Efficient and Portable Digital Rights Management for Preserving User Privacy Using Smart Cards

Hsing-Bai Chen Wei-Bin Lee*

Department of Information Engineering and Computer Science

Feng Chia University

Taichung 407, Taiwan, ROC

{P9317594, wblee}@fcu.edu.tw

Received 20 October 2007; Revised 15 December2007; Accepted 10 January 2008

Abstract. Digital Rights Management (DRM) system is a mechanism used to control the use of digital content according to the usage rules specified in the license. To avoid unauthorized accesses, DRM systems would bind the unique hardware ID to the license. In this way, however, it is inconvenient for consumers to play the purchased content on another device, and is uneasy about the information about what he/she purchases can be collected easily according the unique hardware ID. Recently, some novel DRM systems based on smart cards have been introduced to address these problems. However, some drawbacks on the security, practicability, and efficiency exist in these systems, which will be shown in this paper. In this paper, a smart card based DRM scheme addressing their drawbacks is proposed to provide portability, user privacy, security, practicability, and efficiency.

Keywords: digital rights management, e-commerce, security, smart, card, user privacy

1 Introduction

To avoid digital piracy and protect commercial digital intellectual property, Digital Rights Management (DRM) is recognized as a practical mechanism that prevents unauthorized access to digital contents and control their distribution and usage. Microsoft Media DRM system [1] and Apple iTunes [2] are the most famous DRM systems. These DRM systems apply a set of policies and techniques to facilitate the content owner to specify the desired ownership rights of the content and to guide the proper use of digital content.

However, the protection of user privacy is absent from the design of these DRM systems while the number of people who suffer from events caused by the infringement of individual information dramatically increases, such as installing spyware on rendering device happened to Sony BMG in late 2005 [3,4]. Besides that, consumers are also concerned for that their personal information is revealed over online transactions due to uncontrolled and public networks. Consequently, consumers look forward to a good DRM solution that can regard privacy protection and grant anonymous access to digital contents [5].

In addition, in Microsoft Media DRM system and Apple iTunes, a digital content is only permitted to be played on a specific device according to the license purchased. This model is classified as device-based DRM system [6]. In this device-based DRM system, the purchased license is bound to the hardware ID of this specific device. A consumer who has two or more devices is required to purchase two or more license for rendering a desired content on these devices he/she owned. Contrary to the digitalized world, the consumer whishes to play physical contents purchased, such as CD, DVD, etc., on any player in the real world. This business model does not accord with the model in the real world and will decline sharply the aspiration for consuming.

For the convenience, the smart card based DRM systems [6,7,8] in which the purchased license is bound to a secret element stored into particular smart card not to a specific device are presented recently. The consumer who owns the smart card can render the desired content on any compliant device that installs a card reader and complies with a given standard and adheres to certain usage rules in a DRM system. Succinctly, due to the portability character of smart cards, the license bound to a particular smart card can be floated around among all compliant devices to make the consumer enjoy the rendered content anywhere and any time. However, we find that these schemes have several drawbacks involving its practicality, its lack of content protection, no user privacy, and low no efficiency. For addressing these problems, this paper presents an efficient scheme with smart cards to permit that the consumer can purchase the rights to render the corresponding content anonymously and efficiently in a typical DRM architecture.

* Correspondence author

The rest of this paper is organized as follows. In Section 2, the investigation of related smart card based DRM systems is given. In Section 3, the proposed DRM scheme is introduced. In Section 4, discussions on security and efficiency are given. Finally, concluding this paper is given in Section 5.

2 Overview of Smart Card Based DRM Systems

In this section, an investigation into related some smart card based DRM systems [6,7,8] based on their practicality, content protection, user privacy, and efficiency are made as follows:

1. **Conrado *et al.*'s DRM scheme** [8]
   Overview: Conrado *et al*.'s scheme is the fist user privacy protection DRM system based on smart cards. This scheme consists of two roles—the consumer and the license server. For anonymous transactions, a secret key and the public key of the consumer are used together to generate an anonymous verifier. The license server does not authenticate the consumer but just check whether the payment is made. Whenever the content is rendered, the anonymous verifier is generated. Only if the smart card can generate a correct verifier, the content can be enjoyed the consumer.
   Drawback: Sun *et al*. [6] pointed out that there are two drawbacks in the Conrado *et al*.'s scheme. The first is that the content is not protected, which may be accessed on an incompliant device. The second is that the DRM architecture only involving the consumer and the content provider is not practical in the real world.

