Anonymous Pairing-Free and Certificateless Key Exchange Protocol for DRM System

Hisham Abdalla[1], Xiong Hu[1], Abubaker Wahaballa[1], Philip Avornyo[2] and Qin Zhiguang[1]
(Corresponding author: Hisahm Abdalla)
School of Information and Software Engineering, University of Electronic Science and Technology of China[1]
School of Management Science and Engineering, University of Electronic Science and Technology of China[2]
2006 Xiyuan Avenue, Gaoxin West Zone, Chengdu 611731, China.
(Email: hisham_awaw@hotmail.com)
(Received Feb. 13, 2015; revised and accepted May 5 & June 29, 2015)

Abstract

Mostly, current security architectures for Digital rights management (DRM) systems use either Public Key Cryptography (PKC) or Identity-based Public Key Cryptography (ID-PKC). However, PKC has a complex certificate management and ID-PKC has a key escrow problem. Certificateless Public Key Cryptography (CL-PKC) has some attractive properties which seem compatible with the requirements of DRM systems. In this paper, we present anonymous pairing-free certificateless authenticated key exchange (CL-AKE) protocol for DRM system which provides a mechanism for distributing licenses in a flexible and secure manner. Furthermore, the analyses demonstrate that our scheme is efficient and secure.

Keywords: Anonymity, authentication, certificateless public key cryptography, digital rights management

1 Introduction

Digital rights management (DRM) is a famous mechanism for protecting content copyright [18]. Current DRM systems mostly encrypt the digital contents with a content-key from the content providers first. They then provide licenses to the users. The licenses authorize the users to play the digital contents according to the usage rights in the license. Consequently, illegal copies of the content are available over the network which causes a significant loss of revenue to the right holders. In preventing other users from using digital content file without content-key, the existing DRM mechanisms need to manage content/content-key on a server provider and to provide encrypted content-key with the user-key for the user. This mechanism also ensures the server provider manages all user's licenses, manages encrypted digital content files and protects copyrights against unlawful content distribution.

Ideally, DRM systems should also be able to provide flexible and secure content distribution mechanisms. For the purpose of resolving the above loopholes in DRM systems, it is necessary to apply an efficient mutual authentication and key agreement protocol. In this case the concerned parties can authenticate each other and create a secure session key. The session key is established with the information shared by the concerned parties which is used to achieve its purpose of confidentiality and data integrity.

The existing DRM systems mostly rely on two approaches. The first approach is the Public Key Cryptography (PKC) [17]. In this approach, the schemes apply PKC to authenticate public key [3, 11]. The PKC manages a Certificate Authority (CA). CA authenticates the concerned parties and their public key. Furthermore, it administrates certificate management involving distribution, storage and revocation. However, CA becomes infeasible because it suffers from a huge computational cost of certificate verification especially for a large network. The second approach, on the other hand, is referred to as Identity based Public Key Cryptography (ID-PKC) [1]. The schemes in this approach [12, 13, 14] use an identity based infrastructure where concerned parties get their full private key from Private Key Generator (PKG). Public key is then generated from their public identity using an email address or a physical IP address. Another scheme proposed also uses an identity based authenticated key agreement protocol which manages secure communication [15]. However, this scheme suffers from the key escrow problem, because the PKG knows the full private key of each user. This implies PKG can easily break the user privacy. Mishra et al. [7] proposed certificateless authenticated key agreement protocol for DRM system using the elliptic curve bilinear pairings. Since the pairing over elliptic curve is regarded as one of the highly expensive cryptography primitives [10], the use of such pairings makes the scheme [7] less applicable in practical applications, even secure in standard model. Therefore, to improve the efficiency of Mishra's scheme, we propose anonymous pairing-free CL-AKE protocol for DRM system, that does

not depend on the pairings and based on ECC. Elliptic Curve Cryptography (ECC) is commonly used for highly secure authentication protocols [9], because it's more applicable from the efficiency point of view.

This paper introduces anonymous pairing-free CL-AKE protocol for DRM system. Our scheme can eliminate the use of trusted certificate authority, solve key escrow problem and avoid the high computation of pairings operation. Furthermore, the symmetric key encryption is adopted in our scheme. This reduces computational costs and communication overheads significantly compared with public key encryption.

The rest of this paper is organized as follows: In next Section the preliminaries required in this paper are presented. Our anonymous pairing-free CL-AKE for DRM system is presented in Section 3. Section 4 presents the security analysis and performance evaluation of our scheme. Finally, the conclusion is introduced in Section 5.

