A Certificateless Authenticated Key Agreement Protocol for Digital Rights Management System

Dheerendra Mishra and Sourav Mukhopadhyay

Department of Mathematics
Indian Institute of Technology
Kharagpur–721302, India
{dheerendra,sourav}@maths.iitkgp.ernet.in

Abstract. Digital rights management (DRM) is the system which tries to ensure authorized content consumption. Current DRM systems either adopt public key cryptography (PKC) or identity based public key cryptography (ID-PKC). PKC associates certificate management which includes revocation, storage, distribution and verification of certificate, as a result, certificate authority becomes the bottleneck for the large network. While, ID-PKC has drawback of key escrow. However, for secure and authorized content distribution, evacuation from these problems is needed. In this paper, we present a certificateless authenticated key agreement protocol for DRM system, which ensures flawless mutual authentication and establishes a session key between user and license server. Furthermore, we analyzed proposed scheme to show that proposed scheme is secured.

Keywords: Digital rights management, Certificateless public key cryptography, Bilinear pairing, Authentication.

1 Introduction

With the advancement of Internet technology, e-commerce industry achieves a scalable infrastructure for digital content distribution at low cost. Internet facilitates an online trade of digital content (text, music, movies, and software). However, the digital contents can be easily copied and redistributed over the network without any drop in the quality of contents. As a result, illegal copies of content are available over the network which causes a huge loss of revenue to the right holders. Therefore, rights holders are demanding a mechanism which can regulate the authorized content consumption so that copyright protection would be achieved. One such a mechanism is digital rights management (DRM)system which is developed to ensure the copyright protection [11].

DRM system also tries to maintain flexible and secure content distribution. For flexible and secure communication, an efficient mutual authentication and key agreement protocol is needed where the involved parties can authenticate each other and establish a secure session key. The session key is generated with the information shares of involve parties which used to achieve its goal of confidentiality and data integrity. Current DRM systems [9,12,16,6,7,8,15] basically

K. Singh, A.K. Awasthi, and R. Mishra (Eds.): QSHINE 2013, LNICST 115, pp. 568–577, 2013.

apply two approaches, namely, public key cryptography (PKC) [14] and identity based public key cryptography (ID-PKC) [13]. Schemes [9,12] use PKC to authenticate public key where PKC maintains a certificate authority (CA). CA proofs the relation between entity and its public key. Moreover, it manages certificate management including storage, distribution and revocation. However, CA becomes bottleneck for large network. Therefore, computational cost of certificate verification become infeasible. While, schemes [6,7,15] apply identity based infrastructure where involve parties achieve their private key from private key generator (PKG) and public key is derived from their public identity such as email address. Yen et al. [15] also presented an ID based authenticated key agreement protocol which manage secure communication. However, in ID-PKC, PKG knows the private key of each user, that means PKG could generates forge signature of any entity. This causes the key escrow problem.

In this paper, we will apply the pairing based certificateless authenticated key agreement protocol for DRM system which is introduced by Al-Riyami and Paterson [1]. In this scheme, license server and user achieve their private keys using PKG generated partial private key share and self generates secret value. Further, both parties establish authenticated session key to communicate securely. Even more, They can establish different session keys for different sessions to achieve security. The proposed protocol eliminated the use of trusted certificate authority and solve key escrow problem. Moreover, user adopts symmetric key encryption to achieve content license which requires less computation compare to public key encryption.

The rest of the paper is organized as follows: In section 2, we discuss a typical DRM model, recall the concept of public key cryptography, identity based cryptography and certificateless cryptography, and define some notation that we will use throughout the paper. We present our content distribution scheme in section 3. We present security analysis in section 4. Finally, in section 5, we draw a conclusion.

2 Preliminaries

2.1 Basic DRM System

A general DRM architecture involves four core component: content provider (owner), license server, distributor and user [11].

Content Provider. Content provider holds the digital rights of the content and wants to protect the content. It works as a packager. To protect the content from unauthorized user, it encrypts the content. It provides protected content with content information to the distributor and content key with usage rules to the license server.

