6-2013

Attribute-based access to scalable media in cloud-assisted content sharing

Yongdong WU

Zhuo Wei

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

DOI: https://doi.org/10.1109/TMM.2013.2238910

Citation
WU, Yongdong; Wei, Zhuo; and DENG, Robert H.. Attribute-based access to scalable media in cloud-assisted content sharing. (2013). *IEEE Transactions on Multimedia*. 15, (4), 778-788. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/1639

Published in IEEE Transactions on Multimedia, 2013 June, Volume 15, Issue 4, Pages 778-788
https://doi.org/10.1109/TMM.2013.2238910

Attribute-Based Access to Scalable Media in Cloud-Assisted Content Sharing Networks

Yongdong Wu, Zhuo Wei, and Robert H. Deng

Abstract—**This paper presents a novel Multi-message Ciphertext Policy Attribute-Based Encryption (MCP-ABE) technique, and employs the MCP-ABE to design an access control scheme for sharing scalable media based on data consumers' attributes (e.g., age, nationality, or gender) rather than an explicit list of the consumers' names. The scheme is efficient and flexible because MCP-ABE allows a content provider to specify an access policy and encrypt multiple messages within one ciphertext such that only the users whose attributes satisfy the access policy can decrypt the ciphertext. Moreover, the paper shows how to support resource-limited mobile devices by offloading computational intensive operations to cloud servers while without compromising data privacy.**

Index Terms—**Access control, cloud computing, data security and privacy, scalable media content.**

I. Introduction

CONTENT sharing environments such as social networking are very dynamic in terms of the number of on-line users, storage requirement, network bandwidth, computational capability, applications and platforms, thus it is not easy for a service provider to allocate resources following the traditional client-server model. As cloud computing offers application developers and users an abstract view of services that hides much of the system details and inner workings, it is more and more popular in content-sharing applications. However, the weak security provision of cloud computing services is delaying their adoption [1]. As a result, it is imperative for cloud computing based service providers, private or public, to build security functionalities into their services and manage their services following prudent security practices [2].

Access control is the fundamental security mechanism to facilitate information sharing in a controllable manner. It exerts control over which user can access which resource based on a permission relationship between user attributes and resource attributes, where attributes can be any information deemed relevant for granting access, such as user's job function and resource quality, and permission is specified in terms of requirements on the attributes of resource and user. Any user with attributes that meet the requirements has access to that resource. However, it is challenging to design a suitable access control mechanism in content sharing services due to: (1) any individual is able to freely produce any number and any kind of online media such as text, image, sound, video, and presentation; (2) any individual is able to grand any access to his media to anyone, at any time; (3) an individual may reveal a large number of attributes (e.g., name, age, address, friendship, classmate, fans, hobby, personal interest, gender, and mobility), and some of them can be very dynamics; and (4) individuals may share contents using various devices and bandwidth, and hence demand different access privileges for the same media.

A promising approach to access control in content sharing services is to empower users to enforce access controls on their data directly, rather than through a central administrator. However, this requires flexible and scalable cryptographic key management to support complex access control policies. A naïve access control solution is to assign one key for each user attribute, distribute the appropriate keys to users who have the corresponding attributes, and encrypt the media with the attribute keys repeatedly, e.g., the ciphertext is produced as $c = \mathcal{E}(\mathcal{E}(m, sk_1), sk_2)$ to protect message m with attribute key pair $\langle sk_1, sk_2 \rangle$ and cipher $\mathcal{E}(\cdot)$. This naïve solution is flexible, but it is vulnerable to collusion attack. Technically, a user having key sk_1 for one attribute and another user having key sk_2 for another attribute can collude to decrypt ciphertext c to m. In other words, two users having one attribute each are able to conspire to have the same capability as a user having those two attributes in the näive scheme. Another method is to classify users into different roles based on their attributes, assign role keys to users, and then encrypt the content using the role keys. However, this approach results in high complexity, i.e., the number of keys for each user and the number of ciphertexts for one message are on the order of $O(2^n)$ where n is the number of all possible user attributes. Both of these solutions suffer from the rigid and inflexible definition of the underlying access control policies. A remedy to this problem is employing Ciphertext Policy Attribute-Based Encryption (CP-ABE) [3]. In CP-ABE, a ciphertext is embedded with an access control policy, or access policy for short, associated with user attributes. A recipient of the ciphertext is able to decrypt the ciphertext only if her attributes satisfy the access policy in the ciphertext. CP-ABE can be viewed as a one-to-many public key encryption scheme and hence enables a data owner to grant access to an unknown set of users. Nonetheless, existing CP-ABE schemes merely deliver one encrypted message per ciphertext to all authorized users and are not optimal for efficient sharing of scalable media.

Manuscript received March 01, 2012; revised July 08, 2012; accepted November 13, 2012. Date of publication January 11, 2013; date of current version May 13, 2013. This work was supported in part by A*STAR SERC under Grant 102 101 0027 in Singapore. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chang Wen Chen.

Y. Wu is with the Cryptography and Security Department, Institute for Infocomm Research, 138632 Singapore (e-mail: wydong@i2r.a-star.edu.sg).

Z. Wei and R. H. Deng are with the School of Information Systems, Singapore Management University, 178902 Singapore (e-mail: phdzwei@gmail.com; robertdeng@smu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2013.2238910

In this paper we present an access control scheme for scalable media. The scheme has several benefits which make it especially suitable for content delivery. For example, it is extremely scalable by allowing a data owner to grant data access privileges based on the data consumers' attributes (e.g., age, nationality, gender) rather than an explicit list of user names; and it ensures data privacy and exclusiveness of access of scalable media by employing attribute-based encryption. For this purpose, we introduce a novel Multi-message Ciphertext Policy Attribute-Based Encryption (MCP-ABE) technique. MCP-ABE encrypts multiple messages within one ciphertext so as to enforce flexible attribute-based access control on scalable media. Specifically, the scheme constructs a key graph which matches users' access privileges, encrypts media units with the corresponding keys, and then encrypts the key graph with MCP-ABE; only those data consumers with the required user attributes can decrypt the encryption of the key (sub)graph and then decrypt the encrypted media units. To cater for resource-limited mobile devices, the scheme offloads computational intensive operations to cloud servers while without compromising user data privacy. Attribute-based encryption schemes, such as CP-ABE and MCP-ABE, are designed to be secure against user collusion attacks. The present scheme is also secure against user collusion attacks due to use of attribute-based encryption. The experiments demonstrate that the present scheme is applicable on smartphone, especially when a cloud platform is available.

The reminder of the paper is arranged as follows. Section II provides an overview of the related work. Section III presents the necessary cryptographic and security primitives. Section IV shows the architecture and operations of cloud-assisted media sharing. Section V introduces MCP-ABE and a novel access control scheme based on MCP-ABE. Section VI analyzes the access control scheme in terms of security, flexibility and operational overhead. Section VII presents the experiment results. Finally, Section VIII presents some concluding remarks and future work.

II. Related Work

Fundamental to usage control model [8] is the concept of attributes attached to both users and resources. In content sharing applications, as mapping between user identity and resource is dynamic, access control methods related to our work can be classified into two categories.

A. User Attribute Oriented Access Control

EASiER [9] is an architecture that supports fine-grained access control policies and dynamic group membership by using CP-ABE scheme. In addition, EASiER is able to revoke a user without issuing new keys to other users or re-encrypting existing ciphertexts by using a proxy.

Yu *et al.* [10] employed KP-ABE (Key Policy Attribute based Encryption [11]) to enforce access policies based on data attributes. Their scheme allows data owners to delegate most of the computation tasks involved in fine-grained data access control to untrusted cloud servers without disclosing the underlying data contents by combining techniques of attribute-based encryption, proxy re-encryption, and lazy re-encryption.

Pirretti *et al.* [12] proposed an information management architecture using CP-ABE and optimized security enforcement efficiency. Furthermore, they employed the architecture and optimization method on two example applications: an HIPAA (Health Insurance Portability and Accountability Act) compliant distributed file system and a content delivery network.

Akinyele *et al.* [13] designed and implemented a self-protecting electronic medical records (EMRs) using both CP-ABE and KP-ABE. In order to protect individual items within an EMR, each item is encrypted independently with its own access control policy.

Persona [14] enables access control by employing a combination of traditional public key cryptography and attribute-based encryption scheme. The combination of classical public-key schemes and ABE schemes has the drawback of increased key management complexity.

The above attribute-based access control methods enable flexible access policies for the users. However, they treated media content as a single monolithic object, ignoring the structure of the content. Hence, these schemes are not suitable for access control to scalable multimedia content.