2 Preliminaries

Our scheme relies on a certificateless authenticated key agreement protocol. We will briefly introduce the basic DRM System, the basic definitions and some properties related to this technique.

2.1 Basic DRM System

The basic architecture of DRM consists of four parts: content provider, content server, license server and user.

1) **Content Provider**: This is an entity that holds the digital content and protects the content from unauthorized user.

2) **Content Server**: It is an entity that keeps the encrypted content over the storage server and provides the encrypted content to user.

3) **License Server**: It is an entity which generates and distributes the licenses for authorized users.

4) **User**: This is an entity that wants to get the encrypted content from content server and acquires the license from license server.

2.2 Background

Elliptic Curve (EC): An elliptic curve E over a prime finite field $\mathbb{F}_P$ denoted as $E/\mathbb{F}_P$ satisfies an equation of the form.

$$y^2 = x^3 + ax + b \quad a, b \in \mathbb{F}_P \quad \text{and} \quad 4a^3 + 27b^2 \neq 0. \quad (1)$$

The condition that $4a^3 + 27b^2 \neq 0$ implies that the $\mathbb{F}_P$ is non-singular. Our scheme is based on the following computational assumptions:

1) Computational Diffie-Hellman Problem (CDH): Suppose G is a cyclic group of a prime order P. For a given generator P of G and $\{P, aP, bP\} \in G$, where $a, b \in \mathbb{Z}_P$, computing abP is hard.

2) Elliptic Curve Discrete Logarithm Problem (ECDLP): Suppose an elliptic curve E over a prime finite field $\mathbb{F}_P$, a point $P \in E(\mathbb{Z}_P)$ of order n, and a point $Q \in \langle P \rangle$. To find the integer $k \in [0, n-1]$ such that $Q = kP$ is hard.

2.3 AL-Riyami and Paterson CL-AKA Scheme

In 2003, Al-Riyami and Paterson [2] proposed certificateless public key cryptography (CL-PKC) to successfully remove the necessity of certification using user-chosen secret information. Certificateless public key cryptography is an intermediary between identity-based and traditional PKI-based cryptography. A generic two-party CL-AKE scheme consists of two phases. The first phase is the setup which runs between KGC (Key Generator Center) and entities. It includes the following five Probabilistic Polynomial Time (PPT) algorithms: Setup, Partial-Private-Key-Extract, Set-Secret-Value, Set-Private-Key and Set-Public-Key. The second phase is the key agreement phase which runs between two entities and depends on session key agreement algorithm.

2.4 System Model

Problem Statement. Users usually purchase software licenses from license server. They also might have downloaded copies of the encrypted software from the server providers. It is necessary to provide flexible and secure content distribution mechanism to protect both the software providers' intellectual property rights and users' privacy.

Architecture and Basic Approach. The architecture and approach of our DRM system are shown in Figure 1. The user and license server first register in the server provider (server provider act as PKG) and obtain their corresponding partial private keys. They then compute their own public/private keys. After this process user can anonymously acquire a license for a software from the license server. To execute the software, the user will decrypt the encrypted license using the session key (k) provided and obtain the valid license.

Assumptions. We assume that none of the parties, i.e. service provider, software provider, and license server can get any user's personal information like which software is bought and who buys the software.

Requirements.

1) Content Protection: Digital contents should be encrypted, and then the encrypted contents and

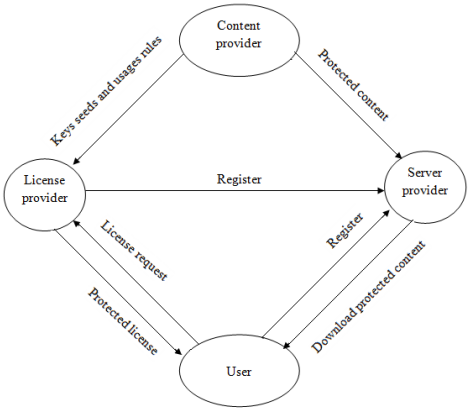

Figure 1: Architecture and basic approach of our DRM system

encrypted licenses are separately distributed by server provider.

2) DRM Security: The content provider expects that an authorized user must not be able to play the content. Also content confidentiality against unauthorized users must be created. Meanwhile, server provider and license server must not be able to obtain the plain content and content key.

3) User Privacy/Anonymity: User privacy means the protection of user's personal identification information (PII) [8]. To realize the user privacy 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 can retrieve user's personal information, such as user identity, IP address, etc.

3 Proposed Protocol

In Section 3.1, we present our anonymous pairing-free CL-AKE for DRM system. Furthermore, the steps in implementing this scheme for DRM system is provided in Section 3.2.

