Integrating the Functional Encryption and Proxy Re-cryptography to Secure DRM Scheme

Hisham Abdalla, Xiong Hu, Abubaker Wahaballa, Ahmed Abdalla,
Mohammed Ramadan and Qin Zhiguang
(Corresponding author: Hisahm Abdalla Sedahmed)

University of Electronic Science and Technology of China
2006 Xiyuan Avenue, Gaoxin West Zone, Chengdu 611731, China.
(Email: hisham_awaw@hotmail.com)

Abstract

The current Digital Rights Management (DRM) systems use attribute-based encryption (ABE) and proxy re-encryption (PRE) to achieve fine-grained access control in cloud computing. However, these schemes have some limitations particularly in terms of security, functionality and also higher decryption time which grows linearly with the complexity of access policies. In this paper, we propose a novel DRM scheme founded on a deterministic finite automata-based functional proxy re-encryption (DFA-based FPRE) scheme which has been proven to be secure against CCA in the standard model. In particular, we leverage the DFA-based FPRE scheme to realize fine-grained access control over encrypted contents among a set of users. Furthermore, a secure content key distribution protocol and efficient revocation mechanism are provided. Moreover, we tackle the critical issue of high computation at the user side, by outsourcing computation into (DFA-based FPRE) scheme for the first time. In comparison, our scheme achieves higher efficiency and smaller computation time against state-of-the-art.

Keywords: Cloud computing, digital rights management, fine-grained access control, privacy preserving

1 Introduction

The rapid development and growth of the Internet have fuelled a trend towards outsourcing data and its management. The excitement of the emerging technology is due to the advancement of internet, whose infrastructure is cloud computing. It brings a flexible, cost effective and reliable way for data owners to deal with their data storage. Storing digital contents to the cloud, enable users to concentrate on their core business issues rather than incurring substantial hardware, software, or personal costs.

However, owners still have to remain cautious to protect their contents from being pirated and illegally distributed [24]. The cloud service provider is semi-trusted [27]. In this sense, the semi-trusted cloud service provider follows the normal flow of the protocol in the system. For instance, during the interaction with the users, a CSP may collect users' personal information and consumption profiles, which inspires a serious security concern for cloud user. Proxy re-encryption (PRE) technique [23] is devised to prevent the CSP from accessing the contents in semi-trusted cloud environment. Also, it's crucial for the CSP to be prevented from knowing exactly which users are accessing certain contents [21].

Digital rights management (DRM) is a famous mechanism to protect content copyright [1] based on the techniques of content encryption, access control, and dynamic licensing [16, 31]. In the past few years, there have been some DRM schemes which deal with confidentiality and privacy preserving of outsourced data in cloud computing. Petrlic et al. [23] introduced a privacy-preserving cloud DRM scheme based on proxy re-encryption, which allows a user anonymously purchase content from a content provider, and in the same time prevents any party from building usage profiles under a pseudonym. Petrlic also presented a privacy-preserving DRM scheme [21], which employs a combination of ring signatures with an anonymous recipient scheme. Secret sharing makes it possible for the content provider to expose the user identities in case of fraud. Perlman et al. proposed a privacy-preserving DRM conception on the basis of anonymous cash and blind decryption. Their scheme allows users to buy digital content without exposing their track [22]. Although these schemes are able to ensure data security, these schemes cannot support fine-grained access control, or limit a set of individual users to access encrypted data.

In order to solve these problems, Muller et al. proposed a new DRM architecture which limits the digital content access to a subset of users who possess certain properties assigned during the encryption process [17]. In their model, the set of rules are divided into two part, static

and dynamic. The static rules are enforced by using ABE before accessing the content, while dynamic rules stored in the license needs to be enforced at run time by the DRM viewer. However, revocation cannot be achieved in this scheme. What's worse is that, this scheme is not applicable for large numbers of users, which may be a huge burden for server.

As a result, attempting to achieve the revocation mechanism, the traditional revocation schemes relying on attribute authority would usually enforce periodically re-encrypt content, and re-generate a new secret keys to legal users as in [8, 30]. However, these schemes always results in key update operation. In practice, a large number of users can access cloud services. Hence, these schemes are far from have been suitable in cloud. Being aware of this problem, this paper follow schemes [4, 5], which permits the delegated key server to revoke the attributes and the malicious users immediately.

Thus, the ABE and PRE are usually employed to solve the problems above. Nevertheless, security and functional problems still exist. Additionally, besides [4, 5, 17], these schemes still suffer from the drawback of high computational cost associated with ABE operations; that is, both the computational cost and the ciphertext size for the users grow linearly with the size of the access formula. Hence, in this paper, we adopt a deterministic finite automata-based functional proxy re-encryption (DFA-based FPRE) scheme [13] to protect the contents stored in the semi-trusted cloud environment. Besides, outsourcing computation into (DFA-based FPRE)to avoid the high computation at the user side.

1.1 Motivation

Although using ABE and PRE can solve practical network problems, this method leaves interesting open problems in terms of security and functionality. As to security, it is not easy for ABE constructions [9, 10, 19, 20] to achieve adaptive security without random oracles [25]. Meanwhile, all existing attribute-based PRE (ABPRE) schemes [11, 12, 15] are proven secure only against chosen-plaintext attacks (CPA) in the selective model, whereas security against chosen ciphertext attack (CCA) is considered an important notion for ABPRE schemes. The functionality of an ABPRE system is another practical issue. All existing ABPRE schemes only support access policy combining with AND gates and fixed size number of boolean variables inputs. Practically, an access policy might be required to combine with AND, OR gates and NOT. Also, in some particular applications, the access policy might be expressed by regular languages with arbitrary size input data.

Referring to the research above, our DRM scheme is founded on a DFA-based FPRE scheme [13]. Having proved to be secure against CCA in the standard model, the DFA-based FPRE scheme also provides unlimited size input for access policy while the functionality of proxy re-encryption remains still.

In a nutshell, our contribution can be summarized as:

1) We propose a secure key management mechanism in DRM based on DFA-based FPRE scheme.

2) We introduce a fine-grained access control mechanism, which allows flexibility in specifying the access rights of individual users.

3) Our scheme provides a scalable revocation mechanism, which allows the delegated key server in the cloud to revoke the attributes and malicious users immediately.

4) We perform an analysis evaluation of our DFA-based FPRE DRM scheme.

The rest of this paper is organized as follows. In next Section the preliminaries required in this paper are presented. Our DRM scheme based on DFA-based FPRE scheme is presented in Section 3. Analysis of our scheme is discussed in Section 4. Finally, the conclusion is introduced in Section 5.

2 Preliminaries

Our scheme relies on a DFA-based functional proxy re-encryption scheme. We will briefly introduce the DFA-based functional proxy re-encryption scheme and the groups underlying our encryption scheme.

2.1 Composite Order Bilinear Groups

Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups of same order $N = P_1P_2P_3$ (where P_1, P_2, P_3 are distinct primes). We call a map $e : \mathbb{G}\times\mathbb{G} \longrightarrow \mathbb{G}_T$ bilinear if it should satisfy the following properties:

- Bilinear. $e(g^a, h^b) = e(g, h)^{ab}$, $\forall g, h \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}_{\mathbb{N}}{}^*$; and
- Non-degenerate. There exists $g \in \mathbb{G}$ such that $e(g, g)$ is a generator of $\mathbb{G}_T$.

We denote by G_{p_1}, G_{p_2} and G_{p_3} the subgroups of $\mathbb{G}$ of respective orders p_1, p_2 and p_3.