B. Media Structure Oriented Access Control

The SSS (Secure Scalable Streaming) encryption method [15] for scalable video is a progressive encryption technique. As SSS encryption may result in decryption failures due to package loss [16], it should be integrated with error correction techniques in practice so as to overcome this problem. By exploiting the JPEG2000 property of " *encode once, decode many ways* ", Wu *et al.* [17] designed an access control scheme which is efficient and secure. More importantly, the scheme is extremely flexible as its " *encrypt once, decrypt many ways* " property is completely compatible with the feature of the JPEG 2000 image code-streams.

An MPEG4 [18] stream may have two types of quality scalabilities—either PSNR or bit rate scalability. Zhu *et al.* [19] proposed access control schemes for streams encoded by the MPEG-4 Fine Granularity Scalability (FGS) standard so as to allow a single encrypted stream to support both types of scalabilities simultaneously.

H.264/SVC [20] is an efficient video codec standard which specifies temporal, quality and spatial scalabilities. Selective encryption (e.g.,[21], [22], [23], [24]) encrypts portions of the bitstream such as sign of motion vector so as to protect the SVC bitstream in a fast and flexible way. However, selective encryption is usually insecure. By exploiting the tree-structures of H.264 SVC bitstreams, schemes in[25], [26] produce secure scalable bitstreams with relatively high overhead.

All the above media structure based access control schemes exploit the format of media data to generate protected objects so that users with the necessary keys can decrypt the corresponding ciphertext. These schemes are limited to efficient key generations and normally assume the existence of an online key distribution center; and they don't deal with access policies, e.g., how to assign user attributes to access privileges.

III. Preliminaries

To make the paper self-contained, this section introduces the basic concepts of one-way hash function, bilinear map, access tree, and CP-ABE [3]. For simplicity, we assume that the data owner is a male while a data consumer is a female in the following.

Notations

For ease of reference, important notations used throughout the paper are listed below.

$\mathbb{T}$	Access tree, the graph representation of an access policy.
$\mathcal{N}_j$	The jth node of tree $\mathbb{T}$, N_1 is the root of the tree.
$\mathcal{L}$	The set of leaf nodes of tree $\mathbb{T}$.
$\mathcal{S}$	The set of user attributes associated with the set $\mathcal{L}$.
$\mathcal{A}$	The set of attributes of a user.
a_i	The ith user attribute, represented as a string.
AA	Attribute authority.
$\hat{e}(\cdot)$	A bilinear map function.
$\mathcal{G}_1$	The input group of bilinear map $\hat{e}(\cdot)$.
p	The order of $\mathcal{G}_1$, i.e., $\mathcal{G}_1$ has p elements.
$\mathbb{Z}_p$	The set of integers $[0, p-1]$.
g	The generator of $\mathcal{G}_1$. $\forall x \in \mathbb{Z}_p,\ g^x \in \mathcal{G}_1$.
$\mathcal{G}_2$	The output group of bilinear map $\hat{e}(\cdot)$.
PK	The public key of an ABE scheme.
MK	The master key of an ABE scheme.
SK	The secret key of a user, issued by AA.
m_j	The jth unit of a media content.
p_j	The jth privilege with p_1 as the lowest privilege.
k_j	The unit key for encrypting unit m_j.
CT	The enabling block of unit keys.
$x \parallel y$	The concatenation of string x and string y.
$\mid \mathcal{X} \mid$	The number of elements in set $\mathcal{X}$.
$\mathcal{H}_1(\cdot)$	One-way hash function for hashing an attribute.
$\mathcal{H}_2(\cdot)$	One-way hash function for hashing a unit key.

A. One-Way Hash Function

A hash function takes a variable-length input string and converts it into a fixed-length output string, called a hash value. A one-way hash function, denoted as $\mathcal{H}(\cdot)$, works in one direction only: it is easy to compute a hash value $y = \mathcal{H}(x)$ from a pre-image x; however, given an image y, it is hard to find a pre-image x such that $\mathcal{H}(x) = y$. There are many one-way hash functions, such as SHA-1 [4].

B. Bilinear Map

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two multiplicative cyclic groups of prime order p, and g is a generator of $\mathcal{G}_1$. A bilinear map $\hat{e} : \mathcal{G}_1 \times \mathcal{G}_1 \longrightarrow \mathcal{G}_2$ has the following properties:

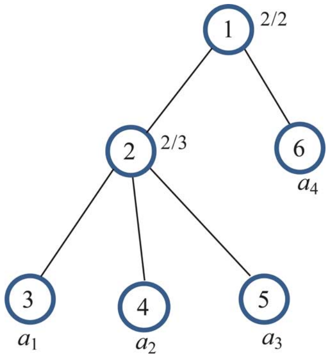


Fig. 1. Access tree—the graph representation of access policy. In this example, there are four attributes, and the Boolean function representation of the access policy is $\breve{a}_4(\breve{a}_1\breve{a}_2 + \breve{a}_2\breve{a}_3 + \breve{a}_1\breve{a}_3)$. If an user has attribute a_i, $\breve{a}_i$ is assigned to be TRUE.

- Bilinearity: for all $u, v \in \mathcal{G}_1$ and $a, b \in \mathbb{Z}_p$, we have $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$;
- Non-degeneracy: $\hat{e}(g, g) \neq 1$.

where both the group operation in $\mathcal{G}_1$ and the bilinear map $\hat{e}(\cdot)$ are efficiently computable. The input group $\mathcal{G}_1$ in a bilinear map is usually a point group over an elliptic (or hyperelliptic) curve.

C. Access Tree

In any access control scheme, there is an access policy which defines the access conditions under which a subject can access an object. An access tree $\mathbb{T}$ is a graph representation of the access policy. Such a tree includes non-leaf nodes and leaf nodes. Each leaf node is associated with a user attribute (e.g., age, gender, profession), while each non-leaf node has child nodes which may be leaf nodes, other non-leaf nodes or both. The root, a special non-leaf node, has no parent node. Without loss of generality, we tag the nodes in an access tree as follows. The root node N_1 is tagged with 1, and all the other nodes N_j are tagged with $j,\ j = 2, 3, \ldots$ sequentially. For simplicity, this paper refers a node using either N_j or j interchangeably unless otherwise stated. Each non-leaf node N_j is associated with a Boolean function derived from the access policy[1]. With reference to Fig. 1, the Boolean function of non-leaf node N_j is represented with n_j/n, which means that N_j has n child nodes, and its Boolean value is evaluated to be TRUE if it has at least n_j child nodes whose Boolean functions are evaluated to be TRUE. For instance, the Boolean function for node N_2 is 2/3, or equivalently $\breve{a}_1\breve{a}_2 + \breve{a}_2\breve{a}_3 + \breve{a}_1\breve{a}_3$, and it's TRUE if $a_1 \in \mathcal{S}$ and $a_2 \in \mathcal{S}$. Note that we use $\breve{a}_i$ to denote a Boolean variable which takes value TRUE if the attribute $a_i \in \mathcal{S}$.

We say that the set of attributes $\mathcal{A}$ of a user satisfies the access tree $\mathbb{T}$ if $\mathbb{T}(\mathcal{A})$=TRUE, which is iteratively defined as follows. For any leaf node N_j which is associated with an attribute $a_i \in \mathcal{A}$, its Boolean value is TRUE. For any non-leaf node, its Boolean value is the value of its Boolean function. If and only if the root node's Boolean value is TRUE, then $\mathbb{T}(\mathcal{A})$=TRUE. For example, given an attribute universe $\{a_1, a_2, a_3, a_4\}$, let the access policy be " if a_4 AND *Any two out of three in set* $\{a_1, a_2, a_3\}$ are included in set $\mathcal{S}$, *the access is granted* ", the

[1] This paper focuses on binary attribute value as a multi-level valued attribute can be represented by several binary valued attributes.

TABLE I
EVALUATION OF THE ACCESS TREE IN FIG. 1.

	$\mathcal{A}$	Node N_2	Node N_1 ($\mathbb{T}(\mathcal{A})$)	Access
1	a_1, a_2, a_3	TRUE	FALSE	No
2	a_3, a_4	FALSE	FALSE	No
3	a_1, a_2, a_4	TRUE	TRUE	Yes
4	a_2, a_3, a_4	TRUE	TRUE	Yes
5	a_1, a_3, a_4	TRUE	TRUE	Yes
6	a_1, a_2, a_3, a_4	TRUE	TRUE	Yes

access tree $\mathbb{T}$ in Fig. 1 is the graph representation of the access policy. Table I lists some results on $\mathbb{T}(\mathcal{A})$ with respect to various user attributes $\mathcal{A}$.

