An Efficient KP-ABE Scheme for Content Protection in Information-Centric Networking

Jinmiao Wang, Bo Lang
State Key Laboratory of Software Development Environment, Beihang University, Beijing, China 100191
School of Computer Science and Engineering, Beihang University, Beijing, China 100191
Email: wangjinmiao@buaa.edu.cn, langbo@buaa.edu.cn

***Abstract*—Media streaming has largely dominated the Internet traffic and the trend will keep increasing in the next years. To efficiently distribute the media content, Information-Centric Networking (ICN) has attracted many researchers. Since end users usually obtain content from indeterminate caches in ICN, the publisher cannot reinforce data security and access control depending on the caches. Hence, the ability of self-contained protection is important for the cached contents. Attribute-based encryption (ABE) is considered the preferred solution to achieve this goal. However, the existing ABE schemes usually have problems regarding efficiency. The exponentiation in key generation and pairing operation in decryption respectively increases linearly with the number of attributes involved, which make it costly. In this paper, we propose an efficient key-policy ABE with fast key generation and decryption (FKP-ABE). In the key generation, we get rid of exponentiation and only require multiplications/divisions for each attribute in the access policy. And in the decryption, we reduce the pairing operations to a constant number, no matter how many attributes are used. The efficiency analysis indicates that our scheme has better performance than the existing KP-ABE schemes. Finally, we present an implementation framework that incorporates the proposed FKP-ABE with the ICN architecture.**

***Index Terms*—Information-Centric Networking; Content protection; Attribute based encryption; Fast key generation; Fast decryption**

I. Introduction

According to the report from Sandvine [1], media streaming has become the largest traffic category on virtually every kind of network. In North America, Netflix and YouTube take up 49% of all fixed downstream Internet traffic and 24% of mobile. Cisco Visual Networking Index forecasts that the global consumer Internet video traffic will be 80% of all consumer Internet traffic in 2019, up from 64% in 2014 [2]. With such a trend, content distribution has become a challenge for the multimedia applications since traditional multicast method is not suitable for the case of content on demand, and the unicast schemes do not scale efficiently in the case of popular content [3]. Recently, there has been a concerted push to redesign the Internet architecture, and the Information-Centric Networking (ICN) is considered as the candidate of the new architecture. Although the existing ICN oriented projects investigate different aspects, caching is an integral component of all these architectures. The use of caches can minimize transmission delays and alleviate network congestion, which can efficiently distribute content on demand, as shown in Fig.1. In the case of several users requesting the same content, suppose Alice is the first requester, then she needs to obtain the content from the storage server because the content is not cached beforehand. During the download process which is indicated by the red path, the content is cached in the ICN nodes that are traversed, i.e. node 1 and 2. Next, when Bob requests the content, he finds the cached content in node 1. During his downloading which follows the blue path, the content is cached in node 3. Finally, Chris can get the same content from node 3 using the green path. Similarly, other users like Tom or Mary in the same domain with Alice, Bob and Chris can get content directly from node 2 or 3.

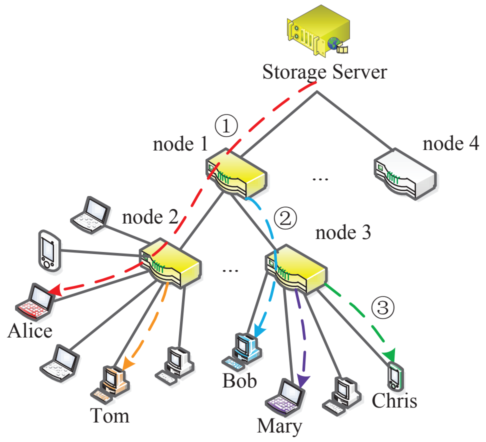


Fig. 1. Content distribution in ICN

Nevertheless, this way of content delivery raises severe security concerns. As end users usually obtain contents from caches which are numerous and unknown to the publisher, it is impossible to achieve security and access control depending on the caches. Furthermore, ICN promotes the notion that content is application-independent, location-independent, etc., namely it requires that the content is totally self-dependent. Thus, the content is rendered self-secure and does not rely on any other third parties. To achieve this goal, the most intuitive solution is to encrypt the content which can only be decrypted by authorized users. Many encryption-based access control schemes have been proposed to ensure the content confidentiality and authenticity in ICN [4]–[6]. Some of them adopt classic encryption schemes, such as public-key encryption, which are designed for one-to-one communication and data encrypted by a public key can only be decrypted

978-1-5090-0679-3/16/$31.00 ©2016 IEEE

by the specific private key. In this case, each content object should be encrypted using different keys for different users, which introduces much redundancy and prevents the efficient use of caches. On the other hand, just adopting the classic encryption cannot enforce access control according to different subscriptions, i.e. it cannot enforce fine-grained access control.

To address these limitations, Papanis et al. [3] and Ion et al. [7] introduced attribute based encryption (ABE) into ICN architecture. In ABE, both user's private key and ciphertext are associated with some attributes [8]. The biggest difference between ABE and classic encryption schemes is that ABE is designed for many-to-many communication, and each public key corresponds to more than one private key. Once there is a match between the attributes of the ciphertext and the attributes in a user's private key, can the user decrypt the ciphertext. ABE is applicable for ICN as the content is encrypted with access policy and the policy is integrated into the protected content, which gives content the ability to enforce access control depending on just itself and achieve self-contained protection. Another important property of ABE is its collusion resistance, i.e. unauthorized users cannot combine their private keys to decrypt the ciphertext that any of them cannot decrypt individually. Hence, ABE can maintain the advantages of caching while ensure the security of the contents. There are two variants of ABE, i.e. ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE). Since KP-ABE is content-centric, i.e. the attributes is associated with content, and CP-ABE is user-centric [9], KP-ABE is more suitable for content protection in ICN architecture.