2.2 Complexity Assumptions

Definition 1. *(*The Source Group l-Expanded Bilinear Diffie-Hellman Exponent (l-Expanded BDHE.) Assumption in a Subgroup [13]*). Given a group generator $\mathscr{G}$ and a positive integer l, we define the following distribution:*

$$\begin{aligned}(N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e) &\xleftarrow{} \mathscr{G},\\ g_1 &\xleftarrow{R} \mathbb{G}_{p_1},\\ g_2 &\xleftarrow{R} \mathbb{G}_{p_2},\\ g_3 &\xleftarrow{R} \mathbb{G}_{p_3},\\ a, b, d, m, n, x, c_0, \cdots, c_{l+1} &\xleftarrow{R} \mathbb{Z}_N,\end{aligned}$$

Table 1: Notations in proposed scheme

Notion	Description	Notion	Description
K	security parameter	PP, MSK	public parameters and master key
U	user	CP	content provider
SP	service provider	LS	license server
CSP	cloud service provider	PK_U, SK_U	public and secret keys of user
PK_{CP}	public key of content provider	$RK_{M\to w}$	re-encryption key
CMK	content master key	AK	assistant key
CEK	content encryption key	CID	content identity
M	plain content data	CT	encrypted content data
UR	user rights	RE	rights expression
T	timestamp	$\sigma()$	signature algorithm
σ_{LS}	license acquisition request signature	σ_{KS}	key acquisition request signature
σ_L	license signature	SK_{CP}	secret key of content provider

$$
\begin{aligned}
D &= (N, \mathbb{G}, \mathbb{G}_T, e, g_1, g_2, g_3, g_2^a, g_2^b, g_2^{ab/dx}, g_2^{b/dx}, g_2^{ab/x}, \\
&\quad g_2^n, \forall i \in [0, 2l+1], i \neq l+1, j \in [0, l+1] g_2^{a^i mn}, \\
&\quad g_2^{a^i bmn/c_j x}, \forall i \in [0, l+1] g_2^{c_i}, g_2^{a^i d}, g_2^{abc_i/dx}, g_2^{bc_i/dx}, \\
&\quad \forall i \in [0, 2l+1], i \neq l+1, j \in [0, l+1] g_2^{a^i bd/c_j x}, \\
&\quad \forall i, j \in [0, l+1], i \neq j g_2^{a^i bc_j/c_i x}), \\
T_0 &= g_2^{a^{l+1} bm}, \\
T_1 &\xleftarrow{R} \mathbb{G}_{p_2}.
\end{aligned}
$$

The advantage of an algorithm $\mathscr{A}$ in breaking this assumption is $Adv_{\mathscr{A}}^{l-BDHE}(1^n) = | Pr[\mathscr{A}(D, T_0) = 1] - Pr[\mathscr{A}(D, T_1) = 1] |$. We say that $\mathscr{G}$ satisfies the l-Expanded BDHE Assumption if $Adv_{\mathscr{A}}^{l-BDHE}(1^n)$ is negligible for any PPT algorithm $\mathscr{A}$.

Definition 2. *(*The Source Group Modified q Bilinear Diffie-Hellman Exponent (q-BDHE) Assumption in a Subgroup [13]*.) Given a group generator $\mathscr{G}$, we define the following distribution:*

$$
\begin{aligned}
(N = p_1 p_2 p_3, \mathbb{G}, \mathbb{G}_T, e) &\longleftarrow \mathscr{G}, \\
g &\xleftarrow{R} \mathbb{G}_{p_1}, \\
g_2 &\xleftarrow{R} \mathbb{G}_{p_2}, \\
g_3 &\xleftarrow{R} \mathbb{G}_{p_3}, \\
c, a, e, f &\xleftarrow{R} \mathbb{Z}_N,
\end{aligned}
$$

$$
\begin{aligned}
D &= (N, \mathbb{G}, \mathbb{G}_T, e, g, g_2, g_3, g_2^e, g_2^a, g_2^{eaf}, g_2^{c+f/c}, g_2^{c^2}, \cdots, \\
&\quad g_2^{c^q}, g_2^{1/ac^q}), \\
T_0 &= g_2^{aec^{q+1}}, \\
T_1 &\xleftarrow{R} \mathbb{G}_{p_2}.
\end{aligned}
$$

The advantage of an algorithm $\mathscr{A}$ in breaking this assumption is $Adv_{\mathscr{A}}^{q-BDHE}(1^n) = | Pr[\mathscr{A}(D, T_0) = 1] - Pr[\mathscr{A}(D, T_1) = 1] |$. We say that $\mathscr{G}$ satisfies the Source Group Modified q-BDHE Assumption if $Adv_{\mathscr{A}}^{q-BDHE}(1^n)$ is negligible for any PPT algorithm $\mathscr{A}$.

2.3 A DFA-based Functional Proxy Re-encryption Scheme

For more details we refer the reader to [28] for the definition of DFA and DFA-based FE. The DFA-based functional proxy re-encryption scheme consists of the following seven algorithms [13]:

1) $(PP, MSK) \leftarrow Setup(1^n, \Sigma)$: The system setup algorithm takes a security parameter n and the description of a finite alphabet Σ as input. It outputs the public parameters PP and a master key MSK, where $n \in \mathbb{N}$. Here, we note that PP implicitly includes Σ.

2) $SK_M \leftarrow k.Gen(MSK, M = (Q, \tau, q_0, F))$: The key generation algorithm takes the master key MSK and a DFA description M as input. It outputs a private key SK_M, where Q is a set of states, τ is a set of transitions, $q_0 \in Q$ is a start state and $F \subseteq Q$ is a set of accept states.

3) $RK_{M\to w} \leftarrow ReKeyGen(SK_M, w)$: This algorithm takes SK_M for a DFA description M and an arbitrary length string $w \in \sigma$ as input. It outputs a re-encryption key $RK_{M\to w}$. Using the re-encryption key any ciphertext under a string $w^{'}$ (in which $ACCEPT(M, w^{'})$), it can converted to another ciphertext under w.

4) $CT \leftarrow DFA.E(PP, w, m)$: The encryption algorithm takes the public parameters PP, a message m and a $w \in \Sigma$ as input. It outputs the ciphertext CT under w.

5) $CT^R \leftarrow ReEnc(Rk_{M\to w}, CT)$: The encryption algorithm takes $Rk_{M\to w}$ and CT (under $w^{'}$). If $ACCEPT(M, w^{'})$, it outputs the ciphertext CT^R under w.

6) $m/\perp \leftarrow DFA.D(SK_M, CT)$: The decryption algorithm takes a secret key SK_M and ciphertext CT

(under w) as input. The decryption can be done if $ACCEPT(M, w)$, then it outputs a message m; otherwise, outputs an error symbol $\perp$.

7) $m/\perp \leftarrow DFA.D_R(SK_M, CT^R)$: The decryption algorithm takes a secret key SK_M and ciphertext CT^R (under w) as input. The decryption can be done if $ACCEPT(M, w)$, then it outputs a message m; otherwise, outputs an error symbol $\perp$.

3 Proposed Scheme

3.1 Security Requirements

We provide the requirements of our proposed scheme into two terms namely security and privacy, as follows:

1) **Efficient**: In cloud computing the user is expected to access various contents through multiple devices any time anywhere without limitation, and also looks for flexible usage model. Therefore, the DRM scheme in cloud computing should provide efficient license distribution models with low computational complexity to support huge number of users.

2) **Security**: The content provider is supposed to ensure that an authorized user is not able to extract and run the content. Also, content confidentiality against unauthorized users must be achieved. Meanwhile, server provider and license server must not be able to get the plain content and content key.