D. *Ciphertext Policy Attribute-Based Encryption*

In a CP-ABE scheme, every user's personal secret key is associated with a set of attributes while every ciphertext is associated with an access policy. A user successfully decrypts a ciphertext only if her set of attributes satisfies the access policy specified in the ciphertext. We briefly describe the CP-ABE in [3] below. We will extend this CP-ABE scheme to MCP-ABE scheme and use the latter in our access control scheme.

- **AB-Setup** (λ): is an initialization algorithm run by an Attribute Authority (AA). It takes as input a security parameter λ and outputs a public key PK and a master secret key MK. Specifically, AA chooses a group $\mathcal{G}_1$ of prime order p with generator g. Next it chooses two random exponents $\alpha, \beta \in \mathbb{Z}_p$, and outputs the public key

$$PK = (\mathcal{G}_1, g, g_1 = g^{\beta}, g_2 = \hat{e}(g,g)^{\alpha})$$

  and the master key $MK = \{\beta, g^{\alpha}\}$.
- **AB-KeyGen** $(MK, \mathcal{A})$: is run by AA to issue a personal secret key to a user. It takes as input MK and the set of attributes $\mathcal{A}$ of the user, and outputs the personal secret key SK associated with $\mathcal{A}$. Specifically, for each user, AA first randomly chooses $r \in \mathbb{Z}_p$, and $r_i \in \mathbb{Z}_p$, $\forall a_i \in \mathcal{A}$. Then AA gives the user

$$SK = (D = g^{(\alpha+r)/\beta}, \{D_i = g^r \mathcal{H}_1(a_i)^{r_i}, D'_i = g^{r_i}\}_{\forall a_i \in \mathcal{A}})$$

  as her personal secret key, where $\mathcal{H}_1(\cdot)$ is a one-way function, and an attribute a_i is represented as a string.
- **AB-Encrypt** $(PK, k, \mathbb{T})$: is run by a data owner to encrypt a message k according to an access tree $\mathbb{T}$. Technically, for each node N_j in tree $\mathbb{T}$, the data owner selects a polynomial f_j, sets the degree $d_j = n_j - 1$ where n_j is a threshold such that node N_j is TRUE if it has n_j child nodes whose Boolean values are TRUE. Furthermore, for the root node N_1, select a random $s \in \mathbb{Z}_p$, let $f_1(0) = s$ and let other points of the polynomial $f_1(\cdot)$ randomly. For any non-root node N_j in the tree, choose a polynomial $f_j(\cdot)$ by letting $f_j(0)$ be $f_{parent(N_j)}(j)$, and $f_j(x)$ be random $\forall x \neq 0$. The ciphertext is given by

$$\begin{aligned} CT = (B = kg_2^s = k\hat{e}(g,g)^{\alpha s}, \qquad (1)\\ C = g_1^s = (g^{\beta})^s = g^{\beta s}, \\ \{E_j = g^{f_j(0)}, E'_j = \mathcal{H}_1(a_i)^{f_j(0)}\}_{j\in\mathcal{L}}\mathbb{T}) \qquad (2) \end{aligned}$$

  where attribute $a_i \in \mathcal{S}$ is associated with a leaf node $N_j \in \mathcal{L}$, $\mathcal{S}$ is the set of attributes, and $\mathcal{L}$ is the set of leaf nodes.
- **AB-Decrypt** $(CT, \mathbb{T}, SK, \mathcal{A})$: is run by a data consumer in possession of a set of attributes $\mathcal{A}$ and the secret key SK in order to decrypt the ciphertext CT with an access policy $\mathbb{T}$. Specifically, for any node N_j in $\mathbb{T}$,
  (a) If N_j is a leaf node and is associated with attribute $a_i \in \mathcal{A} \cap \mathcal{S}$, set the Boolean value of node N_j to TRUE, and calculate

$$\begin{aligned} V_j &= DeNode_1(CT, SK, j, a_i) \\ &= \frac{\hat{e}(D_i, E_j)}{\hat{e}(D'_i, E'_j)} = \frac{\hat{e}(g^r\mathcal{H}_1(a_i)^{r_i}, g^{f_j(0)})}{\hat{e}(g^{r_i}, \mathcal{H}_1(a_i)^{f_j(0)})} \\ &= \frac{\hat{e}(g^r, g^{f_j(0)})\hat{e}(\mathcal{H}_1(a_i)^{r_i}, g^{f_j(0)})}{\hat{e}(g, \mathcal{H}_1(a_i))^{r_i f_j(0)}} \\ &= \hat{e}(g^r, g^{f_j(0)}) = \hat{e}(g,g)^{rf_j(0)} \qquad (3) \end{aligned}$$

  where the last three equations are due to the bilinearity property given in Section III-C; otherwise set $V_j \neq \perp$ (i.e., Boolean function value of node N_j is not TRUE).
  (b) If N_j is a non-leaf node, let $\mathcal{S}_j$ be its arbitrary n_j-sized set of child nodes N_k whose $V_k \neq \perp$. If no such set exists, i.e., node N_j does not satisfy the access policy, set $V_j = \perp$; otherwise, let Boolean function value of node N_j be TRUE, and compute

$$\begin{aligned} V_j &= DeNode_2(CT, SK, j) \\ &= \prod_{k\in\mathcal{S}_j} V_k^{\Delta_{k,\mathcal{S}_j}(0)} \\ &= \prod_{k\in\mathcal{S}_j} \hat{e}(g,g)^{rf_k(0)\Delta_{k,\mathcal{S}_j}(0)} \\ &= \prod_{k\in\mathcal{S}_j} \hat{e}(g,g)^{rf_j(k)\Delta_{k,\mathcal{S}_j}(0)} \\ &= \hat{e}(g,g)^{r\sum_{k\in\mathcal{S}_j} f_j(k)\Delta_{k,\mathcal{S}_j}(0)} = \hat{e}(g,g)^{rf_j(0)} \qquad (4) \end{aligned}$$

  where the Lagrange coefficient polynomial is

$$\Delta_{k,\mathcal{S}_j}(x) = \prod_{u\in\mathcal{S}_j, u\neq k} \frac{x-u}{k-u}.$$

  (c) For the root N_1, the output in step (b) is

$$\begin{aligned} V_1 &= DeNode_2(CT, SK, 1) \\ &= \hat{e}(g,g)^{rf_1(0)} = \hat{e}(g,g)^{rs}. \end{aligned}$$

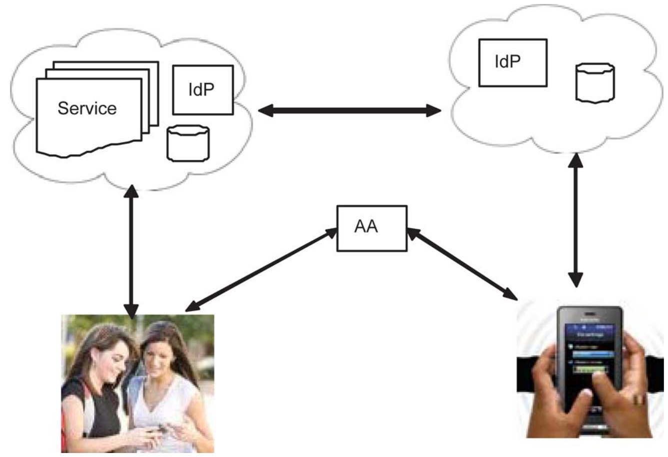


Fig. 2. Architecture of a cloud-assisted network, where IdP (Identity Provider) is used to authenticate users to the media provider.

Therefore, the decrypted message is

$$\begin{aligned}\frac{BV_1}{\hat{e}(C,D)} &= \frac{k\hat{e}(g,g)^{\alpha s}\hat{e}(g,g)^{rs}}{\hat{e}(g^{\beta s}, g^{(\alpha+r)/\beta})} \\ &= \frac{k\hat{e}(g,g)^{(\alpha+r)s}}{\hat{e}(g^{s}, g^{\alpha+r})} = k.\end{aligned}$$

IV. Media Sharing in Cloud Environment

A. System Architecture

With reference to Fig. 2, a media sharing application in cloud environment is composed of the following parties: backend servers, foreground servers, AA, and data consumers (or users).

Backend server is part of the infrastructure of the cloud computing platform. According to the National Institute of Standards and Technology (NIST) [5], cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider's interaction. Cloud computing platforms are assumed to have abundant storage capacity and computation power. Hence, from the viewpoints of network service providers, cloud computing significantly decreases the traffic and storage requirements incurred by their applications.

Foreground server provides the services which are always online. A server is often operated by a cloud service provider (CSP), but sometimes, a user is able to run his/her own services on the cloud platform too. The foreground services may include web service, database service, media maker service, media decoding service, identity management service, etc.