In the large scale user-oriented multimedia applications, a noteworthy feature is that the user group is large and dynamically changing. For example, as of October 2015, Netflix reported 69.17 million subscribers worldwide, and expected to end of 2015 with over 74 million members [10]. At the same time there may be thousands upon thousands new users joining the system and old users updating their subscriptions, which will make key generation quite frequent. In practice, in order to protect the commercial interests and users' privacy, the Key Generation Servers (KGSs) are usually internal servers whose performance may be limited. Hence, each multimedia publisher should take into account the efficiency of private key generation. Efficient key generation not only can save expenses by deploying fewer KGSs, but also decreases the response time of private key requesting which can optimize the end users' experience.

However, although ABE is powerful, it is costly. In the existing ABE schemes, the key generation algorithm needs at least one exponentiation for each attribute of the user. In KP-ABE, the key generation algorithm needs some polynomial or matrix operations at all attributes to share a secret among them, which already consume some computing resources. The exponentiations introduced by the attributes further aggravate the burden of KGS, especially when the user group is large and changing dynamically. All of these heavy burdens centralized at KGS would make it becoming a bottleneck in practice. Hohenberger and Waters [11] proposed an online/offline encryption scheme to alleviate the efficiency problem of key generation by splitting it into two phases. The offline phase finished the majority work before it knew the access policy and the online phase rapidly assembled the private key with few computations. As far as we know, it is the only scheme that gets rid of exponentiations in $real$ key generation, even though the entire process still needs some.

On the other hand, the efficiency of decryption also needs to be improved. In decryption of most ABE schemes, the pairing operations which consume much more time and memory than other calculations are proportional to the number of attributes used for decryption. For example, in the above mentioned online/offline encryption scheme, the decryption requires 3 pairing operations for each attribute involved, which is relatively costly. Nowadays users may access multimedia contents with some lightweight devices which have limited CPU and memory resources, such as mobile phone, tablet, etc. According to the Cisco Forecast [12], mobile video will increase 13-fold between 2014 and 2019, accounting for 72% of total mobile data traffic by the end of the forecast period (i.e. 2019) . Thus the decryption cost is also an important factor to be concerned.

To improve the efficiency of decryption, many researchers focus on reducing the pairing operations in decryption to a constant number. The KP-ABE schemes respectively proposed by Attrapadung et al. [13], Hohenberger et al. [14] and Lai et al. [15] all reduce the pairing operations in decryption to 2 times. However, there is a common drawback in these schemes: the generation of private key requires a large number of exponentiations which approximately equals to the quadratic of attributes. Takashima [16] reduced the pairing operations in decryption to constant 17 times which is much more than [13]–[15]. Also, its key generation is costly and needs 3 exponentiations for each attribute of the user. To our best knowledge, there are still no method that can really get rid of exponentiations in key generation while reduce the pairing operations in decryption to a constant number.

a) Contributions: Due to the inefficiency of key generation and decryption, ABE is still difficult to be applied in ICN architecture and the mobile environment at present. To address these limitations, we propose a more efficient KP-ABE scheme with Fast key generation and decryption (FKP-ABE) in this paper. The main contributions of our work are summarized as follows:

1) The key generation of FKP-ABE only requires one exponentiation, no matter how complex the access policy is. For each attribute it only needs 2 multiplications/divisions which are quite less resource-consuming and the consumption can be ignored. Hence, the key generation of our scheme is much more efficient and faster than other schemes.
2) By converting the pairing operations to exponentiations, we reduce the pairing operations in decryption to only once which is less than the existing KP-ABE schemes.
3) We present a video on demand system framework that incorporates FKP-ABE with the ICN architecture. In this

system, the multimedia content obtains the ability of self-contained protection, and end users can efficiently obtain the required content and access it with lightweight devices.

b) Organization: We review some background knowledge of KP-ABE in Section II. In Section III, we first illustrate the access structure used in our scheme, and then we propose our construction. The security and performance of our scheme are evaluated in Section IV. Section V presents an implementation framework that incorporates FKP-ABE with the ICN architecture. Finally, this paper is concluded in Section VI.

II. Background Knowledge

In this section, we firstly introduce the relevant preliminaries about bilinear maps and Decision Bilinear Diffie-Hellman (DBDH) assumption. Then, we give some background knowledge of KP-ABE.

A. Preliminaries

1) Bilinear Maps: Let $\mathbb{G}_0$ and $\mathbb{G}_1$ be two multiplicative cyclic groups of prime order p. Let g be the generator of $\mathbb{G}_0$. Define a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$. It has the following properties:

- Bilinearity. For all $x, y \in \mathbb{G}_0$ and $a, b \in \mathbb{Z}_p^*$, it has $e(x^a, y^b) = e(x, y)^{ab}$.
- Non-degeneracy. $e(g, g) \neq 1$.

If the group operation in $\mathbb{G}_0$ and the bilinear map e are both computable, the multiplicative cyclic group $\mathbb{G}_0$ is a bilinear group. Notice that the map e is symmetric since $e(x^a, y^b) = e(x, y)^{ab} = e(x^b, y^a)$.

