Comparison-Based Encryption for Fine-grained Access Control in Clouds

Yan Zhu[1,2], Hongxin Hu[3], Gail-Joon Ahn[3], Mengyang Yu[1,2], Hongjia Zhao[1,2]
[1] Institute of Computer Science and Technology, Peking University, Beijing, 100871, China
[2] Beijing Key Laboratory of Internet Security Technology, Peking University, Beijing, 100871, China
[3] Laboratory of Security Engineering for Future Computing (SEFCOM), Arizona State University, Tempe, Arizona, 85281, USA
{yan.zhu,myyu,zhaohj}@pku.edu.cn; {hxhu,gahn}@asu.edu

ABSTRACT

Access control is one of the most important security mechanisms in cloud computing. However, there has been little work that explores various comparison-based constraints for regulating data access in clouds. In this paper, we present an innovative comparison-based encryption scheme to facilitate fine-grained access control in cloud computing. By means of forward/backward derivation functions, we introduce comparison relation into attribute-based encryption to implement various range constraints on integer attributes, such as temporal and level attributes. Then, we present a new cryptosystem with dual decryption to reduce computational overheads on cloud clients, where the majority of decryption operations are executed in cloud servers. We also prove the security strength of our proposed scheme, and our experiment results demonstrate the efficiency of our methodology.

Categories and Subject Descriptors

D.4.6 [**Operation Systems**]: Security and Protection—*Access controls, Cryptographic controls*; E.3 [**Data Encryption**]: Public key cryptosystems

General Terms

Security, Theory, Verification

Keywords

Access Control, Cryptography, Integer Comparison, Dual Decryption, Attribute-Based Encryption, Cloud

1. INTRODUCTION

The emerging cloud-computing paradigm is rapidly gaining momentum as an alternative to traditional information technology due to the reason that it provides an extensible and powerful environment for growing amounts of services and data. One fundamental aspect of this paradigm shifting is that data storage and processing are being outsourced into the cloud. However, cloud computing is also facing many challenges for data security as the users outsource their sensitive data to clouds, which are generally beyond the same trusted domain as data owners.

To address such a problem, access control is considered as one of critical security mechanisms for data protection in cloud applications. Unfortunately, traditional data access control approaches usually assume that data is stored in a trusted data server for all users. This assumption however no longer holds in cloud computing since the data owners and cloud servers are very likely to be in different domains. Hence, attribute-based encryption (ABE) has been introduced into cloud computing to encrypt outsourced sensitive data in terms of access policy on attributes describing the outsourced data, and only authorized users can decrypt and access the data [5, 9, 10, 12, 14, 20]. Since the access control policy of every object is embedded within it, the enforcement of policy becomes an inseparable characteristic of the data itself. This is in direct contrast to most currently access control systems, which rely upon a trusted host to mediate access and maintain policies.

Challenges. Although there have been some attempts to construct fine-grained access control systems in clouds, existing work lacks a systematic mechanism to support a complete comparison relation, $<, >, \leq, \geq$, in policy specification. In particular, to realize integer comparisons in ABE, Bethencourt *et al.* [5] proposed a naive approach, called as BSW's scheme, by using Bitwise-comparison operators based on AND/OR operators. However, this scheme has following shortcomings:

- It cannot support dual comparative expressions, where two range-based comparative constraints must be embedded into the outsourced files as well as the user's private key. For example, we cannot generate a user's private key with a range $4 \leq Month \leq 10$, which is particularly useful for representing fine-grained policies.
- It cannot support efficient cryptographic comparison methods. In Bitwise-comparison, the sizes of user's key and ciphertext are very large because the integer must be split into bits, and this causes higher computational costs of both encryption and decryption.
- All algorithms in existing scheme are run in a stand-alone mode, and the overheads of running those algorithms are big due to the sophisticated bilinear pairing operations,

CODASPY'12, February 7–9, 2012, San Antonio, Texas, USA.

especially for decryption. Hence, such a system is unsuitable for lightweight cloud clients, such as mobile devices, in a cloud environment.

To address those limitations, it is critical to investigate a more comprehensive solution to enable fine-grained expressions of range constraints in ABE-based systems.

Contributions. In this paper, we attempt to construct a new cryptosystem to explore richer attribute expressions in access control policies, especially for range constraints, and efficient support for lightweight clients in clouds.

Our contributions in this work are summarized as follows:

- We define and construct two new cryptographic functions, forward and backward deviation functions, to solve integer comparison problem. By avoiding complex bitwise comparison, our comparison method incorporates the expression of integer range in user's private key, as well as enables the security based on one-way function.
- We present a novel comparison-based encryption (CBE) scheme to enable fine-grained access control in cloud computing, which not only provides O(1) size of private-key and ciphertext for each range attribute, but also supports the provable security under RSA and CDH assumption.
- We introduce a cryptosystem with key delegation and dual decryption structure to reduce computational overheads on lightweight devices by shifting the majority of decryption operations to cloud servers. We prove that this structure is secure against various chosen derivation-key attacks.
- We implement a prototype of CBE system to evaluate our proposed approach. Our experimental results not only validate the efficiency of our scheme and algorithms, but also verify that the decryption overheads is effectively apportioned over cloud servers and clients.

This paper is organized as follows. We define the basic notation in Section 2. In Section 3, we introduce the framework of CBE cryptosystem and corresponding security requirements. Section 4 shows how the CBE scheme can be constructed. In Section 5, we discuss how to apply CBE for achieving fine-grained access control in clouds. Section 6 gives the security analysis of our scheme. We evaluate the performance of our scheme in Section 7. Finally, we overview the related work in Section 8, and conclude this paper in Section 9.

2. PRELIMINARIES

First, we establish the notation used in this paper. In many practical scenarios, the users may be restricted to access resources at a predefined level, range or period. For example, a user wishes to send an important notice which remains valid until a certain date, or a university permits the plumbers to check water-pipe into some areas during the first three days of each month. Hence, *range (or period) constraints* are used to specify the exact intervals during which an action can be enabled or disabled for a certain resource. We can represent the constraint by an integer attribute A_t with interval $[t_i, t_j]$, where $[t_i, t_j]$ is a range (or interval) denoting the lower (e.g. beginning time) and upper (e.g. ending time) bounds for the instants in A_t.

On the other hand, in order to realize comparison-based access control, a user is also assigned a digital certificate (called *access privilege*) which includes an integer attribute A_t. For example, as the definition in the X.500 standard, we assume that each user is assigned a licence with a time interval $[t_a, t_b]$ for a certain attribute A_t. Specially, given a range constraint $[t_i, t_j]$ and an access privilege $[t_a, t_b]$ on the same attribute A_t, we must satisfy the following criterion:

Definition 1 (Comparison Criterion). *Given an access constraint $t_i \leq A_t \leq t_j$ for the protected resources and a privilege $t_a \leq A_t \leq t_b$ in the user's certificate, secure data access control must guarantee that the user can be permitted to access the resources if and only if $[t_i, t_j] \cap [t_a, t_b] \neq \emptyset$.*

This requirement is necessary for integer or level attributes in attribute-based access control, in which we define the policy with range constraints to specify the exact intervals during which an event can be enabled or disabled by matching the user's certificate. Further, we introduce this requirement into attribute-based encryption to define the comparison criterion of integer or level attributes.

3. COMPARISON-BASED ENCRYPTION

3.1 Definition of Fine-grained Access Control with Comparison

In mathematics, the ordering imposed on a set of elements U is said to be a *total ordering relation* or *chain* if and only if every two elements are comparable in U. The set of integer, ordered usually by the $\leq, \geq$ (or $<, >$) relations, is totally ordered as the subsets of natural numbers and rational numbers. It is obvious that some attributes, such as level, time, and position location, also satisfy the total ordering relation or monotone, which can be mapped into consecutive integers. So that we consider the values of these attributes as a countable set constituted in the range $[0, Z]$, $U = \{t_1, \cdots, t_T\} \subseteq [0, Z]$. Based on this ordering relation on U, we define an attribute-based access control with comparison operations as follows:

- $\mathcal{A}$: the set of attributes $\mathcal{A} = \{A_1, \cdots, A_m\}$;
- $A_k(t_i, t_j)$: the range constraint of attribute A_k on $[t_i, t_j]$, i.e., $t_i \leq A_k \leq t_j$;
- $\mathcal{P}$: the access control policy expressed as a Boolean function on AND/OR logical operations, generated by the grammar: $\mathcal{P} ::= A_k(t_i, t_j) | \mathcal{P}$ AND $\mathcal{P} | \mathcal{P}$ OR $\mathcal{P}$; and
- $\mathcal{L}$: the access privilege assigned into the user's certificate, generated by $\mathcal{L} ::= \{A_k(t_a, t_b)\}_{A_k \in \mathcal{A}}$.

3.2 Framework of CBE Cryptosystem

With the focus on comparison-based access control and dual-decryption mechanism in cloud environment, a comparison-based encryption (CBE) consists of six algorithms as follows:

- Setup$(1^\kappa, \mathcal{A})$: Takes a security parameter κ as input, outputs the master key MK and the public-key $PK_{\mathcal{A}}$;
- GenKey$(MK, u_k, \mathcal{L})$: Takes the user's ID number u_k as the input, the access privilege $\mathcal{L}$ and MK, outputs the user's private key $SK_{\mathcal{L}}$;

- Encrypt$(PK_{\mathcal{A}}, \mathcal{P})$: Takes a comparable access policy $\mathcal{P}$ and PK as input, outputs the ciphertext header $\mathcal{H}_{\mathcal{P}}$ and a random session key ek;
- Delegate$(SK_{\mathcal{L}}, \mathcal{L}')$: Takes a private key $SK_{\mathcal{L}}$ and a specified privilege requirement $\mathcal{L}'$ as input, outputs a derivation key $\widetilde{SK}_{\mathcal{L}'}$ if each attribute in $\mathcal{L}$ and $\mathcal{L}'$ satisfies the above-mentioned comparison criterion;
- Decrypt1$(\widetilde{SK}_{\mathcal{L}'}, \mathcal{H}_{\mathcal{P}})$: Takes a user's private key $\widetilde{SK}_{\mathcal{L}'}$ and a ciphertext header $\mathcal{H}_{\mathcal{P}}$ as input, outputs a new ciphertext header $\widetilde{\mathcal{H}}_{\mathcal{P}}$ if $\mathcal{L}'$ satisfies the constraint of $\mathcal{P}$; and
- Decrypt2$(SK_{\mathcal{L}}, \widetilde{\mathcal{H}}_{\mathcal{P}})$: Takes a user's private key $SK_{\mathcal{L}}$ and a ciphertext header $\widetilde{\mathcal{H}}_{\mathcal{P}}$ as input, outputs a session key ek which can be used to decrypt the stored data.

With the help of this framework, a workflow of CBE-based cryptosystem for clouds is depicted in Figure 1. For sake of clarity, the operations on the data are not shown in the framework since data owner could easily employ traditional symmetric key cryptosystem to encrypt and then outsource data with the help of a random session key ek.