3) **Privacy preserving**: To realize the user privacy preserving, the user should stay anonymous towards the content provider that deals with user's content purchase and the license server that receives acquisition request. Therefore, neither content provider nor license server will be able to retrieve user's personal information, such as user identity, IP address, etc.

4) **Collusion-resistance**: The group of non revoked yet unauthorized users should not be able to pull together their information (DFA) to decrypt an encrypted content in that each of them is unable to decrypt it individually.

3.2 Basic DRM System Model

The basic architecture of DRM consists of seven entities as shown in Figure 1 and the notations are shown in Table 1.

1) **Cloud storage**: This entity provides a storage service based on cloud computing, which holds the encrypted contents from the content providers.

2) **Key server**: It is an entity that generates the public/private key pair for content provider and user, and keeps the encrypted content master key and assistant key issued by content provider. Further, key server re-encrypts the assistant key to the license server when users acquire to consume the content.

3) **Public authority**: This entity generates the public parameters PP and a master key MSK for the system. It also works as a key authority and issues secret keys associated with DFA to users. A key server is delegated by the public authority to perform a revocation task, revoking DFA of user and illegal users immediately. In addition, it allows flexibility in specifying the access rights of individual users according to their attributes description.

4) **Cloud service provider**: This entity keeps the encrypted content in the cloud storage. It is in charge of computing the transformed data, providing corresponding encrypted contents and license distribution to the user.

5) **License server**: This entity generates and distributes the license for authorized users whenever receiving the license acquisition from the CSP. The license includes the encrypted CMK.

6) **Content provider**: This is an entity that holds the digital contents and protect the contents from unauthorized user by encrypting their own contents with the content encryption key. Then, content providers outsource their encrypted contents to cloud storage provided by the CSP.

7) **User**: This is an entity that can get the encrypted content from the CSP. If a user owns the DFA that is satisfying the string w of the ciphertext, he will be able to recover the content encryption key. Then, he can decrypt and play the contents.

3.3 Intuition

In order to achieve a secure DRM scheme, we leverage the DFA-based FPRE scheme as the basic cryptographic tool and combine the outsourcing techniques and efficient revocation to tackle the focusing issues on efficiency, immediate revocation and fine-gained access control.

Our proposed construction operates as follows:

1) In system setup and key generation, the public authority generates the public parameters PP and a master key MSK. It also generates a user DFA and secret keys SK_M denoted as $SK_M = (M, \epsilon, \Omega, DK)$ for each user. Further, it also generates the re-encryption key $Rk_{M \to w}$ for authorized user and sends it to the key server in secure channel. Moreover, the content provider generates the CEK with random CMK and AK.

2) In content packaging and encryption, the content provider encrypts the contents C_x with content encryption key CEK_x, It then gets the encrypted contents in the following form:

$$E_{sym}(C_x|CEK_x), \quad \text{where} \quad x = 1, 2, 3, .., n$$

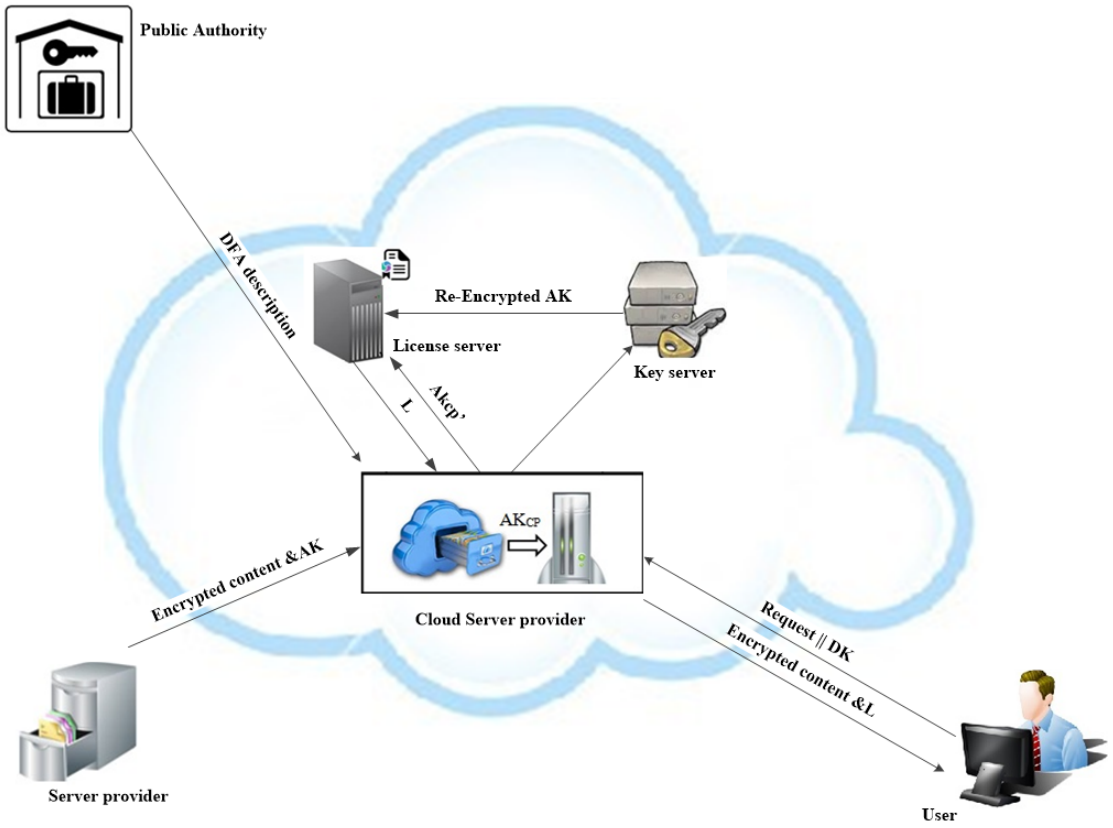

Figure 1: The system model of the proposed scheme system

Content provider later outsources the encrypted content to content server provider. Padding is employed to the contents before the encryption to make sure that each content has the same length. The sequence diagram of content encryption steps are shown in Figure 2.

3) In license acquisition, the user chooses the interesting content C_x with a unique identifier CID from the CSP, which is allowed to download the encrypted content. After downloading the DFC header from the content service provider, the user extracts the CMK_M from the DFC header and make sure that his/her DFA satisfy the string w of the content. A user cannot play the content without the valid license. Meanwhile, in order to acquire the license, the user first sends his partial decryption key DK to the content service provider, which is part of his private key. In fact, he just needs to send it once, unless his private key is regenerated, then content service provider transforms the ciphertext AK_{CP} to $AK_{CP}{}'$ and finally sends license acquisition request ($LSQ = \{CID \parallel UR \parallel T \parallel \sigma_{LS} \parallel AK_{CP}{}'\}$) including the user's rights UR, CID, T, $AK_{CP}{}'$ and $\sigma_{LS} = \sigma(SK_{CSP}, CID \parallel UR \parallel T \parallel AK_{CP}{}')$ to license server.

   Upon receiving the user's license acquisition request, the license server checks the signature σ_{LS} and T, and then acquires the assistant key from the key server. The key acquisition request includes $\{CID \parallel T \parallel \sigma_{KS}\}$, where $\sigma_{KS} = \sigma(SK_{LS}, CID \parallel UR \parallel T)$. The key server checks the signature σ_{KS} and T. Then, key server computes the re-encrypted (AK_{CP}^{R}) and sends it to the license server. After that, the license server generates the right expression RE from the UR according to the right expression language and also generates the license $L = \{CID \parallel RE \parallel AK_{CP}^{R} \parallel \sigma_L \parallel AK_{CP}{}'\}$, which includes content identity CID, AK_{CP}^{R}, $AK_{CP}{}'$, right expression UE and signature $\sigma_L = \sigma(SK_{LS}, CID \parallel AK_{CP}^{R} \parallel AK_{CP}{}' \parallel RE)$. Finally, the license server sends L to user through CSP. Upon receiving L, user checks the signature and keeps the license.