2) Decisional Bilinear Diffie-Hellman (DBDH) Assumption: Let $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$ be a computable bilinear map and g is the generator of $\mathbb{G}_0$. Choose random integer $a, b, c, z \in \mathbb{Z}_p^*$. The DBDH assumption is that no probabilistic polynomial-time algorithm can distinguish the tuple $D_{bdh} = (g, g^a, g^b, g^c, e(g, g)^{abc})$ from the tuple $D_{rand} = (g, g^a, g^b, g^c, e(g, g)^z)$ with more than a negligible advantage ε:

$$\varepsilon = |\Pr[\beta(D_{bdh}) = 0] - \Pr[\beta(D_{rand}) = 0]|$$

B. KP-ABE

1) KP-ABE algorithm: The KP-ABE scheme consists of the following algorithms. The main notations and their definitions used in this paper are listed in Table I.

Setup(γ**,** U**).** The setup algorithm takes a security parameter γ and an attribute universe U as input and outputs the public parameters pk and the master key mk.

Encrypt(pk, M, w**).** The encryption algorithm takes the public parameters pk, a message M and a set of attributes w as input. It will publish a ciphertext C_w associate with w.

KeyGen($mk, \mathcal{T}$**).** The key generation algorithm takes the master key mk and a user's access policy tree $\mathcal{T}$ as input and outputs a secret key $sk_{\mathcal{T}}$ associated with $\mathcal{T}$.

Decrypt($sk_{\mathcal{T}}, C_w$**).** The decryption algorithm takes a secret key $sk_{\mathcal{T}}$ and a ciphertext C_w as input. If the attribute set w satisfies the access tree $\mathcal{T}$, the algorithm will decrypt the ciphertext and return the plaintext M. Otherwise, it will return an error symbol $\perp$.

TABLE I
Main notations and definitions

Notation	Definition
pk	public key of KP-ABE
mk	master key of KP-ABE
U	attribute universe in system
M	plaintext message
w	the set of encryption attribute
C_w	ciphertext encrypted under w
$\mathcal{T}$	access tree
$sk_{\mathcal{T}}$	private key associated with $\mathcal{T}$
num_z	number of children of node z in $\mathcal{T}$
k_z	threshold value of node z in $\mathcal{T}$
$parent(z)$	parent of node z in $\mathcal{T}$
$index(z)$	index of node z in $\mathcal{T}$
$att(z)$	attribute associated with leaf node z

2) Security Model: The semantic security under chosen-plaintext attack (CPA) is modeled by a game between a challenger and an adversary. It includes six phases detailed as follows:

Init. The adversary chooses a challenge set of attributes w^* and sends it to the challenger.

Setup. The challenger runs $Setup$ algorithm to generate public parameters pk and master key mk. Then he sends pk to the adversary.

Phase 1. The adversary is allowed to make secret key request for any access tree $\mathcal{T}$, with the restriction that $w^* \notin \mathcal{T}$. The challenger returns $sk_{\mathcal{T}}$ to the adversary.

Challenge. The adversary sends two equal length message M_0, M_1 to the challenger. The challenger chooses a random $b \in \{0, 1\}$, and encrypts M_b with the attributes w^*. Then the ciphertext C_{w^*} is returned to the adversary.

Phase 2. **Phase 1** is repeated with the same restriction that $w^* \notin \mathcal{T}$.

Guess. The adversary outputs a guess $b' \in \{0, 1\}$.

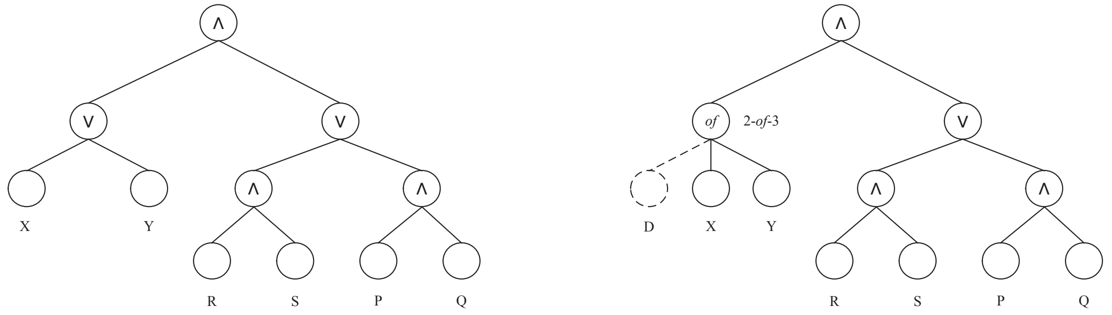
Definition 1: A KP-ABE scheme is said to be secure against a chosen plaintext attack (CPA) if any polynomial-time adversaries have at most a negligible advantage in the above game. The advantage of an adversary is defined as $\varepsilon = |\Pr[b' = b] - 1/2|$.

III. Our Construction

In this section we propose a KP-ABE scheme with Fast key generation and decryption (FKP-ABE). Firstly, we illustrate the access structure used in our scheme. Next, we describe the construction of FKP-ABE.

A. Policy Representation

Restricted Access Tree (RAT). To enhance the security, the access policy is expressed by a restricted access tree in our scheme. Let $\mathcal{T}$ be a tree representing an access structure. Each leaf node of $\mathcal{T}$ is associated with an attribute. Except the parents of leaf nodes, each internal node of $\mathcal{T}$ represents a threshold operator including AND, OR and of (i.e. $k -$



(a) Traditional access tree

(b) Corresponding restricted access tree

Fig. 2. The policy representation of restricted access tree

$of - n$ where $k < n$), which is described by its children and a threshold value. The parents of leaf nodes are the same as other internal nodes except that they cannot represent OR operator. This kind of tree $\mathcal{T}$ is called a restricted access tree (RAT).

Let num_z denote the children number of a node z and k_z denote its threshold value, then we have $1 < k_z \leq num_z$ for each parent of leaf nodes and $1 \leq k_z \leq num_z$ for other internal nodes. When $k_z = 1$, the threshold is an OR operator (i.e. the parents of leaf nodes cannot represent OR operator), and when $k_z = num_z$ it is an AND operator.