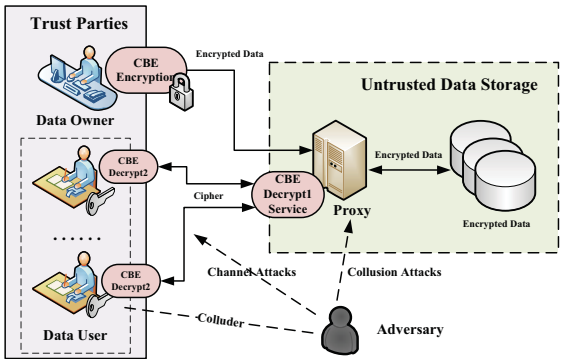

Figure 1: Workflow of CBE-based Cryptosystem for Clouds.

- First, the system manager establishes a CBE cryptosystem by invoking the *Setup* algorithm, and then assigns a private key $SK_{\mathcal{L}}$ on a specified access privilege $\mathcal{L}$ to each user in this system by the *GenKey* algorithm;
- For each file needing to store in the cloud, the data owner specifies an access control policy $\mathcal{P}$ to encrypt data by using the *Encrypt* algorithm before it leaves from the cloud client;
- Anytime a user can send a request to the proxy to access a stored file in the cloud.
  1. After obtaining the policy $\mathcal{P}$ embedded in ciphertext header $\mathcal{H}_{\mathcal{P}}$, the proxy extracts the necessary privilege $\mathcal{L}'$ from $\mathcal{P}$ and sends $\mathcal{L}'$ to the user.
  2. The user invokes the *Delegate* algorithm to generate a temporary derivation key $\widetilde{SK}_{\mathcal{L}'}$ for $\mathcal{L}'$ and returns it to the proxy.
  3. The proxy makes use of $\widetilde{SK}_{\mathcal{L}'}$ to convert $\mathcal{H}_{\mathcal{P}}$ into a new $\widetilde{\mathcal{H}}_{\mathcal{P}}$ by using the *Decrypt*1 algorithm and sends it to the user.
  4. The user invokes the *Decrypt*2 algorithm to decrypt $\widetilde{\mathcal{H}}_{\mathcal{P}}$ to get the session key ek.

To reduce the user's decryption overheads, decryption in this framework is converted into an interactive decryption protocol consisted by three algorithms: Delegate, Decrypt1, and Decrypt2.

3.3 Security Requirements

In our framework, we are concerned with the security risks from data users or service providers as follows:

Data users: In our framework, the malicious users cannot observe the encrypted data stored in outsourced storages, thus they cannot directly attack to the ciphertext header $\mathcal{H}_{\mathcal{P}}$. However, the malicious users could try to make use of the *Delegate* algorithm to access files. To do so, they can change the range of his privileges independently or cooperatively. We are certainly more concerned with the second case, which is called *collusion privilege attack*.

Another attack is based on the fact that the malicious users can increase their capabilities of attack by observing the derivation keys from channel. It is a potential threat because the derivation keys, directly derived from the valid private keys, involves enough information of access privileges. Based on this threat, we define a security game to describe *key security under chosen derivation-key attacks* (KS-CDA):

Setup: The challenger runs the Setup algorithm and gives the public parameters to the adversary.

Learning: The adversary is allowed to choose a range attribute A_t and query the *Delegate* algorithm with the polynomial number of users $u_{k_1}, \cdots, u_{k_s}$ with any interval $A_t[t_{k_i}, t_{k_i'}] \in \mathcal{L}_k$. The challenger responds the corresponding keys $\{\widetilde{SK}_{\mathcal{L}_k}\}$ to the adversary.

Challenge: The challenger sends a challenge private key $SK_{\mathcal{L}^*}$ of user u^* to the adversary, where $A_t[t_i, t_j] \in \mathcal{L}^*$ and the user u^* is not queried before.

Response: The adversary outputs a private key $SK_{\mathcal{L}'}$ corresponding to u^*. If this key is valid and $\mathcal{L}'$ has more privileges than $\mathcal{L}^*$, the adversary wins this game.

Proxy: Similarly to the solution proposed in [20], we just consider the "Honest but Curious" proxy server assumption, that is, the proxy will honestly follow our proposed algorithms in general, but try to find out as much secret information as possible based on the inputs. More specifically, we assume the attacker is more interested in the stored data (by obtaining the session key to decrypt the data) and the user's private key than other secret information.

Further, attackers will also try to obtain as much prior knowledge as possible to help them break the encryption or forge the private key. To better evaluate this attack, we also define a security game to describe the *semantical security under chosen derivation-key attacks* (SS-CDA):

Setup: The challenger runs the Setup algorithm and gives the public parameters to the adversary.

Learning: The adversary is allowed to choose a range attribute A_t and query the *Delegate* algorithm with the polynomial number of users $u_{k_1}, \cdots, u_{k_s}$ with any interval $A_t[t_{k_i}, t_{k_i'}] \in \mathcal{L}_k$. The challenger responds the corresponding keys $\{\widetilde{SK}_{\mathcal{L}_k}\}$ to the adversary.

Challenge: The challenger sends a challenge ciphertext $\mathcal{H}_{\mathcal{P}}{}^*$ to the adversary, where all $A_t[t_i, t_j] \in \mathcal{P}$ are queried before.

Response: The adversary outputs a session key ek^* corresponding to $\mathcal{H}_{\mathcal{P}}{}^*$. If this key is valid, the adversary wins this game.

In our framework, the proxy need not to keep track of all access queries, or the system works in an anonymous manner. Therefore, we consider that the attackers also work in an anonymous environment.

4. CONSTRUCTION OF CBE SCHEME

In this section, we propose a novel construction for integer comparison to overcome the limitations of BSW's CP-ABE scheme. We first give the background on composite order bilinear groups. Then, we present two key constructions: forward and backward derivation functions. Finally, we present the construction of our CBE scheme based on those techniques.

4.1 Composite Order Bilinear Map

We set up our systems using bilinear pairings introduced by Boneh and Franklin [6, 7]. We define a bilinear map group system $\mathbb{S} = (N = pq, \mathbb{G}, \mathbb{G}_T, e)$, where $N = pq$ be the RSA-modulus, p, q are two large primes, $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic groups with order $n = sp'q'$ [1], and e be a computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties:

- Bilinearity: for any $g, h \in \mathbb{G}$ and all $a, b \in \mathbb{Z}$, $e(g^a, h^b) = e(g, h)^{ab}$;
- Non-degeneracy: $e(g, h) \neq 1$ whenever g and h are the generators of group $\mathbb{G}$; and
- Computability: $e(g, h)$ is efficiently computable.

In this system, we make N public and keep n, s, p', q' secret.

Let $\mathbb{G}_s$ and $\mathbb{G}_{n'}$ denote the subgroups of order s and $n' = p'q'$ in $\mathbb{G}$, respectively. We note that when $g \in \mathbb{G}_s$ and $h \in \mathbb{G}_{n'}$, $e(g, h)$ is the identity element in $\mathbb{G}_T$. To see this, suppose w denote a generator of $\mathbb{G}$, then, $w^{n'}$ generates $\mathbb{G}_s$ and w^s generates $\mathbb{G}_{n'}$. Hence, for some k_1, k_2, $g = (w^{n'})^{k_1}$ and $h = (w^s)^{k_2}$, we have

$$e(g, h) = e((w^{n'})^{k_1}, (w^s)^{k_2}) = e(w^{k_1}, w^{k_2})^{sn'} = 1.$$

This orthogonality property of $\mathbb{G}_s$ and $\mathbb{G}_{n'}$ will be used to implement the comparison mechanism in our constructions.

4.2 Forward/Backward Derivation Functions

CBE scheme utilizes "one-way" property to represent the total ordering relation of integers. This means that given the integer relation $t_i \leq t_j$ and two corresponding values v_{t_i}, v_{t_j}, there exists an efficient algorithm to obtain v_{t_j} from v_{t_i}; however, it is hard to compute v_{t_i} from v_{t_j}. Based on this idea, we formally define the forward and backward derivation functions.

[1]Without loss of generality, we have $n = sn' = s_1 s_2 p'q' | lcm(p+1, q+1)$, $n' = p'q' | n$, $s = s_1 s_2$, $p = 2p's_1 - 1$, $q = 2q's_2 - 1$, and s_1, s_2, p', q', p, q are some secret large primes.

Let comparable variables be denoted as a countable set $U = \{t_1, t_2, \cdots, t_T\}$ constituted from the discrete consecutive integers with total ordering $0 \leq t_1 \leq t_2 \leq \cdots \leq t_T \leq Z$, where Z is the maximum integer. In order to construct a cryptographic algorithm for integer comparison, we use a cryptographic map $\psi : U \to V$, where $V = \{v_{t_1}, \cdots, v_{t_T}\}$ is a set of cryptographic values. It is obvious that ψ must be an order-preserving map, that is a map such that if $t_i \leq t_j$ in U implies there exists a partial-order relation $\preceq$ to ensure $v_{t_i} \preceq v_{t_j}$ in V, where $v_{t_i} = \psi(t_i)$ and $v_{t_j} = \psi(t_j)$. In order to setup this kind of relation over V, we consider the partial-order relation in V as the "one-way" property in cryptography, which is defined as a forward derivation function:

DEFINITION 2 (FORWARD DERIVATION FUNCTION,FDF). *Given a function $f : V \to V$ based on a set $(U, \leq)$, it is called a forward derivation function if it satisfies the conditions:*

- ***Easy to compute**: the function f can be computed in a polynomial-time, if $t_i \leq t_j$, i.e., $v_{t_j} \leftarrow f_{t_i \leq t_j}(v_{t_i})$;*
- ***Hard to invert**: it is infeasible for any probabilistic polynomial (PPT) algorithm to compute v_{t_i} from v_{t_j} if $t_i < t_j$.*

Similarly, we also define a function $\bar{f}$ for the derivation in opposite direction, which is called *Backward Derivation Function* (BDF). In order to avoid interference between f and $\bar{f}$, we use a different sign $\bar{\psi} : U \to \bar{V}$, and then the BDF $\bar{f}$ is defined as follows:

DEFINITION 3 (BACKWARD DERIVATION FUNCTION,BDF). *Given a function $\bar{f} : \bar{V} \to \bar{V}$ based on a set $(U, \leq)$, it is called a forward derivation function if it satisfies the conditions:*

- ***Easy to compute**: the function $\bar{f}$ can be computed in a polynomial-time, if $t_i \geq t_j$, i.e., $\bar{v}_{t_j} \leftarrow \bar{f}_{t_i \geq t_j}(\bar{v}_{t_i})$;*
- ***Hard to invert**: it is infeasible for any probabilistic polynomial (PPT) algorithm to compute $\bar{v}_{t_i}$ from $\bar{v}_{t_j}$ if $t_i > t_j$.*

4.3 Cryptographic Construction of FDF/BDF

The cryptography construction for integer comparisons is constructed based on forward/backward derivation functions. This construction is built on a special multiplicative group $\mathbb{G}_{n'}$ of RSA-type composite order $n' = p'q'$, where p', q' are two large primes. First, we choose two different random generators $\varphi, \bar{\varphi}$ in a group $\mathbb{G}_{n'}$, where $\varphi^{n'} = \bar{\varphi}^{n'} = 1$. Next, we choose two different random λ and μ in $\mathbb{Z}^*_{n'}$, where the order of λ, μ are sufficiently large in $\mathbb{Z}^*_{n'}$.