Distributor. Distributor works as a service provider. It associates a media server and sets up a website. It keeps protected content over the media server and display content information over the website. The distributor provides encrypted content and content detailed information (file size, file type, player, etc.) to the users.

License Server. License server generates the license by using key seeds where license comprises of content key, usage rules and constraints. It authenticates the user by using standard authentication mechanisms such as password based, smart card based, etc. It issues the license only for authorized users.

User. A DRM user downloads the protected content from the media server and acquires the license from the license server. A user always wants that content should be easy to play and easy to download besides secure payment mechanism and privacy.

2.2 Bilinear Pairings

Let $(G_1, +)$ and $(G_2, .)$ be the additive and multiplicative cyclic groups of order q respectively where q is a k-bit large prime. The bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$ defined by $e(\cdot, \cdot)$ has the following properties as discussed in [2,4]:

- **Bilinear:** $e(aP, bQ) = e(P, Q)^{ab}$, for all $P, Q \in G_1$ and $a, b \in Z_q^*$;
- **Non-degenerate:** There exist $P, Q \in G_1$ such that $e(P, Q) \neq 1$;
- **Computable:** There exists an efficient algorithm to compute $e(P, Q)$, for all $P, Q \in G_1$.

The security of CL-PKC in $e(\cdot, \cdot)$ based on the hardness of following computational problems:

Discrete Logarithm Problem: For a given generator P of G_1 and $Q \in G_1$, find an element $a \in Z_q^*$ such that $aP = Q$.
Computational Diffie-Hellman (CDH) Problem: Let P be a generator of G_1. Given $\langle P, aP, bP \rangle \in G_1$ compute abP for $a, b \in Z_q^*$.
Bilinear diffie-Hellman (BDH) Problem: Let P be a generator of G_1. Given $\langle P, aP, bP, cP \rangle \in G_1$ compute $e(P, P)^a bc$, for $a, b, c \in Z_q^*$.

2.3 Public Key Cryptography

The public key cryptography (PKC) is introduced by Diffie and Hellman [5]. PKC involves two different keys for encryption and decryption instead of single key as symmetric key system. Since, public key is random string in PKC. Therefore, To prove the relation between entity and its public key, PKC adopts certificate mechanism where certificate-based protocols work by considering that each entity has a public and private key pair. These public keys are authenticated via certificate authority (CA) which issue a certificate. When two entities wish to establish a session key, a pair of ephemeral (short term) public keys are exchanged between them. The ephemeral and static keys are then combined in a way so as to obtain the agreed session key. The authenticity of the static keys provided by signature of CA assures that only the entities who posses the static keys are able to compute the session key.

2.4 Identity Based Public Key Cryptography (ID-PKI)

Identity-based cryptosystem eliminates the generation and distribution of entities public key problem by making each entity public key derivable from some known aspects of his identity, such as email address. Here, entities achieve their private key from a trusted third party called a Private Key Generator (PKG), after their authenticity verification. Shamir [13] introduced the concept of identity-based encryption (IBE) to simplify public key management procedures (or the public key distribution problem) by eliminating certificate-based public key infrastructure. However, the first fully functional pairing-based IBE scheme was proposed in [2]. Shortly after this, many pairing based cryptographic protocols were proposed. A survey over pairing based cryptography is presented in [4]. The identity-based PKI can be an efficient alternative of certificate-based PKI, especially when efficient key management and moderate security are required for large networks.

In an ID-based encryption scheme consists of four algorithms (*i*) Setup, (*ii*) Extract (Key generation), (*iii*) Encryption, and (*iv*) Decryption. For more details, one can refer [3].