4) In content consumption, whenever a user want to play the content, the user will compute the content encryption key. Then, decrypts the content and play the content according to the usage rules in the license.

5) In revocation scheme, the public authority delegates the key server in the cloud to perform the DFA revocation and user revocation. The DFA revocation will revoke a user's one or more DFAs that he has possessed, which will not influence other users' DFA. However, the user revocation will revoke all of a user's

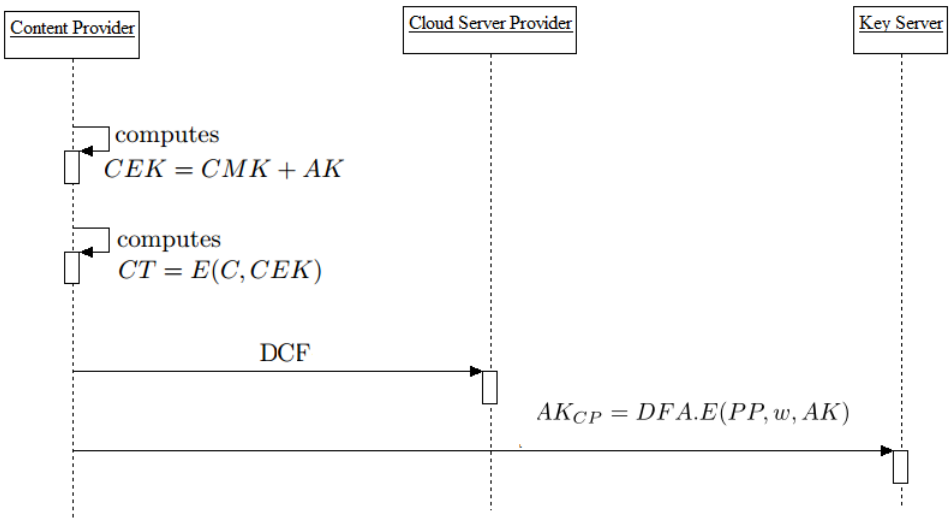

Figure 2: The sequence diagram of content encryption steps

DFA. The revocation scheme operates as follows: In DFA revocation, whenever a user's DFA revocation event is triggered and when the user intends to access the encrypted content, the public authority will inform the key server in the cloud to verify the user's DFA in advance. If the user's DFA cannot satisfy the string w of the encrypted content, the key server refuses to re-encrypt the assistant key for the user. Thus, the user cannot access the content.

6) In user revocation, whenever there is a revoked user intending to access the encrypted contents, the public authority will inform the key server in the cloud to refuse to re-encrypt the assistant key for the user. Hence, immediately after the revocation request is made, our scheme realizes the DFA and user revocations, and in both cases the unauthorized user will not get the licensee.

3.4 Concrete Construction

In this section, we will explain a detailed construction for the proposed scheme as follows:

3.4.1 System Setup and Key Generation

System setup. In this phase, The public authority runs the *Setup* algorithm to generate the public parameters PP and a master key MSK as the following:

$Setup(1^n, \Sigma)$ The setup algorithm selects random group elements $g, g_0, z, h_0 \in \mathbb{G}_{p_1}$ and randomly chooses an exponents $\alpha, k, a, b, \alpha_{End}, \alpha_{Start} \in \mathbb{Z}_N^*$. Then set $H_{Start} = g^{\alpha_{Start}}, H_{End} = g^{\alpha_{End}}$ and $H_k = g^k$. In addition, $\forall \sigma \in \Sigma$ it chooses random $\alpha_\sigma \in \mathbb{Z}_N^*$ and set $H_\sigma = g^{\alpha_\sigma}$. After that it choose a one-time symmetric encryption scheme $Sym = (sym.Enc, sym.Dec)$, a one-time signature scheme Ots and two target collision resistant (TCR) hash functions namely H_1 and H_2, where $H_1 : \mathbb{G}_T \longrightarrow \mathbb{Z}_N^*$ and $H_2 : \mathbb{G}_T \longrightarrow \{0,1\}^{poly(n)}$. Finally, the public authority publishes the public parameters $PP = \{e(g,g)^\alpha, g, g^{ab}, g_0, z, h_0, H_{Start}, H_{End}, H_k, \forall_{\sigma \in \Sigma} H_\sigma, Sym, Ots, H_1, H_2\}$ along with the description of the group $\mathbb{G}$ and the alphabet Σ, while the $MSK = (g^{-\alpha}, X_3)$ is kept secretly by the public authority. Here, X_3 is a generator of $\mathbb{G}_{p_3}$.

Key generation. The public authority runs the $k.Gen(MSK, M = (Q, \tau, q_0, F))$, the key generation algorithm takes the master key MSK and a DFA description M as input. It outputs a private key SK_M, where Q is a set of states $q_0, ..., q_{|Q|-1}$, τ is a set of transitions, for each transition $t \in T$ is a triple $(x, y, \sigma) \in Q \times Q \times \Sigma$. $q_0 \in Q$ is a start state and $F \subseteq Q$ is a set of accept states. The algorithm chooses random group elements $D_0, D_1, ..., D_{|Q|-1} \in \mathbb{G}_{p_1}$, where D_i is associated with state q_i, for each transition $t \in T$ it randomly selects $r_t \in \mathbb{Z}_N^*$, for all $q_x \in F$ it randomly selects $r_{End_x} \in \mathbb{Z}_N^*$, and selects $u \in \mathbb{Z}_N^*$. It also randomly selects $R_{Start_1}, R_{Start_2}, R_{Start_3}, R_{t,1}, R_{t,2}, R_{t,3}, R_{End_{x,1}}$, $R_{End_{x,2}} \in \mathbb{G}_{p_3}$, it also selects randoms $\epsilon, \Omega \in \mathbb{G}_T$ and random $r_{Start} \in \mathbb{Z}_N^*$. The algorithm computes the private key as follows.

Firstly it computes:

$$K_{Start_1} = D_0 \cdot (H_{Start})^{r_{Start}} \cdot R_{Start_1},$$
$$K_{Start_2} = g^{r_{Start}} \cdot R_{Start_2},$$
$$K_{Start_3} = g^u \cdot R_{Start_3}.$$

Secondly, for each transition $t = (x, y, \sigma) \in \tau$ it com-

putes:

$$K_{t,1} = D_x^{-1} \cdot z^{r_t} \cdot R_{t,1},$$
$$K_{t,2} = g^{r_t} \cdot R_{t,2},$$
$$K_{t,3} = D_y \cdot (H_\sigma)^{r_t} \cdot R_{t,3}$$

Thirdly, for all $q_x \in F$ the algorithm sets:

$$K_{End_{x,1}} = g^{-\alpha} \cdot D_x \cdot (H_{End} \cdot g^{ab})^{r_{End_x}} \cdot g^{ku} \cdot R_{End_{x,1}},$$
$$K_{End_{x,2}} = g^{r_{End_x}} \cdot R_{End_{x,2}}.$$

Finally, the key public authority generates the SK_M and sends it to the authorized user in secure channel. SK_M is denoted as: $SK_M = (M, \epsilon, \Omega, DK)$, where $DK = (DK_1 = {K_{Start_1}}^{H_1(\epsilon)}, DK_2 = {K_{Start_2}}^{H_1(\epsilon)}, DK_3 = {K_{Start_3}}^{H_1(\epsilon)}, \forall t \in \tau(DK_{t,1} = {K_{t,1}}^{H_1(\epsilon)}, DK_{t,2} = {K_{t,2}}^{H_1(\epsilon)}, DK_{t,3} = {K_{t,3}}^{H_1(\epsilon)}), \forall q_x \in F(DK_{End_{x,1}} = {K_{End_{x,1}}}^{H_1(\epsilon)}, DK_{End_{x,2}} = {K_{End_{x,2}}}^{H_1(\epsilon)}))$.