Based on RSA cryptography system, we define two mapping functions $(\psi(\cdot), \bar{\psi}(\cdot))$ from an integer set $U = \{t_1, \cdots, t_T\}$ into $V = \{v_{t_1}, \cdots, v_{t_T}\}$ and $\bar{V} = \{\bar{v}_{t_1}, \cdots, \bar{v}_{t_T}\}$ as follows:

$$\begin{aligned} v_{t_i} &\leftarrow \psi(t_i) = \varphi^{\lambda^{t_i}} \in \mathbb{G}_{n'}, \\ \bar{v}_{t_i} &\leftarrow \bar{\psi}(t_i) = \bar{\varphi}^{\mu^{Z-t_i}} \in \mathbb{G}_{n'}. \end{aligned}$$

where, φ^{λ^t} denotes $\varphi^{(\lambda^t)}$ rather than $(\varphi^{\lambda})^t$. Note that, the values, $w_{t_i} = \lambda^{t_i}$ and $\bar{w}_{t_j} = u^{Z-t_j}$, can only be computed in the integer $\mathbb{Z}$ if n' are unknown. Next, according to the definition of $\psi(\cdot)$ and $\bar{\psi}(\cdot)$, it is easy to define the FDF $f(\cdot)$ and BDF $\bar{f}(\cdot)$ as

$$\begin{aligned} v_{t_j} &\leftarrow f_{t_i \leq t_j}(v_{t_i}) = (v_{t_i})^{\lambda^{t_j - t_i}} \in \mathbb{G}_{n'}, \\ \bar{v}_{t_j} &\leftarrow \bar{f}_{t_i \geq t_j}(\bar{v}_{t_i}) = (\bar{v}_{t_i})^{\mu^{t_i - t_j}} \in \mathbb{G}_{n'}. \end{aligned}$$

It is easy to show that $(\varphi^{\lambda^{t_i}})^{\lambda^{t_j-t_i}} = \varphi^{\lambda^{t_j}} = v_{t_j} \in \mathbb{G}_{n'}$ and $(\bar{\varphi}^{\mu^{Z-t_i}})^{\mu^{t_i-t_j}} = \bar{\varphi}^{\mu^{Z-t_j}} = \bar{v}_{t_j} \in \mathbb{G}_{n'}$. But it is intractable to obtain v_{t_i} from v_{t_j} for $t_i \leq t_j$ under the RSA assumption that λ^{-1} and μ^{-1} cannot be efficiently computed based on the secrecy of n' [2].

4.4 Proposed CBE Scheme

We use the above-mentioned FDF/BDF functions to construct an efficient CBE scheme with the range comparisons on integer attributes, as follows:

- *Setup*$(1^\kappa, \mathcal{A}) \rightarrow (MK, PK_{\mathcal{A}})$: Given a bilinear map system $\mathbb{S}_N = (N = pq, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot))$ of composite order $n = sn'$ and two subgroups $\mathbb{G}_s$ and $\mathbb{G}_{n'}$ of $\mathbb{G}$. This algorithm chooses the random generators $w \in \mathbb{G}$, $g \in \mathbb{G}_s$, and $\varphi, \bar{\varphi} \in \mathbb{G}_{n'}$, as well as two random $\lambda, \mu \in \mathbb{Z}_n^*$ as described in Section 4.3. Thus, we have $e(g,\varphi) = e(g,\bar{\varphi}) = 1$ but $e(g,w) \neq 1$. Additionally, the setup algorithm employs a hash function $H : \{0,1\}^* \rightarrow \mathbb{G}$, mapping any attribute described as a binary string to a random group element. Next, the setup algorithm chooses two random exponents $\alpha, \beta \in \mathbb{Z}_n^*$ and set $h = w^\beta, \eta = g^{1/\beta}, \zeta = e(g,w)^\alpha$. Finally, the setup algorithm outputs the public key

$$PK = (\mathbb{S}_N, g, h, \zeta, \eta, w, \varphi, \bar{\varphi}, \lambda, \mu, H(\cdot))$$

and the master key $MK = (g^\alpha, \beta, p, q, n')$.

- *GenKey*$(MK, u_k, \mathcal{L}) \rightarrow SK_{\mathcal{L}}$: Given a user u_k with license $\mathcal{L}$ on a set of attributes $S = \{A_t\} \subseteq \mathcal{A}$, the GenKey algorithm first chooses a unique integer τ_k to distinguish the different users. Assume that the user u_k is assigned a range attribute $A_t \in \mathcal{L}$ with the constraint $A_t[t_a, t_b]$, this algorithm chooses a random $r \in \mathbb{Z}$ and sets the user's attribute key as

$$(D_t, D'_{t_a}, \bar{D}'_{t_b}, D''_t)_{A_t[t_a,t_b]} = (g^{\tau_k} H(A_t)^r, (v_{t_a})^r, (\bar{v}_{t_b})^r, w^r),$$

where $v_{t_a} = \varphi^{\lambda^{t_a}} \in \mathbb{G}_{n'}$ and $\bar{v}_{t_b} = \bar{\varphi}^{\mu^{Z-t_b}} \in \mathbb{G}_{n'}$. The private key is

$$SK_{\mathcal{L}} = (D = g^{(\alpha+\tau_k)/\beta}, \{(D_t, D'_{t_a}, \bar{D}'_{t_b}, D''_t)\}_{A_t[t_a,t_b] \in \mathcal{L}}).$$

- *Encrypt*$(PK_{\mathcal{A}}, \mathcal{P}) \rightarrow (\mathcal{H}_{\mathcal{P}}, ek)$: Given an access policy tree $\mathcal{T}$ over access policy $\mathcal{P}$, the ciphertext can be composed of a ciphertext header

$$\mathcal{H}_{\mathcal{P}} = (\mathcal{T}, C = h^s, \{((\bar{E}_{t_i}, E'_{t_i}), (E_{t_j}, E'_{t_j}))\}_{A_t[t_i,t_j] \in \mathcal{T}})$$

and a session key $ek = \zeta^s = e(g^\alpha, w)^s$, where

$$\begin{aligned} &((\bar{E}_{t_i}, E'_{t_i}), (E_{t_j}, E'_{t_j}))_{A_t[t_i,t_j]} \\ &= (((\bar{v}_{t_i} w)^x, H(A_t)^x), (((v_{t_j} w)^y, H(A_t)^y)), \end{aligned}$$

where s is a main secret in $\mathbb{Z}_N$ for this tree $\mathcal{T}$, $\Delta_s(A_t) = x + y$ is the secret share of s in the tree $\mathcal{T}$ for an attribute A_t (see BSW's scheme).

- *Delegate*$(SK_{\mathcal{L}}, \mathcal{L}') \rightarrow \widetilde{SK}_{\mathcal{L}'}$: Given the private key $SK_{\mathcal{L}}$ and a specified $\mathcal{L}'$, this algorithm checks whether each attribute $A_t \in \mathcal{L}'$ holds $t_a \leq t_j$ and $t_b \geq t_i$ for $A_t[t_a, t_b] \in \mathcal{L}$ and $A_t[t_i, t_j] \in \mathcal{L}'$. If so, it computes

$$\begin{aligned} D'_{t_j} &\leftarrow f_{t_a \leq t_j}(D'_{t_a}) \cdot D''_t = f_{t_a \leq t_j}(v^r_{t_a}) \cdot w^r = v^r_{t_j} \cdot w^r, \\ \bar{D}'_{t_i} &\leftarrow f_{t_b \geq t_i}(\bar{D}'_{t_b}) \cdot D''_t = \bar{f}_{t_b \geq t_i}((\bar{v}_{t_b})^r) \cdot w^r = \bar{v}^r_{t_i} \cdot w^r, \end{aligned}$$

where,

$$\begin{aligned} f_{t_a \leq t_j}(v^r_{t_a}) &= (\varphi^{r\lambda^{t_a}})^{\lambda^{t_j - t_a}} = \varphi^{r\lambda^{t_j}} = v^r_{t_j}, \\ f_{t_b \geq t_i}(\bar{v}^r_{t_b}) &= (\bar{\varphi}^{r\mu^{Z-t_b}})^{\mu^{t_b - t_i}} = \bar{\varphi}^{r\mu^{Z-t_i}} = \bar{v}^r_{t_i}. \end{aligned}$$

Next, it chooses a random $\delta \in \mathbb{Z}$ and computes $\widetilde{SK}_{\mathcal{L}'} = \{\widetilde{D}_t, \widetilde{D}'_{t_j}, \widetilde{\bar{D}'}_{t_i}\}_{A_t \in \mathcal{L}'}$, where, $\tau'_k = \tau_k + \delta$, $r' = r + \delta$,

$$\begin{aligned} \widetilde{D}_t &= D_t \cdot (gH(A_t))^\delta = g^{\tau_k + \delta} H(A_t)^{r+\delta} = g^{\tau'_k} H(A_t)^{r'}, \\ \widetilde{D}'_{t_j} &= D'_{t_j} \cdot (v_{t_j} w)^\delta = (v_{t_j} w)^{r+\delta} = (v_{t_j} w)^{r'}, \\ \widetilde{\bar{D}'}_{t_i} &= \bar{D}'_{t_i} \cdot (\bar{v}_{t_i} w)^\delta = (\bar{v}_{t_i} w)^{r+\delta} = (\bar{v}_{t_i} w)^{r'}. \end{aligned}$$

Finally, it outputs $\widetilde{SK}_{\mathcal{L}'}$ as the derivation key for $\mathcal{L}'$.

- *Decrypt*1$(\widetilde{SK}_{\mathcal{L}'}, \mathcal{H}_{\mathcal{P}}) \rightarrow \widetilde{\mathcal{H}}_{\mathcal{P}}$: Given the private key $\widetilde{SK}_{\mathcal{L}'}$ and a ciphertext header $\mathcal{H}_{\mathcal{P}}$, this algorithm also check whether each range attribute $A_t[t_i, t_j] \in \mathcal{L}'$ is consistent with $A_t[t_i, t_j] \in \mathcal{P}$. If true, the secret share $\Delta_s(A_t)$ of s over $\mathbb{G}_T$ is reconstructed by using

$$\begin{aligned} F_1 &\leftarrow \frac{e(\widetilde{D}_t, E_{t_j})}{e(\widetilde{D}'_{t_j}, E'_{t_j})} = \frac{e(g^{\tau'_k} H(A_t)^{r'}, (v_{t_j} w)^x)}{e((v_{t_j} w)^{r'}, H(A_t)^x)} \\ &= \frac{e(g^{\tau'_k}, (v_{t_j} w)^x) \cdot e(H(A_t)^{r'}, (v_{t_j} w)^x)}{e((v_{t_j} w)^{r'}, H(A_t)^x)} \\ &= e(g^{\tau'_k}, v^x_{t_j}) \cdot e(g^{\tau'_k}, w^x) = e(g^{\tau'_k}, w)^x, \\ F_2 &\leftarrow \frac{e(\widetilde{D}_t, \bar{E}_{t_i})}{e(\widetilde{\bar{D}'}_{t_i}, E'_{t_j})} = \frac{e(g^{\tau'_k} H(A_t)^{r'}, (\bar{v}_{t_i} w)^y)}{e((\bar{v}_{t_i} w)^{r'}, H(A_t)^y)} \\ &= \frac{e(g^{\tau'_k}, (\bar{v}_{t_i} w)^y) \cdot e(H(A_t)^{r'}, (\bar{v}_{t_i} w)^y)}{e((\bar{v}_{t_i} w)^{r'}, H(A_t)^y)} \\ &= e(g^{\tau'_k}, \bar{v}^y_{t_i}) \cdot e(g^{\tau'_k}, w^y) = e(g^{\tau'_k}, w)^y, \\ F_t &= F_1 \cdot F_2 = e(g^{\tau'_k}, w)^{\Delta_s(A_t)}, \end{aligned}$$

where, $e(g^{\tau'_k}, v^x_{t_j}) = e(g^{\tau'_k}, \bar{v}^y_{t_i}) = 1$ due to $g^{\tau'_k} \in \mathbb{G}_s$ and $v^x_{t_j}, \bar{v}^y_{t_i} \in \mathbb{G}_{n'}$. Next, the value $T = e(g^{\tau'_k}, w)^s$ is computed from $\{e(g^{\tau'_k}, w)^{\Delta_s(A_i)}\}_{A_i \in \mathcal{T}}$ by using the aggregation algorithm (see BSW's scheme). Finally, the new ciphertext header $\widetilde{\mathcal{H}}_{\mathcal{P}} = (C, T)$ is returned.

