in T_{ESP} :

We also provided a comparison of the number of exponentiations, multiplications, inversion, and pairing operations incurred by decryption outsourcing on DSP side and user side as shown in the Table V-B, where a_1 is the number of attributes in T_{ESP} :

From the above analysis, we can see that the computation overhead is linear for service providers (ESP and DSP) and constant for the user. Among all operations, pairing and ECC operations are most computationally intensive. We conducted the experimental evaluation of cryptographic pairing and ECC operations on a wireless Mote sensor (8 bit-7.37 MHz AT-Mega128L, 4KB RAM), a pocket pc (600 MHz CPU) and a PC (1GHZ CPU). The testing environments and results are listed in the Table V-B:

The result in Table V-B showed that, without outsourcing, it is rather infeasible for the resource constrained devices to perform the operations. To show that PP-CP-ABE can offload most of the computation overhead from user to service providers., we implemented and evaluated the PP-CP-ABE on a PC with 1.6GHz Intel Atom processor running Linux 2.6.32. The computation time was measured using clock ticks returned by `clock_t clock(void)` function in standard C library. To illustrate that most of the computation overhead is outsourced to service providers, we run the user and server on the same platform and recorded the number of clock ticks. In the Figure 5, we compared computation overhead incurred on service providers and users in encryption and decryption

TABLE IV
COMPUTING TIME OF CRYPTOGRAPHIC OPERATIONS ON EMBEDDED DEVICES

	Pairing	Exp \mathbb{G}_0	Mul \mathbb{G}_0
PC (1GHZ CPU)	20 ms	5 ms	0.7 ms
Pocket PC (600 MHz CPU)	550 ms	177 ms	26 ms
Sensor (8 × 8MHZ)	31250 ms	10720 ms	196 ms

outsourcing. The computation overhead was calculated in terms of 10 based logarithms, i.e., \log_{10} , of thousands (K) clocks ticks. As we can see from the figure, more than 90% of encryption and more than 99% of decryption computation are performed by the service providers.

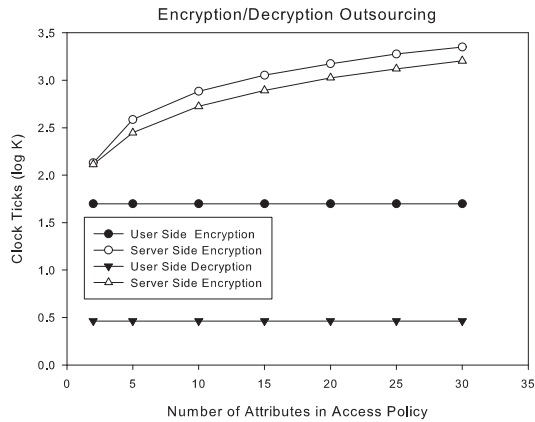


Fig. 5. Performance evaluation of the encryption and decryption outsourcing.

VI. RELATED WORKS

Existing works related to our proposed schemes includes (i) attribute based encryption and (ii) cryptographic access control over untrusted storage.

Attribute Based Encryption (ABE) was first proposed as a fuzzy version of IBE in [27], where an identity is viewed as a set of descriptive attributes. There are two main variants of ABE proposed so far, namely Key Policy Attribute Based Encryption (KP-ABE [16]) and Ciphertext Policy Attribute Based Encryption (CP-ABE [3]). In KP-ABE, each ciphertext is associated with a set of attributes and each user's private key is embedded with an access policy. Decryption is enabled only if the attributes on the ciphertext satisfy the access policy of the user's private key. In CP-ABE [3], [7], [21], [32], each user has a set of attributes that associate with user's private key and each ciphertext is encrypted by an access policy. To decrypt the message, the attributes in the user private key need to satisfy the access policy. CP-ABE is more appealing since it is conceptually closer to the Role Based Access Control (RBAC) [28] model.

Cryptographic access control over untrusted storage is investigated in both cryptography community and networking community. In cryptography community, Broadcast Encryption (BE) was introduced by Fiat and Naor in [13]. Compared with traditional one-to-one encryption schemes, BE is very efficient. Based on tradeoffs between key storage and ciphertext storage overhead, existing BE schemes can be generally categorized into the following classes: (i) constant ciphertext, linear public and/or private key on number of total receivers [5]; (ii) linear ciphertext on number of revoked receivers, constant (or logarithm) public and/or private key, [10], [25], [4]; (iii) sub-linear ciphertext, sub-linear public and/or private key [5]. In this work, I proposed a new construction of attribute based data storage (ABDS) scheme to address the deficiency of all 3

class existing works. Particularly, ABDS supports any arbitrary number of receivers with much lower complexity of storage and communication.

In networking community, various encrypted file systems [20], [2], [11] were proposed to secure data over untrusted storage. Particularly, in [2], the authors proposed a distributed storage scheme where users outsource encryption to a semi-trusted re-encryption server. However, if the server colludes with some malicious user, the data secrecy will be compromised completely. Compared with this scheme, the proposed PP-CP-ABE is secure even if service providers and malicious users collude. Recently, Yu et al. [33] proposed a security framework for cloud computing based on CP-ABE. Compared with the proposed work, their solution requires the users to disclose part of original private key to the cloud while the proposed solution only send blinded private keys. Moreover, the proposed solution specially considers mobile cloud environments and their work.

Data security in public cloud is an emerging research area [31], [35], [8], [30], [12], [23], [34], [15], [14], [17], [26], [22]. With the fast development of wireless technology, mobile cloud has become an emerging cloud service model [18], in which mobile devices and sensors are used as the information collecting and processing nodes for the cloud infrastructure. This new trend demands researchers and practitioners to construct a trustworthy architecture for mobile cloud computing, which includes a large numbers of lightweight, resource-constrained mobile devices.