The public authority also runs the re-encryption algorithm $ReKeyGen(SK_M, w)$ to generate the re-encryption key $Rk_{M\to w}$ for authorized user and sends it to the key server in secure channel as follows:

Firstly, the public authority selects random $\beta_r \in \mathbb{Z}_N^*$ for all $q_x \in F$. Then computes $Rk_1 = K_{start_1}^{H_1(\Omega)}$, $Rk_2 = K_{start_2}^{H_1(\Omega)}$, $Rk_3 = K_{start_3}^{H_1(\Omega)}$, for all $t \in \tau(RK_{t,1} = K_{t,1}^{H_1(\Omega)}, RK_{t,2} = K_{t,2}^{H_1(\Omega)}, RK_{t,3} = K_{t,3}^{H_1(\Omega)})$, for all $q_x \in F(Rk_{End_{x,1}} = K_{End_{x,1}}^{H_1(\Omega)} \cdot H_{End}^{\beta_r}, Rk_{End_{x,2}} = K_{End_{x,2}}^{H_1(\Omega)} \cdot g^{\beta_r})$.

Finally, the $Rk_{M\to w} = (M, Rk_1, Rk_2, Rk_3, \forall t \in \tau(RK_{t,1}, RK_{t,2}, RK_{t,3}), \forall q_x \in F(Rk_{End_{x,1}}, Rk_{End_{x,2}}))$.

3.4.2 Content Packaging and Encryption

The content provider process in this phase are represented as follows:

- Firstly, the content provider computes CEK such as $CEK = CMK + AK$. Then using symmetric encryption algorithm encrypts the content such as $CT = E_{sym}(C, CEK)$.
- Secondly, the content provider encrypts the CMK using *sym.Enc* encryption algorithm and obtains the CMK_M as follows: $CMK_M = sym.Enc(H_2(CID), CMK)$, where the CID is the content identity. Then outsources the DCF to the CSP.
- Finally, the content provider encrypts the AK using $DFA.E(PP, w, AK)$ encryption algorithm and obtains the AK_{CP} as follows:

  The content provider randomly selects $\lambda_0, \lambda_1, ..., \lambda_l \in \mathbb{Z}_N^*$, run $(ssk, svk) \longleftarrow KeyGen(1^n)$ and computes AK_{CP} as

  First set: $C_{AK} = AK \cdot e(g,g)^{\alpha \cdot \lambda_l}$, $C_{Start_1} = C_{0,1} = g^{\lambda_0}$, $C_{Start_2} = (H_{Start})^{\lambda_0}$, $C_{Start_3} = (g_0^{svk} h_0)^{\lambda_0}$, for $i = 1$ to l, set: $C_{i,1} = g^{\lambda_i}, C_{i,2} = (h_{wi})^{\lambda_i} \cdot z^{\lambda_i - 1}$, finally, set:
  $C_{End_1} = C_{l,1} = g^{\lambda_l}$, $C_{End_2} = (H_{End} \cdot g^{ab})^{\lambda_l}$, $C_{End_3} = (H_k)^{\lambda_l}$,
  $C_{End_4} = Sign(ssk, (w, C_{AK}, C_{Start_1}, C_{Start_2}, C_{Start_3}, (C_{1,1}, C_{1,2}), ..., (C_{End_1}, C_{l,2}), C_{End_2}, C_{End_3}))$.

  The ciphertext AK_{CP} is
  $AK_{CP} = (svk, w, C_{AK}, C_{Start_1}, C_{Start_2}, C_{Start_3}, (C_{1,1}, C_{1,2}), ..., (C_{l,1}, C_{l,2}), C_{End_2}, C_{End_3}, C_{End_4})$.

3.4.3 License Acquisition

Upon receiving the user's license acquisition request, the key server computes the re-encrypted AK_{CP} with $Rk_{M\to w'}$ using the re-encryption algorithm $ReEnc(Rk_{M\to w'}, AK_{CP})$. The process in this phase represented as follows:

- The key server checks $verify(svk, (w, C_{AK}, C_{Start_1}, C_{Start_2}, C_{Start_3}, (C_{1,1}, C_{1,2}), ..., (C_{End_1}, C_{l,2}), C_{End_2}, C_{End_3})) = 1$ and $e(C_{Start_1}, g_0^{svk} h_0) = e(g, C_{Start_3})$, outputs "True" if valid and "False" otherwise. If the verification is hold then proceed.
- The string $w = (w_1, ..., w_l)$ is associated with the AK_{CP} and the DFA $M = (Q, \tau, q_0, F)$ is associated with the user's re-encryption key $Rk_{M\to w'}$ where $ACCEPT(M, w)$. There must exist a sequence of $l+1$ states $\mu_0, \mu_1, ..., \mu_l$ and l transitions $t_1, ..., t_l$ where $\mu_0 = q_0$ and $\mu_l \in F$, we have $t_i = (\mu_{i-1}, \mu_i, w_i) \in \tau$. The key server re-encrypts AK_{CP} as follows.
  - First computes:

    $$\begin{aligned} \phi_0 &= e(C_{Start_1}, Rk_1) \cdot e(C_{Start_2}, Rk_2)^{-1} \\ &= e(g, D_0)^{\lambda_0 \cdot H_1(\Omega)}. \end{aligned}$$

  - For $i = 1$ to l, computes:

    $$\begin{aligned} \phi_i &= \phi_{i-1} \cdot e(C_{(i-1),1}, Rk_{t_i,1}) \\ &\quad \cdot e(C_{i,2}, Rk_{t_i,2})^{-1} \cdot e(C_{i,1}, Rk_{t_i,3}) \\ &= e(g, D_{\mu_i})^{\lambda_i \cdot H_1(\Omega)}. \end{aligned}$$

    whenever M accepts w, we have that $\mu_l = q_x$ for some $q_x \in F$ and $\phi_l = e(g, D_x)^{\lambda_l \cdot H_1(\Omega)}$.
  - Then sets:

    $$\begin{aligned} \phi_{End} &= \phi_l \cdot e(C_{End_{x,1}}, Rk_{End_{x,1}})^{-1} \\ &\quad \cdot e(C_{End_{x,2}}, Rk_{End_{x,2}}) \cdot e(C_{End_{x,3}}, Rk_3) \\ &= e(g,g)^{\alpha \cdot \lambda_l \cdot H_1(\Omega)}. \end{aligned}$$

  - The key server selects random $\gamma \in \mathbb{G}_T$ and sets $\pi_1 = sym.Enc(H_2(\gamma), A)$ and $\pi_2 = DFA.E(PP, w', \gamma)$, where $A = (AK_{CP} \parallel \phi_{End})$. Finally, the key server sends $AK_{CP}^R = (\pi_1, \pi_2)$ to the license server.