3.1 The Anonymous Pairing-Free CL-AKE

In this paper, we propose anonymous pairing-free CL-AKE protocol for DRM system based on Xiong. et. al.'s protocol [19], it has been proven to be secure in the mBR model and it seems suitable for DRM system. To achieve the user anonymity in the key-agreement phase, we use pseudonym instead of sending the real identity of the entities. It allows a user to generate a session key with the license server in anonymous way.

3.2 Implementation Steps of Our Scheme for DRM System

The content provider encrypts the content with a content encryption key (K_{CE}). The content provider then outsources the encrypted content to the service provider and provides the content encryption key with usage rules to the license server. Whenever a user initiates a buying process, the license server authenticates the user, receives the payment, and generates the license. The license server then sends the license through the service provider to the user. Our DRM system consists of the following four parties:

- Private key Generator PKG;
- Content Provider CP;
- Service Provider SP;
- License Server LS;
- User U.

We define the proposed scheme by describing the following four phases:

- **Key Generation**: In this phase the service provider acts as a Private key Generator (PKG) for our anonymous pairing-free CL-AKE protocol. PKG generates the system public key and system master key, while both U and LS compute their public keys and full private keys.
- **Content Packaging**: Content provider generates a set of symmetric keys as content encryption keys. Content provider then encrypts the content with content encryption key, and outsources the encrypted content to service provider. Padding is employed to the software before encryption.
- **License Acquisition**: The user chooses the right content from the service provider which is allowed to download the encrypted content. A user cannot use the software without the valid license. Meanwhile, in order to acquire the license, a user needs to establish a secured communication from the license server by using an anonymous pairing-free CL-AKE protocol with license server.
- **Content Consumption**: Whenever a user wants to use the content, the user will decrypt the message using session key k and get the valid license.

Next, the algorithms of the four phases of the proposed scheme are shown in Figure 2 and the following:

1) **Key Generation**: In this phase, the system uses five algorithms: Setup, Partial private key extract, Set secret value, Set private key and Set public key. Illustration of key generation phase is as follows:

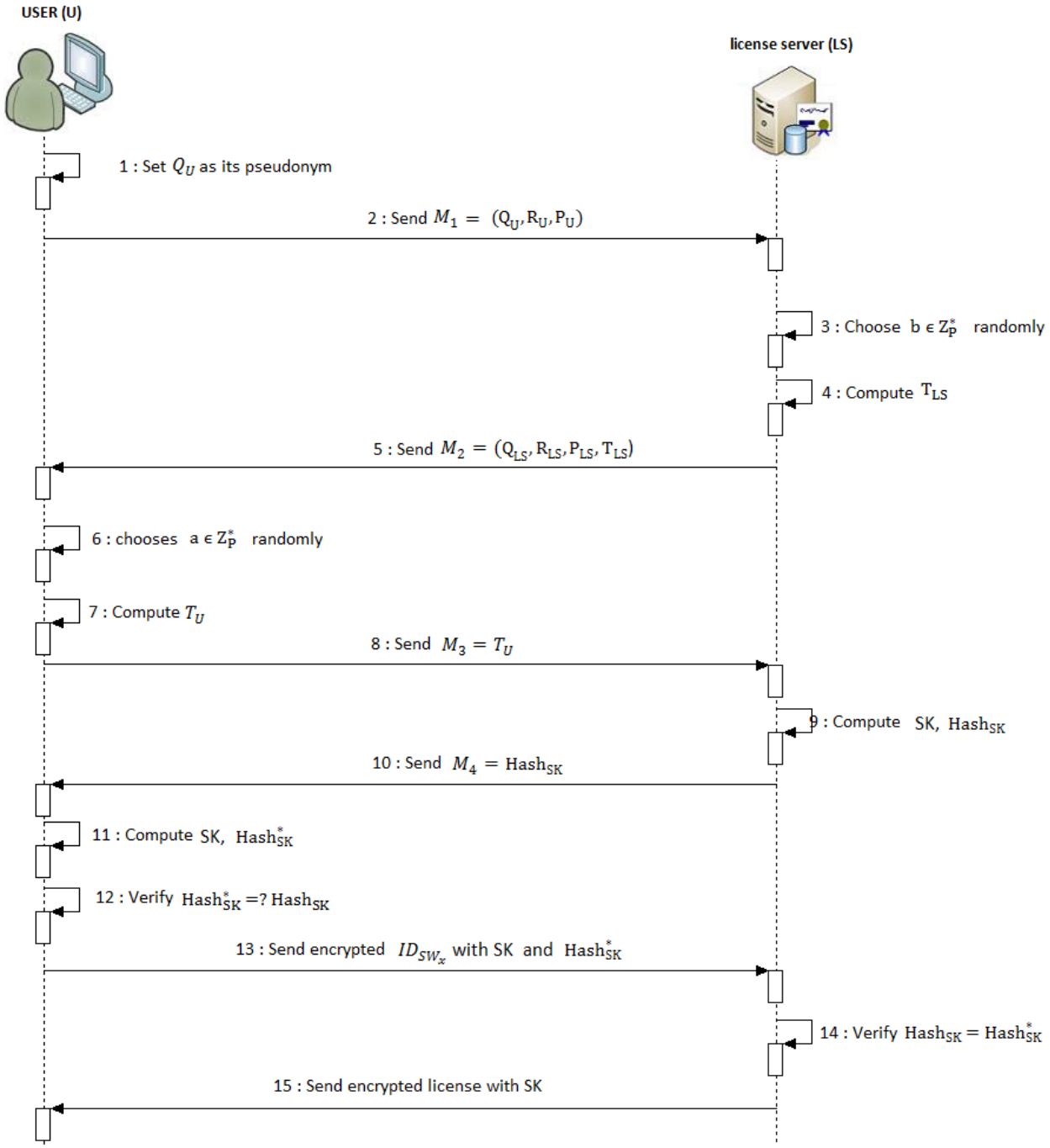

Figure 2: Proposed an anonymous license distribution mechanism

- `Setup`(run by the PKG): The Private key Generator (PKG) chooses a security parameter $k \in \mathbb{Z}$ and determines the tuple $\{G, P, \mathbb{F}_P, E/\mathbb{F}_P\}$ similar to how it is determined in Section 3. The PKG also chooses a master private key $s \leftarrow \mathbb{Z}_P^*$ then computes the master public key $P_0 = s \cdot P$ and two cryptographic hash functions namely H_1 and H_2, where $H_1 : \{0,1\}^* \times G \rightarrow \mathbb{Z}_P^*$ and $H_2 : \{0,1\}^{*2} \times G^9 \rightarrow \{0,1\}^K$. Finally, the PKG publishes the system parameters $(params) = \{G, P, \mathbb{F}_P, E/\mathbb{F}_P, P_0, H_1, H_2\}$, while the master key s is kept secretly by the PKG.
- `Set-public-Key`(run by U and LS):
  - U randomly selects $x_U \in \mathbb{Z}_P^*$, computes $X_U = x_U P$, then takes $P_U = X_U$ as its public key and keeps x_U secret.
  - LS randomly selects $x_{LS} \in \mathbb{Z}_P^*$, computes $X_{LS} = x_{LS} P$, then takes $P_{LS} = X_{LS}$ as its public key and keeps x_{LS} secret.
- `Partial-Private-Key-Extract` (run by the

PKG): This algorithm takes master key s, a user's ID_U identifier, license server's ID_{LS} identifier and system parameters as inputs. It then returns the corresponding partial private keys. PKG works as follows:

- PKG chooses two random numbers r_U, $r_{LS} \in \mathbb{Z}_p^*$, and computes $R_U = r_U P$, $Q_U = H_1(ID_U||R_U)$, and then computes $R_{LS} = r_{LS}P$, $Q_{LS} = H_1(ID_{LS}||R_{LS})$.
- PKG computes $d_U = (r_U + Q_U s)^{-1}$, $d_{LS} = (r_{LS} + Q_{LS}s)^{-1}$. It issues partial keys $\{d_U, R_U\}$, $\{d_{LS}, R_{LS}\}$ to the user U and license server LS respectively through a secret channel.

Upon receiving their partial private keys U and LS can validate their private keys respectively by checking whether the following equations holds: $d_U(R_U + Q_U P_0) = P$, $d_{LS}(R_{LS} + Q_{LS}P_0) = P$.

- `Set-Private-Key` (run by U and LS): When the U and LS receives their partial private keys from the PKG, they can compute their full private keys as follows:
  - U takes $SK_U = (d_U, x_U, R_U)$ as its private key.
  - LS takes $SK_{LS} = (d_{LS}, x_{LS}, R_{LS})$ as its private key.

Based on the fact that there is limited validity period to maintain forward secrecy of this pair of keys, U and LS will have to repeat this process after a period is ended. However this process does not involve the PKG but can be repeated individually by the LS and U using their respective partial private keys, $\{d_U, R_U\}$ and $\{d_{LS}, R_{LS}\}$. More details about forward secrecy can be seen in Section 4.1.