While data integrity and retrievability in the cloud are also important security requirements, they are not the focuses of this dissertation. Readers can refer to research works in the provable data possession (PDP) [1], [12].

VII. CONCLUSION

In conclusion, we proposed a secure data inquiry service architecture for mobile cloud computing. Especially, our solution enables lightweight wireless devices to securely store and retrieve their data in public cloud with minimal cost. To this end, we proposed a novel Privacy Preserving Cipher Policy Attribute-Based Encryption (PP-CP-ABE) to protect users' encrypted data. Using PP-CP-ABE, light-weight devices can securely outsource intensive encryption and decryption operations to cloud service providers, without revealing the data content and used security keys. Also, we proposed an Attribute Based Data Storage (ABDS) system as a cryptographic access control mechanism. ABDS achieve information theoretically optimal in terms of minimizing computation, storage and communication overheads. Especially, ABDS minimize cloud costs charged by cloud service providers as well as communication overhead for data managements. Our performance assessments demonstrate the security strength and efficiency of our solution in terms of computation, communication, and storage.

ACKNOWLEDGEMENT

This work is supported by the Office of Naval Research's (ONR) Young Investigator Program (YIP).

REFERENCES

- [1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 598–609. ACM, 2007.
- [2] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, 2006.
- [3] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pages 321–334, Washington, DC, USA, 2007. IEEE Computer Society.
- [4] D. Boneh, X. Boyen, and E.J. Goh. Hierarchical identity based encryption with constant size ciphertext. *Advances in Cryptology—EUROCRYPT 2005*, pages 440–456, 2005.
- [5] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Advances in Cryptology—CRYPTO 2005*, pages 258–275. Springer, 2005.
- [6] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. pages 535–554. Springer, 2007.
- [7] L. Cheung and C. Newport. Provably secure ciphertext policy abe. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 456–465, New York, NY, USA, 2007. ACM.
- [8] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina. Controlling data in the cloud: outsourcing computation without outsourcing control. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 85–90. ACM, 2009.
- [9] B.G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. Clonecloud: Elastic execution between mobile device and cloud. In *Proceedings of the sixth conference on Computer systems*, pages 301–314. ACM, 2011.
- [10] C. Delerablée, P. Paillier, and D. Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. *Pairing-Based Cryptography—Pairing 2007*, pages 39–59.
- [11] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Over-encryption: management of access control evolution on outsourced data. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 123–134. VLDB Endowment, 2007.
- [12] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia. Dynamic provable data possession. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 213–222. ACM, 2009.
- [13] A. Fiat and M. Naor. Broadcast Encryption, *Advances in Cryptology—Crypto93. Lecture Notes in Computer Science*, 773:480–491, 1994.
- [14] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 169–178. ACM, 2009.
- [15] C. Gentry. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3):97–105, 2010.
- [16] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98, 2006.
- [17] F. Hsu and H. Chen. Secure file system services for web 2.0 applications. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 11–18. ACM, 2009.
- [18] D. Huang, X. Zhang, M. Kang, and J. Luo. Mobicloud: A secure mobile cloud framework for pervasive mobile computing and communication. In *Proceedings of 5th IEEE International Symposium on Service-Oriented System Engineering*, 2010.
- [19] Dijiang Huang. Mobile cloud computing. *IEEE COMSOC Multimedia Communications Technical Committee (MMTC) E-Letter*, 6(10):27–31, October 2011.
- [20] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. Plutus: Scalable secure file sharing on untrusted storage. In *FAST '03: Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, pages 29–42, 2003.
- [21] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT'08: Proceedings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology*, pages 146–162, Berlin, Heidelberg, 2008. Springer-Verlag.
- [22] E. Keller, J. Szefer, J. Rexford, and R.B. Lee. NoHype: virtualized cloud infrastructure without the virtualization. In *Proceedings of the 37th annual international symposium on Computer architecture*, pages 350–361. ACM, 2010.
- [23] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou. Fuzzy keyword search over encrypted data in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–5. IEEE.
- [24] E.J. McCluskey. Minimization of Boolean functions. *Bell System Technical Journal*, 35(5):1417–1444, 1956.
- [25] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. *Lecture Notes in Computer Science*, pages 41–62, 2001.
- [26] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 199–212. ACM, 2009.
- [27] A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. *Advances in Cryptology—Eurocrypt*, 3494:457–473.
- [28] RS Sandhu, EJ Coyne, HL Feinstein, and CE Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996.
- [29] A. Shamir. How to Share a Secret. *Communications of the ACM*, 22(11):612–613, 1979.
- [30] C. Wang, Q. Wang, K. Ren, and W. Lou. Privacy-preserving public auditing for data storage security in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.
- [31] W. Wang, Z. Li, R. Owens, and B. Bhargava. Secure and efficient access to outsourced data. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 55–66. ACM, 2009.
- [32] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. *ePrint report*, 290, 2008.
- [33] S. Yu, C. Wang, K. Ren, and W. Lou. Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. *INFOCOM'1010. Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, pages 1–9, 2010.
- [34] S. Yu, C. Wang, K. Ren, and W. Lou. Attribute based data sharing with attribute revocation. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 261–270. ACM, 2010.
- [35] A. Yun, C. Shi, and Y. Kim. On protecting integrity and confidentiality of cryptographic file system for outsourced storage. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 67–76. ACM, 2009.