2010 Fourth International Conference on Genetic and Evolutionary Computing

An Efficient and Flexible Authentication Scheme with User Anonymity for Digital Right Management

Jen-Ho Yang
Department of Information and Electronic Commerce,
Kainan University,
No. 1, Kannan Road, Luzhu, Taoyuan County, 33857, Taiwan
E-mail: jenhoyang@mail.knu.edu.tw

Chih-Cheng Hsueh
Department of Information Application,
Aletheia University,
70-11, Pei-Shi-Liao, Matou, Tainan County, 33857, Taiwan
E-mail：jrcheng@mt.au.edu.tw

Chung-Hsuan Sun
Department of Information and Electronic Commerce,
Kainan University,
No. 1, Kannan Road, Luzhu, Taoyuan County, 33857, Taiwan
E-mail: m09808004@ms1.knu.edu.tw

***Abstract*—In recent years, various Digital Right Management (DRM) schemes have been proposed to protect and manage access rights of digital contents. Nevertheless, most DRM schemes do not care for protecting user privacy so the user's private information is easily compromised. To solve this problem, Zhang et al. proposed a license management scheme with anonymous trust for digital right management. However, we find that their scheme has high computation cost. Besides, their scheme is insecure because the decryption key of the digital content can be easily computed by an attacker. Thus, we propose an efficient and flexible authentication scheme with user anonymity for DRM in this paper. Compared with Zhang et al.'s scheme, the proposed scheme has lower computation cost. In addition, the user privacy can be well-protected without revealing the decryption key. Therefore, the proposed scheme is efficient and practical for the DRM applications.**

***Keywords**: Digital Right Management, authentication, user privacy, anonymity, access right*

I. INTRODUCTION

The Digital Right Management (DRM) scheme is a digital protection technique that protects and manages the access rights of digital contents. It can prevent the confidential information of a digital content from unauthorized usages by illegal users. Generally, there are four roles in the DRM scheme: a content provider (author), a consumer (client), a clearing house, and a distributor [1]. The content provider creates the digital content and encrypts it using some proper cryptosystems, such as RSA [2], ElGamal [3], and ECC [4]. Then, the content provider sends the encrypted content to the distributor (e.g., web server or online shop) for online distribution. Next, the content provider sends the usage rules to the clearing house, such as the copy permit, the pay-per-view, and the usage fee, to specify how to use the digital content. Note that the clearing house is responsible for issuing the digital license and handling the financial transactions for the content provider, the distributor, and the consumer.

Next, the consumer downloads the digital content from the web server. To access the encrypted content, the consumer requests the clearing house to issue a valid license, which contains the decryption key, usage rules, and descriptions of the digital content. Then, the clearing house performs the user authentication mechanisms [5-11] to verify the identity of the consumer. Next, the clearing house can charge the consumer account for the digital content. After the consumer has paid the money, the clearing house sends the license to the consumer. Finally, the consumer has the access rights to use the digital content. Due to the digital content can be easily obtained and distributed via the Internet, the DRM scheme becomes a popular research topic in recent years. Therefore, many DRM schemes [12] have been proposed to protect and manage the access right of a digital content. In 2005, Zhang et al. [12] proposed a license management scheme with anonymous trust for digital rights management (LMSAT). Their scheme allows the client access the content anytime and anywhere by using any permissive device. Thus, their scheme provides a more powerful and flexible license acquisition and usage tracking for the digital right management.

However, we find that Zhang et al.'s scheme has two drawbacks. First, their scheme has large computation costs because it utilizes the public-key cryptosystem [2-4]. Thus, their scheme does not suitable for the mobile device with low computation ability. Second, their scheme has a security flaw because the encryption/decryption key of the digital content may be revealed. To solve the above problems, we propose an efficient and flexible authentication scheme with user anonymity for digital rights management in this paper. In the proposed scheme, we use one-way hash functions for user authentication to reduce the computation costs. In addition, we fix the security problem of Zhang et al.'s scheme so that the decryption key of the digital content would not be revealed by any illegal user. In addition, the user privacy can be well-protected in our scheme. Therefore, the proposed scheme is more efficient and practical than Zhang et al.'s scheme for the digital right management in e-commerce.

