Scalable Access Control For Privacy-Aware Media Sharing

Changsha Ma, Zhisheng Yan, *Member, IEEE* and Chang Wen Chen, *Fellow, IEEE*

***Abstract*—The prevalence of social networks has made it easier than ever for users to share their photos, videos, and other media content with anybody from anywhere. However, the easy access of user-generated media content also brings about privacy concerns. Traditional access control mechanisms, where a single access policy is made for a specific piece of content, cannot satisfy the user privacy requirements in large-scale media sharing systems. Instead, configuring multiple levels of access privileges for the shared media content is desired. On one hand, it conforms to the principle of social networks in information propagation. On the other hand, it accords with the diverse and complex social relationship among social network users. In this paper, we propose a scalable media access control (SMAC) system to enable such a configuration in a secure and efficient manner. The proposed SMAC system is empowered by the scalable ciphertext policy attribute-based encryption (SCP-ABE) algorithm as well as a comprehensive key management scheme. We provide formal security proof to prove the security of the proposed SMAC system. Additionally, we conduct intensive experiments on mobile devices to demonstrate its efficiency.**

***Index Terms*—Social media sharing, privacy, access control, SCP-ABE, scalable media format**

I. Introduction

The prevalence of social networks has boosted the advancement of a variety of user generated content (UGC) such as texts, photos, and videos. The popularity and the easy access of UGC brings about new opportunities for numerous applications such as personal branding and commercial advertising. For example, photographers can utilize Instagram and Flickr to promote their works. Similarly, users can advertise products, ideas, and themselves by creating YouTube channels.

However, UGC sharing also results in privacy concerns [1], [2]. One of the primary privacy concerns is content repurposing by third parties [3]. For example, the content shared on social networks can be plagiarized by others and served for their own profitable purpose. Additionally, displaying informative media content such as photos and videos on the social networks can easily disclose sensitive user information, such as friendships, hobbies, and footprints, to untrusted ones. With the advancement of image/video processing and artificial intelligence techniques that might uncover more personal information [4], [5], the privacy concerns caused by UGC sharing will become critical. Ultimately, privacy preservation will be a mandatory feature to keep the prosperity of social media. Therefore, it is imperative to build privacy awareness in existing social media sharing system.

The root cause of the privacy issues in UGC sharing is that users have little control on the information propagation in social networks, i.e. who will be viewing their shared content. Although users can usually enable or disable other users to access their shared content by configuring privacy settings on social networks, they cannot prevent social network servers from leaking their content to third parties without their authorization. Some existing work proposes to protect user privacy through visual obfuscation [6], [7] or transmorphing [8] on the image region of interests (ROI), such as human faces. However, it is hardly feasible to determine the ROI for arbitrary media content, since different content consumers might have different ROI of the same media content. Such a method is therefore not applicable for social media sharing. Alternatively, encryption based access control can be leveraged for privacy-aware media sharing [9], [10]. Specifically, the content owner sets an access policy for the encrypted content. Whether a content consumer can decrypt the content depends on the availability of the access privilege [11], [12], [13]. Unfortunately, the traditional single access policy based access control mechanism cannot satisfy the user privacy requirements in large-scale media sharing systems, which in turn results in a barrier of populating privacy preservation on social media sharing [14]. On one hand, the mechanism sacrifices the content popularity. A single access policy can easily block a large number of users, which significantly degrades the breadth and depth of propagation as well as the degree of interactions on the content. On the other hand, a single on-off access privilege cannot accommodate the diverse multi-level social relationship since it simply divides all users into two groups.

Therefore, instead of utilizing traditional access control mechanisms, it is necessary to develop a mechanism that is able to flexibly balance privacy preservation and content propagation, and to support multiple levels of access privileges for the shared content in large-scale social media systems [15].

In this paper, we propose a scalable media access control (SMAC) system to achieve this goal. In the proposed system, a media stream is encoded into multiple levels of perceptual quality by exploiting techniques such as JPEG 2000 [16] and scalable video coding (SVC) [17]. Specifically, low-quality media content has relatively lower resolution, lower signal-noise ratio (SNR), or lower frame rate, compared to high-quality media content. In the SMAC system, sacrificing the breath of media content propagation is not the only choice

Changsha Ma is with Department of Computer Science and Engineering, University of Buffalo, email: changsha@buffalo.edu.

Zhisheng Yan is with Department of Computer Science, Georgia State University, email: zyan@gsu.edu. Zhisheng Yan is the corresponding author.

Chang Wen Chen is with School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, and Department of Computer Science and Engineering, University of Buffalo, email: chencw@cuhk.edu.cn, chencw@buffalo.edu

for user privacy preservation. Indeed, the system can allow widespread propagation for the low-quality media content while enforce a more restricted access policy on the high-quality media content. For example, a content owner can enable every user in the media sharing system to access the low quality version of the content, but only allow the trusted ones to view a high quality version. The rationality of the mechanism relies on that the low-quality media content is less commonly used in re-purposing [18], [19] and is more robust in resisting analysis based attacks than the high-quality media content [20], [21]. Such a mechanism is especially expected when the users do not need a very restricted access policy on their shared media content, which is the most common case in social networks [14]. Additionally, the SMAC system is able to support arbitrary levels of trust relationship by configuring the same number of access policies efficiently. This way, we can ensure that the content with a specific quality is propagated along the corresponding trusted chain.

Developing the SMAC system is faced with two non-trivial challenges: 1) how to securely enforce multiple access policies for a scalable media stream; 2) how to reliably authenticate the access privileges of content consumers and manage their dynamics. To tackle these challenges, we first propose a scalable ciphertext policy attribute-based encryption (SCP-ABE) algorithm that can securely encrypt a multi-dimensional scalable media stream. Under a set of social attribute-based access policies, the media stream can be decoded into media content with various levels of quality from multiple dimensions. Thus a content consumer whose social attributes satisfy the access policy will obtain the right access keys to decrypt the media stream, and decode and view the content with a specific quality. If a consumer's attributes match more than one access policy of the media stream, the individual will enjoy a higher access privilege and a higher viewing quality of the content. Furthermore, we propose a comprehensive key management scheme to handle the access key distribution and revocation. It is able to reliably authenticate the attributes of consumers, and distribute and revoke their corresponding access keys. In addition, the proposed scheme shifts most of the key management cost from the content distributer side to the more powerful social network server side. In this way, the privacy preservation cost on the distributor side does not increase with the number of content consumers but only depends on the number of shared contents. Through formal security analysis, we prove the security and reliability of the SMAC system. Furthermore, we conduct practical experiments on mobile devices to demonstrate its efficiency.

To summarize, we make the following contributions.

- We present the first access control scheme that protects user privacy in large-scale media sharing systems, satisfying two essential user requirements, i.e., widespread content propagation and multiple-level access privileges.
- We propose a SCP-ABE algorithm that is able to securely enforce multiple access policies on multi-dimensional scalable media streams.
- We propose a comprehensive key management scheme that facilitates the reliable and efficient access privilege authorization and revocation.

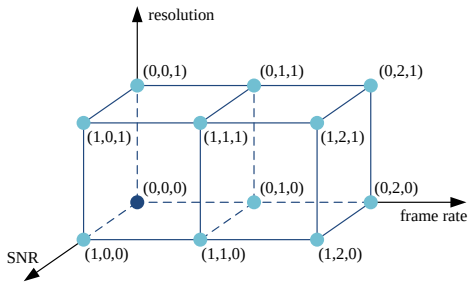


Fig. 1. An illustration of the 2-by-3-by-2 scalable media data structure

This paper is organized as follows. In Section II, we introduce the background and the related work. In Section III, we present the overview of the SMAC system. We then introduce the implementation details of SMAC from two aspects, i.e., how to enforce multiple access policies for the scalable media data, and how to authorize and revoke the access privileges of media content consumers, in Section IV and Section V, respectively. In Section VI, we evaluate the performance of the proposed system in terms of security and efficiency. Finally, we conclude our work in Section VII.

II. Background And Related Work

In this section, we introduce the scalable media format as the background, and review the state-of-the-art access control schemes for scalable media data.