2. **Sun *et al.*'s DRM scheme** [6]
   Overview: Generally, this paper follows the design of Conrado *et al*.'s scheme but addresses the drawback of Conrado *et al*.'s scheme. Compared with the design of the Conrado *et al*.'s scheme, the public key pair of the consumer is replaced with secret keys and the participant is extended to four roles—the consumer, content provider, content server, and license server—in the Sun *et al*.'s scheme.
   Drawback: We have some comments on the efficiency and the architecture. First, following the design of the Conrado *et al*.'s scheme causes some redundancies while the component of anonymous verifier is secret key that is confidential instead of public key pair that can be used to identify the owner. These redundancies will spend more computation resource and storage space. Secondly, from the view point of the architecture, all roles can obtain the content key except only the content provider. This means that the content provider cannot confirm its ownership while the content server owning both the content and the content key is malicious. To address this problem, a typical DRM architecture in which the content server plays the role of only the distributor like [5] is preferable.

3. **Lee *et al.*'s DRM scheme** [7]
   Overview: Lee *et al*.'s scheme is implemented in a typical DRM architecture [5] and presents a new method applying the subliminal channel technique. Compared with Conrado *et al*.'s and Sun *et al*.'s schemes, Lee *et al*. presents an efficient way to control the content access in which the content key is directly generated and is used to check the authorization according to whether the content can be rendered successfully.
   Drawback: The user privacy is not considered in to the design of Lee *et al*.'s scheme.

Accordingly, still now, no DRM system based on smart cards can fully achieve the practicality, content protection, user privacy, and efficiency.

3 The Proposed Scheme

In this section, a portable and privacy-preserved DRM scheme based on smart cards is presented. The proposed DRM scheme consists of four phases: the initialization phase, the purchase phase, the play phase, and the play-on-another-device phase. Before illustrating this scheme, the environment of our DRM system is described.

3.1 Overview

Architecture of a typical DRM system consisting of information and money flows is shown in Fig. 1 [5]. Four major components involved in a typical DRM system are described as follows:

1. **The content provider** such as a music record label, a movie studio, e-publisher, etc. who holds the ownership rights of the content and is concerned with piracy of these content. The content provider encrypts the content using a content key to protect the content from outsiders. The content provider also creates appropriate usage rules specifying the permitted ways to render the corresponding content.
2. **The content distributor** provides various distribution channels including the delivery of content either on-demand over the Internet or offline on CDs and DVDs, such as a web retailer, an online shop, etc. After receiving content from the content provider, the content distributor packages the received content into an appropriate form for easy distribution and creates a web catalogue displaying the packaged content for user's downloading.
3. **The license server** creates a license for certain content according to the corresponding usage rules specified by the content provider. In general, the license server handles the financial transaction for issuing the license to the consumer and then provides royalty fees to the content provider and appropriate remuneration for the content distributor. Note that the details in this financial transaction are beyond the scope of this paper.
4. **The consumer** used the DRM system to consume and render the desired content through a compliant device. In the smart card based DRM system, the compliant device is equipped with a card reader and requires a smart card to enable all operations of this DRM system. In our scheme, assume that the compliant device and the smart card both possess protected memory to confidentially store secret information.