Attribute Authority (AA), a trusted third party [6], sets up the system parameters of attribute-based encryption system (such as system-wide public key PK and master key MK), verifies every user's attributes (e.g., group membership, role, security clearance or authorization information) and issues personal secret key corresponding to the set of attributes of the user. In practice, there could be multiple AAs in a system. For example, a university or corporate may run an AA, and a user may act as an AA for his/her extended family members. To keep the presentation simple, we assume a single AA in the rest of the paper.

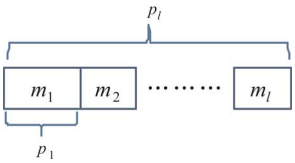


Fig. 3. Mapping between media units and privileges.

User may be a data owner, or a data consumer, or both. A data owner produces (protected or unprotected) media content (text, voice, video, etc.), and uploads the media content to cloud servers. To enforce access control to his data, the data owner assigns access privileges to data consumers whom the data owner may or may not know. A data consumer downloads media content of her interest from cloud servers, and obtains the content based on her attributes and the access policy of the data owner. To this end, the data consumer must obtain from AA a personal secret key SK bound to her set of attributes.

In this data owner-consumer model, the backend servers provide the fundamental platform for storage, networking, etc; the foreground servers provide the interface for media generation, transmission, and computational assistance to users; while AA issues personal secret keys so that access control can be enforced flexibly based on user attributes and media scalability.

B. Data Structure of Media

In media sharing applications, files of almost all formats could be exchanged. Particularly, for some media file formats such as text, PDF, Microsoft word, JPEG2000, H.26x, SVC files, and presentations, their content can be segmented into logical units. Each unit itself is meaningful, and more units provide more information. Thus, different sets of units can be viewed by different groups of consumers. We refer to such media content as *scalable* media content. For example, assuming that an SVC video file includes one base layer and two enhancement layers, we may assign three access privileges to users. If the unit for base layer is assigned to a consumer, she obtains a video experience of basic quality, but if the base layer and the two enhancement layers are available, the video shown to her will be of full fidelity. Fig. 3 illustrates a mapping between access privileges and media units, where a consumer with privilege p_j is allowed to access media units $(m_1, m_2, \ldots, m_j)$. To simplify the description and without loss of generality, we will assume the linear mapping given in Fig. 3 in the following.

V. Access Control on Scalable Media

Components in an attribute-based access control scheme includes subjects each specified by a set of attributes, objects and access policies. For example, a user's age, reputation, role are the subject attributes, while SVC stream files or presentation files are objects. An access policy defines the minimal attribute set which a subject should have in order to access the object. Therefore, the challenge in attribute-based access control is how to provide flexible and fine-grained access control at low cost.

In this section, we first present a security model. We then introduce an attribute-based access control scheme for scalable media based on a novel MCP-ABE scheme, an extension of the CP-ABE in [3].

A. Security Model

In the architecture described in Section IV-A, AA is a trusted authority which is responsible for issuing personal secret keys for consumers. Cloud servers are assumed to be honest but curious [7]. That is, they perform system operations faithfully, but try to find out as much information about consumers as possible. A consumer may be under the control of an adversary, and hence compromised such that she would attempt to access data beyond the scope of her access privileges. In addition, cloud servers might collude with a polynomial number of consumers. Communication channels between the data owners, data consumers and the cloud servers are assumed to be insecure.

B. Overview of the Access Control Scheme

Without loss of generality, assuming a media content consists of l units $\{m_1, m_2, \dots, m_l\}$, and its first j units are organized into an access level L_j which is accessible by a user with privilege $p_j, j = 1, \dots, l$, define a binary dominant relation $\preceq$ on the set of privileges $\{p_1, \dots, p_l\}$: $p_i \preceq p_j$ if and only if $\mathsf{L}_i \subseteq \mathsf{L}_j$, i.e.,

1) $\mathsf{L}_1 = \{m_1\}$;
2) $\mathsf{L}_{j+1} = \mathsf{L}_j \cup \{m_{j+1}\}, j = 1, 2, \dots, l-1$;
3) $p_1 \preceq \cdots \preceq p_j \preceq \cdots \preceq p_l$

In order to implement flexible access control for the scalable media content $\{m_1, m_2, \dots, m_l\}$, a data owner uses a unit key k_i to encrypt the media unit m_i into a ciphertext unit c_i with a standard symmetric cipher such as AES, $i = 1, 2, \dots, l$. Therefore, the access control problem is: given that a consumer has the attributes corresponding to access privilege p_j and the ciphertext units $\{c_1, c_2, \dots, c_j\}$, how can she obtain all the unit keys $\{k_1, k_2, \dots, k_j\}$?

The solution to the above problem is as follows. Initially, AA runs the algorithm **AB-Setup** in Section III-E to obtain the public key

$$PK = (\mathcal{G}_1, g, g_1 = g^{\beta}, g_2 = \hat{e}(g, g)^{\alpha})$$

and the master key $MK = \{\beta, g^{\alpha}\}$. Then AA publishes the public key to all users, and issues a personal secret key SK to every user using the algorithm **AB-KeyGen** . When a data owner wants to share a media content $\{m_1, m_2, \dots, m_l\}$ with others, he creates the access policy by mapping attributes of potential data consumers to the l privilege levels $\{p_1, p_2, \dots, p_l\}$, and calls algorithm *Data packing* in Section V-C to produce a package including protected media content and an enabling block. After receiving the package, a data consumer whose attributes match the access policy of privilege p_j, calls the algorithm *Data unpacking* in Section V-D to recover the unit keys $(k_1, \dots, k_j)$ and further decrypts the protected media in the package to the media units $(m_1, m_2, \dots, m_j)$.

C. Data Packing

As stated in Section V-B, the data owner aims to create a media content with l units $\{m_1, m_2, \dots, m_l\}$ that can be accessed by others at l privilege levels. To this end, he selects the unit key k_l from $\mathbb{Z}_p$ randomly, and generates a hash chain for unit keys as

$$k_j = \mathcal{H}_2(k_{j+1} \parallel j),\ j = l-1, \dots, 2, 1 \tag{5}$$

where $\mathcal{H}_2(\cdot)$ is a one-way function, $x \parallel y$ is the concatenation of string x and string y.

Next, the data owner encrypts the data unit m_j using the corresponding k_j and a symmetric cipher $\mathcal{E}(\cdot)$. Finally, he selects an attribute set $\mathcal{S}$ according to the access policy, and builds an access tree $\mathbb{T}$ whose leaf nodes are associated with $\mathcal{S}$. Specifically, the access tree $\mathbb{T}$ has a "trunk" and some "branches", where the "trunk" consists of l nodes $\{N_1, N_2, \dots, N_l\}$; node N_{l-j+1} is mapped to privilege level $p_j, j = 1, 2, \dots, l$. Each node in the "trunk" is the root of a subtree which has leaf nodes corresponding to a subset of $\mathcal{S}$. In order to protect all the unit keys k_j, we extend CP-ABE in Section III-E to MCP-ABE as follows. The data owner chooses a random value $s = f_1(0) \in \mathbb{Z}_p$ and publishes the ciphertext as $EM \parallel CT$:

$$EM = \{\mathcal{E}(m_j, k_j)\}_{j=1}^{l} \tag{6}$$

$$\begin{aligned} CT = (\{B_j = k_j g_2^{f_j(0)} = k_j \hat{e}(g, g)^{\alpha f_j(0)}\}_{j=1}^{l}, \qquad (7)\\ \{C_j = g_1^{f_j(0)} = (g^{\beta})^{f_j(0)} = g^{\beta f_j(0)}\}_{j=1}^{l},\\ \{E_j = g^{f_j(0)}, E'_j = \mathcal{H}_1(a_i)^{f_j(0)}\}_{j \in \mathcal{L}},\\ \mathbb{T}). \qquad (8) \end{aligned}$$

In (6), $\mathcal{E}(msg, k)$ is the encryption of message msg with unit key k and a standard symmetric cipher $\mathcal{E}(\cdot)$. In (7), CT is the enabling block for recovering unit keys by authorized data consumers, $\mathcal{L}$ is the set of leaf nodes in the tree $\mathcal{T}$ and $a_i \in \mathcal{S}$ is associated with leaf node $N_j \in \mathcal{L}$.

Remark: the difference on CT between CP-ABE and MCP-ABE comes from $\{B_j\}$ and $\{C_j\}$. As there are l messages (i.e., l unit keys) encrypted in the ciphertext CT of MCP-ABE, MCP-ABE is able to encrypt multiple messages within one ciphertext.