2) **Content Packaging**: The proposed DRM system supports the packaging of different types of media contents such as video, audio, text and image files. In the first stage of content packaging, there is need to restrain the service provider from analyzing the encrypted software by its length. To achieve this, padding is employed to the software prior to encryption. The second stage of content packaging is the encryption of the content. This resolves the owner's fear over security of content and distributors' fears over unlawful download of content from their SP.

Suppose the content provider has n contents, denoted by $SW_1, SW_2, .., SW_n$ with their unique identifiers $ID_{SW_1}, ID_{SW_2}, .., ID_{SW_n}$ respectively. The content provider then can generate n symmetric keys $CEK_1, CEK_2, .., CEK_n$ and individually encrypt each content with a corresponding unique symmetric key. It then obtains the encrypted contents in the following form:

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

Content provider later provides protected content with content information to the content server, provides content encryption keys $CEKs$ and provides usage rules to the license server via a secure channel.

3) **License Acquisition**: User chooses the interesting software SW_x with identifier ID_{SW_x} from the service provider which is allowed to download the encrypted content. A user cannot use the software without obtaining a valid license. To obtain the license, user creates a secure channel between U and LS by using an authenticated key agreement protocol with license server. Based on our anonymous pairing-free CL-AKE protocol, we allow a user to generate a session key with the license server without leaking his/her identity. The process in this phase is represented as follows:

- U sets Q_U as its pseudonym, then sends $M_1 = \{Q_U, R_U, P_U\}$ to the license server.
- Upon receiving the user's message M_1, LS randomly chooses the ephemeral key $b \in \mathbb{Z}_P^*$ and computes the key token $T_{LS} = b(R_U + Q_U P_0)$. Finally, the message $M_2 = \{Q_{LS}, P_{LS}, R_{LS}, T_{LS}\}$ is sent to U.
- Upon receiving M_2, U randomly chooses the ephemeral key $a \in \mathbb{Z}_P^*$ and computes the key token $T_U = a(R_{LS} + Q_{LS}P_0)$. Then sends $M_3 = T_U$ to LS.
- Upon receiving M_3, LS computes $d_{LS}T_U = aP$, $K^1_{LSU} = aP + bP$, $K^2_{LSU} = b \cdot aP$ and $K^3_{LSU} = b \cdot P_U + SK_{LS} \cdot aP$. Then computes the session key $SK = H_2(Q_U, Q_{LS}, R_U, R_{LS}, P_U, P_{LS}, T_U, T_{LS}, K^1_{LSU}, K^2_{LSU}, K^3_{LSU})$ and computes $Hash_{SK} = H_1(SK, T_U, T_{LS})$. Finally, LS sends the message $M_4 = \{Hash_{SK}\}$ to U.
- Then U can computes $d_U T_{LS} = bP$, $K^1_{ULS} = bP + aP$, $K^2_{ULS} = a \cdot bP$ and $K^3_{ULS} = a.P_{LS} + SK_U \cdot bP$. Then computes the session key $SK = H_2(Q_U, Q_{LS}, R_U, R_{LS}, P_U, P_{LS}, T_U, T_{LS}, K^1_{ULS}, K^2_{ULS}, K^3_{ULS})$ and the authentication token $Hash^*_{SK} = H_1(SK, T_U, T_{LS})$. Obviously, the two parties get the same session key because $K^1_{ULS} = aP + bP = K^1_{LSU}$, $K^2_{ULS} = abP = K^2_{LSU}$, $K^3_{ULS} = a \cdot P_{LS} + SK_U \cdot bP = SK_{LS} \cdot aP + b \cdot P_U = K^3_{LSU}$.

  Then U verifies the condition $Hash^*_{SK} \stackrel{?}{=} Hash_{SK}$. If the condition holds, U accepts the session key SK and anonymously purchases interesting content with identity ID_{SW_x} within the service provider, and service provider sends license acquisition request to license server. The license acquisition request involves the encrypted ID_{SW_x} which uses session key SK with $Hash^*_{SK}$ value.
- Upon receiving the license acquisition request, LS checks the condition $Hash_{SK} \stackrel{?}{=} Hash^*_{SK}$.

LS gets ID_{SW_x} by decrypting the encrypted content's identity using the shared SK.

- Finally, LS receives the payment and generates the license $L_{ID_{SW_x}}$ which includes content identity, content encryption key CEK, usage rules and user's pseudonym. It then encrypts the license using symmetric session key SK and sends encrypted license $E_{SK}(L_{ID_{SW_x}})$ to U through service provider. Furthermore, LS also keeps a record of usage license statistics for commercial use in the future.