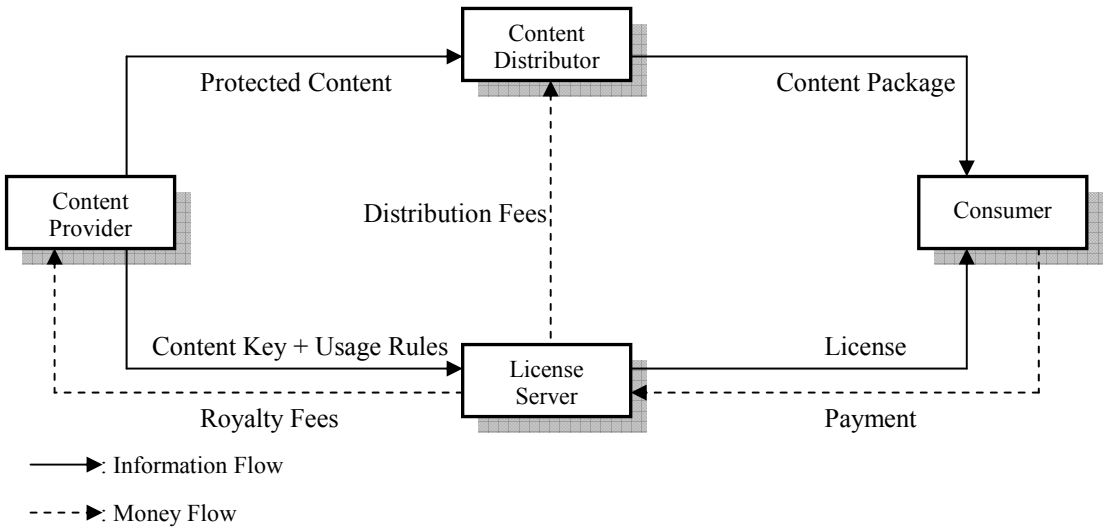

Fig. 1. High-level architecture and major components of a typical DRM system [5]

3.2 Initialization Phase

Before performing any procedures, the initialization phase (see Fig. 2) must be performed as follows to prepare for the content delivery and licenses issuing:

Step 1— The content provider creates a raw content, denoted as *content*, and sets the corresponding *usage_rules* that guide the proper use of this content. After that, the content provider works as follows:

1. Encrypt this content using a content key *CK* which is randomly generated by the content provider, as $E_{CK}(content)$ for secure transmission, where $E_K(.)$ denotes a symmetric encryption with key *K*, such as AES [9] or TDES [10].

2. Send {*usage_rules*|| *H*(*CPid*|| *SN*)|| *CK*} to the license server through SSL (Secure Socket layer) protocol, where *H*(.) denotes a secure one-way hash function, such as SHA-512 [11], || denotes the concatenation operation, *CPid* is the identity of content provider, and *SN* is serial number generated by the content provider.
3. Transmit the encrypted content {*License_URL*|| E_{CK}(*content*)} to the content distributor, where *License_URL* is the address directing to the license server and would involve information about *H*(*CPid*|| *SN*) used to be bridge of usage rules, *CK*, and E_{CK}(*content*).

Step 2— The content distributor processes the following operations for distributing the encrypted content:
1. Package the received content from the content provider as {*Cid*|| *License_URL*|| E_{CK}(*content*)}, where *Cid* is content identity.
2. Create a web catalogue presenting this package for potential consumers.

Step 3— The license server stores {*usage_rules*|| *H*(*CPid*|| *SN*)|| *CK*} into the database securely.

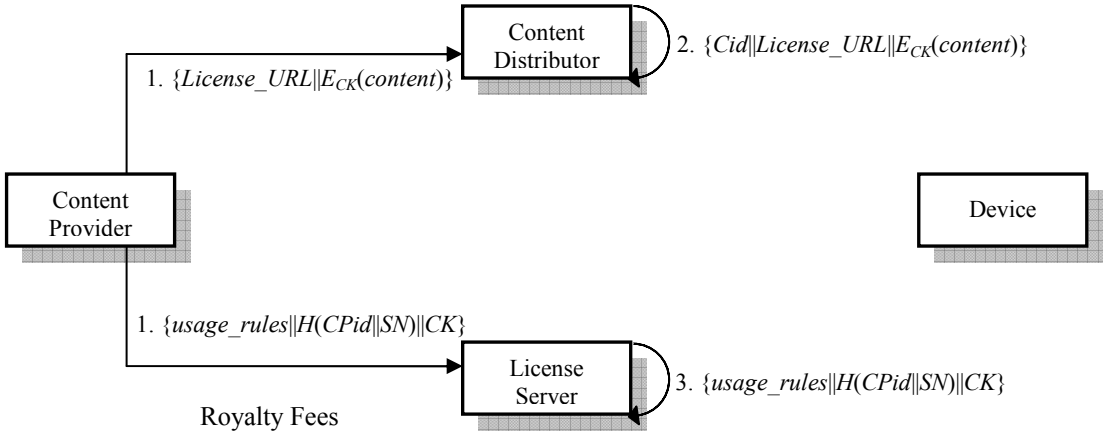

Fig. 2. The procedure of the initialization phase

In addition, each consumer receives a unique smart card in this DRM scheme. Each smart card stores a secret key *Ku*, used to protect the content key involved in the license, into the protected storage secure against burglary inside the smart card.