2.5 Certificateless Public Key Cryptography (CL-PKI)

Certificateless cryptography is introduced in 2003 by Al-Riyami and Paterson [1]. It eliminates the necessity of certificate authority (CA) and removes key escrow problem in the system. It comprises of seven algorithms which are as follows:

Setup: It is a probabilistic algorithm run by the private Key Generator (PKG) which takes security parameter, randomly chosen master key and a list of public parameters such as description of message space and ciphertext space.
Partial Private Key-Extract: It is a probabilistic algorithm which run by the PKG. It takes input as user's identity $ID \in \{0,1\}^*$ and the master key. It returns partial private key.
Set Secret Value: It is a probabilistic algorithm which is perform by the entity. It takes list of public parameters and produces a random secret value for entity.
Set Private Key: It is a deterministic private key generation algorithm which is run by the entity. It takes input as entity partial private key and secret value, then outputs a private key.
Set Public Key: It is a deterministic public key generation algorithm which run by entity. It takes parameter and entity secret value, then computes entity public key.
Encrypt: It is a probabilistic algorithm which takes input as message, receiveridentity and public key. It outputs ciphertext.
Decrypt: It is a deterministic algorithm which takes a ciphertext and receiver private key. It returns original message.

3 Proposed Protocol

The basic architecture of proposed DRM system is similar to Liu et al. [10] system. Here, the content provider handles the content packing (encryption) work. Once the content encryption is over, it provides the content key with usage rules to the license server and protected content with content information to the distributor. License server authenticates the user, receives the payment, and generates the license. While, Distributor works as a service provider and facilitates the protected content distribution in the system. Parties involved in our DRM model are:

- Private key generator (PKG)
- Content provider (C)
- Distributor (D)
- License server (L)
- DRM User (U)

Content provider keeps the original unprotected digital contents and provides these contents for business use after their encryption. If it has r contents, namely, $M_1, M_2, \ldots, M_r$ with their unique identity $\mathsf{id}_{M_1}, \mathsf{id}_{M_2}, \ldots, \mathsf{id}_{M_r}$. Then, he generates r symmetric keys K_1, K_2, $K_3, \ldots, K_r$ and encrypts each content with an unique symmetric key and gets

$$E_{\mathsf{sym}}(M_i|K_i),\ i = 1, 2, 3, \ldots, r.$$

Content provider provides content decryption keys (key seeds) with usage rules and permissions to the license server through a secure channel. Distributor achieves encrypted contents $\{E_{\mathsf{sym}}(M_i|K_i),$ for all $i = 1, 2, 3, \ldots, r\}$ with content information from the Packager. Distributors keep protected contents over the media server and display content details over the website. To communicate securely in the system, entities achieve their secret partial keys with the help of packager and generates their public and private keys. In this process system usages five algorithms: Setup, Partial private key extract, Set secret value, Set private key and Set public key. Description of key generation process is as follows:

Setup: Private key generator (PKG) chooses an arbitrary generator $P \in G_1$, selects a master key $\mathsf{mk} \in Z_q^*$ and sets $\mathsf{PK} = \mathsf{mk}P$. It chooses hash functions $H_1 : \{0,1\}^* \to G_1^*$, $H_2 : \{0,1\}^k \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$, and $H : \{0,1\}^* \times \{0,1\}^* \times G_1 \times G_1 \times G_2 \to \{0,1\}^k$. Then, PKG publishes system parameters $\langle G_1, G_2, e(.,.), k, P, \mathsf{PK}, H_1, H_2, H\rangle$ and Keep master key mk secret.
Partial Private key extraction: License server (L) and user U submit their public identities ID_L and ID_U to the PKG. Then, PKG verifies the proof of identities. If verification succeeds, then generates the partial private keys in the following way:

- Compute $Q_L = H_1(ID_L)$ and$Q_U = H_1(ID_U) \in G_1^*$.
- By using its master key mk, PKG generates the partial private keys $W_L = \mathsf{mk}Q_L$ and $W_U = \mathsf{mk}Q_U$ and delivers these partial keys W_L and W_U to L and U respectively through a secure channel.