4) **Content Consumption**: In the content consumption phase, user checks the license, decrypts the encrypted license with shared key SK and obtain the content encryption key. The user can decrypt the content with content encryption key and consumes the content according to usage rules in the license. The user needs to create a session key only once. Once the session has been established, a user can acquire any number of license during that session. For security enhancement, user can create a separate session key for each session. An overview of our anonymous pairing-free CL-AKE protocol is shown in Figure 2.

4 Analysis

The security analysis of our key exchange protocol are discussed in Section 4.1, the DRM security requirements analysis are discussed in Section 4.2 and Section 4.3 deals with efficiency comparison.

4.1 Security Analysis of Our Key Exchange Protocol

This section attempts to demonstrate that our protocol has managed to achieve almost all of the known desirable security attributes as defined by Blake-Wilson et al. [4].

4.1.1 Passive Attack

Attacker can get the information (P, T_U, T_{LS}, R_{LS}, P_{LS}, R_U, P_U, Q_U, Q_{LS}) transferred through the public channel. Indeed, it is more complicated for an adversary E to compute the session key SK, because the adversary does not know the secret keys for the concerned entities. Recalling that computing the values K^3_{ULS} or K^3_{LSU} is required to compute the correct session key SK, where the secret values SK_U or SK_{LS} are required respectively to find out K^3_{ULS} or K^3_{LSU}. Furthermore, the adversary may obtains the information (d_U, x_U, R_U),(d_{LS}, x_{LS}, R_{LS}),$d_{LS}T_U = aP$ and $d_UT_{LS} = bP$ for unknown a, b the value abP is required to obtain the correct session key SK. To compute abP without the knowledge of either a or b is equivalent to CDH problem which is slightly hard.

4.1.2 Man-in-the-Middle Attack

The most likely attack during the run of a key agreement protocol is the man-in-the-middle attack. Enabling the license server and user to authenticate with each other through exchanging $Hash_{SK}$ and $Hash^*_{SK}$ values, our proposed protocol is able to resist against the man-in-the middle attack. Therefore, there is no way to try man-in-the-middle attack by sending the forged message. It is necessary to compute the secret session key SK to find out $Hash_{SK} = H_1(SK, T_U, T_{LS})$. However, computing SK an adversary requires computing the value K^3_{ULS} or K^3_{LSU}, where the secret value SK_U or SK_{LS} is essential to find out K^3_{ULS} or K^3_{LSU}. Moreover, computing SK an adversary also requires finding out the ephemeral values a and b, which are not known to an adversary or malicious PKG.

4.1.3 Known Key Attack

If an adversary E obtains the secret keys of U and LS, it would be infeasible for E to recover any past session keys. The reason is as follows: Each session key involves two random ephemeral secrets a and b. Thus, it is not possible to derive a, b from T_U, T_{LS}, as ECDLP is not solvable in a polynomial time algorithm. On the other hand, it is also impossible to commutate abP given (P, aP, bP) due to the difficulties of CDH problem.

4.1.4 Forward Secrecy

If the secret key of PKG is disclosed, information about the session key is not revealed. This is because in order to get a session key, the values (x_U, x_{LS}) and (a, b) are required. These values cannot be computed by using master key since the secret values (a, x_U) and (b, x_{LS}) are randomly chosen by U and LS respectively. Furthermore, computation of abP from given (P, aP, bP) is hard due to difficulties of CDH problem.

4.1.5 Key Off-set Attack (KOA)

In our protocol, user U sends the message $M_1 = \{Q_U, R_U, P_U\}$ and $M_3 = T_U$ to LS. An adversary E can modify it to $M_3 = T^*_U$, where $T^*_U = tialT_U$. When, LS computes the session key $SK_1 = H_2(Q_U, Q_{LS}, R_U, R_{LS}, P_U, P_{LS}, T^*_U, T_{LS}, K^{1*}_{ULS}, K^{2*}_{ULS}, K^{3*}_{ULS})$ and $Hash_1$. LS sends the message $M_2 = \{Q_{LS}, P_{LS}, R_{LS}, T_{LS}\}$ to U. Again, the adversary E modifies T_{LS} to $T^{**}_{LS} = tialT_{LS}$, but does not change the $Hash_1$, because the LS's secret is required. Now U computes the session key $SK^*_1 = H_2(Q_U, Q_{LS}, R_U, R_{LS}, P_U, P_{LS}, T_U, T^{**}_{LS}, K^{1**}_{ULS}, K^{2**}_{ULS}, K^{3**}_{ULS})$ and the authentication token $Hash^*_1 = H_1(T_U||T^{**}_{LS}||SK^*_1)$. It then compares it with the received $Hash_1$ and concludes that $Hash^*_1 \neq Hash_1$. User U therefore turns off the session key-agreement and sends an authentication-failed message to LS. So the KOA attack is impossible.