3.3 Purchase Phase

Once the consumer attempts to purchase contents, the purchase phase is triggered as follows:

Step 1— The consumer connects with content distributor by using his device to survey what he/she wants and then press the "Download" button.

Step 2— The consumer's device stores the downloaded content package {*Cid*|| *License_URL*|| E_{CK}(*content*)} from the content distributor.

Step 3— The smart card performs the following processes to purchase the corresponding license:
1. Recognize the consumer as the card owner in some appropriate ways, such as entering password, personal identification number (PIN), etc. If the consumer cannot be authenticated, terminate this service.
2. Calculate an anonymous key *AK* = *H*(*Ku*|| *Cid*), used to encrypt the content key in the license, with *Ku*.
3. Compute E_{PKls}(*SSI*|| *AK*) to protect the confidentiality of *SSI* and *AK*, where *PKls* is the public key of the license server, and *SSI* (Secret Security Identifier) corresponds to the pre-paid amount of money and is a kind of e-cash [12].
4. Connect with *License_URL* to transmit {*Cid*|| E_{PKls}(*SSI*|| *AK*)|| *H*(*CPid*|| *SN*)} to the license server for purchasing the corresponding license.

Step 4— The license server generates a license as follows:
1. Extract *SSI* and *AK* by decrypting E_{PKls}(*SSI*|| *AK*) with the private key of the license server.

2. Verify *SSI* and check whether the amount of money is correct. Only if they hold, the following operations progress.
3. Encrypt *CK* with *AK* as ($CK \oplus AK$), where $\oplus$ denotes bitwise XOR operation.
4. Create a corresponding license as $License = Sign_{LS}\{Cid \| (CK \oplus AK) \| usage\_rules\}$, where $Sign_{LS}\{.\}$ denotes a signature of the message sealed by the license server.
5. Transmit *License* to the device.
6. Generate the corresponding *index* by computing $\{Cid \| H(AK)\}$ for easy search because the same *Cid* will correspond to numerous *License* purchased by different consumers.
7. Store $\{index \| License\}$ in database and then establish relationship with the stored $\{usage\_rules \| H(CPid \| SN) \| CK\}$.

After receiving *License*, the consumer stores *License* in the device. The procedure of the purchase phase is illustrated in Fig. 3

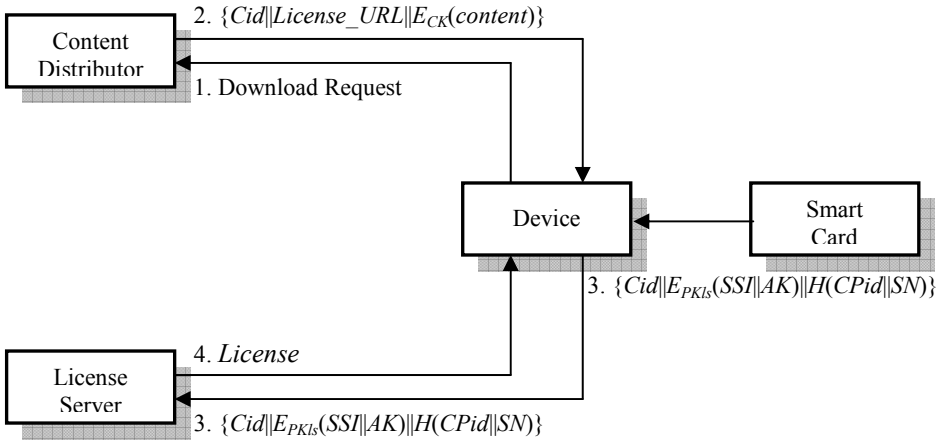

Fig. 3. The procedure of the purchase phase

3.4 Play Phase

Whenever the consumer chooses the *Cid* of the desired content from the playlist collecting all downloaded package into the device to render the corresponding *content*, he/she presses the "Play" button to trigger the following events (shown in Fig. 4) to bring the consumer to the enjoyment of the content:

Step 1— The device looks for the corresponding *License* according to *Cid*. If *License* does not exist in the device, it will trigger the following events for requesting the *License*:
1. The smart card computes $AK = H(Ku \| Cid)$ and then sends $\{Cid \| H(AK)\}$ as request for *License* to license server.
2. The license server treats the request as *index* to finds the corresponding *License* which is concatenated by the same *index*. If *License* is found, the license server will respond *License*. Otherwise, it means that the consumer who holds this smart card never purchase *License* and is required to go to **Step 3** in **Purchase Phase** if he/she wants to enjoy the content.