A. Scalable Media Format

In scalable media format, a media stream is encoded into a base layer providing the basic quality and multiple enhancement layers enhancing the quality. The quality can be enhanced from multiple dimensions such as resolution, SNR, and frame rate [17]. Such kind of multi-dimensional scalability is a special characteristic of media content. As an illustration, we show the data structure of a 2-by-3-by-2 scalable media stream in Fig. 1. With the base layer denoted by (0, 0, 0), a media consumer can view the basic quality with the lowest SNR, frame rate, and resolution. By receiving two more enhancement layers denoted by (1, 0, 0) and (1, 0, 1), or (0, 0, 1) and (1, 0, 1), the consumer can enjoy higher SNR and resolution. Under such a data structure, the media consumption experience can be effectively controlled by adjusting the transmitted media layers upon sharing [22].

B. Access Control for Scalable Media Data

Based on the data structure of scalable media streams, a typical access control mechanism will encrypt each media layer with an individual access key, and issue the access keys to the authorized consumers according to their access privileges [23]. If the media stream is encoded into M layers and the number of consumers is N, then the amount of access keys that have to be distributed will be $O(MN)$.

To decrease the key distribution cost, some related works try to reduce the number of access keys sent to each data consumer. A popular scheme is to enable an access key computable from an arbitrary access key that corresponds to a higher-level access privileges. This can be achieved through

a one-way hash chain [23]. In this way, just one access key needs to be sent to each content consumer. Using Fig. 1 as an example, if a content consumer is supposed to access the base layer (0, 0, 0) and enhancement layers (1, 0, 0), (1, 0, 1), and (1, 1, 1), then only the access key k_{111} for layer (1, 1, 1) needs to be distributed to the consumer. The access keys for the other three layers can be computed as $k_{101} = H(k_{111})$, $k_{100} = H(H(k_{111}))$, and $k_{000} = H(H(H(k_{111})))$. Theoretically, such a strategy can reduce the key distribution cost from $O(MN)$ to $O(N)$. However, to prevent the user collusion attack in practice, where two users collude with each other to generate a valid but illegal access key corresponding to an unauthorized higher level access privilege, a secure scheme can only achieve a cost of $O(M'N)$ [24], [25], [26]. Specifically, M' is the ratio between M and the maximum levels of an encoding dimension for a scalable media stream, which is $\frac{2\times3\times2}{3} = 4$ in Fig. 1. Although this mechanism decreases the key distribution cost to some extent, it does not help in large-scale media sharing systems, where the number of potential content consumers is the efficiency bottleneck.

Another approach to improve system efficiency is to shift the key distribution process from the resource constraint user side to the resourceful server side [12], [27], [28]. Since the social networks servers may not be trusted by the media content owners, such an approach should keep the access keys confidential to the server. In addition, it needs to guarantee that only the content consumers with the desired access privileges can obtain the corresponding access keys. Only a few existing works have adopted this approach by leveraging attribute-based encryption [11], [13]. In [29], a multi-message ciphertext policy attribute-based encryption (MCP-ABE) scheme was proposed to encrypt multiple messages into one ciphertext according to the attribute-based access policy. A content owner can encrypt multiple access keys under MCP-ABE, store the ciphertext on the server, and delegate the server to distribute access keys to the content consumers. Whether or not a content consumer can decrypt the access keys, and how many access keys the consumer can decrypt, are determined by the individual's attributes [30]. Despite the security, this scheme only supports one-dimension scalability, and hence is not able to enforce access control policies on multi-dimensional scalable media streams that varies in resolution, frame rate, SNR, etc. Similarly, an algorithm that is able to encrypt two-dimensional (2D) scalable data based on the 2D scalable access policies was proposed in [31]. However, it cannot encrypt a general scalable media stream with more than two dimensions.

To enable access control on arbitrary dimensional scalable media streams, we propose a preliminary scheme in [32]. In particular, we proposed the initial version of the SCP-ABE algorithm, and on the basis designed an access control scheme that is able to enforce multiple access policies on multi-dimension scalable media stream. In this new research, by examining the characteristics of social networks more closely, we have substantially re-designed the access control scheme. First, we optimize the access structure construction method. As a result, the proposed method reduces the number of non-leaf nodes in the access tree compared to the previous method, making it more interpretable. Second, we solve the attribute authorization problem in social networks that was not addressed in [32]. Except for the traditional attribute revocation problem that has been well studied from the literature [33], we are faced with a new challenge, which is how to authorize user attributes depending on the untrusted authority, i.e. the social network server. In particular, we propose a comprehensive key management scheme that leverages the SCP-ABE access structure to tackle this challenge. Thanks to these new features, the SMAC system proposed in this paper is able to perform secure and reliable access control on multi-dimensional scalable social media streams according to content consumers' diverse attributes.

III. SMAC: System Overview

In this section, we present the overview of the proposed SMAC system. Specifically, we first introduce the trust model of the system, i.e., the trust relationship among different parties in the system. Then we clarify the assumptions of access policy for scalable data. Based on the models and assumptions, we describe the system architecture.

A. Trust Model

As shown in Fig. 2, there are four entities in the SMAC system, i.e., the media content distributor who owns the content, the media content consumer who consumes (e.g., views, downloads, and shares) the content, the social network server that stores the content, and the attribute authority (AA) that authorizes the access privileges of the content.

To begin with, the trust relationship between the distributor and the consumer may vary significantly from weak to strong. This is because users share various social relationship with each other in the large-scale media sharing system. To accommodate such diverse trust relationship with the consumers, the distributor configures multiple access privileges for the shared media content according to the consumer's social attributes.

In addition, the social network server is semi-trusted by the users. On one hand, the social network server is trusted to authenticate the social attributes of the consumers and assist the access control. On the other hand, the social network server is not trusted by the media content distributor to access the shared content. Hence, the media content is stored in the social network servers as ciphertext, and cannot be decrypted by the server by default. In practice, the content distributor can configure the access policy to enable an access privilege for the server, e.g., accessing the low-quality media content.

The AA is a trusted party for other parties in the system. Its functions include setting up the SCP-ABE parameters for the distributor, and managing attribute authorization.

To enforce the trust model, it should be guaranteed that the social network server and the unqualified consumer without the desirable attributes cannot obtain the shared content of the distributor. We will prove in Section 5 that the SMAC system is able to provide the desirable security and reliability.

B. Access Policy Assumption

For ease of presentation, we will use the 2-D scalable (M-by-N) data structure to illustrate our scheme starting from this section. However, it is important to note that our work is general for scalable data with arbitrary dimensions. Suppose that a

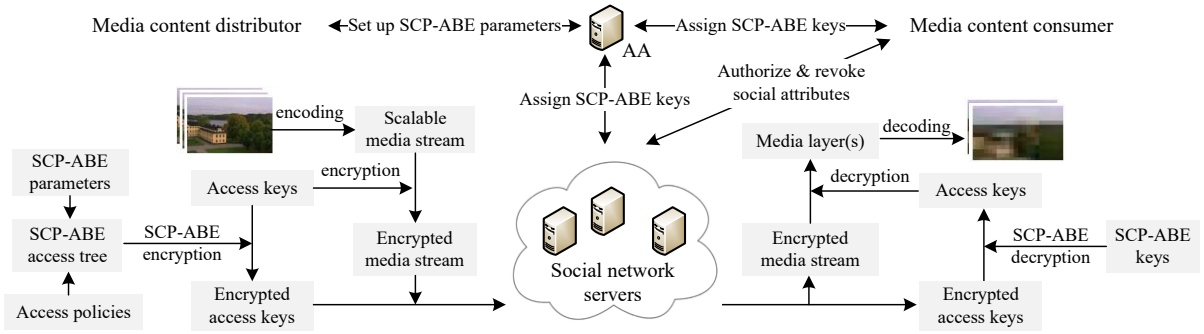


Fig. 2. Architecture of the SMAC system

data layer is denoted by m_{ij} $(i = 1, 2, ..., M; j = 1, 2, ..., N)$, where m_{11} is the base layer and others are the enhancement layers. We group these layers into $M + N - 1$ *access levels*. A layer m_{ij} with larger i and j would have a higher access level compared to a layer with smaller i and j. The access level of two layers m_{ij} and m_{pq} are the same if and only if $i + j = p + q$. Take the example shown in Fig. 3, there are four access levels, where m_{23} is at the highest access level, followed by m_{13} and m_{22}, m_{12} and m_{21}, and m_{11}. In the SMAC system, we assume that the access policy of a higher-access-level layer contains all the attributes in the access policy of a lower-access-level layer. Only in this way, it can be guaranteed that a user who is able to obtain the layer at a higher access level can also obtain the layer at a lower access level while the opposite does not hold.