Table 1: Efficiency comparison

	Operations						Total Running Time m/s
	Multiplication		Pairing		One-way hash		
	Number	Running time	Number	Running time	Number	Running time	
Ref. [7]	17	37.57	4	80.16	8	24.32	142.05
Our	21	46.41	0	0	6	18.24	64.65

4.1.6 Known Session-specific Temporary Information Attack (KSTIA)

If the session ephemeral secrets a and b are compromised by an adversary, then session key will not be revealed. Because, a user cannot compute SK, and the user can generate the session key if and only if it is possible to compute U's or LS's secret values.

4.1.7 No Key Control (NKC)

Both entities, U and LS have an input each into the session key. No entity can force the full session key to be a preselected value. It is determined jointly by both entities U and LS. Whenever $SK = H_2(Q_U, Q_{LS}, R_U, R_{LS}, P_U, P_{LS}, T_U, T_{LS}, K^1_{ULS}, K^2_{ULS}, K^3_{ULS})$ it involves T_U and T_{LS} and these are computed by U and LS respectively.

4.1.8 Reflection Attack (RA) and Unknown Key-share Attack (UKA)

In our scheme, the session key is computed not only by using K^1_{ULS},K^2_{ULS},K^3_{ULS} but also by using the pseudonyms of the entities Q_U, Q_{LS} and other session dependent tokens T_U, T_{LS}. According to Wang et al. [16], our scheme provides the resilience against unknown key-share attack and reflection attack.

4.2 DRM Security Requirement Analysis

Based on the DRM security requirements that have been discussed in Section 2, this section endeavors to manifest that our scheme satisfies all the following requirements

4.2.1 Content Protection

The DRM content is encrypted separately from the license, which increases the flexibility of management. That is to say, if a DRM content is modified, the corresponding license will not be affected. Even if an unauthorized user downloads a DRM encrypted content, he could not be able to play it without the valid license, due to the reason that the safe performance is also optimized to prevent the unauthorized access.

4.2.2 DRM Security

The user is limited to purchase the content from service provider and obtain the license from the license server. With the license, the user can get the content encryption key. Thus, only a legal user can decrypt the content with a valid license.

4.2.3 User Privacy/Anonymity

In our method, an anonymous user directly communicates with the license server. Since the user is giving out Q_U as its pseudonym instead of its real identity ID_U, which prevents the other parties such as an adversary from getting any user's personal information like which software is bought and who buys the software. In this sense, the user's privacy is maintained.

4.3 Performance Evaluation

In this section, the efficiency comparison of our scheme against Mishra *et al.* [7] scheme is presented. This comparison is prepared based on experimental results in [5, 6], for various cryptographic operations using MIRACLE [10] 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 is 20.04 m/s, ECC-based scalar multiplication is 2.21 m/s, one-way hash function is 3.04 m/s and pairing-based scalar multiplication is 6.38 m/s. For convenience, we define the following notations: T_H(the time complexity of one-way hash function); T_e (the time complexity of pairing operation) and T_{mul} (the time complexity of a scalar multiplication operation of point). As indicated in Table 1, the computational costs of Mishra *et al.* scheme is increasingly higher. Furthermore, this scheme requires 4 times bilinear pairing operation. However, the time consumed in pairing operation is more than other operations over elliptic curve group. Moreover, Figure 3 shows the efficiency comparison of our scheme versus Mishra *et al.* based on running time for each operation.

5 Conclusions

Based on our anonymous pairing-free CL-AKE protocol for DRM system, we put forward a mechanism for distributing licenses in a flexible and secure manner. In our scheme, the license server authenticates an anonymous user and creates session key to communicate securely, which achieves not only user anonymity, but also preserved user privacy. Moreover, compared to public key

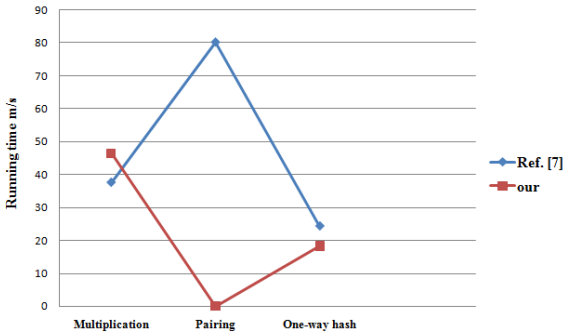

Figure 3: Efficiency comparison by running time

encryption, our method applies symmetric key encryption to achieve content license, which needs less computation. As a result, it is safe to draw the conclusion that our present work could be considered as the most efficient and scalable for DRM system.