On receiving their partial private keys L and U can verify their partial keys respectively as follows:

$$e(W_L, P) = e(\mathsf{mk}Q_L, P) = e(Q_L, \mathsf{mk}P) = e(Q_L, \mathsf{PK})$$

$$e(W_U, P) = e(\mathsf{mk}Q_U, P) = e(Q_U, \mathsf{mk}P) = e(Q_U, \mathsf{PK}).$$

Private and public key extraction: L and U achieve their private and public keys as follows:

- L selects a secret value $x_L \in Z_q^*$ at random and keeps x_L secret. Then, L generates its private key SK_L by computing $SK_L = x_L W_L = x_L \mathsf{mk} Q_L$. L constructs its public key $PK_L = \langle X_L, Y_L \rangle$ where $X_L = x_L P$ and $Y_L = x_L \mathsf{PK} = x_L \mathsf{mk} P$.
- U selects a secret value $x_U \in Z_q^*$ at random and keeps x_U secret. Then, U generates its private key SK_U by computing $SK_U = x_U W_U = x_U \mathsf{mk} Q_U$. U constructs its public key $PK_U = \langle X_U, Y_U \rangle$ where $X_U = x_U P$ and $Y_U = x_U \mathsf{PK} = x_U \mathsf{mk} P$.

3.1 License Acquisition

User visits the distributor's website and selects some content with identity id_{M_t} and downloads encrypted content $E_{\mathsf{sym}}(M_t|K_t)$ from media server where media server provides free download of encrypted content. However, the encrypted content can not be played without the valid license where license server issues the license for authorized users. To acquire the license, user first establishes an authenticated key agreement protocol with license server, then achieves the license by using established secure communication from the license server. The detailed process is as follows:

Step 1. U chooses a random value $u \in Z_q^*$ and computes $T_U = uP$. Then, sends $\langle ID_U, T_U, PK_U \rangle$ to L.
Step 2. On receiving the user message, L selects a random value $l \in Z_q^*$ and gets $T_L = lP$. Then, L computes $Q_U = H_1(ID_U)$ and achieves S_L as follows:

$$S_L = e(Q_U, Y_U)^l \cdot e(SK_L, T_U) = e(Q_U, P)^{x_U \mathsf{mk} l} \cdot e(Q_L, P)^{x_L \mathsf{mk} u}.$$

Step 3. L computes $lT_U = luP$ and $x_L X_U = x_L x_U P$, then gets the session key as:

$$sk = H(ID_U || ID_L || luP || x_L x_U P || S_L).$$

Then, L sends $\langle ID_L, P_L, T_L, \mathsf{mac} \rangle$ to U where $\mathsf{mac} = H_2(sk || ID_U || ID_L)$.
Step 4. On receiving the message, U computes $Q_L = H_1(ID_L)$ and achieve S_U as follows:

$$S_U = e(Q_L, Y_L)^u \cdot e(SK_U, T_L) = e(Q_L, P)^{x_L \mathsf{mk} u} \cdot e(Q_U, P)^{x_U \mathsf{mk} l}.$$

Step 5. U computes $uT_L = ulP$ and $x_U X_L = x_U x_L P$, then gets the session key as

$$sk^* = H(ID_U||ID_L||ulP||x_U x_L P||S_U).$$

Then, U compute $\mathsf{mac}^* = H_2(sk^*||ID_L||ID_U)$ and checks the condition $\mathsf{mac}^* =?\ \mathsf{mac}$. If the condition hold, U selects his required content with identity id_{M_t}. Then, U encrypts id_M using session key sk and sends encrypted id_M with mac^* to L.
Step 6. On receiving the message, L verifies $\mathsf{mac} =?\ \mathsf{mac}^*$. If verification success, L decrypts the encrypted identity using sk and gets id_{M_t}. Then, L receives the payment. L allows two types of payment system which are as follows:

- *Prepayment*: user deposits an initial amount to the license server and gets a membership. A member can engage in a virtual finite number of interactions with the license server, to get the license at the total cost, which does not exceed the initial deposit amount.
- *Pays per item*: User need not to deposit any initial amount as an advance. In this case, the user pays at the time of license acquisition.