- *Decrypt*2$(SK_{\mathcal{L}}, \widetilde{\mathcal{H}}_{\mathcal{P}}) \rightarrow ek$: After receiving $\widetilde{\mathcal{H}}_{\mathcal{P}} = (C, T) = (w^{\beta s}, e(g^{\tau'_k}, w)^s)$, the decryptor uses the secret δ to compute $D' = D \cdot \eta^\delta = g^{(\alpha+\tau_k)/\beta} g^{\delta/\beta} = g^{(\alpha+\tau_k+\delta)/\beta} = g^{(\alpha+\tau'_k)/\beta}$. Next, the session key is computed by

$$ek = \frac{e(C, D')}{T} = \frac{e\big(g^{(\alpha+\tau'_k)/\beta}, (w^\beta)^s\big)}{e(g^{\tau'_k}, w)^s} = e(g^\alpha, w)^s.$$

For improving the efficiency, the output of this algorithm is a random session key ek instead of a plaintext because this key can be used to encrypt the object files using symmetrical-key cryptosystem.

Note that, it is also easy to combine *Decrypt*1 and *Decrypt*2 into one decryption algorithm, so that we can directly use that to decrypt the ciphertexts by using private keys.

[2]The secrecy of n' is similar to that of Euler's totient function $\phi(N)$ for RSA-type $N = pq$.

Table 1: Attribute lists for employee's working hours.

People	Period-of-Validity	Job	Level	Day	Hour
Anderaon	2009/01-2011/06	Manager,	4	Mon.-Fri.	9:00AM-14:00PM
Grant	2010/04-2010/12	Accountant	3	Thu.-Fri.	10:00AM-16:00PM
Kidman	2010/04-2011/06	Engineer	2	Mon.-Fri.	9:00AM-16:00PM
Coolidge	2010/01-2010/12	Retailer	2	Mon.-Wed.	9:00AM-16:00PM
Jones	2010/08-2010/12	Retailer	1	Mon.-Sat.	10:00AM-17:00PM

Table 2: Schedule for outsourced storage systems.

Files	Period-of-Validity	Job	Level	Day	Hour
Tech. Archive	2009/11-2010/03	Engineer	≥ 3	Mon.-Fri.	9:00AM-16:00PM
		Manager	≥ 4	(All)	16:00PM-9:00AM
Sales Record	2010/01-2010/03	Accountant OR Manager	≥ 3	Thu.-Fri.	(All)
Salary History	2010/05-2010/11	Manager	≥ 4	Mon.-Fri.	9:00AM-16:00PM
Service Log	2009/06-2010/04	Retail	≥ 1	Mon.-Fri.	9:00AM-16:00PM
		Engineer	≥ 3	(All)	16:00PM-9:00AM
Contact Info.	2010/11-2011/05	(All)	≥ 1	Mon.-Sat.	9:00AM-16:00PM

5. CBE FOR FINE-GRAINED ACCESS CONTROL IN CLOUDS

In this section, we demonstrate the usability of our CBE scheme for fine-grained access control in the cloud-based application systems. Especially, we discuss how our CBE scheme can be used to support various temporal constraints, including simple temporal constraints and periodic constraints.

5.1 Fine-grained Access Control in Clouds

Cloud-based service relieves the client's burden for storage management and maintenance by providing a comparably low-cost, scalable, location-independent platform. However, the fact that clients no longer have physical possession of data, indicates that they are facing a potentially formidable risk for abusing, coping, missing or corrupting data. Hence, an important issue for outsourced storage systems is to design an efficient approach to prevent the data stored at remote servers from unauthorized access.

The CBE scheme provides exactly an effective approach to regulate outsourced sensitive data, which enables only authorized users to access data based on the various attributes. We next give an example to show how CBE scheme can provide a fine-grained access control in clouds. We assume that a small business company, which has many retailers distributed across US, constructs their data centers using CBE on a platform as a service (PaaS) environment, e.g., Google's App Engine or Amazon's Elastic. These retailers may use the mobile handheld devices to access the data centers.

According to the employee's working hours, Table 1 illustrates a simple schedule for some employees, which consists of five attributes: Period-of-Validity, which is a time attribute on month basis; Job, which is a string attribute to denote employee's position; Level, which is an integer attribute to denote secret level for documents; Day and Hour, which are two period attributes to denote working days and hours. The system manager assigns the attribute values into the employee's private key by using CBE scheme.

In this example, the manager wishes to define the access policies to protect each component of their business information: Technique Archive, Sales Record, Salary History, Service Log, Contact Information. An access policy can be viewed as a description of attributes, which is used to match the attribute values in the employee's private key. As illustrated in Table 2, these attributes describe various functions and temporal constraints for these business information.

A document, like a technique archive, reposits the core technology of latest products, which is being manufactured in 2009/11-2010/03. Its policy therefore stipulates that only engineers with Level ≥ 3 can view this document during regular working hours (at 9:00AM-16:00PM from Monday to Friday), as well as the managers can view it at other times. Another example is the sales-record from 2010/01 to 2010/03, which can be accessed by accountant or manager with Level ≥ 3 from Thursday to Friday. As described above, these policies are represented as follows:

Technique Archive: *(2009/11 $\leq$ Period-of-Validity $\leq$ 2010/03)* ***AND*** *(((Engineer)* ***AND*** *(Level $\geq$ 3)* ***AND*** *(Monday$\leq$ Day $\leq$ Friday)* ***AND*** *(9:00AM $\leq$ Hour $\leq$ 16:00PM))* ***OR*** *((Manager)* ***AND*** *(Level $\geq$ 4)* ***AND*** *(16:00PM $\leq$ hour $\leq$ 9:00AM)))*

Sales Record: *(2010/01 $\leq$ Period-of-Validity $\leq$ 2010/03)* ***AND*** *(Accountant OR Manager)* ***AND*** *(Level$\geq$ 3)* ***AND*** *(Thursday$\leq$ Day $\leq$ Friday)*

It is obvious that our CBE construction can effectively implement these access policies by using integer comparison and temporal constraints. After the documents are encrypted in accordance with the above policies, only Manager Anderaon can access the technique archive, and Engineer Kidman cannot view it due to his invalid Period-of-Validity. Similarly, both Manager Anderaon and Accountant Grant can access Sales Record. The above-mentioned example implements the protection of encrypted files by using the attribute matching between ciphertext-policy and user's private key. In order to ensure the security of secrets in these files, the session key (see Figure 2) must be renewed whenever these files are updated, so that the employees whose private keys had lapsed cannot access them.

5.2 File Structure Based on CBE

In the CBE scheme, since the access control policy is embedded within each protected object, the enforcement of policy becomes an inseparable characteristic of the data itself. This is in direct contrast to most currently available systems, which depend on a trusted host to govern the data access and maintain policies. Such a file header $H_{\mathcal{P}}$ is shown in Figure 2, in which "Cipher" consists of constant-size C, "Access Policy" consists of $\mathcal{T}$, and "Encrypted Attribute List" denotes the attribute set $\{(E_{t_i}, E'_{t_i}), (E_{t_j}, E'_{t_j})\}$ in CBE-type

ciphertext. In this case, the number of users is not limited. Moreover, no file needs be changed to permit the access of existing files for a new user.

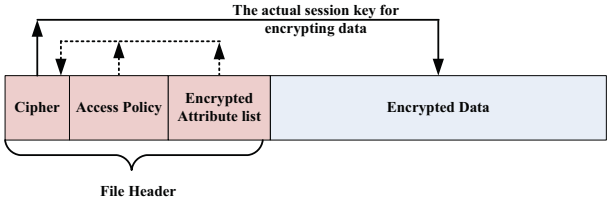

Figure 2: File structure based on CBE scheme.

5.3 Supporting Temporal Constraints with CBE

In this subsection, we illustrate how our CBE scheme can be utilized to construct various temporal constraints, enabling fine-grained access control in clouds.

5.3.1 Simple Temporal Constraint

In order to realize the temporal constraint on the single time axis, we need a function to convert the time into a nonnegative integer in $[0, Z]$. For instance, the function

$$f_1(Y, M) := 12 \cdot (Y - 2000) + M$$

transforms (Year, Month) into an integer and its inversion still holds, where $Z = 600$ if the cryptosystem can be used for 50 years. Similarly, we define a function $f_2(Y, M, D) := 365 \cdot (Y - 2000) + 31 \cdot M + D$ to transform (Year, Month, Day).

For a simple temporal system (or called a single temporal coordinate system), we categorize the relationships between constraints and permissions into four cases:

1. The encrytor uses the current time t'_c (time-stamp) to encrypt a file. Any user can decrypt it if and only if $t_1 \le t'_c \le t_2$, where $[t_1, t_2]$ is the granted time range. This means that we need to check $(t_1 \le t'_c)$ and $(t'_c \le t_2)$ by using $v_{t'_c} \leftarrow f_{t_1 \le t'_c}(v_{t_1})$ and $\bar{v}_{t'_c} \leftarrow \bar{f}_{t'_c \le t_2}(\bar{v}_{t_2})$;

2. The encrytor assigns a period of validity $[0, t']$ to encrypt a file. Any user can decrypt it if and only if $(t_1 \le t' \le t_2)$OR$(t_1 \le t_2 \le t')$ for a licence $[t_1, t_2]$. This means that we just need to check $(t_1 \le t')$ by using $v_{t'} \leftarrow f_{t_1 \le t'}(v_{t_1})$, as shown in Figure 3;

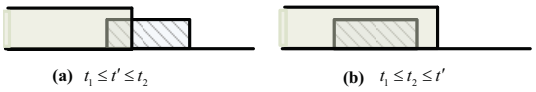

Figure 3: The range relations on $[0, t] \bigcap [t_1, t_2]$.