Acknowledgments

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

References

[1] S. Adi, "Identity-bsed cryptosystems and signature schemes," in *Advance in Cryptography (CRYPTO'84)*, LNCS 196, pp. 47–53. Springer, 1985.

[2] S. S. Al-Riyami and K. G. Paterson, " Certificateless public key cryptography," in *Advance in Cryptography (ASIACRYPT'03)*, LNCS 2894, pp. 452–473, Springer, 2003.

[3] S. Amit, S. Emmanuel, A. Das and M. S. Kankanhalli, "Privacy preserving multiparty multilevel DRM architecture," in *IEEE Consumer Communications and Networking Conference (CCNC'09)*, pp. 1–5, 2009.

[4] Z. Cheng, M.Nistazakis, R.Comley, L.Vasiu, "On the indistinguishability-based security model of key agreement protocols-simple cases", Cryptology ePrint Archieve, Report 2005/129, 2005.

[5] H. Debiao, J. Chen and R. Zhang. "An efficient identity-based blind signature scheme without bilinear pairings," *Computers & Electrical Engineering*, vol 37, no. 4, pp. 444–450, July 2011.

[6] H. Debiao and C. Jianhua. "An efficient certificateless designated verifier signature scheme," *The International Arab Journal of Information Technology*, vol 10 no. 4, pp.389–396, 2013.

[7] M. Dheerendra and S. Mukhopadhyay, "A certificateless authenticated key agreement protocol for digital rights management system," in *Quality, Reliability, Security and Robustness in Heterogeneous Networks*, pp. 568–577, Springer, 2013.

[8] M. Erika, T. Grance and K. Kent, "Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)," *National Institute of Standards and Technology (NIST)*, Special Publication, pp. 800–122, Apr. 2010.

[9] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Systems Journal*, vol. 9, no. 3, pp. 816–823, 2014.

[10] MIRACL, *Multiprecision Integer and Rational Arithmetic C/C++ Library*, July 14, 2015. (`http://indigo.ie/mscott/`)

[11] H. S. Oun, K. S. Yoon, K. P. Jun and K. H. Lee, "Modeling and implementation of digital rights," *Journal of Systems and Software*, vol 73, no. 3, pp. 533–549, 2004.

[12] D. Ratna, D. Mishra and S. Mukhopadhyay, "Vector space access structure and ID based distributed DRM key management," in *Proceedings of First International Conference on Advances in Computing and Communications (ACC'11)*, CCIS 193, pp. 223–232, Springer, 2011.

[13] D. Ratna, D. Mishra and S. Mukhopadhyay, "Access policy based key management inmulti-level multidistributor DRM architecture," in *Proceedings of First International Conference on InfoSecHiComNet*, LNCS 7011, pp. 57–71, Springer, 2011.

[14] D. Ratna, S. Mukhopadhyay and T. Dowling, "Key management in multi-distributor based DRM system with mobile clients using IBE," in *Second International Conference on the Applications of Digital Information and Web Technologies*, pp. 597–602, 2009.

[15] Y. C. Ta, H. T. Liaw and N. W. Lo, "Digital rights management system with userprivacy usage transparency, and superdistribution support," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 1714–1730, 2014.

[16] S. Wang, Z. Cao, K. K. R. Choo and L. Wang, "An improved identity-based key agreement protocol and its security proof," *Information Sciences*, vol 179, no. 3, pp. 307–318, 2009.

[17] D. Whitfield and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol 22, no. 6, pp. 644–654, 1976.

[18] K. William and C. H. Chi, "Survey on the technological aspects of digital rights management," *Information Security*, pp. 391–403, 2004.

[19] H. Xiong, Q. Wu and Z. Chen, "Toward pairing-free certificateless authenticated key exchanges," in *Proceedings of 14th International Conference on Information Security (ISC'11)*, LNCS 7001, pp. 79–94, Springer, 2011.

[20] Z. Zhiyong, Q. Pei, J. Ma and L. Yang, "Security and trust in digital rights management: a survey," *International Journal of Network Security*, vol 9, no. 3, pp. 247–263, 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.

Philip Avornyo is a PhD. degree candidate in school of management science and engineering at University of Electronic Science and Technology of China (UESTC). His research area is Electronic Business (e-Business) with focus on Mobile Banking (M-Banking).

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.