3.4.4 Content Consumption

The user recovers the CMK from the ciphertext CMK_M as follows:

$$CMK = sym.Dec(H_2(CID), CMK_M) \quad (1)$$

The user whose the DFA associated with the his/her secret key accepts the string w will recover the AK as follows:

- Firstly, computes γ as follows: The user sends his partial decryption key DK to the cloud service provider for partial decryption, the cloud service provider works the following:

  If $verify(svk, (w, C_{AK}, C_{Start_1}, C_{Start_2}, C_{Start_3}, (C_{1,1}, C_{1,2}), \cdots, (C_{End_1}, C_{l,2}), C_{End_2}, C_{End_3})) = 1$ and $e(C_{Start_1}, g_0^{svk} h_0) = e(g, C_{Start_3})$, outputs "True" if valid and "False" otherwise. If the verification hold, proceed.

  Then cloud service provider computes $AK_{CP}{}'$ as follows:

  - $\theta_0 = e(C_{Start_1}, DK_1) \cdot e(C_{Start_2}, DK_2)^{-1} = e(g, D_0)^{\lambda_0 \cdot H_1(\epsilon)}$.
  - For $i = 1$ to l, compute:

    $$\begin{aligned} \theta_i &= \theta_{i-1} \cdot e(C_{(i-1),1}, DK_{t_i,1}) \\ &\quad \cdot e(C_{i,2}, DK_{t_i,2})^{-1} \cdot e(C_{i,1}, DK_{t_i,3}) \\ &= e(g, D_{\mu_i})^{\lambda_i \cdot H_1(\epsilon)}, \end{aligned}$$

    whenever M accepts w, we have that $\mu_l = q_x$ for some $q_x \in F$ and $\theta_l = e(g, D_x)^{\lambda_l \cdot H_1(\epsilon)}$.
  - Finally compute:

    $$\begin{aligned} \theta_{End} &= \theta_l \cdot e(C_{End_{x,1}}, DK_{End_{x,1}})^{-1} \\ &\quad \cdot e(C_{End_{x,2}}, DK_{End_{x,2}}) \\ &\quad \cdot e(C_{End_{x,3}}, DK_3) \\ &= e(g, g)^{\alpha \cdot \lambda_l \cdot H_1(\epsilon)} \end{aligned}$$

    and sends the message $AK_{CP}{}' = \theta_{End}$ to the user within his licence. The user then can retrieve γ as follows:

    $$\gamma = C_\gamma / \{AK_{CP}{}'\}^{H_1(\epsilon)^{-1}}$$

    We note that $C_\gamma = \gamma \cdot e(g, g)^{\alpha \cdot \lambda_l}$, since it encrypted using the $DFA.E(PP, w, m)$ encryption algorithm.

- Secondly, the user computes A as follows:

  $$A \longleftarrow sem.Dec(H2(\gamma), \pi_1)$$

  Where $A = (AK_{CP} \parallel \phi_{End})$.

- Thirdly, computes Key as follows:

  $$Key = \phi_{End}^{H_1(\Omega)^{-1}}$$

- Finally, outputs the AK as follows:

  $$AK = C_{AK} / Key$$

Then if the user's UR are effective, the user can computes the CEK such as:

$$CEK = CMK + AK$$

Finally the user decrypts the encrypted content and plays the content according to the RE in the license.

$$C = D(CEK, CT)$$

4 Analysis

4.1 Correctness

The correctness of our scheme is extremely straightforward. After downloading the DFC header from the content service provider, the user extracts the CMK_M from the DFC header. The CMK_M is a ciphertext of using a $sym.Enc$ encryption algorithm. So, the user recovers the CMK from the ciphertext CMK_M as follows:

$$CMK = sym.Dec(H_2(CID), CMK_M)$$

In the license acquisition phase, the key server re-encrypts the AK_{CP} to the AK_{CP}^R without disclosing the AK. Moreover, AK_{CP} is converted to a re-encrypted ciphertext AK_{CP}^R under w. Thus, if the user has been granted the right DFA, the user should own the corresponding secret key for decryption (SK_M). Hence, only the user can recover the plain text of AK with the private key SK_M as follows:

$$\begin{aligned} AK &= C_{AK} / Key \\ &= AK \cdot e(g, g)^{\alpha \cdot \lambda_l} / \phi_{End}^{H_1(\Omega)^{-1}} \\ &= AK \cdot e(g, g)^{\alpha \cdot \lambda_l} / \{e(g, g)^{\alpha \cdot \lambda_l \cdot H_1(\Omega)}\}^{H_1(\Omega)^{-1}} \\ &= AK \end{aligned}$$

Therefore, with the possibility of recovering the CMK and AK, the user can surely decrypt the content.

4.2 Security

Our proposed scheme relies on K. Liang et al.'s scheme [26]. It has been proven to be secure in the standard model, and it seems suitable for DRM system. Therefore, we focus on the following theorems.

Theorem 1. *It is only feasible for an authorized user to access the contents.*

Proof. As we presented, the user who has a matching of the DFA and effective usage rights only can decrypt the content. Hence, our scheme provides access of contents only for authorized users. The proposed scheme ensures

Table 2: Efficiency comparison

Scheme	**Fine-grained access control**	**Revocation scheme**	**Privacy preserving**	**Access policy size input**	**Complexity of content decryption**
ref[23]	No	No	Yes	None	$8T_a + T_r$
ref[17]	Yes	No	N/A	Limited	$T_b + T_{dec}$
ref[4]	Yes	Yes	Yes	Limited	$T_b + T_a + T_{dec}$
ref[5]	Yes	Yes	Yes	Limited	$T_b + T_{dec} + 2T_{exp}$
our scheme	Yes	Yes	Yes	Unlimited	$3T_{dec} + 2T_{exp}$

Table 3: Efficiency comparison against Huang's scheme

	Operations								Total running time m/s
	Pairing		Pairing-based scalar multiplication		Symmetric decryption		Exponential operation		
	Number	Running time	Number	Running time	Number	Running time	Number	Running time	
Ref. [5]	3	61.2	4	25.52	1	3.04	2	21.28	111.04
Our	0	0	0	0	3	9.12	2	21.28	30.4

confidentiality of the content from the following four aspects. Firstly, if the unauthorized user can retrieve the desired value $e(g,g)^{\alpha \cdot \lambda_t}$, which is required for the decryption into two cases, the DFA revocation and the user revocation, he also cannot recover the AK. In fact, if the user's DFA cannot satisfy the string w of the encrypted content, the key server will refuse to re-encrypt the assistant key for the user. On the other hand, when a user is revoked, he cannot recover AK without the re-encrypt the assistant key part in the user revocation case. Therefore, unauthorized user cannot recover the AK. Secondly, the license server cannot get the content master key. Thirdly, the key server cannot get the plain assistant key. Fourthly, the curious CSP cannot read the contents without the content encryption key since any of private keys is not given to the CSP from the content provider in our scheme. On the other hand, even if CSP colludes with some user transform the ciphertext AK_{CP} to $AK_{CP}{}'$ and obtain $AK_{CP}{}'$ using the user's partial decryption key DK, he still cannot recover the AK, because he does not know the secret values ϵ and Ω. Therefore, neither a curious CSP nor unauthorized users or license server in the cloud can read the contents. □

Theorem 2. *It is infeasible for an illegal user to get the license from the malicious employees of license server.*

Proof. In the license acquisition phase, license server only can receive the re-encrypted ciphertext AK_{CP}^{R} . Therefore, malicious employees of license server cannot issue license to illegal user without the full content encryption key. □

Theorem 3. *It is infeasible for an attacker to replay license acquisition request and key acquisition request.*

Proof. In our scheme, the service provider sends license acquisition request $LSQ = \{CID \parallel UR \parallel T \parallel \sigma_{LS} \parallel AK_{CP}{}'\}$ to the license server. Upon receiving the user's license acquisition request, the license server checks the signature σ_{LS} and T. If the adversary E can modify it to $LSQ' = \{CID \parallel UR \parallel T' \parallel \sigma_{LS} \parallel AK_{CP}{}'\}$ and send LSQ' to the license server, the license server concludes that $T' \neq T$, and rejects the request. Thus, the replaying license acquisition request is impossible.