Step 2— The device transmits *License* to the smart card.
Step 3— The smart card performs the following works with *License* to prepare for playing the content:
1. Compute $AK = H(Ku \| Cid)$.
2. Extract *CK* by decrypting ($CK \oplus AK$) with the computed *AK*.
3. Retrieve *content* by decrypting $E_{CK}(content)$ with the extracted *CK*.

After **Step 3**, the *content* and *CK* are transferred to the protected memory in the device to avoid the access of any unauthorized applications. Then, *content* can be played on this device.

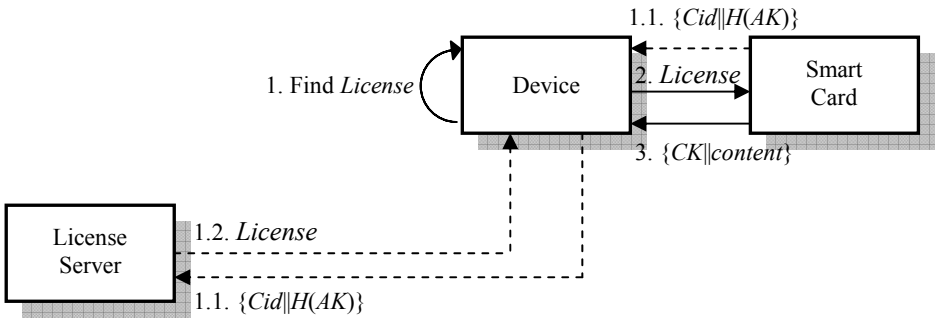

Fig. 4. The procedure of the play phase

From the perspective of dispute between the license server and the consumer when certain *CK* in *License* signed by the license server cannot make *content* play well, this issue can be functionally addressed by applying the non-repudiation property of digital signatures and is ignored here. Under this situation, if *content* fails in playing, the validity of *License* will be verified with the public key of the license server. If the verification is failed, it will re-request *License* from the license server.

3.5 Play-on-Another-Device Phase

Once the consumer wants to play certain content whose header is *Cid* on another compliant device, he/she needs to insert his/her own smart card into this compliant device. Then, the play-on-another phase is done as follows and illustrated in Fig. 5:

Step 1— The device looks for the content package and the corresponding *License* according to *Cid*. If the content package or the corresponding *License* is not found, the following events are performed:

1. The consumer connects with content distributor through this device and then downloads this content package $\{Cid\| License\_URL\| E_{CK}(content)\}$ into this device if the content package does not exist in the device.
2. The smart card calculates key $AK = H(Ku\| Cid)$ with *Ku* and then sends $\{Cid\| H(AK)\}$ as request to license server for *License* if the corresponding *License* does not exist. The license server looks for the *License* according to the request $\{Cid\| H(AK)\}$. If *License* is found, the license server will respond *License*. Otherwise, it means that this consumer never be authorized to possess *License* whose header is *Cid*. The license server will request the smart card to perform **Step 3** in **Purchase Phase** if the content still is desired to play.

Step 2— The device transmits *License* to the smart card while receiving *License*

Step 3— The smart card triggers the following activities:

1. Compute $AK = H(Ku\| Cid)$.
2. Extract *CK* by decrypting $(CK \oplus AK)$ with the computed *AK*.
3. Retrieve *content* by decrypting $E_{CK}(content)$ with *CK*.

After **Step 3**, the *content* and *CK* are transmitted to the protected memory in the device. Then, the device plays *content* to make the consumer enjoyment in *content*.