3. The encrytor assigns a period of validity $[t', T]$ to encrypt a file. Any user can decrypt it if and only if $(t_1 \le t' \le t_2)$OR$(t' \le t_1 \le t_2)$ for a licence $[t_1, t_2]$. This means that we just need to check $(t_2 \ge t')$ by using $\bar{v}_{t'} \leftarrow \bar{f}_{t' \le t_2}(\bar{v}_{t_2})$, see Figure 4;

4. The encrytor assigns a period of validity $[t'_1, t'_2]$ to encrypt a file. Any user can decrypt it if and only if $[t_1, t_2] \bigcap [t'_1, t'_2] \neq \emptyset$. This includes four cases: 1) $t_1 \le t'_1 \le t_2 \le t'_2$; 2) $t'_1 \le t_1 \le t'_2 \le t_2$; 3) $t'_1 \le t_1 \le t_2 \le t'_2$; and 4) $t_1 \le t'_1 \le t'_2 \le t_2$. We can synthesize these cases into

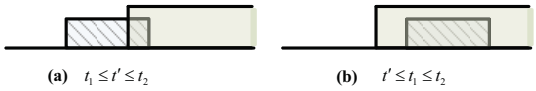

Figure 4: The range relations on $[t, T] \bigcap [t_1, t_2]$.

a simple format: $((t_1 \le t'_2)$ AND $(t'_1 \le t_2))$. This means that we need to check $((t_1 \le t'_2)$ AND $(t'_1 \le t_2))$ by using $v_{t'_2} \leftarrow f_{t_1 \le t'_2}(v_{t_1})$ and $\bar{v}_{t'_1} \leftarrow \bar{f}_{t'_1 \le t_2}(\bar{v}_{t_2})$, as depicted Figure 5;

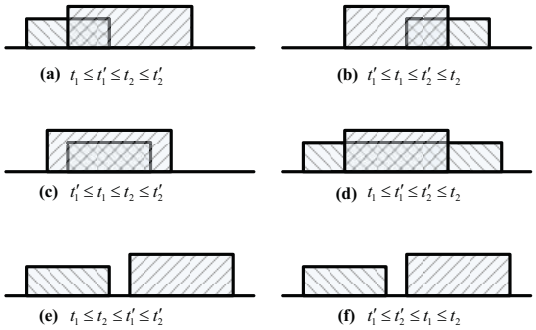

Figure 5: The range relations on $[t'_1, t'_2] \bigcap [t_1, t_2]$.

For four afore-mentioned cases, we show corresponding cryptographic operations in Table 3, in which each case can be implemented by using at most two derivation operations. Hence, with the help of f and $\bar{f}$, we can realize the simple temporal constraint in an attribute-based cryptosystem.

5.3.2 Periodic Constraint

Periodic constraints are another important way for expressing temporal constraints. Alien from the common constraints with continuous-time range, this kind of constraints can be divided into some intervals. That is, given a set of time $U = \{t_1, \cdots, t_T\}$, the permitted time is defined as $U_p = \bigcup_{i=I} [t_{i_b}, t_{i_e}]$, where I is an index set, each index $i \in I$ corresponds to an interval $U_i = [t_{i_b}, t_{i_e}]$, and $U_i \bigcap U_j = \emptyset$ for different $i, j \in I$. By the same token, we also extend the granted time in licence to periodic expression, that is, $U_g = \bigcup_{j=J} [t_{j_b}, t_{j_e}]$, where J is an index set of granted intervals. It is easy to see that a user can also be authorized if and only if $U_p \bigcap U_g \neq \emptyset$ in terms of Definition 1.

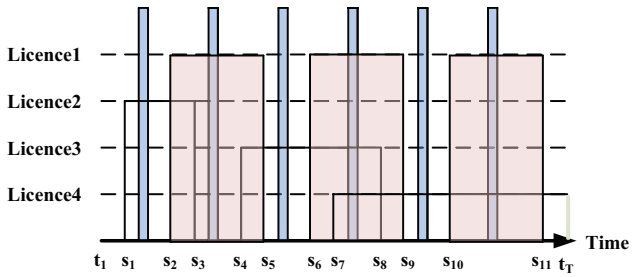

Figure 6: An example for periodic time.

In Figure 6, we show some examples for periodic constraints. We assume that a policy assigns the permission during periodic intervals on blue boxes. The black lines above the time axis indicate the users' granted-times in their

Table 3: Cryptographic operations for deferent cases.

	Cases	Logical Representations	Cryptographic Operations
1	$t_1 \le t'_c \le t_2$	$(t_1 \le t'_c) \wedge (t'_c \le t_2)$	$v_{t'_c} \leftarrow f_{t_1 \le t'_c}(v_{t_1})$ and $\bar{v}_{t'_c} \leftarrow f_{t'_c \le t_2}(\bar{v}_{t_2})$
2	$[0, t']$	$(t_1 \le t')$	$v_{t'} \leftarrow f_{t_1 \le t'}(v_{t_1})$
3	$[t', T]$	$(t_2 \ge t')$	$\bar{v}_{t'} \leftarrow \bar{f}_{t' \le t_2}(\bar{v}_{t_2})$
4	$[t'_1, t'_2]$	$(t_1 \le t'_2) \wedge (t'_1 \le t_2)$	$v_{t'_2} \leftarrow f_{t_1 \le t'_2}(v_{t_1})$ and $\bar{v}_{t'_1} \leftarrow \bar{f}_{t'_1 \le t_2}(\bar{v}_{t_2})$

licenses. The intersection portions of these lines indicate intervals in which permission-licence assignments are valid. For example, when the licence$_1$ is assigned to the regular intervals on pink boxes, the licence is activated every other blue intervals. The other licences are kind of like licence$_1$.

The aforementioned construction can be used to realize periodic constraints by using OR logical to connect each interval $[t_{i_b}, t_{i_e}]$. Obviously, this kind of exhaustion method is not a perfect solution for large index sets I, J because this method needs to search all intervals. Here, we propose a simple method handling multiple attributes to address this problem as follows: periodic is represented by logic combination of multiple range attributes, where each attribute can be defined by a notation in calendar, such as Hours, Days, Weeks, Months, and Years. For example, $(Months[3, 5])$ and $(Years[2009, 2011])$.

TIME&DAY	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
00:00-08:59							
09:00-10:59	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	
11:00-12:59	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	
13:00-14:59							
15:00-16:59	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta		
17:00-18:59	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta	Stu/Fac/Sta		
19:00-20:59							
21:00-22:59	Fac	Fac	Fac	Fac	Fac	Repairmen	
23:00-23:59	Fac	Fac	Fac	Fac	Fac	Repairmen	

Figure 7: An example for library schedule in a university.

In Figure 7, we show several examples on two attributes: Weeks (W) and Hours (H). We make use of a table to describe all combinations between weeks and hours, i.e., $(w_i, h_j) \in W \times H$, where $W \times H$ denotes Cartesian product on weeks and hours, $w_i \in W$ and $h_j \in H$. Assume that this figure is a schedule for a library in a university. The grey parts denote the working hours for students (Stu), faculties (Fau), and staffs (Sta) every week, i.e., $\big((W[1, 5]) \wedge ((H[9, 13]) \vee (H[15, 19]))\big) \vee \big((W[6, 6]) \wedge (H[9, 13])\big)$, in which the students, faculties, and staffs obtain admissions to enter the library. The orange parts denote the time in which only faculties are permitted to enter the library every week, i.e., $((W[1, 5]) \wedge (H[21, 24]))$. Finally, the cyan parts denote the repairman's working hours for installing and maintaining the equipments every week, i.e., $((W[6, 6]) \wedge (H[21, 24]))$.

6. SECURITY ANALYSIS

We now briefly analyze the security of CBE scheme.[3] First, we describe the hardness assumptions used in our scheme: Given a bilinear map group system $\mathbb{S}_N = (N = pq, \mathbb{G}, \mathbb{G}_T, e(\cdot, \cdot))$ with composite order n. The security of TACE scheme is constructed on three basic assumptions:

Definition 4 (RSA assumption). *Given an RSA public key (N, e) and a ciphertext $C = M^e \in \mathbb{G}_{n'}$, it is intractable to compute the plaintext M.*

Definition 5 (Co-CDH assumption). *For two random $x, y \in \mathbb{Z}^*_n$, given a quadruple $(G_1, G_1^x, G_2, G_2^y) \in \mathbb{G}^4$, it is intractable to compute G_2^{xy}.*

Definition 6 (Bilinear Co-CDH assumption). *For two random $x, y \in \mathbb{Z}^*_n$, given a quintuple $(G_1, G_1^x, G_1^y, G_2, G_2^y) \in \mathbb{G}^5$, it is intractable to compute $e(G_1^y, G_2^{xy})$.*

Since this scheme is constructed based on BSW's CP-ABE scheme, CBE scheme remains semantically secure against chosen plaintext attack (IND-CPA) [5]. In addition, we introduce forward and backward derivation functions $(f, \bar{f})$ into CBE scheme. It is easy to find that one-way property of f and $\bar{f}$ can be guaranteed under the RSA assumption: given an RSA public key (N, e) and a ciphertext $C = M^e \in \mathbb{G}_{n'}$, it is infeasible to compute M. This is based on the fact that it is intractable to compute n, n' and $\frac{1}{e} \pmod{n'}$ by factoring large number $N = pq$.

6.1 Security for Collusion Privilege Attacks

We depend on the confidentiality of r to guarantee the security of scheme against collusion privilege attacks. For sake of clarity, we only consider the collusion attacks by two adversaries to analyze all possible cases. For example, two users, u_i and u_j, intend to transfer the u_i's a range attribute key $(D_t^{(i)}, D_{t_a}^{'(i)}, \bar{D}_{t_b}^{'(i)}, D_t^{''(i)})$ into the u_j's the attribute key $(D_t^{(j)}, D_{t_{a'}}^{'(j)}, \bar{D}_{t_{b'}}^{'(j)}, D_t^{''(j)})$ due to $t_a < t_{a'} < t_{b'} < t_b$, that is,

$$(D_t^{(i)}, D_{t_a}^{'(i)}, \bar{D}_{t_b}^{'(i)}, D_t^{''(i)}) = (g^{\tau_i} H(A_t)^r, v_{t_a}^r, \bar{v}_{t_b}^r, w^r),$$
$$(D_t^{(j)}, D_{t_{a'}}^{'(j)}, \bar{D}_{t_{b'}}^{'(j)}, D_t^{''(j)}) = (g^{\tau_j} H(A_t)^{r'}, v_{t_{a'}}^{r'}, \bar{v}_{t_{b'}}^{r'}, w^{r'}).$$

It is easy to find following two approaches to collude a new private key with more privileges:

$$1.\ (\boxed{D_t^{(j)}}, D_{t_a}^{'(i)}, \bar{D}_{t_b}^{'(i)}, D_t^{''(i)}) = (g^{\tau_j} H(A_t)^r, v_{t_a}^r, \bar{v}_{t_b}^r, w^r),$$
$$2.\ (D_t^{(j)}, \boxed{D_{t_a}^{'(j)}, \bar{D}_{t_b}^{'(j)}}, D_t^{''(j)}) = (g^{\tau_j} H(A_t)^{r'}, v_{t_a}^{r'}, \bar{v}_{t_b}^{r'}, w^{r'}).$$

We called them as CPA-I and CPA-II attacks, respectively.

For CPA-I attacks, we can prove the following theorem (see the proof in Appendix A):

Theorem 1. *Given a CBE cryptosystem over the RSA-type elliptic curve system $\mathbb{S}_N$, it is intractable to extract the values g^{τ_k} or $H(A_t)^r$ from the user's key $SK_{\mathcal{L}} = (D_t = g^{\tau_k} H(A_t)^r, D'_{t_a} = v_{t_a}^r, D'_{t_b} = \bar{v}_{t_b}^r, D''_t = w^r)$ under computational Co-Diffie-Hellman (Co-CDH) assumption.*

[3]The details of the security analysis are omitted in this paper due to the space limitation.