D. Data Unpacking

When a data consumer with personal secret key SK and attribute set $\mathcal{A}$ receives the ciphertext CT, she computes **AB-Decrypt**$(CT, \mathbb{T}, SK, \mathcal{A})$ as described in Section III-E. That is, she calculates V_j of every node N_j using either (3) (for leaf node N_j) or (4) (for non-leaf node N_j). Let p_J be the highest privilege level granted to the data consumer, she calculates

$$\begin{aligned} V_J &= DeNode_2(CT, SK, J) = \hat{e}(g, g)^{r f_J(0)}\\ \frac{B_J V_J}{\hat{e}(C_J, D)} &= \frac{k_J \hat{e}(g, g)^{\alpha f_J(0)} \hat{e}(g, g)^{r f_J(0)}}{\hat{e}(g^{\beta f_J(0)}, g^{(\alpha + r)/\beta})}\\ &= \frac{k_J \hat{e}(g, g)^{(\alpha + r) f_J(0)}}{\hat{e}(g^{f_J(0)}, g^{\alpha + r})} = k_J. \end{aligned} \tag{9}$$

Furthermore, she computes

$$k_j = \mathcal{H}_2(k_{j+1} \parallel j),\ j = J-1, \dots, 2, 1 \tag{10}$$

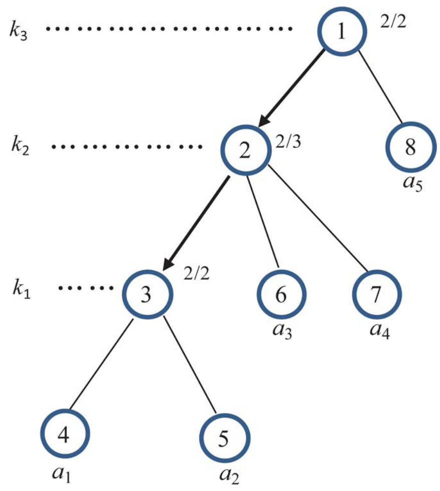


Fig. 4. Access tree for the example, where the arrows in the tree "trunk" indicate dependence of unit keys.

so as to derive all the unit keys granted to her. After obtaining all the necessary unit keys, she can then recover all the media units from EM.

E. Example

We use an example to further illustrate how the proposed access control scheme works. Assume that a data owner intends to share his presentation $\{m_1, m_2, m_3\}$, where m_1 is a text file, m_2 is the base layer (BL) of an SVC video and m_3 is the enhancement layer (EL) of the same SVC video. He builds a unit key chain ($k_3 \longrightarrow k_2 \longrightarrow k_1$) and the access tree as shown in Fig. 4. The data owner encrypts the presentation text, base layer and enhancement layer using the corresponding unit keys.

The present access control scheme is very expressive. Table II shows a portion of the access policies for the example. User 1 who has attribute a_1 only, can not recover any unit key, and hence has no access to the presentation at all. Neither does User 2. User 3 has attributes (a_1, a_2, a_5), hence, she is able to compute V_3 with the algorithm in Section III-E, but can recover neither V_2 nor V_1. Thus, User 3 is able to obtain k_1 from (9) for decrypting the text m_1. User 4 has attributes (a_1, a_2, a_3), and is able to recover (k_1, k_2) because the polynomial $f_2(\cdot)$ has degree 2, and thus she can obtain the text m_1 and the base layer m_2 of the SVC video. User 5 has attributes (a_3, a_4), and is able to obtain k_2, and derive k_1 using (10). User 6 is able to recover k_3 because she has attributes a_3 and a_4 to compute V_2 and attribute a_5 to compute V_8. She can then calculate the key k_3 using (9) and obtain the entire key chain from (10). As a result, User 6 has access to the entire presentation.

TABLE II
SOME ACCESS POLICIES FOR THE EXAMPLE.

	Attributes	Unit keys granted	Privilege	Presentation
User 1	a_1	None	None	Noise
User 2	a_2	None	None	Noise
User 3	a_1, a_2, a_5	k_1	p_1	Text
User 4	a_1, a_2, a_3	k_1, k_2	p_2	Text, BL
User 5	a_3, a_4	k_1, k_2	p_2	Text, BL
User 6	a_3, a_4, a_5	k_1, k_2, k_3	p_3	Text, BL, EL

VI. DISCUSSIONS

A. Cloud-Assisted Decryption

In the data unpacking process, a data consumer has to call the decryption algorithm $DeNode_1(\cdot)$ or $DeNode_2(\cdot)$ repeatedly. For each invocation, $DeNode_1(\cdot)$ (i.e., (3)) has 2 bilinear map operations, while $DeNode_2(\cdot)$ (i.e., (4)) has some modular exponentiation operations. Thus, the unpacking process is computationally intensive for resource constrained devices such as smartphones. To reduce the computational burden of such devices, we propose to offload the decryption operations to a cloud server. In order to guarantee security, it is important that no information is leaked to network attackers and the cloud server. To this end, the data consumer selects a random $z \in \mathbb{Z}_p$ after she registers to the AA, and blinds her secret $\{D_i, D'_i\}_{a_i \in \mathcal{A}}$ as

$$\{\tilde{D}_i, \tilde{D}'_i\}_{a_i \in \mathcal{A}} = \{D_i^z, (D'_i)^z\}_{a_i \in \mathcal{A}}.$$

She sends $\{\tilde{D}_i, \tilde{D}'_i\}_{a_i \in \mathcal{A}}$ and D to the cloud server after registration. In the decryption process, she sends the ciphertext CT to the cloud server. With reference to (3), the cloud server is able to calculate

$$\begin{aligned} \tilde{V}_j &= \frac{\hat{e}(\tilde{D}_i, E_j)}{\hat{e}(\tilde{D}'_i, E'_j)} = \frac{\hat{e}(D_i^z, E_j)}{\hat{e}((D'_i)^z, E'_j)} \\ &= \frac{\hat{e}(D_i, E_j)^z}{\hat{e}(D'_i, E'_j)^z} = V_j^z \end{aligned} \tag{11}$$

for the leaf node N_j associated with attribute a_i, $\forall a_i \in \mathcal{A} \cap \mathcal{S}$, where (11) holds due to the property of bilinear map in Section III-C. Moreover, with reference to (4), the cloud server is able to calculate

$$\begin{aligned} \tilde{V}_j &= \prod_{k \in \mathcal{S}_j} \tilde{V}_k^{\Delta_{k,\mathcal{S}_j}(0)} = \prod_{k \in \mathcal{S}_j} (V_k^z)^{\Delta_{k,\mathcal{S}_j}(0)} \\ &= \prod_{k \in \mathcal{S}_j} V_k^{z\Delta_{k,\mathcal{S}_j}(0)} = V_j^z \end{aligned} \tag{12}$$

for non-leaf node N_j.

If the data consumer has the privilege p_J, the cloud server can further compute

$$\begin{aligned} W_J &= \frac{B_J}{\hat{e}(C_J, D)} = \frac{k_J \hat{e}(g, g)^{\alpha f_J(0)}}{\hat{e}(g^{\beta f_J(0)}, g^{(\alpha+r)/\beta})} \\ &= \frac{k_J}{\hat{e}(g, g)^{r f_J(0)}}. \end{aligned}$$

Then the server returns to the consumer W_J and $\tilde{V}_J$, and the data consumer will calculate

$$\begin{aligned} W_J(\tilde{V}_J)^{1/z} &= W_J V_J \\ &= \frac{k_J}{\hat{e}(g, g)^{r f_J(0)}} \times \hat{e}(g, g)^{r f_J(0)} = k_J. \end{aligned}$$

Finally, the data consumer can recover the rest of the unit keys according to (10).

B. Provable Security

As the CP-ABE scheme [3] is provably secure against user collusion attack, and the difference between MCP-ABE and

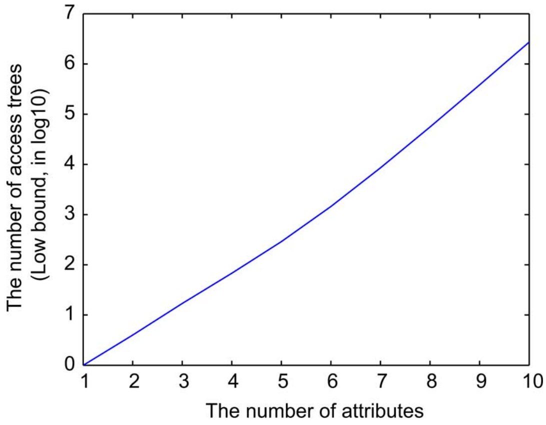


Fig. 5. The low bound of access policies vs the number of attributes.