Under this assumption, we have (1), where P_{ij} is the access policy to access m_{ij}. An example of P_{ij} and P_{pq} in the media sharing scenario could be "having mutual friends" and "having mutual friends and having shares of similar topics" or "being friends or having mutual friends" and "being friends or having mutual friends, and having shares of similar topics".

$$P_{ij} \subseteq P_{pq}, \text{ if } i \leq p, j \leq q \tag{1}$$

Furthermore, we can derive (2) from (1), indicating that the common attributes of the higher-level access policy always contain those of the lower-level access policy.

$$\begin{aligned} &P_{i_1 j_1} \cap P_{i_2 j_2} \subseteq P_{p_1 q_1} \cap P_{p_2 q_2}, \\ &\text{if } max(i_1 + j_1, i_2 + j_2) \leq min(p_1 + q_1, p_2 + q_2) \end{aligned} \tag{2}$$

According to (1), we have $P_{11} \subseteq P_{12} \subseteq P_{13} \subseteq P_{23}$, $P_{11} \subseteq P_{12} \subseteq P_{22} \subseteq P_{23}$, and $P_{11} \subseteq P_{21} \subseteq P_{22} \subseteq P_{23}$ for the 2-D scalable data structure shown in Fig. 3. Additionally, by representing P_{11} as P_{I_1}, $P_{12} \cap P_{21}$ as P_{I_2}, $P_{13} \cap P_{22}$ as P_{I_3}, and P_{23} as P_{I_4}, we have $I_1 \subseteq I_2 \subseteq I_3 \subseteq I_4$ according to (2).

Given the relationship of multiple access policies for scalable data, the SMAC system should not individually enforce each access policy since it causes lots of repeated operations. First, the process of applying the access policy of an enhancement layer m_{ij} repeats all the operations in applying the access policy $P_{f_{ij}}$ of its referees. Specifically, we define *referee* of a layer m_{ij} $(i = 1, 2, ..., M; j = 1, 2, ..., N)$ as the layer(s) in the path from m_{ij} to m_{11} that is (are) the nearest to m_{ij}. Take the example shown in Fig. 3, enforcing P_{12} and P_{21} repeats the operations in enforcing P_{11}, and enforcing P_{22} repeats the operations in enforcing P_{12} and P_{21}. Moreover, if (3) is satisfied, applying access policy of two layers at the same access level k $(k > 1)$ may introduce additional overlapped operations. For example, when enforcing P_{12} and P_{21}, their overlapped operations are from two aspects, i.e. P_{11} and $P_{I_2} \setminus P_{I_1}$.

$$P_{I_k} \setminus P_{I_{k-1}} \neq \emptyset \tag{3}$$

To enhance computation efficiency, multiple access policies are structured into a single SCP-ABE access tree in SMAC. The access policy enforcement process is able to exclude the repeated operations from the two sources mentioned above. We will discuss this in details in Section IV.

C. Architecture

The architecture of SMAC is shown in Fig. 2. The media content distributor encodes the social media content into a multi-dimensional scalable media stream composed of multiple media layers. It also applies access control on the content as follows.

- Set up the SCP-ABE parameters with the AA.
- Select an access policy for each media layer, and configure the access structure, i.e. a SCP-ABE access tree.
- Select an access key for each media layer, and encrypt the access keys using the SCP-ABE encryption algorithm.
- Encrypt each media layer with the corresponding access key under a standard encryption algorithm such as AES.

The distributor can then utilize the same set of access keys to encrypt multiple media streams/files for secure access control, as long as the access policy is not updated. In this way, the distributor just needs to perform the last step of access control upon sharing new media contents in the network.

The ciphertext of both the media stream and the access keys are stored in the social network servers, and are easily accessible by any social network users. However, the ciphertext can only be decrypted by those with the desirable attributes in the SCP-ABE access tree. In fact, the authentication and authorization of the attributes will need to be performed with the cooperation of the social network server and the AA.

A media content consumer who has the desirable attributes can decode the media content in the following steps.

- Obtain the SCP-ABE keys from the AA and the social network server.
- Obtain the ciphertext of access keys, and decrypt the access key(s) using the SCP-ABE decryption algorithm.

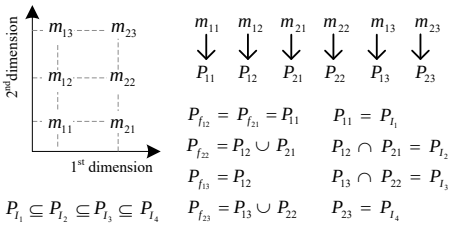


Fig. 3. 2-D scalable data structure and access policies

- Obtain the ciphertext of media contents, and decrypt the media stream with the access key(s).

The consumer only needs to perform the last step upon accessing the new content if the access keys selected by the media content distributor has not been changed.

IV. Access Policy Enforcement

In this section, we introduce the proposed SCP-ABE algorithm, and how it is utilized to enforce access policy in scalable media sharing.

A. The SCP-ABE Algorithm

We design the SCP-ABE algorithm based on the CP-ABE algorithm [13]. In CP-ABE, data is encrypted under an access policy that is composed of attributes. A user could decrypt the cipher-text only if his or her attributes satisfy the access policy. Using a similar mechanism, the SCP-ABE algorithm focuses on efficiently encrypting multi-dimensional scalable data. In particular, SCP-ABE is composed of six sub-algorithms including system setup, access tree construction, encryption, SCP-ABE key generation, delegation, and decryption. We first introduce bilinear map [34] as the preliminary, and then proceed to describe each step of the SCP-ABE algorithm.

1) Preliminary: Similar as CP-ABE, the construction of SCP-ABE algorithm is based on bilinear map. Let G_0 and G_1 be two multiplicative cyclic groups of prime order p. Let g be a generator of G_0 and e be a bilinear map, $e : G_0 \times G_0 \rightarrow G_1$. Then e has the following properties:

- *Bilinearity*: for all $u, v \in G_0$ and $a, b \in Z_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
- *Non-degeneracy*: $e(g, g) \neq 1$.

2) Setup: The setup algorithm chooses a bilinear group G_0 of prime order p with generator g, and two random exponents $\alpha, \beta \in Z_p$. The public key PK and the master key MK are then returned as: $\{PK = G_0, g, h = g^\beta, f = g^{1/\beta}, e(g, g)^\alpha\}$, $\{MK = \beta, g^\alpha\}$.

3) Access Tree Construction: Given a set of attribute-based access policies for the multi-dimensional scalable data, an access tree $\mathcal{T}$ is constructed in this step. Specifically, $\mathcal{T}$ contains leaf nodes and non-leaf nodes. A leaf node represents an attribute, which can either be included or excluded by the access policy. In particular, there are two types of attributes, i.e. *level attributes* and *layer attributes*, that are unrelated to the access policy. For a $M \times N$ data structure, there are $M \times N - 1$ layer attributes and $M + N - 2$ level attributes in $\mathcal{T}$. The functions of the level and layer attributes will be discussed later. A non-leaf node represents a threshold gate, which is described by a threshold value and its children. Specifically, the access tree is constructed from bottom to up as follows.

- Construct the subtree T_{c_k} $(k = 2, ..., M + N - 2)$ according to the access policy $P_{I_k} \setminus P_{I_{k-1}}$, where P_{I_k} is the common access policy for the data layers at access level k. The root of T_{c_k} is denoted as C_k.
- Construct the subtree T_{ij} for each layer m_{ij}. Suppose that k is the access level of m_{ij}, and $P_{f_{ij}}$ is the union of access policies of the referee(s) of m_{ij}. T_{11} is constructed according to P_{11}, and T_{MN} is constructed according to $P_{f_{MN}}$. For other layers, T_{ij} is constructed according to $P_{ij} \setminus (P_{I_k} \cup P_{f_{ij}})$. The root R_{ij} of T_{ij} is a *key node*.
- Construct the subtree T'_{ij}, where the root V_{ij} is an and gate with the children nodes of R_{ij} $(i = 2, ..., M | j = 1; j = 2, ..., N | i = 1)$ and the layer attribute a_{ij}.
- Construct subtree T_k $(k = 2, ..., M + N - 2)$, where the root R_k is an or gate with the children nodes of V_{ij} at the access level k. Let T_1 be T_{11}, and T_{M+N-1} be T'_{MN}.
- Construct T'_k starting from the access level $k = M + N - 2$. Let T'_k $(k = M + N - 1)$ be T'_{MN}. Specifically, we first let an or gate be the mother of the root of T'_{k+1} and a level attribute a_k. Then the root of T'_k is an and gate with the children of or, T_k, and C_k. Repeat this process by decreasing k by one at each step until $k = 1$.