To facilitate working with the restricted access tree, we also define some functions. The parent of the node z in the tree is denoted by $parent(z)$. The function $att(z)$ is defined only if z is a leaf node and denotes the attribute associated with z. The children of a node z are numbered from 1 to num_z. And the function $index(z)$ returns such a number associated with the node z.

Expressing policy with RAT means that we cannot directly express policy like "**X OR Y**". To remedy this limitation, we introduce one or more default attributes in the attribute universe. To express polices like "**X OR Y**", we add a new leaf node associated with a default attribute D to the internal node representing OR operator. Then the internal node is transformed to represent of operator (2-of-3) and the policy "**X OR Y**" is changed to be "**2 of (X, Y, D)**", as shown in Fig.2 where "$\wedge$" denotes AND operator and "$\vee$" denotes OR operator. Except the parents of leaf nodes, other internal nodes can represent OR operator freely.

Since we may introduce default attribute in user's private key, in order to ensure successful decryption we must generate corresponding ciphertext component for each default attribute in encryption phase.

B. KP-ABE with Fast Key Generation and Decryption

Let $\mathbb{G}_0$ be a bilinear group of prime order p with a generator g. And let $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$ denote the bilinear map. We also define the Lagrange coefficient $l_{j,S}$ for $j \in \mathbb{Z}_p^*$ and a subset S of $\mathbb{Z}_p^*$: $l_{j,S}(x) = \prod_{i \in S, i \neq j} \frac{x-i}{j-i}$. The construction of FKP-ABE is detailed as follows:

(1) Setup(γ, U)**.** The setup algorithm takes as inputs a security parameter γ and an attribute universe U which must include at least one default attribute. Then it chooses a bilinear group $\mathbb{G}_0$ of prime order p with a generator g and a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$. For each attribute $A_i \in U (1 \leq i \leq n$ where n denotes the number of attributes in the attribute universe U), choose $t_i \in \mathbb{Z}_p^*$ and set $T_i = g^{t_i}$. Finally, choose a random element $\alpha \in \mathbb{Z}_p^*$ and set $y = e(g, g)^{\alpha}$. The set of public key is:

$$pk = \{e, g, y, T_i (1 \leq i \leq n)\}$$

And the set of master key is:

$$mk = \{\alpha, t_i (1 \leq i \leq n)\}$$

(2) Encrypt(pk, M, w)**.** The encryption algorithm takes as inputs the public parameters pk, a message M and an attribute set w. Especially, the attribute set w must include all default attributes in the attribute universe U. To output the ciphertext of message M which is encrypted under w, the encryption algorithm should choose a random element $s \in \mathbb{Z}_p^*$ and compute $E_0 = My^s$. For each attribute $A_i \in w$, compute $E_i = T_i^s$. Finally, publish the ciphertext:

$$C_w = \{w, E_0, \forall A_i \in w : E_i\}$$

(3) KeyGen($mk, \mathcal{T}$)**.** The key generation algorithm takes as inputs the master key mk and a user's access policy tree $\mathcal{T}$ which must be a RAT. Then the key generation algorithm chooses a polynomial $q_z(x)$ for each node z in the following way, starting from the root node r in a top-down manner.

For each node z in the tree, set the degree d_z of the polynomial $q_z(x)$ to be one less than the threshold value k_z, i.e. set $d_z = k_z - 1$. Then, for the root node r, set $q_r(0) = \alpha$ and randomly choose d_r other points to define the polynomial $q_r(x)$ completely. For any other node z, set $q_z(0) = q_{parent(z)}(index(z))$ and randomly choose d_z other points to completely define $q_z(x)$.

Next, choose a random element $\kappa \in \mathbb{Z}_p^*$ and compute $D_0 = g^{\kappa}$. For each leaf node z, let $att(z) = A_i$ and compute $D_z = q_z(0)/\kappa t_i$. Finally, it returns the private key to the user:

$$sk_{\mathcal{T}} = \{\mathcal{T}, D_0, \forall A_i \in \mathcal{T} : D_z\}$$

(4) Decrypt($sk_{\mathcal{T}}, C_w$). The decryption algorithm first defines a recursive function $DecryptNode(C_w, sk_{\mathcal{T}}, z)$ that takes as inputs the ciphertext C_w, the private key $sk_{\mathcal{T}}$ and a node z from the user's RAT $\mathcal{T}$. If the attribute set w does not satisfy $\mathcal{T}$, the function returns an error symbol $\perp$. Otherwise, if the node z is a leaf node, let $A_i = att(z)$ and compute:

$$DecryptNode(C_w, sk_{\mathcal{T}}, z) = E_i^{D_z} = g^{\frac{sq_z(0)}{\kappa}}$$

If the node z is a non-leaf node, the function $DecryptNode(C_w, sk_{\mathcal{T}}, z)$ will proceed as follows: for all nodes h that are the children of z, it calls the $DecryptNode(C_w, sk_{\mathcal{T}}, h)$ and stores the output as F_h. Let S_z be an arbitrary k_z-sized set of child nodes h such that $F_h \neq \perp$, we compute:

$$\begin{aligned} F_z &= \prod_{h \in S_z} F_h^{l_{j,S'_z}(0)} \\ &= \prod_{h \in S_z} g^{\frac{sq_h(0)}{\kappa} \cdot l_{j,S'_z}(0)} \\ &= \prod_{h \in S_z} g^{\frac{s}{\kappa} q_{parent(h)}(index(h)) \cdot l_{j,S'_z}(0)} \\ &= \prod_{h \in S_z} g^{\frac{s}{\kappa} q_z(j) \cdot l_{j,S'_z}(0)} \\ &= g^{\frac{sq_z(0)}{\kappa}} \end{aligned}$$

where $j = index(h)$ and $S'_z = \{index(h) : h \in S_z\}$.