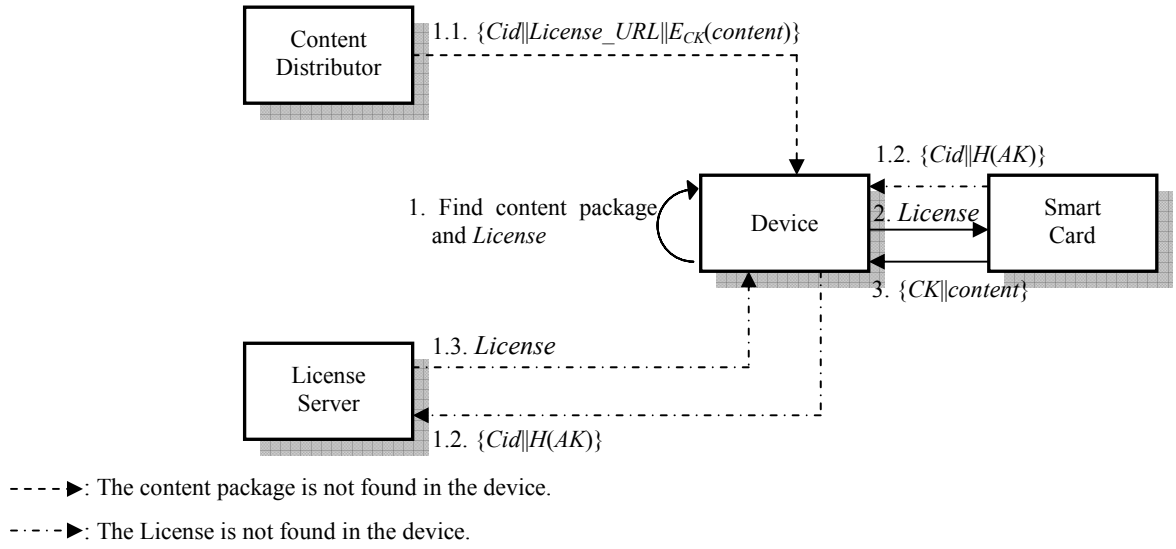

Fig. 5. The procedure of the play-on-another-device phase

4 Discussion

In this section, security and efficiency analysis are given to prove that our DRM scheme is not only secure but also efficient. Besides, a comparison of the proposed scheme and the related DRM systems is made to illustrate that the proposed scheme is more superior to others.

4.1 Security Analysis

Prior to demonstration of security, some assumptions of security are given.

Assumption 1. *There is no dispute among the content provider, the content contributor, and the license server in a DRM system*. Some mechanisms are applied in the DRM system to provide the availability and non-repudiation of the message transmitted among these servers. If the dispute exists in the DRM system, the content provider does not give the content key and the content to the license server and the content distributor, respectively and vice versa, and therefore the DRM system cannot work well. Hence, it is reasonable to keep the DRM successfully working.

Assumption 2. *The consumer's secret key stored into the smart card is kept secret*. In practice, the smart card has the protected mechanism that is secure against the burglary inside the smart card. As the result, any attacker has no way to get the information stored in the smart card even though the consumer who owns this card.

We also highlight the following facts to facilitate the security proof. First, a one-way hash function possesses the properties of collision-free and irreversibility in which it is hard to input two different values to generate the same hash value and computationally unfeasible to find the input the hashes to that value according to the hash value, respectively [13]. Second, every symmetric key cryptosystem is secure to protect confidential information against any cracking.

Based on these assumptions, the security of the proposed protocol is further examined as follows:

Proposition 1. *Content protection: unauthorized consumer cannot render the protected content with his/her smart card*.

Proof. Whenever the content is desired to be played, the consumer's smart card will enable the process looking for the corresponding *License* to check whether the consumer has been authorized to render the content. The following potential cases are discussed to hold this claim:

Case 1. The consumer does not purchase the corresponding *License*. The consumer's smart card will not find the corresponding *License*. Deservedly, the corresponding *License* does also not exist in the license server. Without *License*, it is obvious that it will fail in playing the content.

Case 2. The consumer makes effort but not pay to get the *License*. With the *License* purchased by someone but not the consumer, the consumer can derive a content key from the *License* by using his/her anonymous key $AK' = H(Ku' \| Cid)$. Because the correct content key involved in *License* is encrypted with an purchased consumer's anonymous key $AK = H(Ku \| Cid)$. Without the purchased consumer's smart card stored *Ku* not *Ku'*, it is clear that an incorrect content key will be derived by using *AK'*. Hence, the unauthorized consumer will retrieve an incorrect content key and cannot render the content when he/she gains a valid *License*.