The final access structure is T'_1. The root of $\mathcal{T}$ will be an and gate. The access structure strictly conforms to the access policy. Then we choose a polynomial p_x for each tree node x in $\mathcal{T}$ using a top-down manner: (1) Set the degree d_x of the polynomial p_x to be $d_x = k_x - 1$, where k_x is the threshold value of node x; (2) Choose a random $s \in Z_p$ for the root node $root$. Set $p_{root}(0) = s$, and randomly choose other points of polynomial p_{root}; (3) For any other node x, set $p_x(0) = p_{parent(x)}(index(x))$, and randomly choose other points of p_x. Access tree construction for Fig. 3 is illustrated in Fig. 4.

In addition to the attributes in the access policy, additional attributes are included in the access structure. Specifically, level attribute a_k $(k = 1, 2, ..., M + N - 2)$ is used to enable users to perform decryption from access level k, instead of from the bottom of the tree that corresponds with the highest access level. For example, if the attributes of a user only satisfy P_{11}, the user performs decryption starting from access level one, i.e., where a_1 is located in the tree. If the user has attributes that satisfy P_{12}, he or she can instead perform decryption starting from access level two. Additionally, layer attribute $a_{ij} (i = 2, ..., M | j = 1; j = 2, ..., N | i = 1)$ is employed to guarantee the uniqueness of each key node and the security of access policy enforcement. For example, without a_{12} and a_{21}, $p_{R_{12}}(0)$ will be equal to $p_{R_{21}}(0)$ since their mother node has the degree of zero and $index(R_{12})$ is equal to $index(R_{21})$.

The SCP-ABE access tree has the following features:

- It can be utilized to securely encrypt multiple data layers in a single access structure.
- It enables users to start decryption from any access level according to their access privileges.

4) Encryption: The encryption algorithm encrypts the data layers under $\mathcal{T}$ using the public key PK. Let L be the set

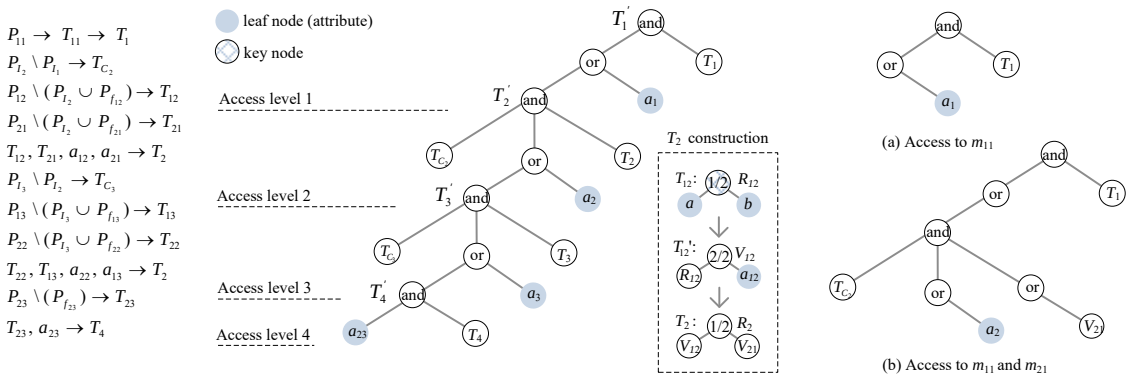


Fig. 4. The SCP-ABE access structure for the 2-D scalable data in Fig. 3

of leaf nodes in $\mathcal{T}$, K be the set of key nodes V_{ij} $(i = 1, 2, ..., M; j = 1, 2, ..., N)$, and m_{ij} be the corresponding data layer, the ciphertext is given as

$$CT = (\mathcal{T}, \forall i \in L : E_i = g^{p_i(0)}, E_i' = H(att(i))^{p_i(0)}$$
$$\forall R_{ij} \in K : \tilde{C_{ij}} = m_{ij} e(g,g)^{\alpha(p_{R_{ij}}(0)+s)}, C_{ij} = h^{p_{R_{ij}}(0)+s})$$

5) SCP-ABE Key Generation: Taking a set of attributes S and MK as input, the user key generation algorithm outputs a SCP-ABE key. The algorithm selects a random $r \in Z_p$ and random $r_x \in Z_p$ for every attribute x in S. Note that S always includes a_i $(i = 1, 2, ..., M+N-2)$, and a_{ij} $(i = 2, ..., M | j = 1; j = 2, ..., N | i = 1)$. The SCP-ABE secret key is

$$SK = \{D = g^{(\alpha+r)/\beta},$$
$$\forall x \in S : D_x = g^r \cdot H(attri_x)^{r_x}, D_x' = g^{r_x}\}$$

6) Delegation: Given a SCP-ABE key SK with the attribute set S, the delegation algorithm creates a SCP-ABE key $\tilde{SK}$ with the attribute set $\tilde{S} \subseteq S$. Specifically, the algorithm selects a random number $\tilde{r} \in Z_p$ and also $\tilde{r_x} \in Z_p$, $\forall x \in \tilde{S}$. Then $\tilde{SK}$ is created as

$$\tilde{SK} = \{\tilde{D} = D f^{\tilde{r}},$$
$$\forall x \in \tilde{S} : \tilde{D_x} = D_x g^{\tilde{r}} H(attri_x)^{\tilde{r_x}}, \tilde{D_x}' = D_x' g^{\tilde{r_x}}\}$$

7) Decryption: The decryption algorithm employs three inputs, i.e., the encrypted data layers, a SCP-ABE key SK, and the public key PK. The number of data layers that a user can decrypt depends on the extent to which the attributes can satisfy the access policies. Suppose that the access level of a user is k. Starting from a_k (or the bottom of $\mathcal{T}$ for the highest access level) to the root, the user needs to perform computations for each leaf node x in the path as in (4).

$$\begin{aligned} F_x &= \frac{e(D_x, E_x)}{e(D_x', E_x')} \\ &= \frac{e(g^r \cdot H(attri_x)^{r_x}, g^{p_x(0)})}{e(g^{r_x}, H(attri_x)^{p_x(0)})} \\ &= \frac{e(g^r, g^{p_x(0)}) \cdot e(H(attri_x)^{r_x}, g^{p_x(0)})}{e(g^{r_x}, H(attri_x)^{p_x(0)})} \\ &= \frac{e(g,g)^{r p_x(0)} \cdot e(g^{r_x}, H(attri_x)^{p_x(0)})}{e(g^{r_x}, H(attri_x)^{p_x(0)})} \\ &= e(g,g)^{r p_x(0)} \end{aligned} \quad (4)$$

Then the user computes F_x for each non-leaf node in a bottom-up manner using polynomial interpolation technique [13]. In this process, the user obtains $F_{R_{ij}}$ for each key node as in (5).

$$F_{R_{ij}} = e(g,g)^{r p_{R_{ij}}(0)} (i = 1, 2, ..., M, j = 1, 2, ..., N) \quad (5)$$

The computation is repeated until the root is reached and the user obtains F_{root} as in (6).

$$F_{root} = e(g,g)^{r p_{root(0)}} = e(g,g)^{rs} \quad (6)$$

Suppose that user Alice has attributes satisfying P_{11}, and user Bob has attributes satisfying P_{12}. Then Alice can compute $F_{R_{11}}$, and utilize F_{a_1}, $F_{R_{11}}$ to compute F_{root}. Bob can compute $F_{R_{12}}$, F_{C_2}, and $F_{R_{11}}$. He can utilize $F_{a_{21}}$ and $F_{R_{21}}$ to compute $F_{V_{21}}$, and then utilize F_{a_2}, $F_{V_{21}}$, and $F_{R_{11}}$ to compute F_{root}. The access behavior of Alice and Bob is illustrated in Fig. 4. F_{root} is not computable unless F_x of all nodes x in the path to the root of $\mathcal{T}$ are computed.