Now we have defined the $DecryptNode$ function, the decryption algorithm should firstly call the function on the root of $\mathcal{T}$. If the attribute set w satisfies the access tree $\mathcal{T}$, we will get

$$A = DecryptNode(C_w, sk_{\mathcal{T}}, r) = g^{\frac{s\alpha}{\kappa}}$$

Next, we compute

$$B = e(A, D_0) = e(g, g)^{\alpha s}$$

Then the algorithm returns the plaintext M', where

$$M' = \frac{E_0}{B} = \frac{My^s}{e(g,g)^{\alpha s}} = M$$

IV. Security and Performance

In this section, we firstly prove our scheme is secure against chosen-plaintext attack (CPA). Then we evaluate the performance of our scheme and compare it with other existing schemes.

A. Security Proof

The security of FKP-ABE is proved under the security model presented in Section II-B2. Since its security is based on DBDH assumption, the advantage of breaking through FKP-ABE is reduced to the advantage of solving the DBDH problem.

Theorem 1: Suppose the DBDH assumption holds. Then there is no adversary can break FKP-ABE scheme in polynomial time.

Proof: Suppose there is a polynomial-time adversary who can win the game described in Section II-B2 with a non-negligible advantage ε. Then we can build a simulator $\mathcal{S}$ who can distinguish the DBDH tuple D_{bdh} from a random tuple D_{rand} with a non-negligible advantage $\varepsilon/2$, which indicates DBDH assumption does not hold. The simulation is detailed as follows.

Firstly, the challenger generates public parameters which include groups $\mathbb{G}_0$ and $\mathbb{G}_1$ with an efficient bilinear map e, and a generator g of $\mathbb{G}_0$. Suppose the attribute universe in this proof contains just one default attribute d. Then the challenger flips a fair coin outside the simulator's view and gets a random $\lambda \in \{0, 1\}$. The challenger chooses random elements $a, b, c, z \in \mathbb{Z}_p^*$. If $\lambda = 0$, the challenger sets $D_{bdh} = (g, A, B, C, Z) = (g, g^a, g^b, g^c, e(g, g)^{abc})$. If $\lambda = 1$, the challenger sets $D_{rand} = (g, A, B, C, Z) = (g, g^a, g^b, g^c,$ $e(g, g)^z)$. Then the challenger sends $D = (g, A, B, C, Z)$ to the simulator $\mathcal{S}$, and $\mathcal{S}$ will play the role of challenger in the next game.

Init. The adversary $\mathcal{A}$ sends to $\mathcal{S}$ a set of attributes w^* that he wants to challenge in this game. Notice that the attributes set w^* must contain the default attribute d.

Setup. The simulator $\mathcal{S}$ sets $\alpha = ab$. Thus the parameter y is set to be $y = e(g, g)^{ab} = e(A, B)$. For each attribute $A_i \in U$, $\mathcal{S}$ chooses a random $h_i \in \mathbb{Z}_p^*$. If $A_i \in w^*$, set $T_i = g^{h_i}$ (thus, $t_i = h_i$). If $A_i \notin w^*$, set $T_i = g^{ah_i} = A^{h_i}$ (thus, $t_i = ah_i$). For the default attribute d, it sets $T_i = g^{h_i}$ because $d \in w^*$. Then $\mathcal{S}$ sends the public parameters to $\mathcal{A}$.

Phase 1. The adversary $\mathcal{A}$ requests private keys corresponding to any RAT $\mathcal{T}$ with the restriction that w^* cannot satisfy $\mathcal{T}$. Then $\mathcal{S}$ responds to $\mathcal{A}$'s quires in two steps. First, it creates a valid but not well-distributed private key. Then, it re-randomizes the private key to ensure that it is well distributed.

To create a valid private key, $\mathcal{S}$ assigns $q_r(0) = 1$. For each leaf node z, let $A_i = att(z)$. If $A_i \in w^*$, $\mathcal{S}$ sets $q_{parent(z)}(index(z)) = 0$ and randomly choose other points to define the polynomial $q_{parent(z)}(x)$ completely. If no such $q_{parent(z)}(x)$ exists (only when those $A_i \in w^*$ satisfy the subtree rooted at $parent(z)$), it returns $\perp$. For other nodes, $\mathcal{S}$ normally calculates the value of $q_z(0)$ as detailed in **Keygen**.

Then $\mathcal{S}$ sets $Q_z = ab \cdot q_z(0)$ for each node z of $\mathcal{T}$. Note that the actual value shared among all nodes of $\mathcal{T}$ is $\alpha = ab$. Next, $\mathcal{S}$ chooses a random element $\kappa' \in \mathbb{Z}_p^*$ and sets $\kappa = b\kappa'$. Then it sets $D_0 = g^{b\kappa'} = B^{\kappa'}$. Through the secret sharing process we get that for each $A_i \in w^*$, it has $D_z = 0$. For each leaf node $A_i \notin w^*$, since $\mathcal{S}$ sets $t_i = ah_i$, it computes

$$D_z = \frac{Q_z}{\kappa t_i} = \frac{abq_z(0)}{b\kappa' \cdot ah_i} = \frac{q_z(0)}{\kappa' h_i}$$