Step 7. On receiving the payment, L generates the license $license_{id_{M_t}}$ where license includes serial number, content key, usages rules and user's identity.
Step 8. L encrypts the license using symmetric session key sk and sends encrypted license to U. In addition, license server also maintain the record of usages license statistic for future business use.
Step 9. On receiving the message, U decrypts the message using session key sk and achieve the desired license $\mathsf{license}_{id_M}$. With the help of license, user can play the content.

User needs to established a session key only once. Once the session has established, a user can achieve any number of license in that session. To enhance the security, user can establish independent session keys for each session. An overview of pairing based authenticated key agreement protocol is given in figure 1.

4 Security Analysis

In this section, we will justify that proposed scheme provides authorized and secure communication between license server and user.

Passive attack: Eavesdropper can collect the information $\langle P, uP, lP, x_U\mathsf{mk}P, x_L\mathsf{mk}P, Q_U, Q_L, x_UP, x_LP\rangle$ which transmits via public channel. However, to compute $e(Q_L, P)^{x_L\mathsf{mk}u}$ and $e(Q_U, P)^{x_U\mathsf{mk}l}$ from given $\langle Q_L, uP, x_L\mathsf{mk}P\rangle$ and $\langle Q_U, lP, x_U\mathsf{mk}P\rangle$ respectively is equivalent to BDH problem. Where, BDH problem is hard to compute. Moreover, to achieve session key $sk = H(ID_U||ID_L||ulP||S_U)$, the values ulP is needed. But, to compute ulP from given $\langle P, uP, lP\rangle$ is equivalent to CDH problem which is hard.

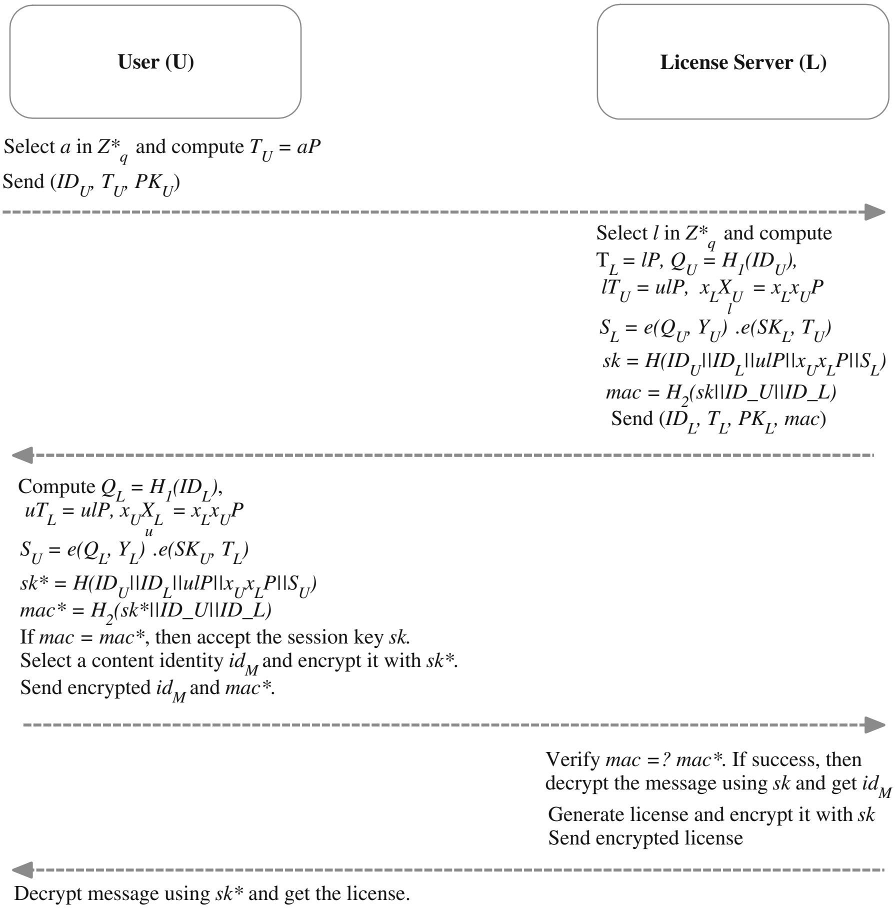
User (U)
License Server (L)
Select a in Z^*_q and compute $T_U = aP$
Send (ID_U, T_U, PK_U)
Select l in Z^*_q and compute
$T_L = lP$, $Q_U = H_1(ID_U)$,
$lT_U = ulP$, $x_L \underset{l}{X_U} = x_L x_U P$
$S_L = e(Q_U, Y_U) .e(SK_L, T_U)$
$sk = H(ID_U||ID_L||ulP||x_U x_L P||S_L)$
$mac = H_2(sk||ID\_U||ID\_L)$
Send (ID_L, T_L, PK_L, mac)
Compute $Q_L = H_1(ID_L)$,
$uT_L = ulP$, $x_U \underset{u}{X_L} = x_L x_U P$
$S_U = e(Q_L, Y_L) .e(SK_U, T_L)$
$sk^* = H(ID_U||ID_L||ulP||x_U x_L P||S_U)$
$mac^* = H_2(sk^*||ID\_U||ID\_L)$
If $mac = mac^*$, then accept the session key sk.
Select a content identity id_M and encrypt it with sk^*.
Send encrypted id_M and mac^*.
Verify $mac =? mac^*$. If success, then
decrypt the message using sk and get id_M
Generate license and encrypt it with sk
Send encrypted license
Decrypt message using sk^* and get the license.