Furthermore, the user computes $K_{ij} = F_{R_{ij}} \cdot F_{root} = e(g,g)^{r(p_{R_{ij}}(0)+s)}$. Each data layer m_{ij} $(i = 1, 2, ..., M, j = 1, 2, ..., N)$ can then be decrypted as in (7).

$$\begin{aligned} &\frac{\tilde{C_{ij}}}{e(C_{ij}, D)/K_{ij}} \\ &= \frac{m_{ij} e(g,g)^{\alpha(p_{R_{ij}}(0)+s)}}{e(h^{p_{R_{ij}}(0)+s}, g^{(\alpha+r)/\beta}) / e(g,g)^{r(p_{R_{ij}}(0)+s)}} \\ &= \frac{m_{ij} e(g,g)^{(\alpha+r)(p_{R_{ij}}(0)+s)}}{e(g^{\beta(p_{R_{ij}}(0)+s)}, g^{(\alpha+r)/\beta})} \\ &= \frac{m_{ij} e(g,g)^{(\alpha+r)(p_{R_{ij}}(0)+s)}}{e(g,g)^{\beta(p_{R_{ij}}(0)+s)\cdot(\alpha+r)/\beta}} \\ &= m_{ij} \end{aligned} \quad (7)$$

B. Access Control on The Shared Media Content

As introduced in Section III, the media content distributor performs the four-step access control process on a scalable media stream based on the SCP-ABE algorithm. We now introduce the implementation details of this process.

The access control starts with running the SCP-ABE setup algorithm by the distributor. The generated public key PK is

accessible by all other parties in the system. In practice, PK can be stored in either the social network server or the AA so that all consumers can access it. The master key MK is shared with and only with the AA.

Second, the distributor configures the access policy by selecting the desirable social attributes, which are authorized by the social network servers. In addition, the distributor selects the level attributes and the layer attributes that are excluded by the access policy. These attributes can be simply set as time stamps or level/layer indexes, which needs no authorization by a third party. After that, the distributor runs the SCP-ABE access tree construction algorithm to build the access structure.

The last two steps are encryptions. Specifically, the distributor applies a symmetric encryption algorithm such as AES to encrypt the media layers, and then utilizes SCP-ABE to encrypt the symmetric keys, i.e., the access keys, for these layers. This is a standard operation of combining symmetric encryption and asymmetric encryption which SCP-ABE algorithm belongs to, because the former one is more efficient to encrypt large volume of media stream [42].

If the distributor wants to update the access policies, the individual needs to reconstruct the SCP-ABE access tree, and to reselect the access keys to ensure security.

V. Access Privilege Authorization

We have introduced in Section III that a media content consumer can perform the three-step process to access the media content with a specific quality as long as the individual has the corresponding attributes. In this section, we introduce how to authenticate the attributes of the consumers and authorize/revoke their access privileges accordingly. This is achieved by utilizing the proposed key management schemes, i.e, the SCP-ABE key distribution scheme, and the SCP-ABE key revocation scheme.

A. SCP-ABE Key Distribution

The access privilege of a media content consumer is enabled by distributing the individual with the SCP-ABE key based on the authenticated attributes. The SCP-ABE key for a consumer is divided into two parts. One is related to the social attributes while the other is related to the layer and level attributes. Both of them are distributed separately by the social network server and the AA. On the premise that the social attributes of consumers are authenticated by the social network server, the SCP-ABE key distribution process in the SMAC system is proceeded as follows.

- The AA generates SK_s as in (8) based on set S_s of all social attributes in the access policy, and SK_n as in (9) based on set S_n of all layer and level attributes in the access structure.
- The AA assigns SK_s to the social network server. No attribute authentication is required for the server, since it manages all social attributes of all users in the network.
- The server authenticates the social attributes of the consumer, runs the delegation algorithm and generates SK_{s_u} for the consumer as in (10), where S_{s_u} is the set of consumer social attributes, and $S_{s_u} \subseteq S_s$. The access behavior of the consumer is then confirmed accordingly, and is shared with the AA.
- The AA selects the set S_{n_u} of consumer's layer and level attributes such that $S_{n_u} \subseteq S_n$. For example, S_{n_u} is equal to $\{a_{12}, a_{21}, a_2\}$ if the consumer is able to access m_{11}, m_{12}, and m_{21}. On the other hand, S_{n_u} is equal to $\{a_{12}, a_{21}, a_{22}, a_3\}$ if the consumer is able to additionally access m_{22}. The AA then runs the delegation algorithm using SK_n as input, and generates $SK_{n_{u,1}}$ as in (11), and sends it to the server.
- The server runs the delegation algorithm using $SK_{n_{u,1}}$ as input, and generates $SK_{n_{u,2}}$ as in (12), and sends it back to the AA.
- The AA generates SK_{n_u} as in (13), where $\tilde{D}_n(i)$ is obtained by dividing $\tilde{D}_{n,2}(i)$ in $SK_{n_{u,2}}$ with $g^{\tilde{r}_1} H(attri_i)^{\tilde{r}_{1,i}}$, and $\tilde{D}_n(i)'$ is equal to $\tilde{D}'_{n,2}(i)/g^{\tilde{r}_1}$. By combining SK_{s_u} and SK_{n_u}, the SCP-ABE key SK_u of the consumer can be derived in (14).

$$\begin{aligned} SK_s = \{ & D = g^{(\alpha+r)/\beta}, \\ & \forall i \in S_s : \\ & D_s(i) = g^r H(attri_i)^{r_i}, D'_s(i) = g^{r_i} \} \end{aligned} \tag{8}$$

$$\begin{aligned} SK_n = \{ & \forall i \in S_n : \\ & D_n(i) = g^r H(attri_i)^{r_i}, D'_n(i) = g^{r_i} \} \end{aligned} \tag{9}$$

$$\begin{aligned} SK_{s_u} = \{ & \tilde{D} = D f^{\tilde{r}}, \\ & \forall i \in S_{s_u} : \\ & \tilde{D}_s(i) = D_s(i) g^{\tilde{r}} H(attri_i)^{\tilde{r}_i}, \\ & \tilde{D}'_s(i) = D'_s(i) g^{\tilde{r}_i} \} \end{aligned} \tag{10}$$

$$\begin{aligned} SK_{n_{u,1}} = \{ & \forall i \in S_{n_u} : \\ & \tilde{D}_{n,1}(i) = D_n(i) g^{\tilde{r}_1} H(attri_i)^{\tilde{r}_{1,i}}, \\ & \tilde{D}'_{n,1}(i) = D'_n(i) g^{\tilde{r}_{1,i}} \} \end{aligned} \tag{11}$$

$$\begin{aligned} SK_{n_{u,2}} = \{ & \forall i \in S_{n_u} : \\ & \tilde{D}_{n,2}(i) = \tilde{D}_{n,1}(i) g^{\tilde{r}} H(attri_i)^{\tilde{r}_i}, \\ & \tilde{D}'_{n,2}(i) = \tilde{D}'_{n,1}(i) g^{\tilde{r}_i} \} \end{aligned} \tag{12}$$

$$\begin{aligned} SK_{n_u} = \{ & \forall i \in S_{n_u} : \\ & \tilde{D}_n(i) = D_n(i) g^{\tilde{r}} H(attri_i)^{\tilde{r}_i}, \\ & \tilde{D}'_n(i) = D'_n(i) g^{\tilde{r}_i} \} \end{aligned} \tag{13}$$

$$\begin{aligned} SK_u = \{ & D_u = g^{(\alpha+r+\tilde{r})/\beta}, \\ & \forall i \in S_u \ (S_{s_u} \cup S_{n_u}) : \\ & D_u(i) = g^{r+\tilde{r}} H(attri_i)^{r_i+\tilde{r}_i}, \\ & D'_u(i) = g^{r_i+\tilde{r}_i} \} \end{aligned} \tag{14}$$

It is worth to mention that the SCP-ABE key distribution process needs not to disclose the social attributes of social

network users to the third party AA. One way to keep social attributes confidential is to only share the hash values of the social attributes of a consumer (by the social network server) and those in the access structure (by the content distributor) with the AA. However, such a method enables the AA to apply brute-force attack to guess the social attributes of a consumer. For example, in order to refer the friendship of the content distributor and the content consumer, the AA can compare the hash value of "friend of the content distributor" with the hash values of the social attributes shared by the social network server. The success rate of brute-force attack can be rather high when the social attribute set has a small size. To reduce the feasibility of Brute-force attack, we can let the social network server and the social network users share one or more secret values. The content distributor can then concatenate the secret value s_a with a social attribute in the access structure $\mathcal{T}$, and replace $H(attri_x)$ $(x \in S)$ with $H(attri_x||s_a)$ $(x \in S)$.