This theorem shows that the colluders cannot forge a new key by exchanging g^{τ_k} or $H(A_t)^r$ from some known private-keys. Hence, our scheme can resist the CPA-I type attacks.

For CPA-II attacks, the attackers try to replace $(v_{t'_a}^{r'}, \bar{v}_{t'_b}^{r'})$ by $(v_{t_a}^{r'}, \bar{v}_{t_b}^{r'})$ according to $(v_{t_a}^{r}, \bar{v}_{t_b}^{r})$, where $t_a < t_{a'} < t_{b'} < t_b$. However, the confidentiality of r and r' can guarantee the security of scheme against this attack in terms of the following theorem (see the proof in Appendix B).

THEOREM 2. *Given a multi-tuple $(N, \varphi, \lambda, t_i, (\varphi^r)^{\lambda^{t_i}})$ over the RSA-type elliptic curve system $\mathbb{S}_N$, where $r \in_R \mathbb{Z}$. It is intractable to compute $(t_j, (\varphi^r)^{\lambda^{t_j}})$ with $t_j < t_i$ for all PPT algorithms under the RSA assumption.*

6.2 Security for KS-CDA Attacks

In addition to collusion attack, chosen derivation-key attack (CDA) is a more easy-to-implement approach to break our CBE scheme, in which the adversary only needs to eavesdrop the channel via the proxy server. In this way, the adversary can obtain as much prior knowledge as possible from the stolen derivation keys, and attempt to forge a new private-key with the help of a known private-key.

Our scheme can prevent the CDA attack from two aspects: 1) the derivation key retains the user's unique identity τ_k, so that other users cannot use this key according to Theorem 1, and 2) a new random variant σ is also introduced into the derivation key to wrap the original private key under the Diffie-Hellmen assumption. Hence, we prove that our scheme is KS-CHA secure under the Bilinear co-CDH assumption (see the proof in Appendix C) as follows:

THEOREM 3. *Given a RSA-type elliptic curve system $\mathbb{S}_N = (N = pq, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot))$ with order $n = sn'$, CBE cryptosystem over $\mathbb{S}_N$ is key secure against chosen derivation-key attacks (KS-CDA) under the Bilinear co-CDH assumption on $\mathbb{G}$ even if the secret s and n' is known.*

6.3 Security for SS-CDA Attacks

When a "honest but curious" service provider tries to reveal the encrypted contents, it can explore potential security issues of our scheme. First, we consider the ciphertext-only attack. We will present our CBE scheme is as strong as the BSW's scheme. In order to demonstrate that the cloud service providers cannot compromise the ciphertext without private keys, we compare the difference between the ciphertext of our scheme and that of BSWąŕs scheme in Table 4.

Table 4: Difference between our CBE scheme and BSW's scheme

Scheme	Ciphertext		
BSW's scheme	$g^{(\alpha+\tau_k)/\beta}$	$g^{\tau_k} H(A_l)^r$	w^r
Our scheme	$g^{(\alpha+\tau_k)/\beta}$	$g^{\tau_k} H(A_l)^r$	$(v_{t_a} w)^r$

It is easy to find that the different between them is merely the value $v_{t_a}^r$ which is introduced into ciphertexts. In fact, our scheme is compatible with BSW's CP-ABE scheme for string-based matching. Hence, our scheme can be considered as an extension of BSW's scheme in this point. Thus, our scheme remains the same security properties as of BSW's scheme, i.e., semantically secure against chosen plaintext attack (IND-CPA) [5]. This means that the cloud service providers cannot obtains the contents of ciphertexts without the knowledge of private keys.

Next, we analyze whether the derivation keys $\{\widetilde{SK}_{\mathcal{L}'}\}$ observed by the adversary (or proxy) increase the adversary's advantage against our scheme. Although $\{\widetilde{SK}_{\mathcal{L}'}\}$ are delegated from the private key $\{SK_{\mathcal{L}}\}$, it cannot be used to decrypted the ciphertexts because 1) they contain only part of information of the private-keys, and 2) the random δ is used to avoid revealing the decryption information to the adversary. In order to verify the validity of this method, we prove that any (polynomial) number of derivation keys observed by the adversary cannot increased the advantage of attacks under the Bilinear co-CDH assumption. This theorem is described as follows (see the proof in Appendix D):

THEOREM 4. *Given a RSA-type elliptic curve system $\mathbb{S}_N = (N = pq, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot))$ with order $n = sn'$, CBE cryptosystem over $\mathbb{S}_N$ is semantically secure against chosen derivation-key attacks (SS-CDA) under the Bilinear co-CDH assumption on $\mathbb{G}$ even if the secret s and n' is known.*

7. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our CBE scheme. We first examine the complexity of our CBE scheme. Then, we discuss the parameter generation for a specific level of security. We also demonstrate the computational cost of performing comparison operations in our experiments.

7.1 Complexity Analysis

In this subsection, we will analyze the complexity of our CBE scheme. For simplification, we give several notations to denote the time for various operations in our scheme. $E(\mathbb{G})$ and $E(\mathbb{G}_T)$ are used to denote the exponentiation in $\mathbb{G}$ and $\mathbb{G}_T$, respectively. B is used to denote the bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. We neglect the operations in $\mathbb{Z}_N$, the hash function $H : \{0,1\}^* \to \mathbb{G}$ and the multiplication in $\mathbb{G}$ and $\mathbb{G}_T$, since they are much more efficient than exponentiation and paring operations. In Table 5, we analyze the computation and communication complexity for each phase, where $|\mathcal{T}|$ denotes the number of the leaf nodes in the tree, S denotes the set of attributes of encryptor and decryptor, and $l_{\mathbb{Z}_n}, l_{\mathbb{G}}, l_{\mathbb{G}_T}$ denote the length of elements in $\mathbb{Z}_n^*, \mathbb{G}, \mathbb{G}_T$, respectively.

In the tradition cryptosystem, decryption is an algorithm executed by a single party. But decryption in our scheme is converted into an interactive decryption protocol consisting with three algorithms: Delegate, Decrypt1, and Decrypt2. Although the sum of overheads of these algorithms is slightly larger than that of one single algorithm, we try to shift the mainly computational overheads of decryption into cloud servers, which have more computing power. In Table 5, the overheads of Decrypt1 are far larger than the sum of the other algorithms as a result that the bilinear pairing operation consumes more memory usage and CPU time than other operations. In addition, the sum of communication overheads of three algorithms is also consistent with that of one single algorithm. In particular, the output of Decrypt1 in a cloud server is a fixed data package size.

7.2 Parameter Generation

The security of CBE scheme is based on the RSA and CDH assumptions. Thus, we define the security parameters as follows: Let $N = pq$ be the RSA-modulus, we construct

Table 5: Performance Analysis of CBE Scheme

	Computation Complexity	Communication Complexity
Setup	$1 \cdot B + 3 \cdot E(\mathbb{G})$	$6 \cdot l_{\mathbb{G}} + 1 \cdot l_{\mathbb{G}_T} + 2 \cdot l_{\mathbb{G}}$
KeyGen	$(1 + 5\|S\|) \cdot E(\mathbb{G})$	$(1 + 4\|S\|) \cdot l_{\mathbb{G}}$
Encrypt	$(1 + 4\|\mathcal{T}\|) \cdot E(\mathbb{G}) + 1 \cdot E(\mathbb{G}_T)$	$4\|\mathcal{T}\| \cdot l_{\mathbb{G}} + 1 \cdot l_{\mathbb{G}_T}$
Delegate	$(1 + 7\|S\|) \cdot E(\mathbb{G})$	$3\|S\| \cdot l_{\mathbb{G}}$
Decrypt1	$2\|S\| \cdot B + \|\mathcal{T}\| \cdot E(\mathbb{G}_T)$	$1 \cdot l_{\mathbb{G}} + 1 \cdot l_{\mathbb{G}_T}$
Decrypt2	$1 \cdot B + 1 \cdot E(\mathbb{G})$	

DCOE scheme using composite order bilinear groups based on RSA-type Cryptosystem $\mathbb{S}_N$ over elliptic curve (EC) [8]. To ensure the security of $\mathbb{S}_N$, we assume that $\#E_p(a,b) = p+1$ and $\#E_q(a,b) = q+1$. Hence, there exists a group $\mathbb{G}_{N_n}$ of order $N_n = lcm(p+1, q+1)$ in $\mathbb{S}_N$. According to the above theorem, we define $n = s_1 s_2 p' q'$, $p+1 = 2s_1 p'$ and $q+1 = 2s_2 q'$, where p', q' are two sufficiently large primes and $|p'| = |q'| = 512$ bits. Such that, we can generate a bilinear map system $\mathbb{S}_N = (N, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot))$ of composite order n, where $\mathbb{G}$ is a subgroup of order n in $\mathbb{G}_{N_n}$ due to $n|N_n$. Further, there exists the subgroup $\mathbb{G}'$ of order $n' = p'q'$ in $\mathbb{G}$, where $n'|n$. These parameters guarantee that our CBE system is secure against the cycling attack.

Finally, given two random generators $g, w \in \mathbb{G}$, we require that the Discrete Logarithm problems on $g' = g^{n'}$ and $w' = w^s$ are difficult in $\mathbb{G}$, that is, two orders, $ord_n(g')$ and $ord_n(w')$, are also sufficiently large. This is also the precondition of the Co-Diffie-Hellman assumption. On the other hand, it is easy to find that the DDH problem is easy in $\mathbb{G}$ because there exists a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Hence, the CDH assumption holds under the above-mentioned parameters. Beyond this, the security of CBE scheme also depends on the Discrete Logarithm assumption in $\mathbb{G}_s$ where the length of $|s|$ is at less 160 bit.

7.3 Experimental Results

We have implemented our scheme on an experimental cloud computing environment (called M-Cloud). We simulated the encryption service and the storage service by using two local IBM servers with two Intel Core 2 processors at 2.16 GHz and 500M RAM running Windows Server 2003 and 64-bit Redhat Enterprise Linux Server 5.3, respectively. These two servers were connected into the M-Cloud via 250 MB/sec of network bandwidth. The storage server was responsible for managing a 16TB storage array based on Hadoop distributed file system (HDFS) 0.20 cluster with 8 worker nodes located in our laboratory. Using GMP and PBC libraries, we have developed a cryptographic library upon which our CBE systems can be constructed. This C library contains approximately 5,500 lines of code and has been tested on both Windows and Linux platforms.

We show the practical computational costs of algorithms for our scheme in Figure 8 under the effective calculation length is $L = 2048$-bits. In this example, for a certain comparison range $[1, Z]$, we generate a secret-key with licence $[t_1, t_2]$, where $t_1 \in_R [1, Z/4]$ and $t_2 \in_R [3Z/4, Z]$; and a message is encrypted by the time $t \in_R [Z/4, 3Z/4]$. So, we ensure that $\max(t - t_1, t_2 - t) \geq Z/4$. As the value of Z is changed from 4 to 65,536, the computational costs should keep pace with the growth of comparison ranges. However, this kind of growth is not significant by comparison with bilinear operations.