CP-ABE is that MCP-ABE uses the intermediate result of CP-ABE encryption to encrypt messages, this modification does not disclose any information. Hence, MCP-ABE is secure even if a group of user launches the collusion attack. For example, in Fig. 4, although User 1 has the personal secret key for attribute a_1 and User 2 has the personal secret key for attribute a_2, and hence they can calculate the value $V_4 = \hat{e}(g, g)^{rf_3(4)}$ and $V_5' = \hat{e}(g, g)^{r' f_3(5)}$; however, they can not calculate V_3 (or V_3') because the random values r and r' are selected differently, randomly and securely with their personal secret keys.

A greedy data consumer may try to elevate her lower privilege level to a higher level. For example, suppose she has the privilege to recover the key k_j and tries to obtain the key k_{j+1}. However, this is computationally infeasible due to the one-wayness of the hash function used in the key chain generation.

In addition, Section VI-A enables that a consumer to offload the time-consuming decryption to the cloud server. This offloading algorithm is similar to the provably-secure scheme in [27]. Hence, it does not undermine the security level of the MCP-ABE.

C. Flexible Access Control

In the access control scheme, a data owner defines his access policy based on the attributes of data consumers without explicitly list their names. If the access policy can be any Boolean functions defined with "AND", "OR" and "NOT" operations on attributes, the number of access policies supported is $2^{2^n-1} - 1$ for n attributes. However, CP-ABE restricts that the attribute negation does not exist in the Boolean functions such that the number of supported access policies is much smaller. Nonetheless, the number of access policies supported by the present access control scheme is still very large. Fig. 5 shows the low-bound of the number of supported Boolean funcstions which includes each attribute once only. For example, the number of access trees is more than 8553 when there are 7 attributes in the system. If it is necessary to handle all the access policies (or Boolean functions), both attributes and their negations must be included in the access tree. Ostrovsky *et al.*. [28] generalized CP-ABE so as to solve this extended access control problem. For simplicity, this paper ignores how to extend MCP-ABE to the generalization of CP-ABE. In all, the present scheme is capable of realizing flexible access control to media content.

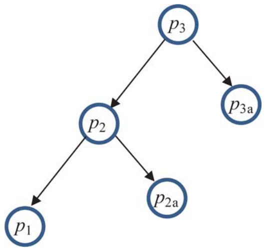


Fig. 6. Privilege tree for representing the relationship among the privileges, where $p_i \longrightarrow p_j$ indicates that privilege p_i dominates p_j.

Sometimes, new attributes are added into the content delivery networks. Our scheme can cope with new attributes easily. Specifically, once AA updates the personal secret keys for the corresponding data consumers, any data owner is able to incorporate the new attributes in the access policy. Hence, It can constantly adapt to the change of user profiles.

D. Fine-Grained Access Control

The proposed scheme allows different access privileges for the same media content. Hence, it matches the double-blindness property in cloud computing where data consumers might be unknown to the policy maker (i.e., the data owner) and vice verse. In addition, from the viewpoint of data consumers, the scheme supports flexible and scalable content rendering. For example, if a data consumer renders an SVC video on a small-size smartphone, she only needs to compute the unit key for the base layer and decrypts the base layer. However, if she likes to render the video on a desktop computer, she can decrypt all the SVC layers so long as she has the attributes corresponding to the highest access privilege.

In the above presentation we assumed that the access privileges are totally ordered such that the unit keys form a chain. Our access control scheme can be easily extended to support privileges having a tree hierarchy, as shown in Fig. 6, in which there are privileges p_i and p_j such that $p_i \not\preceq p_j$ and $p_j \not\preceq p_i$. An example of privileges with a tree hierarchy is when a JPEG2000 image is encoded according to both quality and resolution. The privileges for such a JPEG2000 image can be defined in two dimensions such that the privileges form a tree rather than a chain. To deal with this access policy, we can associate the unit keys to the non-leaf nodes which corresponding to the privilege tree, and the privilege tree is a sub-tree of the access tree elaborated in Section III-D. The packing and unpacking algorithms are similar to those given in Section V.

E. Lightweight Communication Overhead

Section II-A introduces the existing attribute-based access control schemes. As those schemes employ either CP-ABE (e.g., [9], [13]), or KP-ABE (e.g., [10]) or both (e.g., [12], [14]), and KP-ABE has the same communication overhead as CP-ABE, we compare CP-ABE with MCP-ABE only. To encrypt l messages, the payload for the enabling block in CP-ABE and MCP-ABE are

$$\begin{aligned} P_{CP} &= l(2d_1 + \mid \bar{\mathcal{L}} \mid (d_1 + d_2) + \mid \mathbb{T} \mid) \\ P_{MCP} &= 2ld_1 + \mid \mathcal{L} \mid (d_1 + d_2) + \mid \mathbb{T}' \mid \end{aligned}$$

respectively, where d_1 (or d_2) is the size of the elements in the group $\mathcal{G}_1$ ($\mathcal{G}_2$ respectively), $| \bar{\mathcal{L}} |$ is the number of leaf nodes in CP-ABE access tree on average, and $| \mathcal{L} |$ is the number of leaf nodes in the access tree for full privilege, hence, $| \bar{\mathcal{L}} |= 0.5 | \mathcal{L} |$. Their difference is

$$\begin{aligned}\delta P &= L_{CP} - L_{MCP} \\ &= (l \mid \bar{\mathcal{L}} \mid - \mid \mathcal{L} \mid)(d_1 + d_2) + (l \mid \mathbb{T} \mid - \mid \mathbb{T}' \mid) \\ &\approx (0.5l - 1) \mid \mathcal{L} \mid (d_1 + d_2) + (l \mid \mathbb{T} \mid - \mid \mathbb{T}' \mid) \\ &> (0.5l - 1) \mid \mathcal{L} \mid (d_1 + d_2)\end{aligned}$$

where the last inequality holds because the access trees in CP-ABE must have all the nodes in the access tree in MCP-ABE. Thus, MCP-ABE is much more efficient than CP-ABE for access control of scalable media content if the multiple-priority access policy can be represented with access tree. But the advantage varies with the access policy. In the worse case that each access priority is represented with one access tree, MCP-ABE has the same communication overhead as CP-ABE.

VII. Experiments

A. Configuration

We set up a private cloud with three computers supporting BIOS virtualization technology so as to simulate a group of computers. We also set up a console with *Ubuntu Desktop* 11.04 and use *OpenStack Flat Network* mode to configure the computer network. In the experiments, a virtual PC (over a Dell Precision 390 with Intel Core2 Duo@2.4GHz) is used as a cloud server for assisting decryption operations for mobile devices, and a smartphone SAMSUNG (s5830@800MHz) is used as the platform for a data consumer. Besides, a desktop PC is used as AA for generating keys, and pack media data for a data owner.

To implement the access control scheme, we adopted the bilinear map software pbc[2][29] which uses a 160-bit elliptic curve group based on the supersingular curve $y^2 = x^3 + x$ over a 512-bit finite field. In addition, the hash function SHA-1 is used to generate the key chain.

B. Experiments on Performance

In the experiments, we selected 23 video clips randomly, and encoded them into 1 base layer and up to 3 enhancement layers, and defined 23 access policies for distributing the content to consumers with different access priorities. The bilinear map software package pbc is installed at the smartphone to carry out all the decryption operations of MCP-ABE.

1) Packing Time: Each media content used in our experiment is a presentation with one text file and an SVD sequence. The data owner uses the notebook to generate the key chain, and calculate the package $EM \parallel CT$ as elaborated in Section V-C. The presentation is packed into 3 privileges. A desktop E7200 (Intel Core2 Duo @2.53GHz) takes 327 ± 172 ms (excluding video encoding) to pack a presentation.

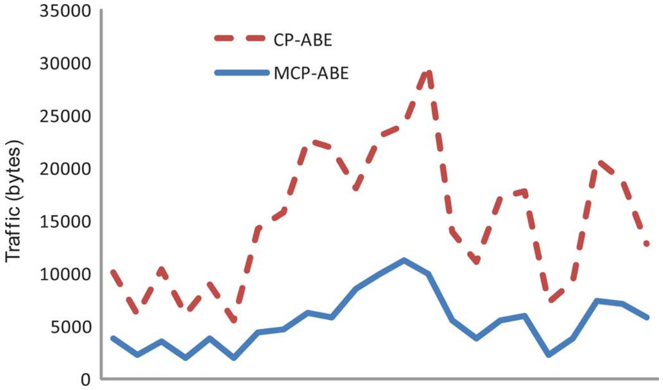


Fig. 7. Communication overhead for 3-level acess.

2) Unpacking Time: Assuming a user have the privilege p_3. She uses the smartphone to recover the content. We adopted two decryption scenarios for this purpose. In the first scenario, the smartphone carries on the decryption directly to recover all the units keys k_1, k_2 and k_3. The decryption time is 312.8 ± 103 ms (mean ± standard deviation).