Fig. 1. Proposed license distribution mechanism

Man in the middle attack: User and license server authenticate each other without knowing each other. An adversary or malicious PKG can try man in the middle attack by sending the forge message. However, to authenticate each other message, license server and user exchange mac and mac^* to each other. Where, to compute mac $= H_2(sk||ID_U||ID_L)$ requires to compute secret session key sk. To compute sk an adversary needs to compute $x_U x_L P$ and S_L or S_U, where to compute S_S or S_U require the secret share x_U and x_L and session secret values u and l information which are not known to An adversary or malicious PKG.

Known key attack: If an adversary achieves a session key, where session key $sk = H(ID_U||ID_L||ulP||S_U)$. It does not mean that other session keys can compromise. Because, each session key involves independent short-term secret values u and l which are different for each session.

Forward secrecy If an adversary achieves entities private keys then,

- *H*alf forward Secrecy: Compromise of the private key (x_U, SK_U) of user does not reveal previously established session keys because to achieve a session key, short time secret keys information is needed. Moreover, for given $\langle P, uP, lP \rangle$ to computation ulP is equivalent to CDH problem.
- *PKG forward secrecy:* Compromise of PKG master key does not reveal any information about session key because to achieve season key, the value (x_U, x_L) and (u, l) are needed, which can not be computed by using master key. Because, secret values (u, x_U) and (l, x_L) are randomly generate by L and U respectively. Moreover, for given $\langle P, x_UP, x_LP \rangle$ and $\langle P, uP, lP \rangle$ computation of x_Lx_UP and ulP are equivalent to CDH problem respectively.

Known session-specific temporary information attack: If short term secret keys u and l are compromised, then session keys does not reveals. Because, with short term secret keys u and l and given information $\langle Y_U = x_U\mathsf{mk}P, Y_L = x_L\mathsf{mk}P, Q_U, Q_L \rangle$ one can achieve S_U or S_L as:

$$e(uQ_L, Y_L) \cdot e(lQ_U, Y_U) = e(Q_L, P)^{x_L\mathsf{mk}u} \cdot e(Q_U, P)^{x_U\mathsf{mk}l}$$

However, to achieve $sk = H(ID_U||ID_L||luP||x_Lx_UP||S_L)$, the value x_Lx_UP is needed, where for given $\langle P, x_UP, x_LP \rangle$ computation of x_Lx_UP is equivalent to CDH problem.