II. ZHANG ET AL.'S LMSAT SCHEME FOR DRM

In this section, we introduce Zhang et al.'s scheme [12] and point out the drawbacks of their scheme. There are five roles in their scheme: the content producer, the content provider, the clearing house (CH), the client, and the certificate authority (CA). Zhang et al.'s scheme has the following assumptions for LMSAT:

978-0-7695-4281-2/10 $26.00 © 2010 IEEE
DOI 10.1109/ICGEC.2010.161

- Each participant in the LMSAT system knows the all cryptography algorithms used in their scheme.
- Each participant has the public/private key pair, which is verified by a certificate issued from the CA..
- The client has paid the money for the digital content, and then the client obtains an anonymity identity (*Anonymity ID*) and the encrypted content.
- DRM agent (DA) is an agent loaded in the Client's device. DA is responsible for paying the digital content, acquiring the digital license from CH, authenticating the license and the content, decrypting the encrypted content, and reporting the usage to CH.

TABLE I. THE NOTATIONS USED IN ZHANG ET AL.'S SCHEME

Notations	Explanations
S_X/Q_X	The private/public key of the entity X on ECC [4]
$H(\ )$	One-way hash function
$[m]_k$	The symmetric encryption with the message m using key k
License	The license of a digital content
K_B	The symmetric key for encrypting the digital license
Decryption Key	The decryption key for the encrypted digital content
Content ID	The identity of a digital content
Usage Rules	The usage rules of a digital content
$\|\|$	The concatenation of strings
SN	The sequence number of the license
Sig_X	The digital signature signed by the entity X
Anonymity ID	The anonymity identity of a client
Other Data	The other data of a digital content

The License Acquisition Phase:

Step 1. DA generates a temporal private key S_{DA} to compute the corresponding public key as $Q_{DA} = S_{DA} \cdot G$, where G is a public point on ECC [4]. Then, DA sends Q_{DA} to CH.

Step 2. CH generates a temporal private key S_{CH} to compute $Q_{CH} = S_{CH} \cdot G$ and the session key $K = S_{CH} \cdot Q_{DA}$. Then, CH computes $Sig_{CH}(H(Q_{DA} \| Q_{CH}))$ and uses K to encrypt it. Next, CH sends $\{Sig_{CH}(H(Q_{DA} \| Q_{CH}))\}_K$ and Q_{CH} to DA.

Step 3. DA computes the session key as $K = S_{DA} \cdot Q_{CH}$ and uses it to decrypt $\{Sig_{CH}(H(Q_{DA} \| Q_{CH}))\}_K$. And, DA verifies $Sig_{CH}(H(Q_{DA} \| Q_{CH}))$. If the verification is correct, then DA signs $H(Q_{DA} \| Q_{CH})$ and H (*Anonymity ID* || *Content ID* || *Usage Rules*). Next, DA encrypts the above signatures by using K and sends the encrypted result to CH.

Step 4. CH decrypts the above message and verifies $Sig_{DA}(H(Q_{DA} \| Q_{CH}))$. If the verification is correct, then CH verifies $Sig_{DA}(H$ (*Anonymity ID* || *Content ID* || *Usage Rules*)) to check whether the client is valid or not according to *Anonymity ID*. If the above authentication is correct, then CH generates the digital license as: *License*={*SN*, *ContentID*, [*Decryption Key*, *Usage Rules*, *Other Data*]$_{KB}$, Sig_{CH} (*H*(*SN*, *ContentID*, [*Decryption Key*, *Usage Rules*, *Other Data*]$_{KB}$))}, where K_B =*H*(*Anonymity ID* || *Content ID*).

Step 5. CH computes $\{License \| Sig_{CH}(H(License))\}_K$ and sends it to DA. Next, DA decrypts the above message to get *License* and verifies $Sig_{CH}(H(License))$. Then, DA computes K_B =*H*(*Anonymity ID* || *Content ID*) and decrypts *License* to get *Decryption Key*. Finally, DA can decrypts the protected content and send it to the trusted rendering agent for rendering.

Usage Tracking Phase:

Step 1. CH generates a temporal private key S_{CH} to compute the corresponding public key as $Q_{CH} = S_{CH} \cdot G$. Then, CH sends Q_{CH} to DA.