B. SCP-ABE Key Revocation

The SCP-ABE key revocation is utilized to disable the access privileges. It takes effects when a consumer's social attributes have changed, e.g. the friendship no longer exists, or the access structure is updated by the distributor.

In particular, in the first case, the revocation is initiated by the social network server, who manages the social attributes of users in the network, and hence can be aware of their changes at the earliest stage. Upon receiving the revocation initiation notification, the distributor needs to change the access keys and re-encrypt them to ensure secure access control. The revocation is only required for the consumer whose attributes have changed, i.e. SK_s and SK_n will remain the same, and the specific SCP-ABE key SK_u of the consumer will be revoked.

To perform the revocation, the social network server needs to re-authenticate the social attributes of the consumer and redefine his or her access level. Then the server re-generates SK_{s_u} by selecting new $\tilde{r}'$ and $\tilde{r_i}'$ $\forall i \in S'_{s_u}$. The updated SK_{s_u} and the updated consumer access level information are shared with the AA. The AA re-generates $SK_{n_{u,1}}$ by re-selecting the attribute set S'_{n_u}, $\tilde{r_1}'$, and $\tilde{r}'_{1,i}$, and sends $SK_{n_{u,1}}$ to the server. Upon receiving $SK_{n_{u,1}}$, the server re-generates $SK_{n_{u,2}}$ using $\tilde{r}'$ and $\tilde{r_i}'$ $\forall i \in S'_{s_n}$, and sends it back to the AA. The consumer is then assigned with the new SCP-ABE key SK_u, which is used to access the newly shared content.

In the second case, the revocation is initiated by the distributor. The SCP-ABE keys of all consumers in the network are revoked and re-distributed. Specifically, the AA re-selects r, and r_i for each member i in the new attribute sets $S'_s \cup S'_n$ and re-generate SK_{s_u} and SK_{n_u}. The updated SK_{s_u} is re-distributed to the social network server. Then the social network server and the AA perform the same SCP-ABE key revocation process as in the first case.

VI. Performance Evaluation

In this section, we evaluate the performance of the SMAC system from two essential aspects, i.e., security and efficiency. The security of the system ensures that a shared media content is always propagated along the trusted chain in the social networks. The system efficiency is necessary for practical implementations on the resource constrained user equipments.

A. Security Analysis

We use two metrics to prove the system security of media content confidentiality [43], i.e. SCP-ABE algorithm security and the reliability of access privilege authorization. The first metric guarantees that access key decryption succeeds if and only if the correct SCP-ABE key is used. The other metric ensures that only the users with the specific access privilege can obtain the corresponding SCP-ABE key.

1) SCP-ABE Security: The proposed SCP-ABE algorithm differentiates itself from the existing CP-ABE algorithm by enabling multiple access behaviors under a single access structure, i.e., users can decrypt different cipher-texts by starting computation from different levels of the access tree. However, this feature introduced by SCP-ABE does not corrupt the security of CP-ABE. We consider the chosen plaintext attack as in [13]. Specifically, the challenge ciphertext $\tilde{C_{ij}}$ $(i = 1, ..., M; j = 1, ..., N)$ is either $m_{ij}e(g,g)^{\alpha(p_{R_{ij}(0)}+s)}$ or $m'_{ij}e(g,g)^{\alpha(p_{R_{ij}(0)}+s)}$. The attacker needs to differentiate whether the ciphertext is for m_{ij} or m'_{ij}. Since both $p_{R_{ij}(0)}$ and s belong to Z_p, we have $p_{R_{ij}(0)} + s \in Z_p$. By treating $p_{R_{ij}(0)} + s$ as $s' \in Z_p$, the challenge turns into differentiating $m_{ij}e(g,g)^{\alpha s'}$ and $m'_{ij}e(g,g)^{\alpha s'}$, which is exactly the challenge faced in CP-ABE. Since the proposed SCP-ABE algorithm uses the same parameter setup, key generation, encryption and decryption mechanisms, and utilizes the mathematical properties of elliptic curve groups [36] and cryptographic hash functions[37] as in the CP-ABE algorithm, it actually inherits the security of the CP-ABE. The security is provable under the generic group heuristic. More details of the security proof can be referred as in [13].

2) Access Privilege Authorization Reliability: The proposed key management schemes enable reliable access privilege authorization for both the media content consumers and the social network server.

First, a media content consumer cannot obtain the SCP-ABE key that requires the attributes out of his or her attribute set.

On one hand, the consumer access privilege authorization is reliable in resisting collusions, where two consumers collude with each other to obtain the access privilege that they are not supposed to have. Suppose that a consumer Bob owns the attributes satisfying the access policy P_{11}, and another consumer Clark owns the attributes satisfying $P_{21} \setminus P_{11}$. The system security guarantees that Bob can obtain m_{11} through SCP-ABE decryption, while Clark should not obtain any access keys due to the lack of attributes satisfying P_{11}. The goal of their collusion is to obtain m_{21}. According to (7), the ciphertext of m_{ij} can be decrypted if Bob and Clark obtain the intermediate computation result K_{21}. In particular, K_{21} is computable from $F_{R_{21}}$ and F_{root} using the correct SCP-ABE key. However, it is infeasible for them to obtain the correct SCP-ABE key by exchanging their SCP-ABE keys. This is because the social network server selects $\tilde{r}$ randomly when distributing SCP-ABE keys to the consumers,

TABLE I
COMPUTATION TIME OF SMAC OPERATIONS (PER ATTRIBUTE)

Operation	Entity	Platform	Time (ms)	Attri. type
SCP-ABE key gener.	AA	server	10	all
SCP-ABE encryption	distr.	phone	60	all
SCP-ABE encryption	distr.	laptop	35	all
SCP-ABE decryption	consu.	phone	5	all
SCP-ABE decryption	consu.	laptop	2.5	all
SCP-ABE key distr.	AA	server	$10 \times N_u$	non-social
SCP-ABE key distr.	SNS	server	$10 \times N_u$	all
SCP-ABE key revoc.	AA	server	10	social
SCP-ABE key revoc.	AA	server	$10 + 10N_u$	non-social
SCP-ABE key revoc.	SNS	server	$10 \times N_u$	all

which prevents the combination of two SCP-ABE keys from two different consumers. Furthermore, it does not work if Bob and Clark exchange their intermediate computational results. Suppose that r_B and r_C are two random parameters selected by the server for Bob and Clark, respectively. After separately running the SCP-ABE decryption algorithm, Bob has $F_{root} = e(g,g)^{(r+\tilde{r}_B)s}$ and $F_{R_{11}} = e(g,g)^{(r+\tilde{r}_B)p_{R_{11}}(0)}$, and Clark can obtain $F_{R_{21}} = e(g,g)^{(r+\tilde{r}_C)p_{R_{21}}(0)}$. Since r_B and r_C are different, the F_{root} computed by Bob and the $F_{R_{21}}$ computed by Clark cannot be used together to compute K_{21}.

On the other hand, the SCP-ABE key revocation scheme is able to accord with the dynamics of user attribute changes in the network by re-authenticate user attributes and re-generating SCP-ABE keys. The old SCP-ABE key and the new SCP-ABE key cannot be combined since $\tilde{r}$ in two keys are different. This prevents consumers from accessing the media contents utilizing expired attributes.

Second, we prevent the social network server from accessing the media contents by avoiding authorization of level attributes. This utilizes the property of the SCP-ABE access structure that the decryption from any access level requires both the social attributes and the level attributes. In the access structure shown in Fig. 4, for example, the level attribute a_{11} is mandatory in obtaining F_{root}. Without F_{root}, the server cannot compute any K_{ij} even though it can compute $F_{R_{ij}}$ for the access key denoted by (i,j). Combing $F_{R_{ij}}$ with the F_{root} computed by others is infeasible as aforementioned explanation. Additionally, the server cannot derive a valid SCP-ABE key in the process of distributing SCP-ABE keys to the consumers as well. This is because the AA protects SK_n by running the delegation algorithm and sending the output $SK_{n_{u,1}}$ to the server. To obtain SK_n, the server needs to derive $g^r H(attri_i)^{r_i}$ from $g^{r+r^1} H(attri_i)^{r_i+r_i^1}$, and g^{r_i} from $g^{r_i+r_i^1}$ $(i \in S_n)$. However, it is computationally infeasible without knowing $\tilde{r}^1$ and $\tilde{r}_i^1$, according to the computational Diffie-Hellman assumption [35].