Now, $\mathcal{S}$ has constructed the components of a valid private key. But it is not well distributed. To re-randomize the private key, $\mathcal{S}$ assigns $\rho_r(0) = 0$ and calculates the value of $\rho_z(0)$ as detailed in **Keygen**. For all leaf nodes $z \in \mathcal{T}$, $\mathcal{S}$ computes $R_z = \rho_z(0)/\kappa' h_i$. Then $\mathcal{S}$ re-randomizes the private key by setting

$$D'_z = D_z + R_z = \frac{q_z(0) + \rho_z(0)}{\kappa' h_i}$$

Through the two steps, we claim that we have got a valid and well-distributed private key. A valid key is generated from the sharing of α, but it is not a well distributed one. The re-randomization procedure generates a fresh secret sharing for α used in D'_z. After applying the re-randomization, any valid private key generated under access policy $\mathcal{T}$ has been redistributed properly and will has the same distribution as a fresh key generated by running **KeyGen(**mk**,** $\mathcal{T}$**)**.

Finally, $\mathcal{S}$ sends the private key $sk_{\mathcal{T}}$ to $\mathcal{A}$.

Challenge. The adversary $\mathcal{A}$ submits two equal length message M_0, M_1 to $\mathcal{S}$. $\mathcal{S}$ chooses a random $b \in \{0, 1\}$ through flipping a fair coin and encrypts the message M_b with the challenge policy w^*. Then $\mathcal{S}$ sets $s = c$. Thus, $E_0 = M_b e(g, g)^{\alpha s} = M_b Z$. For each attribute $A_i \in w^*$, compute $E_i = g^{ch_i} = C^{h_i}$. Then the ciphertext C_{w^*} is sent to $\mathcal{A}$ as the challenge ciphertext.

Phase 2. $\mathcal{A}$ continues to send the private key requests to $\mathcal{S}$ as in **Phase 1**.

Guess. $\mathcal{A}$ gives a guess $b' \in \{0, 1\}$.

If $b' = b$, $\mathcal{S}$ outputs its guess $\lambda' = 0$ which indicates $Z = e(g, g)^{abc}$. Otherwise, $\mathcal{S}$ will guess $\lambda' = 1$ which indicates $Z = e(g, g)^z$.

When $Z = e(g, g)^{abc}$, $\mathcal{S}$ performs a reasonable simulation and C_{w^*} is a valid ciphertext. Since $\mathcal{A}$ has advantage ε to win the above game, $\mathcal{S}$ will solve the DBDH assumption with the following advantage:

$$Pr[b' = b \mid \lambda' = 0] = \frac{1}{2} + \varepsilon$$

When $Z = e(g, g)^z$, the ciphertext C_{w^*} is a random group element for $\mathcal{A}$ and $\mathcal{A}$ cannot get any information about M_b. Then $\mathcal{S}$ will solve the DBDH assumption with the following advantage:

$$Pr[b' \neq b \mid \lambda' = 1] = \frac{1}{2}$$

Since $\mathcal{S}$ will guess $\lambda' = 0$ when $b' = b$ and $\lambda' = 1$ when $b' \neq b$, he will solve the DBDH assumption with the following advantage:

$$\frac{1}{2} Pr[\lambda' = \lambda \mid \lambda = 0] + \frac{1}{2} Pr[\lambda' = \lambda \mid \lambda = 1] - \frac{1}{2} = \frac{\varepsilon}{2}$$

Hence, if the adversary has the advantage ε to win the challenge game, the simulator will solve the DBDH assumption with advantage $\varepsilon/2$ by the help of the adversary's advantage. However, there is no effective polynomial that can solve the DBDH problem with a non-negligible advantage according to the DBDH assumption. Therefore, the adversary also cannot win the game with the non-negligible advantage $\varepsilon/2$, namely, the adversary has no advantage to break through the FKP-ABE scheme. □

B. Performance Evaluation

In our scheme, the encryption algorithm needs $|w|$ exponentiations on $\mathbb{G}_0$ and 1 exponentiation on $\mathbb{G}_1$, where $|w|$ denotes the number of attributes in the attribute set w. The key generation algorithm needs 1 exponentiation on $\mathbb{G}_0$ and $2t$ multiplications/divisions, where t denotes the number of leaf nodes in user's access tree $\mathcal{T}$. Compared with exponentiations, the time consumption of multiplications/divisions is much less and can be ignored. Hence, the key generation of FKP-ABE is relatively efficient. The decryption algorithm requires 1 pairing operation and $|w'|$ exponentiations on $\mathbb{G}_0$, where w' is the attribute set of w that satisfies the user's access tree $\mathcal{T}$, and $|w'|$ is the number of attributes in w'. The performance comparisons of our scheme with other schemes are shown in Table II. Note that only our scheme and HW14 list the consumption of multiplications/divisions while others' are ignored.

In the classic GPSW06 scheme, the exponentiations in key generation and pairing operations in decryption both increase linearly with the number of attributes involved. Though ALP11, HW13 and LDL+14 all reduce the pairing operations in decryption to constant 2 times, they greatly increase the exponentiations, especially ALP11 and LDL+14. Hence, their decryption is still less efficient than our scheme. Besides, the number of exponentiations in their key generation approximately equals to the quadratic of the number of attributes, which will lead to the private key generator becoming a bottleneck of the system. HW14 is an online/offline encryption scheme. The elements within braces $\{\}$ denotes the consumption of offline while the others denote the ones of online. Though the online consumption of key generation and encryption is less than our scheme, the sum of online and offline is much more than ours. Besides, its decryption is much costly and requires 3 pairing operations and exponentiations for each attribute involved in decryption.