In the second scenario, the decryption operation is offloaded from smartphone to the cloud server which runs on a virtual machine. In this case, the bilinear map software packages `pbc` and `cpabe`[30] are used at the cloud server. As a result, the smartphone spends only 166 ms. Hence, offloading significantly reduces the computational latency.

3) Communication Overhead: When a data owner bradcasts the encrypted content, he has to broadcast the enabling block which is the communication overhead. As a comparison, CP-ABE is used 3 times so as to provide the multi-level access.

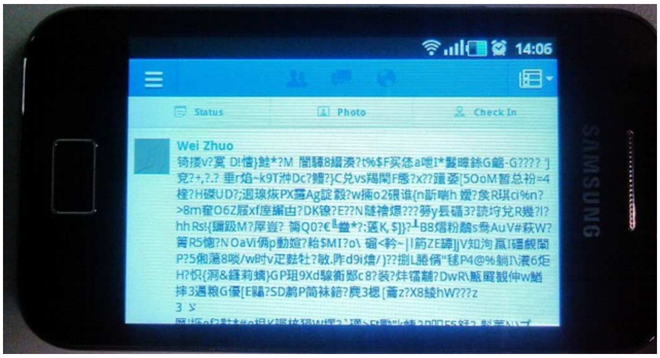
Fig. 7 shows the communication overhead for 3-level presentation media. We observe that MCP-ABE has smaller overhead than CP-ABE. Statiscally, the overhead of MCP-ABE is 5437 ± 2627 bytes, while the overhead of CP-ABE is 9588 ± 4452 bytes. We also know that the traffic overhead varies with the access tree and the number of attributes. This is easily available from the formulas of both MCP-ABE and CP-ABE. Besides, the traffic overhead varies with the number of access levels too. For example, in the experiments for the 4-level policies defined with 7 attributes, the total traffic cost is 7579 ± 2633 bytes for MCP-ABE, and 13601 ± 3606 bytes for CP-ABE.

C. Experiments on Access Control

To demonstrate the multi-level access control, we constructed a presentation with one text file and a fireman sequence (30 frames, encoded into one base layer and two enhancement layers, with a total 103,871 bytes), and generated the encrypted presentation and its enabling block with the method elaborated in Section VII-B1. A smartphone is used to carry out all the decryption operations by exploiting the JAVA bilinear map software package [31] after obtaining the encrypted presentation and its enabling block.

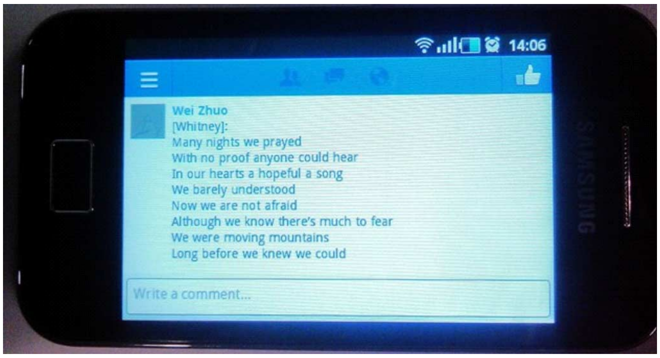
With respect to Table II, neither User 1 nor User 2 is authorized to access anything; hence, their handphones output garbled data shown in Fig. 8. User 3 is able to view the text file shown in Fig. 9, and User 4 obtains the base layer shown in Fig. 10 in addition to the text file in Fig. 9. User 6 gets the full content, i.e.,

[2] The C-language PBC library is recompiled for the dedicated smartphone and wrapped with Java, then the wrapped code is installed on the smartphone for unpacking. The disadvantage of this "native" programming is that one executable may be applicable to one smartphone brand only.



Wei Zhuo
锜掺v?寞 D!懀}鮭*?M 閻驍8繧湊?t%$F买恷a嗺I*�容暲鍒G鎗-G????']
兗?+,?.? 垂r焰~k9T浺Dc?鱧?}C兑vs羯閧F慼?x??躍耍[5OoM暂总衸=4
桎?H磔UD?;迴璅烣PX霹Ag諚斅?w摛o2硍谁{n斵喘h 嫚?矦R琪ci%n?
>8m奃O6Z屜xf座緉由?DK锿?E??N隧禬熼???簩y镸礵3?読埒兌R幾?I?
hh Rs!{鑼蹳M?厡豈? 簐Q0?€╚蠻*?:藹K, $]}?┴B8熠粉鸕s蠢AuV#萩W?
箐R5憁?N OaVi偊p動媗?粭$MI?o\ 碯<矜~|I篛ZE譁]jV知洵 羸I礌靓閫
P?5俐蕩8啖/w时v疋鮭牡?敏.阼d9i熗/)??捌L腃偦"毽P4@%躺I\濮6炬
H?愳{洞&鏠莉螭}GP珇9Xd騄衡鄖c8?装?炷镭輔?DwR\甎靨覩伸w鮨
摔3遖粮G優[E驪?SD鹇P简袜鎯?赓3楒[蕭z?X8綾hW???z
3 ゞ
[illegible]

Fig. 8. Garbled text file shown to unauthorized User 1 or User 2.



Wei Zhuo
[Whitney]:
Many nights we prayed
With no proof anyone could hear
In our hearts a hopeful a song
We barely understood
Now we are not afraid
Although we know there's much to fear
We were moving mountains
Long before we knew we could

Write a comment...

Fig. 9. Clear text file shown to User 3.

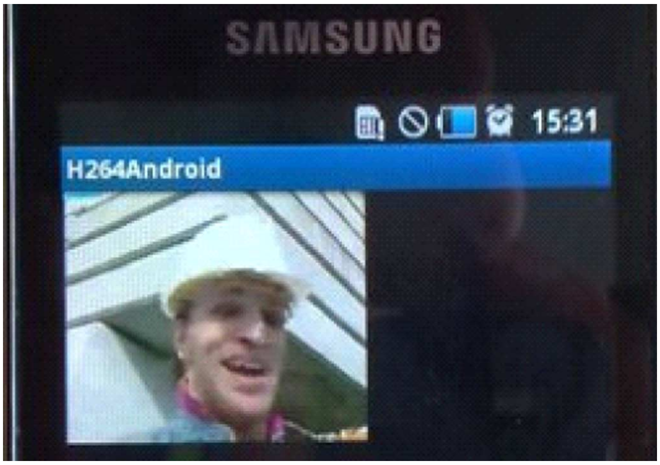


Fig. 10. Base layer shown to User 4 and User 5.

the text file shown in Fig. 9 and the full video shown in Fig. 11. In other words, MCP-ABE realizes the multi-message encryption for meeting the multi-privilege access control requirement.

VIII. Conclusions and Future Work

In order to share media content in a controllable manner, a suitable access control mechanism should be deployed. CP-ABE based access control allows a data owner to enforce access control based on attributes of data consumers without explicitly naming the specific data consumers. However, CP-ABE supports only one privilege level and hence is not suitable for access control to scalable media. In this paper we extended CP-ABE to a novel MCP-ABE and proposed a scheme to support multi-privilege access control to scalable media. As cloud computing is increasingly being adopted and mobile devices are becoming pervasive, the present access control scheme allows a mobile user to offload computational intensive MCP-ABE operations to cloud servers while without compromising user's security. The experimental results indicated that the proposed access control scheme is efficient for securely and flexibly managing media content in large, loosely-coupled, distributed systems.

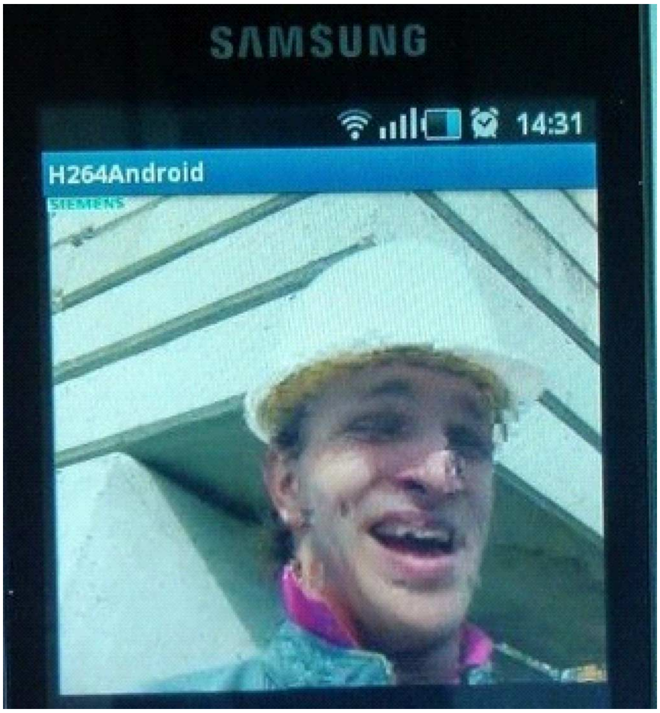


Fig. 11. Full video shown to User 6.