Our experimental results are showed in Figure 8, where

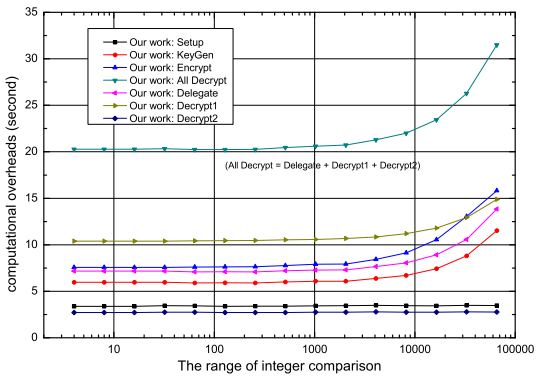

Figure 8: Computational costs of our scheme under different comparison range (the effective calculation length is $L = 2048$-bits).

the curve "All Decrypt" depicts the sum of operation overheads of Delegate, Decrypt1, and Decrypt2 in our scheme. It is obvious that the decryption overhead is well decomposed into these three algorithms, and the overhead of Decrypt1 is the largest of all algorithms. Also, the overhead of Delegate is slightly less than that of KeyGen algorithm, and Decrypt2 has a constant overhead (without regard to the growth of length of data). Hence, these experimental results verify our theoretical analysis in Section 7.1, that is, the decryption overheads can be effectively apportioned over cloud servers and clients.

8. RELATED WORK

In recent years, cryptographic access control [11, 13] has been introduced as a new access control paradigm to manage dynamic data sharing systems. It relies exclusively on cryptography to provide confidentiality and integrity of data managed by the systems, and is particularly designed to run in an untrusted or hostile environment which lacks of trust knowledge and global control [13]. Hence, attribute-based encryption (ABE) is proposed to realize a fine-grained attribute-based access control mechanism. Since Sahai and Waters [16] introduced ABE as a new means for encrypted access control in 2005, ABE has received much attention and many schemes have been proposed in recent years, such as, ciphertext-policy ABE (CP-ABE) [5, 9, 12] and key-policy ABE (KP-ABE) [10, 14].

One part of cryptographic access control focuses on time-based access control. There are a variety of applications requiring time-based access control. For example, a web-based electronic newspaper company could offer several types of subscription packages, covering different topics. Each user may decide to subscribe to one package for a certain period of time (e.g., a week, a month, or a year). Time control is

of particular significance and has been concerned in access control [2, 3, 4]. In [3], the authors gave a temporal access control model and [2, 4] described applications in database systems and secure broadcasting. There have been plentiful time-bound key assignment schemes to set up the period of validity for the cryptographic key [19, 1, 17]. For example, Tzeng [19] used Lucas function and one-way hash function to achieve temporal control for cryptographic key assignment in hierarchical access control and provide the applications in secure broadcasting and cryptographic key backup.

In the context of ABE, there has been little work on studying time control or integer comparison mechanisms. Even though Bethencourt *et al.* [5] gave a bitwise-matching method to implement integer comparison based on CP-ABE scheme, this method unfortunately is not efficient enough for practical applications. In addition, Time-specific encryption (TSE) [15] and multi-dimensional range queries over encrypted data (MRQED) [18] are, in essence, constructed on the similar bitwise approach with BSW's CP-ABE scheme, which makes use of a policy tree (consists of 0/1 branches) on equal matching to realize the integer comparison.

9. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a novel comparison-based encryption scheme to support fine-grained access control in cloud computing. We also prove the security of our proposed scheme and demonstrate the efficiency of our scheme with experimental evaluation. As part of future work, we would extend our work to explore more efficient construction of CBE, the efficient CBE-oriented pairings, as well as the formal methods of security analysis for general binary relation. We will also optimize our solution to improve the performance of integer comparisons.

10. ACKNOWLEDGMENTS

The work of Y. Zhu, Y. Yu and H. Zhao was supported by the National Natural Science Foundation of China (Project No.61170264 and No.10990011) and the NDRC under Project "A cloud-based service for monitoring security threats in mobile Internet". This work of G.-J. Ahn and H. Hu was partially supported by the grants from US National Science Foundation (NSF-IIS-0900970 and NSF-CNS-0831360) and Department of Energy (DE-SC0004308).

11. REFERENCES

[1] G. Ateniese, A. D. Santis, A. L. Ferrara, and B. Masucci. Provably-secure time-bound hierarchical key assignment schemes. In *ACM Conference on Computer and Communications Security*, pages 288–297, 2006.

[2] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. A temporal access control mechanism for database systems. *IEEE Trans. Knowl. Data Eng.*, 8(1):67–80, 1996.

[3] E. Bertino, P. A. Bonatti, and E. Ferrari. Trbac: A temporal role-based access control model. *ACM Trans. Inf. Syst. Secur.*, 4(3):191–233, 2001.

[4] E. Bertino, B. Carminati, and E. Ferrari. A temporal key management scheme for secure broadcasting of xml documents. In V. Atluri, editor, *ACM Conference on Computer and Communications Security*, pages 31–40. ACM, 2002.

[5] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[6] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology (CRYPTO'2001)*, volume 2139 of LNCS, pages 213–229, 2001.

[7] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In J. Kilian, editor, *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2005.

[8] S. D. Galbraith and J. F. McKee. Pairings on elliptic curves over finite commutative rings. In *10th IMA International Conference of Cryptography and Coding, Cirencester, UK, December 19-21, 2005, Proceedings*, pages 392–409, 2005.

[9] V. Goyal, A. Jain, O. Pandey, and A. Sahai. Bounded ciphertext policy attribute based encryption. In *ICALP (2)*, pages 579–591, 2008.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.

[11] A. Harrington and C. D. Jensen. Cryptographic access control in a distributed file system. In *SACMAT*, pages 158–165. ACM, 2003.

[12] L. Ibraimi, Q. Tang, P. H. Hartel, and W. Jonker. Efficient and provable secure ciphertext-policy attribute-based encryption schemes. In *ISPEC*, pages 1–12, 2009.

[13] A. V. D. M. Kayem. *Adaptive Cryptographic Access Control for Dynamic Data Sharing Environments*. Ph.d thesis, Queenaṕs University Kingston, Ontario, Canada, October 2008.

[14] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM Conference on Computer and Communications Security*, pages 195–203, 2007.

[15] K. Paterson and E. Quaglia. Time-specific encryption. In J. Garay and R. De Prisco, editors, *Security and Cryptography for Networks*, volume 6280 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin / Heidelberg, 2010.

[16] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[17] A. D. Santis, A. L. Ferrara, and B. Masucci. New constructions for provably-secure time-bound hierarchical key assignment schemes. In *SACMAT*, pages 133–138, 2007.

[18] E. Shi, J. Bethencourt, T. H. H. Chan, D. Song, and A. Perrig. Multi-dimensional range query over encrypted data. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 350–364, may 2007.

[19] W. Tzeng. A time-bound cryptographic key assignment scheme for access control in a hierarchy. *IEEE Trans. on Knowledge and Data Engineering*, 14(1):182–188, 2002.

[20] S. Yu, C. Wang, K. Ren, and W. Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *INFOCOM*, pages 534–542, 2010.

APPENDIX

A. PROOF OF CPA-I ATTACK RESISTANT

PROOF. First, let $g^{\tau_k} = w^{\xi}$, $H(A_t) = w^k$, $v_{t_a} = w^{k_1}$ and $\bar{v}_{t_b} = w^{k_2}$ in $\mathbb{G}$, so we use the same generator w to denote $SK_{\mathcal{L}}$ as $D_t = g^{\tau_k} H(A_t)^r = w^{\xi+kr}$, $D_{t_a} = v_{t_a}^r = w^{k_1 r}$, $\bar{D}_{t_b} = \bar{v}_{t_b}^r = w^{k_2 r}$, and $D_t = w^r$ in $\mathbb{G}$. Such that, we convert the theorem into the problem: it is intractable to extract the values (w^{ξ}, w^{kr}) from $(w, w^r, w^k, w^{k_1}, w^{k_2}, w^{k_1 r}, w^{k_2 r}, w^{\xi+kr})$. It is obvious that two unknown k_1, k_2 have no concern with this problem, such that the above problem is reduced into $(w, w^r, w^k, w^{\xi+kr}) \rightarrow (w^{\xi}, w^{kr})$.

Assume that there exists a PPT algorithm $\mathcal{A}$ that can breaks this problem. Given a Co-CDH problem $(G_1, G_1^x, G_2, G_2^y) \rightarrow G_2^{xy}$, we can construct an efficient algorithm $\mathcal{B}$ to solve this Co-CDH problem according to the algorithm $\mathcal{A}$ as follows:

(1) $\mathcal{B}$ invokes the algorithm $\mathcal{A}$ on input $(w = G_1, w^r = G_1^x, w^k = G_2^y, w^{\xi+kr} = G_2^z)$, where z is a random integer;

(2) If the output of algorithm $\mathcal{A}$ is (R_1, R_2), $\mathcal{B}$ checks whether

two equations $R_1 \cdot R_2 = G_2^z$ and $e(G_1, R_2) = e(G_1^x, G_2^y)$ hold. If not, $\mathcal{B}$ repeats step (1);

(3) $\mathcal{B}$ computes $G_2^{xy} = R_2$ and returns it as output.

The output of algorithm $\mathcal{B}$ is valid because the input of $\mathcal{A}$ satisfies $r = x$, $w^{kr} = (G_2^y)^r = G_2^{xy} = R_2$, $e(G_1, R_2) = e(G_1, G_2^{xy}) = e(G_1^x, G_2^y)$, and $G_2^z = w^{\xi + kr} = R_1 \cdot R_2$.

This means that the algorithm $\mathcal{B}$ is a PPT algorithm to solve Co-CDH problem only if $\mathcal{A}$ is also a PPT algorithm. But it is well-known that the Co-CDH problem is hard for any PPT algorithms, hence this contradicts the hypothesis. □

B. PROOF OF CPA-II ATTACK RESISTANT

PROOF. Seeking a contradiction, we assume that there exists a PPT algorithm $\mathcal{A}$ that can get a $(t_j, (\varphi^r)^{\lambda^{t_j}})$ under above input $(N, \varphi, \lambda, t_i, (\varphi^r)^{\lambda^{t_i}})$, where $t_j < t_i$. We can use the algorithm $\mathcal{A}$ to construct a PPT algorithm $\mathcal{B}$ that can break the RSA problem over elliptic curve: given the public-key $(\mathbb{G}, N, e)$ and a ciphertext C to compute the plaintext $M = C^{e^{-1}}$. The algorithm $\mathcal{B}$ is described as follows:

(1) Given a RSA problem $(\mathbb{G}, N, e)$, $\mathcal{B}$ invokes the algorithm $\mathcal{A}$ on input $(N, \varphi, \lambda = e, t_i, C)$, where t_i is randomly chosen in integer set and $\varphi = C^{r'}$ is a random element in $\mathbb{G}$;

(2) If the algorithm $\mathcal{A}$ returns a solution (t_j, R), $\mathcal{B}$ first checks if $R^{\lambda^{t_i - t_j}} = C$ (or $\varphi = R^{r'\lambda^{t_i - t_j}}$) and $t_i - t_j - 1 \geq 0$ (or $t_i - t_j > 0$). If not, $\mathcal{B}$ repeats step (1);

(3) $\mathcal{B}$ computes computing $M = R^{e^{t_i - t_j - 1}} \in \mathbb{G}$ in terms of $R^{\lambda^{t_i - t_j}} = C = M^{\lambda}$, and returns the ciphertext M.