In the same way, the license server sends key acquisition request $\{CID \parallel T \parallel \sigma_{KS}\}$ to the key server. Upon receiving the user's key acquisition request, the key server checks the signature σ_{KS} and T. If the adversary E can modify it to $\{CID \parallel T' \parallel \sigma_{KS}\}$ and send it to key server, the key server concludes that $T' \neq T$, and rejects the request. Hence, the replaying key acquisition request is infeasible. □

Theorem 4. *Our key construction mechanism is considered to be secure due to the collusion-resistant for the users.*

Proof. Using symmetric encryption algorithm, the CMK is protected and distributed within the encrypted content. On the other hands, the AK is protected using the DFA-based FPRE encryption algorithm and then stored in the key server. The users who fulfill the string w can get the plain text of the content master key, and then obtain assistant key from key server when they intend to play the

contents. We showed that the CEK is not available for more than one user to collude or access. The conspiring users can retrieve the desired values $e(g,g)^{\alpha \cdot \lambda_l \cdot H_1(\epsilon)}$ and $e(g,g)^{\alpha \cdot \lambda_l \cdot H_1(\Omega)}$ in order to recover the required value $e(g,g)^{\alpha \cdot \lambda_l}$ for decryption operation. However, the two values ϵ and Ω are random and unique exponents for each user, which renders the combination of information in different users' secret keys meaningless. Even if the users obtain the CMK, they cannot compute the CEK, since the CEK is computed by adding the CMK and AK. In this sense, such collusion attack can be precluded in our construction scheme. □

4.3 Privacy Preserving

In our scheme, an anonymous user directly communicates with the cloud service provider and key server, which prevents the other parties from getting any user's personal information, for example, which software is bought and who bought the software. In the key generation phase, an anonymous user register to the CSP and then get it's public/private key issued by key server and secret keys SK_M issued by public authority. In the content decryption phase, the user acquires AK from the key server without giving out any personal information. As a result, the user's privacy is maintained.

4.4 Performance Analysis

For convenience, in this section we define the following notations: T_H (the time complexity of one-way hash function); T_e (the time complexity of pairing operation); T_r (the time complexity of proxy re-encryption); T_{mul} (the time complexity of pairing-based scalar multiplication); T_{exp} (the time complexity of exponential operation); T_b (represents the attribute-based encryption); T_{dec} (the time complexity of symmetric decryption); T_a (represents the asymmetric encryption).

We compare our scheme with existing DRM schemes, in terms of access control, revocation scheme, the access policy size input and privacy preserving. The results are given in Table 2. It is easy to find that Petrlic et al.'s scheme [23] has higher computational cost than our scheme since their scheme uses eight times asymmetric encryption operations at the user side. Whereas, it does not provide a revocation method. For the other related attribute-based encryption DRM schemes [4, 5, 17], the encryption costs are almost the same, which increases linearly with the number of attributes used in the data encryption. However, the decryption cost for the user in our scheme is much less than these schemes, which just includes two modular exponentiation operations and three symmetric decryption operations. Therefore, our scheme has much better efficiency.

In Table 3 the efficiency comparison of our scheme against Huang et al. [5] scheme which has concrete construction is presented. This comparison is prepared based on experimental results in [6, 7], for various cryptographic operations using MIRACLE [18] in PIV 3 GHZ platform processor with memory 512 MB and the Windows XP operating system. From these experimental results, the relative running time of one pairing operation T_e is 20.04 m/s, one-way hash function T_h is 3.04 m/s, pairing-based scalar multiplication T_{mult} is 6.38 m/s and exponential operation T_{exp} is 10.64 m/s, where the symmetric key encryption and decryption running times are very close to hash function running time [26, 29]. Hence, we adopted the same running time of one-way hash function for both encryption T_{enc} and decryption T_{dec}.

Let N_u, N_c and N_a denote the number of users, contents, and attributes respectively. In ABE schemes, the computational cost increases linearly either to $N_u \cdot N_a$ or $N_c \cdot N_a$, but never linearly to the product of the three $N_u \cdot N_c \cdot N_a$ [2, 3]. In our scheme, the computational costs in the content consumption phase at the user side is $3T_{dec} + 2T_{exp}$.

As indicated in Table 3, the computational cost of Huang et al.'s scheme is increasingly higher. Furthermore, the decryption cost in this scheme increases linearly with the number of attributes. Moreover, it requires three times bilinear pairing operation. However, the time consumed in pairing operation is more than other operations over elliptic curve group. Finally, Figure 3 shows the efficiency comparison of our scheme versus Huang et al. based on running time for each operation.

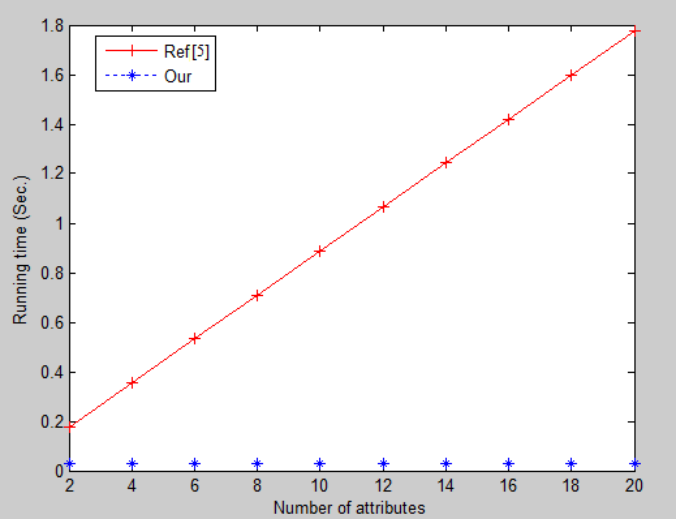

Figure 3: The efficiency comparison against Huang's scheme

5 Conclusions

Based on DFA-based FPRE scheme, we proposed a secure, efficient, and fine-grained access control system for DRM system. Furthermore, we put forward a mechanism for distributing licenses in a flexible and secure manner. In our scheme, the user who has a DFA associated with

his/her secret key accepts the string associated with the ciphertext and has effective usage rights can quite efficiently access the encrypted content with the help of the cloud service provider, and the revocation is both flexible and fine-grained. Moreover, the revocation task can be made immediately without disclosing the ciphertext. Finally, comparing with other DRM schemes in cloud computing, it is safe to draw the conclusion that our present work could be considered a secure and high efficient work for DRM system.

Acknowledgments

The author would like to acknowledge National Natural Science Foundation of China under Grant (No. 61003230, 61370026 and 61202445), the Fundamental Research Funds for the Central Universities under Grant (No. ZYGX2013J073 and ZYGX2012J067).

References

[1] H. Abdalla, X. Hu, A. Wahaballa, P. Avornyo and Q. Zhiguang, "Anonymous pairing-free and certificateless key exchange protocol for DRM system," *International Journal of Network Security*, vol. 18, no. 2, pp. 235–243, 2016.