Key off-set attack: When user send a message $\langle ID_U, T_U, PK_U \rangle$ to L. An adversary can replace it by $\langle ID_U, T_U^*, PK_U \rangle$ where $T_U^* = a^*T_U$. When, L computes

$$S_L^* = e(Q_U, P)^{lx_U\mathsf{mk}} \cdot e(Q_L, P)^{a^*ux_L\mathsf{mk}} \tag{1}$$

and achieves sk_1 and mac_1. L sends the message $\langle ID_L, T_L, P_L, mac_1 \rangle$ to U, adversary again change the message and sends $\langle ID_L, T_{L_1}, P_L, mac_1 \rangle$ where $T_L^* = a^*T_L = a^*lP$. U computes

$$S_U^* = e(Q_L, P)^{ux_L\mathsf{mk}} \cdot e(Q_U, P)^{a^*lx_U\mathsf{mk}} \tag{2}$$

Then, gets sk_1^* and mac_1^*. U, and concludes that $mac_1 \neq mac_1^*$ as by eq.(1) and eq. (2), $S_L^* \neq S_U^*$.

5 Conclusion

In this paper, we proposed a flexible and secure license distribution mechanism. In proposed mechanism, license server and user mutually authenticate each other and establish a session key, which ensures secure communication between them. Moreover, DRM principals communicate to each other using establish symmetric session key instead of public key, which is computationally feasible. Therefore, the proposed scheme is efficient and scalable for DRM system.

References

1. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
2. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
3. Chatterjee, S., Sarkar, P.: Identity-based encryption. Springer-Verlag New York Inc. (2011)
4. Dutta, R., Barua, R., Sarkar, P.: Pairing Based Cryptographic Protocols: A Survey. Manuscript (2004), http://eprint.iacr.org/2004/064
5. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
6. Dutta, R., Mukhopadhyay, S., Dowling, T.: 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)
7. Dutta, R., Mishra, D., Mukhopadhyay, S.: Vector Space Access Structure and ID Based Distributed DRM Key Management. In: Abraham, A., Mauri, J.L., Buford, J.F., Suzuki, J., Thampi, S.M. (eds.) ACC 2011, Part IV. CCIS, vol. 193, pp. 223–232. Springer, Heidelberg (2011)
8. Dutta, R., Mishra, D., Mukhopadhyay, S.: Access policy based key management in multi-level multi-distributor DRM architecture. In: Joye, M., Mukhopadhyay, D., Tunstall, M. (eds.) InfoSecHiComNet 2011. LNCS, vol. 7011, pp. 57–71. Springer, Heidelberg (2011)
9. Hwang, S.O., Yoon, K.S., Jun, K.P., Lee, K.H.: Modeling and implementation of digital rights. Journal of Systems and Software 73(3), 533–549 (2004)
10. Liu, Q., Safavi-Naini, R., Sheppard, N.P.: Digital Rights Management for Content Distribution. In: Proceedings of Australasian Information Security Workshop Conference on ACSW Frontiers 2003, vol. 21 (January 2003)
11. Ku, W., Chi, C.H.: Survey on the technological aspects of digital rights management. Information Security, 391–403 (2004)
12. Sachan, A., Emmanuel, S., Das, A., Kankanhalli, M.S.: Privacy Preserving Multiparty Multilevel DRM Architecture. IEEE Consumer Communications and Networking Conference (CCNC 2009), 1–5 (2009)
13. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
14. Imai, H., Zheng, Y. (eds.): PKC 1998. LNCS, vol. 1431. Springer, Heidelberg (1998)
15. Yen, C.T., Liaw, H.T., Lo, N.W.: Digital rights management system with user privacy, usage transparency, and superdistribution support. Int. J. Commun. Syst., doi:10.1002/dac.2431
16. Zhang, Z.Y., Pei, Q.Q., Ma, J., Yang, L.: Security and trust in digital rights management: a survey. International Journal of Network Security 9(3), 247–263 (2009)