Step 2. DA generates a temporal private key S_{DA} to compute $Q_{DA} = S_{DA} \cdot G$ and $K = S_{DA} \cdot Q_{CH}$. Then, DA computes $Sig_{DA}(H(Q_{DA} \| Q_{CH}))$ and uses K to encrypt it. Next, DA sends $\{Sig_{DA}(H(Q_{DA} \| Q_{CH}))\}_K$ and Q_{DA} to CH.

Step 3. CH computes the session key as $K = S_{CH} \cdot Q_{DA}$ and uses it to decrypt $\{Sig_{DA}(H(Q_{DA} \| Q_{CH}))\}_K$ to verify the signature. If the above verification is correct, then CH signs $H(Q_{DA} \| Q_{CH})$ and H (*Anonymity ID* || *SN*). Next, CH encrypts the above signatures by using K and sends the encrypted result to DA.

Step 4. DA decrypts the above message and verifies $Sig_{DA}(H(Q_{DA} \| Q_{CH}))$ and $Sig_{CH}(H$ (*Anonymity ID* || *SN*)). If the above verifications are correct, then DA collects the usage data *UsageData* according to *Anonymity ID* and *SN*. Next, DA computes $\{UsageData \| Sig_{DA}(H(UsageData)\}_K$ and sends

it to CH. Finally, CH can decrypt $\{UsageData \| Sig_{DA}(H(UsageData)\}_K$ and track the usage of the content.

According to Zhang et al.'s scheme, we find that their scheme has the following drawbacks. First, their scheme is insecure because $K = S_{CH} \cdot Q_{DA} = S_{DA} \cdot Q_{CH} = (S_{CH} S_{DA}) \cdot Q$ can be easily computed by a forgery DA or CH. This is because Zhang et al.'s scheme adopts Elliptic Curve Diffie-Hellman (ECDH) key agreement protocol to negotiate the symmetric key *K* between CH and DA. However, ECDH cannot withstand the man-in-the-middle attack. This causes that the symmetric key *K* can be computed by an attacker who impersonates CH or DA. Thus, the decryption key K_B, which is computed by *Anonymity ID* and *Content ID*, can be also obtained by the attacker. Finally, the attacker can obtain *Decryption Key* from *License* to decrypt the protected digital content.

Second, Zhang et al.'s scheme has high computation costs because its user authentication is designed by public-key cryptosystems [2-4]. To authenticate the validity of each participant, their scheme needs to computes a large amount of digital signatures. Thus, the user authentication greatly increases the computation loads of their scheme. To solve the above problems, we propose an efficient and practical authentication scheme for DRM in the next section.

III. THE PROPOSED AUTHENTICATION SCHEME FOR DRM

There are four roles in the proposed scheme: the content producer, the content provider, the clearing house (CH), and the client. The DRM model of the proposed scheme is similar as Figure 1. The difference of the proposed scheme and Zhang et al.'s scheme is that our scheme does not need the CA to issue the certificate. This is because the proposed scheme does not use the public-key cryptosystem. Besides, the proposed scheme has the following assumptions:

- The client has paid the money to the content provider for the digital content, and then the client obtains an anonymity identity (*Anonymity ID*) and the authentication value $H(Anonymity\ ID \oplus X)$ for the encrypted content. Then, the content provider sends *Anonymity ID*, $H(Anonymity\ ID \oplus X)$, and *Content ID* to CH. Next, the client can use *Anonymity ID* and $H(Anonymity\ ID \oplus X)$ to authenticate by CH and obtain the decryption key of the content.
- There is a DRM agent (DA) loaded in the Client's device. Similar to Zhang et al.'s scheme, the DA is responsible for paying the digital content, acquiring the digital license from CH, authenticating the license and the content, decrypting the encrypted content, and reporting the usage to CH.

The notations used in the proposed scheme are shown in Table 2. Now, we introduce the proposed scheme as follows.