TABLE II
PERFORMANCE COMPARISON OF FKP-ABE WITH OTHER SCHEMES

Scheme	Access Structure	Encryption	Keygen	Decryption
GPSW06 [8]	Tree	$\lvert w\rvert \mathbb{G}_0 + \mathbb{G}_1$	$t\mathbb{G}_0$	$\lvert w'\rvert C_e + \lvert w'\rvert \mathbb{G}_1$
ALP11 [13]	LSSS	$(\lvert w\rvert + 2)\mathbb{G}_0 + \mathbb{G}_1$	$(2W + 1)t\mathbb{G}_0$	$2C_e + (2\lvert w'\rvert + \lvert w\rvert)\mathbb{G}_0$
LDL+14 [15]	LSSS	$(m + 2)\mathbb{G}_0 + \mathbb{G}_1$	$(t + 3)t\mathbb{G}_0$	$2C_e + (m + 1)\lvert w'\rvert \mathbb{G}_0$
HW13 [14]	LSSS	$(\lvert w\rvert + 1)\mathbb{G}_0 + \mathbb{G}_1$	$(t + 2)t\mathbb{G}_0$	$2C_e + 2\lvert w'\rvert \mathbb{G}_0$
HW14 [11]	LSSS	$\Phi + \{(1 + 4\lvert w\rvert)\mathbb{G}_0 + \mathbb{G}_1\}$	$t\Phi + \{5t\mathbb{G}_0\}$	$3\lvert w'\rvert C_e + 3\lvert w'\rvert \mathbb{G}_0 + \mathbb{G}_1$
FKP-ABE(ours)	RAT	$\lvert w\rvert \mathbb{G}_0 + \mathbb{G}_1$	$2t\Phi + \mathbb{G}_0$	$C_e + \lvert w'\rvert \mathbb{G}_0$

Note: Φ denotes the multiplications/divisions in $\mathbb{Z}_p^*$. $\mathbb{G}_0$ and $\mathbb{G}_1$ represent the exponentiations on group $\mathbb{G}_0$ and $\mathbb{G}_1$ respectively. C_e denotes the pairing operations. t stands for the number of attributes in an access structure. $|w|$ ($|w| \leq n$) is the number of attributes associated with a ciphertext. $|w'|$ ($|w'| \leq |w|$) denotes the number of attributes used in decryption. W denotes the maximum number of attributes in a ciphertext. m is the total number of attribute categories in the system.

Except GPSW06 and FKP-ABE, all of other schemes in Table II are based on LSSS matrix. Both tree structure and LSSS are relatively expressive, and can support AND, OR and threshold operations. Specially, the tree structure is more flexible for its hierarchy, which is more legible for users to specify access policy. Table II indicates that our scheme is the first tree-based KP-ABE that greatly improves the efficiency in key generation and decryption.

By the efficiency analysis, we can conclude that the integrated efficiency of key generation and decryption of our scheme is higher than the existing schemes. Especially, the key generation of our scheme consumes much fewer resources and less time than other schemes. Hence, in some large scale user-oriented applications and ICN architecture, our scheme is more suitable than any other schemes.

V. Incorporating FKP-ABE with ICN Architecture

Fig.3 illustrates a video on demand system framework that incorporates FKP-ABE and ICN architecture. The system contains three parts: multimedia publisher, content storage and user end. The multimedia publisher is responsible for encrypting multimedia contents, managing subscriptions and generating private keys for end users. The content storage is used to store the protected multimedia contents. The user end contains the processes that need to be implemented at the user side. The components of the system are detailed as follows.

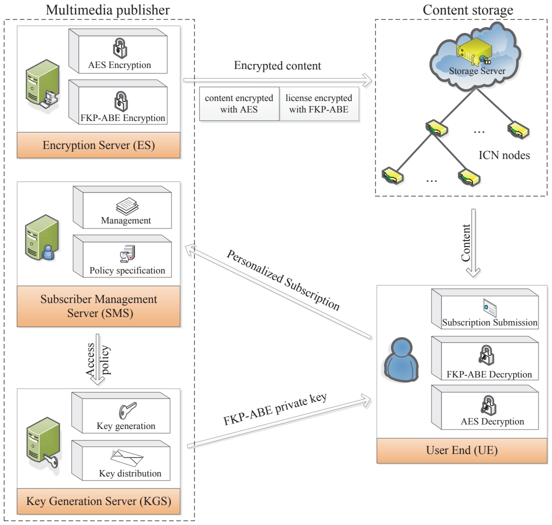


Fig. 3. A video on demand system framework incorporating FKP-ABE with ICN

- Encryption Server (ES). At present, symmetrical encryption, such as AES, is the most efficient encryption mechanism. So we use a hybrid encryption method which includes AES and FKP-ABE. ES consists of two modules: (i) AES encryption module is used to encrypt multimedia contents, and (ii) FKP-ABE encryption module is used to encrypt license which contains AES private key and some other rules.
- Subscriber Management Server (SMS). It consists of two modules: (i) management module is used to manage users' subscriptions, and (ii) policy specification module is used to specify access policy according to the subscriptions.
- Key Generation Server (KGS). It consists of two modules: (i) key generation module is used to generate FKP-ABE private key, and (ii) key distribution module is to distribute the private key through a secure channel.
- User End (UE). It consists of three modules: (i) subscription submission module is used to submit subscriptions to SMS; (ii) FKP-ABE decryption module is used to decrypt the FKP-ABE ciphertext to get AES private key, and (iii) AES decryption module is used to decrypt AES ciphertext to obtain the multimedia content.
- Content storage. It consists of storage servers and ICN nodes.