In the algorithm $\mathcal{B}$, we cannot know the secret $r = \frac{1}{\lambda^{t_i} r'} \pmod{n'}$ for unknown n' (because of the actual difficulty of factoring large number $N = pq$), event through $\varphi = C^{r'}$ and r' is known. This means that the algorithm $\mathcal{B}$ is a PPT algorithm to solve RSA problem only if $\mathcal{A}$ is also a PPT algorithm. But it is well-known that the RSA problem is hard for any PPT algorithms, hence this contradicts the hypothesis. □

C. PROOF OF KS-CDA RESISTANT

PROOF. Assume that there exists a PPT algorithm $\mathcal{A}$ that can breaks this problem over $\mathbb{S}_N$ with the known s, n'. Given a Co-CDH problem $(G_1, G_1^x, G_2, G_2^y) \to G_2^{xy}$ in $\mathbb{G}$, we can construct an efficient algorithm $\mathcal{B}$ to solve this Co-CDH problem according to the algorithm $\mathcal{A}$ as follows:

(1) **Setup**: $\mathcal{B}$ follows the *Setup* algorithm to get the elements $(g, h, \xi, \lambda, \mu)$ and then sets $w = G_2$, $\varphi = G_2^{k_1} \in \mathbb{G}_{n'}$, $\bar{\varphi} = G_2^{k_2} \in \mathbb{G}_{n'}$, where α, β, k_1, k_2 are known by $\mathcal{B}$, $s|k_1$, and $s|k_2$. Therefore, $\mathcal{B}$ sends $PK = (\mathbb{S}_N, g, h, \zeta, w, \varphi, \bar{\varphi}, \lambda, \mu)$ to the adversary $\mathcal{A}$ and $H(\cdot)$ can be obtained by the random Oracle query of $\mathcal{B}$.

(2) **Learning**: $\mathcal{A}$ chooses a range attribute A_t and query Delegate algorithm with the polynomial number of users $u_{k_1}, \cdots, u_{k_s}$ with any time interval $A_{t_i}[t_{k_i}, t_{k_i'}] \in \mathcal{L}_i$. For each query, $\mathcal{B}$ chooses two random τ_i, σ_i and $H_i \in \mathbb{G}$ and sets $r_i = y$, and then computes

$$\widetilde{D}_{t_i} = (g^{\tau_i} H(A_t)^{r_i})^{\sigma_i} = (g^{\tau_i} G_2^{xy})^{\sigma_i} = H_i^{\sigma_i},$$
$$\widetilde{D'}_{t_{k_i}} = (v_{t_{k_i}} w)^{r_i \sigma_i} = (G_2^{y k_1 \lambda^{t_{k_i}}} G_2^y)^{\sigma_i} = (G_2^y)^{\sigma_i (k_1 \lambda^{t_{k_i}} + 1)},$$
$$\widetilde{D'}_{t_{k_i'}} = (\bar{v}_{t_{k_i'}} w)^{r_i \sigma_i} = (G_2^{y k_2 \lambda^{(Z - t_{k_i'})}} G_2^y)^{\sigma_i} = (G_2^y)^{\sigma_i (k_2 \lambda^{(Z - t_{k_i'})} + 1)},$$

and sends these derivation keys $\widetilde{SK}_{\mathcal{L}_i} = \{(\widetilde{D}_{t_i}, \widetilde{D'}_{t_{k_i}}, \widetilde{D'}_{t_{k_i'}})\}$ to the adversary $\mathcal{A}$. Note that, $H(A_t) = G_2^x$ and $\widetilde{SK}_{\mathcal{L}_i}$ is anonymous for $\mathcal{B}$ because τ_i is unknown.

(3) **Challenge**: $\mathcal{B}$ chooses two random τ^*, r^* and defines $r_i = \frac{r^*}{z}$ which is unknown by $\mathcal{B}$. And then it computes $D^* = g^{(\alpha + \tau^*)/\beta}$,

$$\begin{aligned}
D_t^* &= g^{\tau^*} H(A_t)^{r^*} = g^{\tau^*} (G_1^x)^{r^*}, \\
D'^*_{t_i} &= v_{t_i}^{r^*} = G_2^{k_1 \lambda^{t_i} \frac{r^*}{z}} = G_1^{k_1 r^* \lambda^{t_i}}, \\
D'^*_{t_j} &= \bar{v}_{t_j}^{r^*} = G_2^{k_2 \lambda^{Z - t_j} \frac{r^*}{z}} = G_1^{k_2 r^* \lambda^{Z - t_j}}, \\
D''^*_t &= w^{r^*} = G_2^{\frac{r^*}{z}} = G_1^{r^*}.
\end{aligned}$$

Hence, $\mathcal{B}$ sends $SK_{\mathcal{L}} = (D^*, (D_t^*, D'^*_{t_a}, D'^*_{t_b}, D''^*_t))$ as a challenge private key to $\mathcal{A}$, where $\mathcal{L} = A_t[t_i, t_j]$.

(3) **Response**: If the output of algorithm $\mathcal{A}$ is $(\mathcal{L}_i, SK_{\mathcal{L}_i})$, where $SK_{\mathcal{L}_i} = (D^*, (D_{t_i}^*, D'^*_{t_{k_i}}, D'^*_{t_{k_i'}}, D''^*_t))$ and $A_{t_i}[t_{k_i}, t_{k_i'}] \in \mathcal{L}_i$, $\mathcal{B}$ checks whether the equations

$$D'^*_{t_{k_i}} = G_2^{y k_1 \lambda^{t_{k_i}}}, D'^*_{t_{k_i'}} = G_2^{y k_2 \lambda^{(Z - t_{k_i'})}}, D''^*_t = G_2^y$$

and $e(G_1, D_{t_i}^*/g^{r^*}) = e(G_1^x, G_2^y)$ hold. If not, $\mathcal{B}$ repeats step (1), Else, $\mathcal{B}$ computes $G_2^{xy} = D_{t_i}^*/g^{r^*}$ and returns it as output.

The output of algorithm $\mathcal{B}$ is valid because the input of $\mathcal{A}$ satisfies $D_{t_i}^* = g^{r^*} G_2^{xy}$. This means that the algorithm $\mathcal{B}$ is a PPT algorithm to solve Co-CDH problem only if $\mathcal{A}$ is also a PPT algorithm. But it is well-known that the Co-CDH problem is hard for any PPT algorithms, hence this contradicts the hypothesis. □

D. PROOF OF SS-CDA RESISTANT

PROOF. Assume that there exists a PPT algorithm $\mathcal{A}$ that can breaks this problem over $\mathbb{S}_N$ with the known s, n'. Given a Bilinear Co-CDH problem $(G_1, G_1^x, G_1^y, G_2, G_2^y) \to e(G_1^y, G_2^{xy})$, we can construct an efficient algorithm $\mathcal{B}$ to solve this Co-CDH problem according to the algorithm $\mathcal{A}$ as follows:

(1) **Setup**: $\mathcal{B}$ chooses a random integer θ and defines $\alpha = xy, \beta = \frac{\theta}{z}$, where $z = \log_{G_1} G_2$ is unknown. $\mathcal{B}$ chooses the random integers λ, μ, k_1, k_2 to computes $g = G_1^{n'}$, $h = w^y = G_1^{\theta}$, $w = G_2$, $\zeta = e(G_1^x, G_2^y) = e(G_1, G_2)^{xy}$, $\varphi = G_2^{k_1}$, and $\bar{\varphi} = G_2^{k_2}$, where $s|k_1$ and $s|k_2$. So that $\mathcal{B}$ generates $PK = (\mathbb{S}_N, g, h, \zeta, w, \varphi, \bar{\varphi}, \lambda, \mu)$ and sends it to $\mathcal{A}$. $H(\cdot)$ can be obtained by the random Oracle query of $\mathcal{B}$.

(2) **Learning**: $\mathcal{A}$ can send the polynomial number of *Delegate* queries with any time interval $\mathcal{L}_i = \{A_{t_i}[t_{k_i}, t_{k_i'}]\}$. For each query, $\mathcal{B}$ chooses the random τ_i, σ, r_i and computes

$$\begin{aligned}
\widetilde{D}_{t_i} &= (g^{\tau_i} H(A_{t_i})^{r_i})^{\sigma} = G_1^{\sigma \tau_i} G_2^{\sigma k_i r_i}, \\
\widetilde{D'}_{t_{k_i}} &= (v_{t_j}^{r_i} \cdot w^{r_i})^{\sigma} = G_2^{\sigma r_i (k_1 \lambda^{t_{k_i}} + 1)}, \\
\widetilde{D'}_{t_{k_i'}} &= (\bar{v}_{t_i}^{r_i} \cdot w^{r_i})^{\sigma} = G_2^{\sigma r_i (k_2 \lambda^{(Z - t_{k_i'})} + 1)},
\end{aligned}$$

where $H(A_{t_i}) = G_2^{k_i}$ and k_i is random integer. Finally, $\mathcal{B}$ returns $\widetilde{SK}_{\mathcal{L}_i} = \{\widetilde{D}_{t_i}, \widetilde{D'}_{t_{k_i}}, \widetilde{D'}_{t_{k_i'}}\}_{A_{t_i}[t_{k_i}, t_{k_i'}] \in \mathcal{L}_i}$ to $\mathcal{A}$.

(3) **Challenge**: $\mathcal{B}$ sets $s = y$ and chooses a random a and $G_2^b = G_2^y / G_2^a$, where $w^s = G_2^y$ and $s = a + b$. Such that, $\mathcal{B}$ computes $h^s = (G_1^y)^{\theta}$, and

$$\bar{E}_{t_i} = (\bar{v}_{t_i} w)^a = (G_2^a)^{(k_2 \mu^{Z - t_i} + 1)}, \qquad E'_{t_i} = H(A_t)^a = (G_2^a)^{k_i},$$
$$E_{t_j} = (v_{t_j} w)^b = (G_2^b)^{(k_1 \lambda^{t_j} + 1)}, \qquad E'_{t_j} = H(A_t)^b = (G_2^b)^{k_i}.$$

$\mathcal{B}$ outputs $\mathcal{H}_{\mathcal{P}}{}^* = (\mathcal{T}, h^s, \{((\bar{E}_{t_i}, E'_{t_i}), (E_{t_j}, E'_{t_j}))\}_{A_t[t_i, t_j] \in \mathcal{T}})$ as the challenge ciphertext to $\mathcal{A}$.

(4) **Response**: $\mathcal{A}$ outputs a session key ek^* to $\mathcal{B}$, and $\mathcal{B}$ also outputs it as result.

If the output of algorithm $\mathcal{A}$ is valid, $\mathcal{B}$ is also valid because $ek^* = e(g^{\alpha}, w^s) = e(G_1^{xy}, G_2^y) = e(G_1^y, G_2^{xy})$. This means that the algorithm $\mathcal{B}$ is a PPT algorithm to solve Co-CDH problem only if $\mathcal{A}$ is also a PPT algorithm. But it is well-known that the Co-CDH problem is hard for any PPT algorithms, hence this contradicts the hypothesis. □