TABLE II. THE NOTATIONS USED IN THE PROPOSED SCHEME

Notations	Explanations
$H(\ )$	One-way hash function
SK	The session key
$E_{SK}(\cdot) / D_{SK}(\cdot)$	The symmetric encryption/decryption with the session key *SK*
X	The secret key of the content provider
License	License of a digital content
Anonymity ID	The anonymity identity of a client
Content ID	The identity of a digital content
UsageRules	The usage rules of a digital content
UsageData	The usage data of the digital content
$\|$	The concatenation of strings
SN	The sequence number of the license
$\oplus$	Exclusive-or operation
Decryption Key	The decryption key for the encrypted digital content
OtherData	The other information of the license

The Authentication and License Acquisition Phase:

In this phase, the client has downloaded the encrypted digital content and wants to get the license from CH to access the content. In addition, CH authenticates the client and sends the license to the valid client. The steps of this phase are shown as follows.

Step 1. The client (DA) chooses a random number R_{DA} to compute $S_{DA} = H(H(Anonymity\ ID \oplus X)) \oplus R_{DA}$ and $C_{DA} = H(R_{DA})$. Then, DA sends *Anonymity ID*, S_{DA}, and C_{DA} to CH.

Step 2. CH computes $R'_{DA} = S_{DA} \oplus H(H(Anonymity\ ID \oplus X))$ to check if C_{DA} is equal to $C'_{DA} = H(R'_{DA})$. If they are equal, then CH authenticates that DA is valid. Next, CH choose random number R_{CH} to compute $S_{CH} = H(H(Anonymity\ ID \oplus X) \| R_{DA}) \oplus R_{CH}$ and the session key $SK = H(H(Anonymity\ ID \oplus X) \| R_{DA} \| R_{CH})$. Finally, CH computes $C_{CH} = H(R_{DA} \| R_{CH} \| SK)$ and sends S_{CH} and C_{CH} to DA.

Step 3. DA computes $R'_{CH} = S_{CH} \oplus H(H(Anonymity\ ID \oplus X) \| R_{DA})$ and $SK' = H(H(Anonymity\ ID \oplus X) \| R_{DA} \| R'_{CH})$.

Then, DA computes $C'_{CH} = H(R_{DA} \| R'_{CH} \| SK')$ to check if C_{CH} is equal to C'_{CH}. If they are equal, then DA authenticates that CH and *SK* are both valid. Next, DA computes $E_{SK}(Anonymity\ ID \| Content\ ID \| UsageRules \| SK)$ and sends it to CH.

Step 4. CH computes *License* = {*SN*||*Content ID*||*UsageRules*||*DecryptionKey*||*OtherData*}. Then, CH computes $E_{SK}(License \| SK)$ and sends it to DA. Finally, DA can use *SK* to decrypt $E_{SK}(License \| SK)$ and obtain *License*. Thus, the client gets *Decryption Key* from *License* and uses it to decrypt the encrypted digital content.

The Usage Tracking Phase:

In this phase, CH receives the report of the content usage from DA. For fraud prevention, CH needs to authenticate the validity of DA. In addition, the usage information needs to be encrypted for protecting the user privacy. The steps of this phase are shown as follows.

Step 1. First, CH chooses a random number $\overline{R}_{CH}$ to compute $\overline{S}_{CH} = H(H(Anonymity\ ID \oplus X)) \oplus \overline{R}_{CH}$ and $\overline{C}_{CH} = H(\overline{R}_{CH})$. Then, CH sends *Anonymity ID*, $\overline{S}_{CH}$, and $\overline{C}_{CH}$ to DA.

Step 2. DA computes $\overline{R}'_{CH} = \overline{S}_{CH} \oplus H(H(Anonymity\ ID \oplus X))$ to check if $\overline{C}_{DA}$ is equal to $\overline{C}'_{CH} = H(\overline{R}'_{CH})$. If they are equal, then DA authenticates that CH is valid. Next, DA choose random number $\overline{R}_{DA}$ to compute $\overline{S}_{DA} = H(H(Anonymity\ ID \oplus X) \| \overline{R}_{CH}) \oplus \overline{R}_{DA}$ and the session key $\overline{SK} = H(H(Anonymity\ ID \oplus X) \| \overline{R}_{CH} \| \overline{R}_{DA})$. Next, DA computes $\overline{C}_{DA} = H(\overline{R}_{CH} \| \overline{R}_{DA} \| \overline{SK})$ and sends $\overline{S}_{DA}$ and $\overline{C}_{DA}$ to CH.