B. Efficiency Analysis

We evaluate the system efficiency performance by measuring its cost of access policy enforcement and access privilege authorization for different parties in the system. We also compare it with the naive CP-ABE based system. In particular, we implement the user side operations on two mobile platforms, i.e., Google Nexus 4 with 1.5 GHz quad-core CPUs, and Lenovo T430 with 2.6GHz Intel i5 CPU. Operations on the AA and social network server (SNS) side are implemented on Dell PowerEdge 610 server with 2.4 GHz 16-core CPUs.

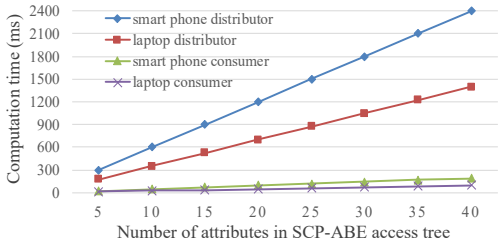


Fig. 5. User side computation cost

1) User Side Cost: The computation cost of distributor side operation mainly comes from two aspects, i.e. symmetric encryption of the media content and SCP-ABE encryption of the access keys. In particular, since media encryption is a routine process of encryption-based access control on media streams, we do not include it in our system performance evaluation. Its computational cost benchmark can be referred from [38]. Instead, we focus on the evaluation of SCP-ABE encryption, which involves pairings, exponentiations, and multiplications computations on G_0 and G_1. We measure the average computation time per attribute on the smartphone platform and the laptop platform[1], and the average results are 60ms and 35ms, respectively. The computation cost linearly increases with the number of attributes. In terms of consumer side computation, it mainly includes pairings and multiplications computations on G_1 resulted from SCP-ABE decryption. The computation cost of SCP-ABE decryption increases approximate-linearly with the number of attributes, but also varies with different access policies. For simplification, we consider that the computation cost grows with the number of attributes in a linear way, and test the average computation time per ten attributes on devices. The results are 48ms on smart phone and 25ms on the laptop.

We then present the experimental results of the computation time on the user side. Fig. 5 shows the computation time on the user side with various number of attributes in the access policies. We can see that the computation performance of the MD-SMAC system for consumers is less than one second even if the number of attributes is large. This satisfies the requirement of general mobile applications [39]. Although the computation cost on the distributor side can reach a few seconds when the number of attributes is relatively large, it is still negligible compared to the cost of media stream encryption. According to the benchmark provide in [38], encryption cost for 25 fps video is about ten times of the video length, which is much higher than encrypting access keys using SCP-ABE encryption. This also indicates that scalable access control can be achieved in the SMAC system with a cheap cost. Therefore, we conclude that the SMAC system can be efficiently implemented in real-world systems.

2) Server Side Cost: The AA side computation cost comes from SCP-ABE key generation, SCP-ABE key distribution, and revocation. All of the three algorithms involve exponen-

[1]As in [13], operations are conducted using a 160-bit elliptic curve group based on the curve $y^2 = x^3 + x$ over 512-bit finite field.

tiations and multiplications on G_0 and G_1. We found in our results that the average computation time of SCP-ABE key generation operation is about 10ms per attribute on the server. The computation time of SCP-ABE key distribution is almost the same, but it is proportional to the number of consumers N_u in the system. If the set of social attributes has M_s elements, and the set of level/layer attributes has M_n elements, then the cost on the AA side is about $(10\times(M_s+M_n)+10\times O(M_n)\times N_u)$ ms for SCP-ABE key generation and distribution for N_u consumers. The SCP-ABE key revocation operation requires re-generation and re-distribution. Therefore, the cost of SCP-ABE key revocation is the sum of SCP-ABE key generation and distribution. On the SNS side, computation cost is from SCP-ABE key distribution and revocation. Specifically, the key distribution cost is $10 \times O(M_n + M_s) \times N_u$, and the key revocation cost is the same. Therefore, in the SMAC system, the privacy preservation cost on the server side linearly increases with the number of media content consumers, which is an acceptable result. The computation cost of operations on the server side and the user side is summarized in Table I. As the power of computation equipment increases while its cost decreases, we believe that the extra key management costs on the social network server side will be acceptable.

3) Comparison with CP-ABE Based System: We compare the cost of utilizing the SCP-ABE algorithm and utilizing the CP-ABE algorithm [13] in the SMAC system. As we have analyzed in Section III, a separate enforcement of multiple access policies for scalable data, e.g., constructing a CP-ABE access tree for each access policy, will cause lots of repeated operations. By structuring all access policies in a single access tree, SCP-ABE can avoid this issue. Although some additional attributes are introduced during access tree construction, the resulted redundancy is negligible, especially when the number of levels in the access structure is large.

Specifically, we assume the media stream can be scalable from 1 to 3 dimensions. In addition, the access policy corresponding to a layer has at least k $(k = 1, 2)$ more attributes than that corresponding to a lower-level layer. By varying the total number of media layers in the stream, we show the number of attributes to be computed by SCP-ABE and CP-ABE for scalable access control. Note that different number of attributes may be needed for a given number of media layers under different scalability dimensions. For example, seven level attributes are introduced in the access structure for the 2×6 scalable media stream, while only six level attributes are introduced in the access structure for the 3×4 scalable media stream or $2 \times 3 \times 2$ scalable media stream. We therefore measure the average number of attributes (over 1-D to 3-D scalability) as well as the upper bound and the lower bound. The results are presented in Fig. 6. We can see that the computation performance of SCP-ABE is superior to CP-ABE, and the advantage becomes more significant as the number of media layers increases.

VII. Conclusion

In this paper, we have presented SMAC, the first access control scheme that protects user privacy in large-scale media sharing systems and satisfies two essential user requirements,

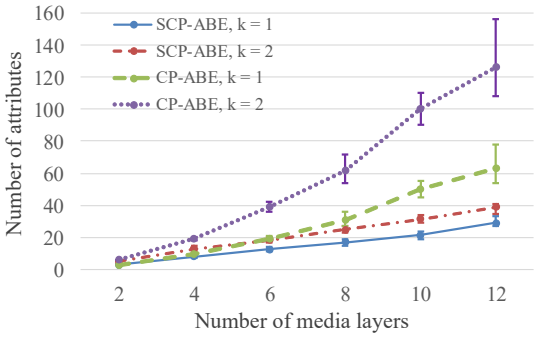


Fig. 6. Comparison between SCP-ABE and CP-ABE

i.e., widespread content propagation and multiple-level access privileges. In particular, we first propose a SCP-ABE algorithm to enable secure enforcement of multiple access policies on the multi-dimensional scalable media streams. In addition, we propose a comprehensive key management scheme to facilitate the reliable and efficient access privilege authorization and revocation. We have proved the security and reliability of the SMAC system. We also demonstrated its efficiency on mobile devices through experiments. We believe these features of the SMAC system will contribute to the wide adoption of privacy preservation in large-scale social networks. For the future work, we will extend the SMAC system to support media sharing across multiple social networks to accord with the trending cloud-based social services [40], [41].

VIII. Acknowledgement

This work is supported by NSF Grant ECCS-1405594.

References

[1] C. Zhang, J. Sun, X. Zhu and Y. Fang, "Privacy and security for online social networks: challenges and opportunities," *IEEE Network*, vol. 24, no. 4, pp. 13-18, 2010.

[2] M. Fire, R. Goldschmidt and Y. Elovici, "Online Social Networks: Threats and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2019-2036, 2014.

[3] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, A. V. Vasilakos, "Security and privacy for storage and computation in cloud computing," *Information Sciences: an International Journal*, 258, p.371-386, 2014.

[4] R. Shokri, V. Shmatikov, "Privacy-Preserving Deep Learning," *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1310-1321, 2015.

[5] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," *CCS*, pp. 308-318, 2016.

[6] L. Yuan, P. Korshunov, T. Ebrahimi, "Privacy-preserving photo sharing based on a secure JPEG," *IEEE Conf. Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 185-190, 2015.

[7] F. Dufaux and T. Ebrahimi, "Scrambling for Privacy Protection in Video Surveillance Systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1168-1174, 2008.

[8] L. Yuan and T. Ebrahimi, "Image Privacy protection with secure JPEG transmorphing," *IET Signal Processing*, vol. 11, no. 9, pp. 1031-1038, 2017.

[9] Z. Yan, X. Li, M. Wang and A. V. Vasilakos, "Flexible Data Access Control Based on Trust and Reputation in Cloud Computing," *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 485-498, 2017.