With the assistance of the cloud server, the decryption operation is accelerated significantly at the consumer side. However, the decryption may be still slow for low-end devices because a modular exponentiation operation is required. Thus, one future work is how to speed-up the decryption operation at low-end devices.

Acknowledgment

Dr. Hui Fang set up the Cloud computing environment, and Mr. Yifan Zhao developed some codes of MCP-ABE and carried on some experiments.

References

[1] E. Messmer, "Are security issues delaying adoption of cloud computing?," *Network World*, Apr. 2009 [Online]. Available: http://www.networkworld.com/news/2009/042709-burning-security-cloud-computing.html

[2] E. Messmer, "Security of virtualization, cloud computing divides IT and security pros," *Networkworld.com*, Feb. 2010 [Online]. Available: http://www.networkworld.com/news/2010/022210-virtualization-cloud-security-debate.html

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

[4] *National Inst. Standards and Technol., Secure Hash Standard (SHS)*, FIPS Publication 180-1, 1995.

[5] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication 800-145, 2011.

[6] M. D. Soete, "Attribute certificate," in *Encyclopedia of Cryptography and Security*, H. C. A. Van Tilborg and S. Jajodia, Eds., 2nd ed. Berlin, Germany: Springer, 2011, p. 51.

[7] B. Carbunar, Y. Yu, W. Shi, M. Pearce, and V. Vasudevan, “Query privace in wireless sensor networks,” *ACM Trans. Sensor Networks*, vol. 6, no. 2, 2010, Art. ID 14.

[8] X. Zhang, M. Nakae, M. J. Covington, and R. Sandhu, “Toward a usage-based security framework for collaborative computing systems,” *ACM Trans. Inf. Syst. Security*, vol. 11, no. 1, pp. 1–36, 2008.

[9] S. Jahid, P. Mittal, and N. Borisov, “EASiER: Encryption-based access control in social networks with efficient revocation,” in *Proc. ACM Symp. Inf. Computer Commun. Security*, Mar. 2011, pp. 411–415.

[10] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in *Proc. IEEE Int. Conf. Comput. Commun.*, 2010, pp. 1–9.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute based encryption for fine-grained access conrol of encrypted data,” in *Proc. ACM Conf. Comput. Commun. Security*, 2006, pp. 89–98.

[12] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, “Secure attribute based systems,” in *Proc. ACM Conf. Comput. Commun. Security*, 2006, pp. 99–112.

[13] J. A. Akinyele, C. U. Lehmanny, M. D. Green, M. W. Pagano, Z. N. J. Petersonz, and A. D. Rubin, “Self-protecting electronic medical records using attribute-based encryption,” in *Proc. ACM Conf. Comput. Commun. Security: Workshop on Security and Privacy in Smartphones and Mobile Devices*, Oct. 2011, pp. 75–86.

[14] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, “Persona: An online social network with user defined privacy,” in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 135–146.

[15] S. J. Wee and J. G. Apostolopoulos, “Secure scalable streaming enabling transcoding without decryption,” in *Proc. IEEE Int. Conf. Image*, 2001, pp. 437–440.

[16] V. Gergely and G. Feher, “Enhancing progressive encryption for scalable video streams,” in *Proc. EUNICE, Open European Summer School and IFIP TC6.6 Workshop on The Internet of the Future*, 2009, vol. 5733, Lecture Notes in Computer Science, pp. 51–58.

[17] Y. Wu, D. Ma, and R. H. Deng, “Flexible access control to JPEG2000 image code-streams,” *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1314–1324, Oct. 2007.

[18] ISO/IEC 14496-2, Coding of Audio-Visual Objects-Part 2: Visual.

[19] B. B. Zhu, C. Yuan, Y. Wang, and S. Li, “Scalable protection for MPEG-4 fine granularity scalability,” *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 222–233, Apr. 2005.

[20] T. W. H. Schwarz and D. Marpe, “Overview of the scalable video coding extension of the h.264/avc standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[21] S.-W. Park and S.-U. Shin, “Efficient selective encryption scheme for the H.264/scalable video coding (SVC),” in *Proc. Int. Conf. Networked Computing Adv. Inf. Manag.*, 2008, vol. 1, pp. 371–376.

[22] S. Lian, “Secure service convergence based on scalable media coding,” *Telecommun. Syst.*, vol. 45, no. 1, pp. 21–35, 2010.

[23] C. Li, X. Zhou, and Y. Zhong, “NAL level encryption for scalable video coding,” in *Proc. Pacific-Rim Conf. Multimedia*, 2008, vol. 5353, Lecture Notes in Computer Sci., pp. 496–505.

[24] G. B. Algin and E. T. Tunali, “Scalable video encryption of H.264 SVC codec,” *J. Vis. Commun. Image Representation*, vol. 22, no. 4, pp. 353–364, 2011.

[25] Y. G. Won, T. M. Bae, and Y. M. Ro, “Scalable protection and access control in full scalable video coding,” in *Proc. Int. Workshop Digital Watermarking*, 2006, pp. 407–421.

[26] H. Hellwagner, R. Kuschnig, T. Stutz, and A. Uhl, “Efficient in-network adaptation of encrypted H.264/SVC content,” *Image Commun.*, vol. 24, no. 9, pp. 740–758, 2009.

[27] M. Green, S. Hohenberger, and B. Waters, “Outsourcing the decryption of ABE ciphertexts,” Usenix Security, 2011.

[28] R. Ostrovsky, A. Sahai, and B. Waters, “Attribute-based encryption with non-monotonic access structures,” in *Proc. ACM Conf. Comput. Commun. Security*, 2007, pp. 195–203.

[29] Pairing-Based Cryptography Library [Online]. Available: http://crypto.stanford.edu/pbc/

[30] J. Bethencourt, A. Sahai, and B. Waters, Ciphertext-Policy Attribute-Based Encryption Mar. 24, 2011 [Online]. Available: http://acsc.cs.utexas.edu/cpabe/

[31] The Java Pairing Based Cryptography Library [Online]. Available: http://gas.dia.unisa.it/projects/jpbc/

Yongdong Wu Received the B.A and M.S. from Beihang University, the Ph.D degree from Institute of Automation, Chinese Academy of Science, and Master for Management of Technology from National University of Singapore. He is currently a senior scientist with cryptography and Security Department, Institute of Infocomm Research ($\mathbf{I}^2\mathbf{R}$), A*STAR, Singapore. He is also an adjunct associate professor with the Singapore Management University. His interests includes multimedia security, e-Business, Digital Right Management and Network security. He has published more than 100 papers, and 7 patents. His research results and proposals was incorporated in the ISO/IEC JPEG 2000 security standard 15444–8 in 2007. He received the Best Paper Award of the 13th Joint IFIP TC6 & TC11 Conference on Communications and Multimedia Security (CMS 2012).

Zhuo Wei received the B.A. degree from Jilin University, China, the M.S. degree and PhD degree from Huazhong University of Science and Technology, China. He is currently a research Fellow in Singapore Management University. His interests include image processing, and video processing. He received the Best Paper Award of the 13th Joint IFIP TC6 & TC11 Conference on Communications and Multimedia Security (CMS 2012).

Robert H. Deng has been a Professor at the School of Information Systems, Singapore Management University since 2004. Prior to this, he was Principal Scientist and Manager of Infocomm Security Department, Institute for Infocomm Research, Singapore. His research interests include data security and privacy, multimedia security, network and system security. He was the Associate Editor of the IEEE Transactions on Information Forensics and Security from 2009 to 2012. He is currently Associate Editor of IEEE Transactions on Dependable and Secure Computing, Associate Editor of Security and Communication Networks (John Wiley), and a member of Editorial Board of Journal of Computer Science and Technology (the Chinese Academy of Sciences). He is the co-chair of the Steering Committee of the ACM Symposium on Information, Computer and Communications Security (ASIACCS).

He received the University Outstanding Researcher Award from the National University of Singapore in 1999 and the Lee Kuan Yew Fellow for Research Excellence from the Singapore Management University in 2006. He was named Community Service Star and Showcased Senior Information Security Professional by (ISC)^2 under its Asia-Pacific Information Security Leadership Achievements program in 2010. He received the Distinguished Paper Award of the 19th Annual Network & Distributed System Security Symposium (NDSS 2012) and the Best Paper Award of the 13th Joint IFIP TC6 & TC11 Conference on Communications and Multimedia Security (CMS 2012).