When an end user joining the system, he/she submits the personalized subscription (such as the multimedia type, published year, etc.) to SMS through UE and pays for them. According to the subscription, SMS specifies access policy and sends to KGS. The access policy contains the user's ID and subscription policy. For example, Alice subscribes action movies published in 2015 and all of the science-fiction movies, her access policy is $(ID_{Alice} \wedge ((action \wedge 2015) \vee science\text{-}fiction))$. When receiving an access policy, KGS generates the corresponding FKP-ABE private key and sends it to the end user. According to the performance analysis in Section IV-B, the key generation of FKP-ABE is much faster than other existing KP-ABE schemes and consumes only a very few resources, so the key generation server is able to generate private keys efficiently even though the subscriber group is large and dynamically changed.

To access a piece of multimedia content, the end user logins the system with his/her ID and downloads the corresponding ciphertext from the content storage. During the previous user's downloading, the content has been cached in the ICN nodes that are traversed. So the subsequent users just need to download from the nearest ICN nodes rather than the storage server. Hence, ICN can distribute the content on demand efficiently. Then the user decrypts the ciphertext with his/her private key. Since the user logins the system with ID, the ID component in user's private key (e.g. ID_{Alice} in Alice's private key) can be matched. Hence, once there is a match between the attributes associated with the content and other attributes in the user's private key, the user can decrypt successfully and get the AES private key which can be used to decrypt the content finally. Therefore, the content protection and fine-grained access control is enforced depending on just the content itself, without relying on any other third parties. Since the decryption of FKP-ABE is more efficient than the existing KP-ABE schemes, the end users can access multimedia content by consuming fewer resources and less time.

Hence, by incorporating FKP-ABE with the ICN architecture, the content obtains the ability of self-contained protection, and end users can efficiently obtain the required contents. Benefit from the high efficiency of FKP-ABE, multimedia publisher is allowed to deploy fewer KGS to save expense without increasing the response time of key requesting, and end users can access multimedia with lightweight devices. Besides, the modular structure of the framework also enables flexible deployment of ES, SMS and KGS.

VI. Conclusions

To improve the efficiency of ABE used in ICN architecture, we propose a tree-based KP-ABE scheme with fast key generation and decryption (FKP-ABE) by respectively reducing the exponentiations and pairing operations in key generation and decryption to a constant number. Based on the DBDH assumption we prove that our scheme is secure against chosen-plaintext attack. The performance analysis indicates that the integrated efficiency of our scheme is more efficient than the existing KP-ABE schemes. Hence, our scheme is more suitable for the large scale user-oriented ICN applications. For future work, it would be interesting to get rid of the attribute universe and construct a KP-ABE scheme where the encryption attributes need not to be specified in advance, while preferably maintaining the high efficiency of our scheme.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No.61170088) and Foundation of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2015ZX-05).

References

[1] Sandvine. (2015) Global internet phenomena report. [Online]. Available: https://www.sandvine.com/downloads/general/global-internet-phenomena/2014/2h-2014-global-internet-phenomena-report.pdf

[2] Cisco. (2015) Visual networking index. [Online]. Available: http://www.cisco.com/web/solutions/sp/vni/vni_forecast_highlights/index.html

[3] J. P. Papanis, S. I. Papapanagiotou, A. S. Mousas, G. V. Lioudakis, D. I. Kaklamani, and I. S. Venieris, "On the use of attribute-based encryption for multimedia content protection over information-centric networks," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 4, pp. 422–435, 2014.

[4] S. Misra, R. Tourani, and N. E. Majd, "Secure content delivery in information-centric networks: design, implementation, and analyses," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, 2013, pp. 73–78.

[5] J. Kuriharay, E. Uzun, and C. Wood, "An encryption-based access control framework for content-centric networking," in *IFIP Networking Conference*, 2015, pp. 1–9.

[6] M. Mangili, F. Martignon, and S. Paraboschi, "A cache-aware mechanism to enforce confidentiality, trackability and access policy evolution in content-centric networks," *Computer Networks*, vol. 76, pp. 126–145, 2015.

[7] M. Ion, J. Zhang, and E. M. Schooler, "Toward content-centric privacy in icn: attribute-based encryption and routing," in *Proceedings of the ACM SIGCOMM 2013 conference*, 2013, pp. 513–514.

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

[9] C. Li, B. Lang, and J. Wang, "Outsourced KP-ABE with chosen-ciphertext security," in *Proceedings of the 6th International Conference on Network & Communications Security*, 2014, pp. 147–160.

[10] Netflix. (2015) Final Q3-15 letter to shareholders with tables. [Online]. Available: http://files.shareholder.com/downloads/NFLX/861339127x0x854558/9B28F30F-BF2F-4C5D-AAFF-AA9AA8F4779D/FINAL_Q3_15_Letter_to_Shareholders_With_Tables_.pdf

[11] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Proceedings of the 17th International Conference on Practice and Theory in Public-Key Cryptography*, 2014, pp. 293–310.

[12] Cisco. (2015) Cisco visual networking index: Global mobile data traffic forecast update 2014-2019 white paper. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html

[13] N. Attrapadung, B. T. Libert, and E. de Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Proceedings of the 14th International Conference on Practice and Theory in Public-Key Cryptography*, 2011, pp. 90–108.

[14] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Proceedings of the 16th International Conference on Practice and Theory in Public-Key Cryptography*, 2013, pp. 162–179.

[15] J. Lai, R. H. Deng, Y. Li, and J. Weng, "Fully secure key-policy attribute-based encryption with constant-size ciphertexts and fast decryption," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, 2014, pp. 239–248.

[16] K. Takashima, "Expressive attribute-based encryption with constant-size ciphertexts from the decisional linear assumption," in *Proceedings of the 9th International Conference on Security and Cryptography for Networks*, 2014, pp. 298–317.