[10] M. Ali et al., "SeDaSC: Secure Data Sharing in Clouds," *IEEE Systems Journal*, vol. 11, no. 2, pp. 395-404, 2017.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," *ACM CCS*, pp. 89-98, 2006.

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

[13] Bethencourt, J.; Sahai, A.; Waters, B., "Ciphertext-Policy Attribute-Based Encryption," *IEEE Symposium on Security and Privacy*, pp. 321-334, 2007.

[14] S. B. Barnes, "A privacy paradox: Social networking in the United State," http://www.firstmonday.org/ISSUES/issue11_9/barnes/.

[15] A. Samuel, M. I. Sarfraz, H. Haseeb, S. Basalamah and A. Ghafoor, "A Framework for Composition and Enforcement of Privacy-Aware and Context-Driven Authorization Mechanism for Multimedia Big Data," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1484-1494, 2015.

[16] M. W. Marcellin, M. J. Gormish, A. Bilgin and M. P. Boliek, "An overview of JPEG-2000," *Data Compression Conf.*, pp. 523-541, 2000.

[17] H. Schwarz, D. Marpe and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Trans. Circuits and Syst. Video Technol.*, vol.17, no.9, pp.1103-1120, 2007.

[18] C. K. Mick, R. R. Shea, K. P. Grundy, J. C. Fjelstad, "Method and apparatus for protecting digital rights of copyright holders of publicly distributed multimedia files," U.S. Patent 20080247543 A1, Oct 9, 2008.

[19] Z.Fu, et al, "Enabling semantic search based on conceptual graphs over encrypted outsourced data," *IEEE Trans. Services Computing*, DOI 10.1109/TSC.2016.2622697, 2017.

[20] S. Dodge and L. Karam, "Understanding How Image Quality Affects Deep Neural Networks," *arXiv preprint*, arXiv:1604.04004, 2016.

[21] S. Karahan, M. Kilinc Yildirum, K. Kirtac, F. S. Rende, G. Butun and H. K. Ekenel, "How Image Degradations Affect Deep CNN-Based Face Recognition?" *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1-5, 2016.

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

[23] Zhu, B.B.; Feng, M.; Li, S., "An efficient key scheme for layered access control of MPEG-4 FGS video," *IEEE ICME*, pp. 443-446, 2004.

[24] Crampton, J.; Daud, R.; Martin, K. M., "Constructing Key Allignment Schemes from Chain Partitions," *IFIP WG 11.3 working conf. on Data and applications security and privacy*, 2010.

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

[26] X. Zhu and C. W. Chen, "A collusion resilient key management scheme for multi-dimensional scalable media access control," *IEEE ICIP*, 2011.

[27] C. Wang, K. Ren and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," *INFOCOM*, pp. 820-828, 2011.

[28] J. Zhou, X. Dong, Z. Cao and A. V. Vasilakos, "Secure and Privacy Preserving Protocol for Cloud-Based Vehicular DTNs," *IEEE Trans. Information Forensics and Security*, vol. 10, no. 6, pp. 1299-1314, 2015.

[29] Y. Wu, W. Zhuo, and R. Deng, "Attribute-Based Access to Scalable Media in Cloud-Assisted Content Sharing Networks," *IEEE Trans. Multimedia,* vol.15, no.4, pp.778-788, 2013.

[30] Soete, M. D.,"Attribute certificate," *Encyclopedia of Cryptography and Security, Springer*, 2011, pp. 51.

[31] C. Ma and C. W. Chen, "Secure media sharing in the cloud: Two-dimensional-scalable access control and comprehensive key management," *IEEE ICME*, 2014.

[32] C. Ma, Z. Yan, and C. W. Chen, "Attribute-Based Multi-Dimension Scalable Access Control For Social Media Sharing," *IEEE ICME*, 2016.

[33] K. Yang, Z. Liu, X. Jia and X. S. Shen, "Time-Domain Attribute-Based Access Control for Cloud-Based Video Content Sharing: A Cryptographic Approach," in IEEE Trans. Multimedia, vol. 18, no. 5, pp. 940-950, 2016.

[34] K. G. Paterson, "ID-based signatures from pairings on elliptic curves," *Electronics Letters*, vol. 38, no. 18, pp. 1025-1026, 2002.

[35] U. Maurer, "Towards proving the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms," *Proc. of Crypto '94*, pp. 271-281.

[36] D. Hankerson, A. Menezes, S. Vanstone, "Guide to Elliptic Curve Cryptography," *Springer-Verlag* New York, Inc. 2004. ISBN 0-387-95273-X.

[37] B. Preneel, "Cryptographic hash functions," *European Trans. Telecom.*, 5 (1994), pp. 431-448, 1994.

[38] D. M. Dumbere and N. J. Janwe, "Video encryption using AES algorithm," *Second International Conference on Current Trends In Engineering and Technology (ICCTET)*, pp. 332-337, 2014.

[39] http://www.infoq.com/news/2014/03/mobile_app_performance benchmark

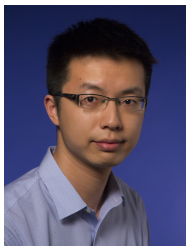
[40] W. Zhu, C. Luo, J. Wang, S. Li, "Multimedia Cloud Computing," Signal Processing Magazine, IEEE, vol.28, no.3, pp.59,69, May 2011.

[41] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, N. Venkatasubramanian, "Mobile Cloud Computing: A Survey, State of Art and Future Directions," *Mobile Netw Appl* (2014) 19: 133. https://doi.org/10.1007/s11036-013-0477-4

[42] Jun Zhou et al, "4S: A secure and privacy-preserving key management scheme for cloud-assisted wireless body area network in m-healthcare social networks," *Inf. Sci.* 314: 255-276, 2015.

[43] Zheng Yan, et al, "Two Schemes of Privacy-Preserving Trust Evaluation," *Future Generation Comp. Syst.* 62: 175-189, 2016.

Changsha Ma received her B.Eng. degree from Southeast University, China in 2010, M.Eng. degree from University of Science and Technology of China in 2013, and Ph.D. degree in Computer Science and Engineering at the State University of New York at Buffalo, USA in 2018. Her research interest includes security, privacy, and data analytics in social networks.



Zhisheng Yan received B.Eng. degree from Shandong University, China in 2010, M.Eng. degree from University of Science and Technology of China in 2013, and Ph.D. degree in Computer Science and Engineering from State University of New York at Buffalo in 2017. Currently, he is an Assistant Professor in the Department of Computer Science at Georgia State University. His research interests lie in mobile and networked systems, including content delivery systems, mobile health, and energy-saving mobile display.

Chang Wen Chen (F'04) received his BS from University of Science and Technology of China in 1983, MSEE from University of Southern California in 1986, and Ph.D. from University of Illinois at Urbana-Champaign in 1992. He is currently a Dean and Professor of School of Science and Engineering at The Chinese University of Hong Kong, Shenzhen and an Empire Innovation Professor of Computer Science and Engineering at the University at Buffalo, State University of New York. He was Allen Henry Endow Chair Professor at the Florida Institute of Technology from July 2003 to December 2007. He was on the faculty of Electrical and Computer Engineering at the University of Rochester from 1992 to 1996 and on the faculty of Electrical and Computer Engineering at the University of Missouri-Columbia from 1996 to 2003.

He has been the Editor-in-Chief for IEEE Trans. Multimedia from 2014 to 2016. He has also served as the Editor-in-Chief for IEEE Trans. Circuits and Systems for Video Technology from 2006 to 2009. He has been an Editor for several other major IEEE Transactions and Journals, including the Proceedings of IEEE, IEEE Journal of Selected Areas in Communications, and IEEE Journal on Emerging and Selected Topics in Circuits and Systems. He has served as Conference Chair for several major IEEE, ACM and SPIE conferences related to multimedia, video communications and signal processing. His research is supported by NSF, DARPA, Air Force, NASA, Whitaker Foundation, Microsoft, Intel, Kodak, Huawei, and Technicolor.

He and his students have received eight (8) Best Paper Awards or Best Student Paper Awards over the past two decades. He has also received several research and professional achievement awards, including the Sigma Xi Excellence in Graduate Research Mentoring Award in 2003, Alexander von Humboldt Research Award in 2010, University at Buffalo Exceptional Scholar - Sustained Achievement Award in 2012, and State University of New York Chancellor's Award for Excellence in Scholarship and Creative Activities in 2016. He is an IEEE Fellow and an SPIE Fellow.