[2] J. Camenisch, M. Dubovitskaya, G. Neven, G. M. Zaverucha, "Oblivious transfer with hidden access control policies" in *Public Key Cryptography*, LNCS 6571, pp. 192–209, Springer, 2011.

[3] J. Camenisch, M. Dubovitskaya, R. R. Enderlein and G. Neven, "Oblivious transfer with hidden access control from attribute-based encryption," in *Conference on Security and Cryptography for Networks,* PP. 1–32, 2012.

[4] Q. Huang, Z. Ma, J. Fu, X. Niu and Y. Yang, "Attribute based DRM scheme with efficient revocation in cloud computing," *Journal of Computers,* vol. 8, no. 11, pp. 2776–2781, 2013.

[5] Q. Huang, Z. Ma, Y. Yang, X. Niu and J. Fu, "Attribute based DRM scheme with dynamic usage control in cloud computing," *Communications,* vol. 11, no. 4, pp. 50–63, 2014.

[6] D. He, J. Chen and R. Zhang, "An efficient identity-based blind signature scheme without bilinear pairings," *Computer Electrical Engineering,* vol. 37, no. 4, pp. 444–450, 2011.

[7] D. He and J. Chen, "An efficient certificateless designated verifier signature scheme," *International Arab Journal of Information Technology,* vol. 10, no. 4, pp. 389–396, 2013.

[8] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in *Information Security Applications,* pp. 309–323, Springer Berlin Heidelberg, 2009.

[9] A. Lewko, T. Okamoto, A. Sahai, K. Takashima and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology (Eurocrypt'10),* pp. 62–91, Springer Berlin Heidelberg, 2010.

[10] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *Advances in Cryptology (Crypto'12)*, pp. 180–198, Springer Berlin Heidelberg, 2012.

[11] X. Liang, Z. Cao, H. Lin and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *ACM Proceedings of the 4th International Symposium on Information, Computer, and Communications Security,* pp. 276–286, 2009.

[12] S. Luo, J. Hu and Z. Chen, "Ciphertext policy attribute-based proxy re-encryption," in *Information and Communications Security,* pp. 401–415, Springer Berlin Heidelberg, 2010.

[13] K. Liang, M. Au, J. Liu, W. Susilo, D. Wong, G. Yang, P. Tran and Q. Xie, "A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing", *IEEE Transactions on Information Forensics and Security,* vol. 9, no. 10, pp. 1667–1680, 2014.

[14] A. Lewko, T. Okamoto, A. Sahai, K. Takashima and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology (Eurocrypt'10),* pp. 62–91, Springer Berlin Heidelberg, 2010.

[15] T. Mizuno and H. Doi, "Hybrid proxy re-encryption scheme for attribute-based encryption," in *Information Security and Cryptology,* pp. 288–302, Springer Berlin Heidelberg, 2010.

[16] Z. Ma, K. Fan, M. Chen, Y. Yang and X. Niu, "Trusted digital rights management protocol supporting for time and space constraint," *Journal on Communications*, vol. 29, no. 10, pp. 153–164, 2008.

[17] S. Muller, S. Katzenbeisser, "A new DRM architecture with strong enforcement," in *Proceedings of the 5th International Conference on Availability, Reliability, and Security,* pp. 397–403, 2010.

[18] MIRACL, Multiprecision Integer and Rational Arithmetic C/C++ Library, 2016. (`http://indigo.ie/mscott/`)

[19] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Advances in Cryptology (Crypto'10),* pp. 191–208, Springer Berlin Heidelberg, 2010.

[20] T. Okamoto and K. Takashima, "Fully secure unbounded inner-product and attribute-based encryption," in *Advances in Cryptology (Asiacrypt'12)*, pp. 349–366, Springer Berlin Heidelberg, 2012.

[21] R. Petrlic and C. Sorge, "Privacy-preserving DRM for cloud computing," in *IEEE 26th International*

Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 1286–1291, 2012.

[22] R. Perlman, C. Kaufman, R. Perlner, "Privacy-preserving DRM," in *Proceedings of the 9th Symposium on Identity and Trust on the Internet,* pp. 69–83, 2010.

[23] P. Ronald, "Proxy re-encryption in a privacy-preserving cloud computing DRM scheme," in *Cyberspace Safety and Security,* pp. 194–211, Springer Berlin Heidelberg, 2012.

[24] M. Ruiz, D. M. L and J. Pedraza, "Privacy risks in cloud computing," in *Intelligent Agents in Data-intensive Computing,* Springer International Publishing, pp. 163–192, 2016.

[25] S. C. Ramanna, "DFA-based functional encryption: Adaptive security from dual system encryption," *IACR Cryptology ePrint Archive*, vol. 2013, pp. 638, 2013.

[26] S. Ramesh, V. M. Bhaskaran, "An improved remote user authentication scheme with elliptic curve cryptography and smart card without using bilinear pairings," *International Journal of Engineering and Technology (IJET),* vol. 5, no. 6, 2014.

[27] A. Wahaballa, Z. Qin, H. Xiong, Z. Qin and M. Ramadan, "A taxonomy of secure electronic english auction protocols," *International Journal of Computers and Applications,* vol. 37, no. 1, pp. 28–36, 2015.

[28] B. Waters, "Functional encryption for regular languages," in *Advances in Cryptology (Crypto'12),* LNCS 7789, pp. 218–235, Springer, 2012.

[29] D. Wang, D. He, P. Wang, et al., "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing,* vol. 12, no. 4, pp. 428–442, 2014.

[30] S. Yu, C. Wang, K. Ren and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *IEEE International Conference on Computer Communications,* pp. 1–9, 2010.

[31] Z. Zhang, Q. Pei, J. Ma, and L. Yang, "Establishing multi-party trust architecture for DRM by using game-theoretic analysis of security policies," *Chinese Journal of Electronics*, vol. 18, no. 3, pp. 519–524, 2009.

Hisham Abdalla is a doctorial student at University of Electronic Science and Technology of China (UESTC). He received his M.Sc. degree from UESTC and BE degree in computer engineering from Karary University in 2006. His research interests include cloud computing security, cryptography and digital right management.

Xiong Hu is an associate professor in the School of Information and Software Engineering, UESTC. He received his Ph.D. degree from UESTC in 2009. His research interests include: information security and cryptography.

Abubaker Wahaballa received his PhD. degree from University of Electronic Science and Technology of China. His current research interests include information security, cryptography, steganography and DevOps.

Ahmed Abdalla was born in 1982. He received the B.S. Degree in Electrical Engineering from Karary University in 2005, Khartoum Sudan and the M.Sc. in Electronic Engineering, Information and signal processing form University of Electronic Science and Technology of China in 2013, Chengdu, China. He is currently working toward the PhD. Degree Electronic Engineering from University of Electronic Science and Technology of China. His current research interests include radar counter countermeasure and radar signal processing.

Mohammed Ramadan received the B.S. Degree in Communication Engineering from Karary University in 2007, Khartoum Sudan and the M.S. Degree in M.Sc. in Computer Engineering, Information Security form University of Electronic Science and Technology of China in 2013, Chengdu, China. He is currently working toward the PhD. degree in Information Security, Mobile Communication Security from University of Electronic Science and Technology of China. His current research interests include Wireless and Mobile Communications security (LTE security).

Qin Zhiguang is a professor at University of Electronic Science and Technology of China (UESTC). Research interest: network security, social network. He has published more than 100 papers on international journals and conference among which more than 50 are indexed by SCI and EI. He has been principal investor of 2 NSF key projects, 2 sub-topics of national major projects and 6 national 863 projects.