Step 3. CH computes $\overline{R}'_{DA} = \overline{S}_{DA} \oplus H(H(Anonymity\ ID \oplus X) \| \overline{R}_{CH})$ and the session key $\overline{SK}' = H(H(Anonymity\ ID \oplus X) \| \overline{R}_{CH} \| \overline{R}'_{DA})$. Then, CH computes $\overline{C}'_{DA} = H(\overline{R}_{CH} \| \overline{R}'_{DA} \| \overline{SK}')$ to check if $\overline{C}_{DA}$ is equal to $\overline{C}'_{DA}$. If they are equal, then CH authenticates DA and $\overline{SK}$ are both valid. Next, CH computes $E_{\overline{SK}}(Anonymity\ ID \| Content\ ID \| SN \| \overline{SK})$ and sends it to DA.

Step 4. DA computes $E_{\overline{SK}}(UsageData \| \overline{SK})$ and sends it to CH. Finally, CH can decrypt $E_{\overline{SK}}(UsageData \| \overline{SK})$ to trace the content usage.

According to the above description, the proposed scheme is designed by one-way hash functions and XOR operations. Thus, the computation cost of the proposed scheme is lower than that of Zhang et al.'s scheme. In addition, the proposed scheme can prevent man-in-the-middle attack even if *Anonymity ID* is revealed. Thus, the proposed scheme is securer than Zhang et al.'s scheme.

IV. CONCLUSIONS

In this paper, we propose an efficient and practical authentication scheme with user anonymity for DRM. The proposed scheme has low computation cost and protects the user privacy by the anonymity identity for the user. In addition, the proposed scheme allows users access the digital contents using any permissive devices. Thus, the proposed scheme is efficient and practical for the DRM applications.

REFERENCES

[1] Q. Liu, S. N. Reihaneh, and N. P. Sheppard, "Digital rights management for content distribution," *Proceedings of Australasian Information Security Workshop 2003 (AISW2003), Conferences in Research and Practice in Information Technology*, Adelaide, Australia., Vol. 21, 2003.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, Vol. 21, pp. 120-126, 1978.

[3] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, Vol. 31, pp. 469-472, 1985.

[4] N. Koblitz, "Elliptic curve cryptosystem", *Mathematics of Computation*, Vol. 48, pp. 203-209, 1987.

[5] T. Kwon and J. Song, "Efficient key exchange and authentication protocols protecting weak secrets," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E81-A, pp. 156-163, 1998.

[6] T. Kwon and J. Song, "Authenticated key exchange protocols resistant to password guessing attacks," *IEEE Proceedings Communications*, Vol. 145, pp. 304-308, 1998.

[7] M. Sandirigama, A. Shimizu, and M. T. Noda, "Simple and secure password authentication protocol (SAS)," *IEICE Transactions on Communications*, Vol. E83-B, pp. 1363-1365, 2000.

[8] M. S. Hwang, C. C. Lee, and Y. L. Tang, "A simple remote user authentication protocol," *Mathematical and Computer Modelling*, Vol. 36, pp. 103-107, 2002.

[9] H. Y. Chien and J. K. Jan, "Robust and simple authentication protocol," *Computer Journal*, Vol. 46, pp. 193-201, 2003.

[10] E. Bresson, O. Chevassut, A. Essiari, and D. Pointcheval, "Mutual authentication and group key agreement for low-power mobile devices," *Computer Communications*, Vol. 27, pp. 1730-1737, 2004.

[11] H. M. Sun and H. T. Yeh, "Password-based authentication and key distribution protocols with perfect forward secrecy," J*ournal of Computer and System Sciences*, Vol. 72, pp. 1002-1011, 2006.

[12] J. Zhang, B. Li, L. Zhao, and S. Q. Yang, "License management scheme with anonymous trust for digital rights management," *Proceedings of 2005 IEEE International Conference on Multimedia and Expo, International Conference on Multimedia &Expo(ICME 2005)*, Amsterdam, Netherlands, pp. 257-260, Jul. 2005.