Proposition 2. *User privacy: the license anonymously links the consumer and the content and no one can recognize who purchase this license*.

Proof. During the transaction, the license server issues *License* according to whether the amount of e-cash *SSI* is correct instead of the authentication of the consumer. The license server authorizes the anonymous key not the specific consumer in a license and the consumer is permitted by the consumer's smart card to render the content only when the correct anonymous key $AK = H(Ku \| Cid)$ is retrieved. It is obvious that the only link between the consumer and the content is *AK*. However, according to **Proposition 1**, no one can gain a valid *AK*. As this result, nobody can recognize who purchases a particular content. In addition, consider the following case: the license server is the only one who knows *AK* and attempts to crash the user privacy. When the signal consumer purchases different contents which do not possess the same *Cid*, the different *AK* can remove the linkable relations between the content and the consumer. Hence, the license server has no way to detect which *AK* is possessed by the same consumer when the identity of his/her device is protected or the device used for purchasing is not the same. For the license server, the secret key *Ku* that is camouflaged in the *AK* may be used to identify which contents are purchased by the same consumer. Unfortunately, if the license server intends to derive *Ku* from *AK*, the license server will face the challenge to break the irreversible property of one-way hash functions. Accordingly, the user privacy can be achieved in this DRM scheme.

Proposition 3. *Portability: a consumer who purchases the content and owns a smart card can render the content on any compliant device*.

Proof. As subsection 3.5, it is clear that the authorized consumer held his/her smart card that stores correct *Ku* can access to certain content through any compliant device anywhere and anytime. The security mechanism of the smart card facilitates the portability of DRM and therefore the consumer is not restrained to only one specific device anymore.

Proposition 4. *Superdistribution:* a *valid license transmitted from the authorized consumer to unauthorized consumers cannot provide the sharing in the protected content*.

Proof. After purchasing *License*, the authorized consumer attempts to share this *License* with his/her friends. The compliant devices that have received *License* from the device of the authorized consumer will not render the desired content successfully because an incorrect content key is retrieved by the inserted smart card without the valid secret key *Ku*. Hence, any unauthorized sharing in the protected content is resisted.

Proposition 5. *Integrity of license: any unauthorized modification in the license can be detected*.

Proof. *License* is publicly transmitted over the Internet. It is possible that a prank is played on the *License* to aim to compromise the progress of a DRM system. In this paper, *License* is sealed with the private key of the license server. Any modification of *License* will fail to retrieve the content key. And the modification will be detected by the consumer by the signature verification on *License* with the public key of the license server.

4.2 Comparisons

The comparison among our scheme and the other smart card based DRM systems involving Sun *et al.*'s and Lee *et al.*'s DRM schemes is given in Table 1. Because Sun *et al.*'s scheme is an improvement of Conrado *et al.*'s, Conrado *et al.*'s scheme is not discussed here. To fit the DRM architecture in the real world, these schemes consist of four roles involving the content provider, content server/distributor, license server, and consumer. However, in Sun *et al.*'s DRM system, only the content provider does not have the knowledge of the content key.

Compared with Lee *et al.*'s and our methods, the content server not only is a distributor but also protects the content instead of the content provider in the Sun *et al.*'s scheme. It implies that the implementation the Sun *et*

al.'s scheme is costlier than Lee *et al*.'s and our methods because finding a trusted party as the content server and protecting the content server against any compromise are required.

From the view point of security, all schemes use a cryptographic technique to protect the content. Compared with others, Lee *et al*.'s scheme does not provide user privacy.

From the perspective of secure storage needed, our and Lee *et al*.'s schemes demand to keep only one secret key in the smart card. It is clear that our and Lee *et al*.'s schemes can save storage space in smart card more than the Sun *et al*.'s scheme. In addition, for computational cost, the operation in play phase is the major burden in a DRM system because performing the phase is the most frequent by the consumer. In play phase, our scheme requires to perform one symmetric key decryption such as AES, one hash function operation such as SHA-512, and one XOR operation. According to [14], the benchmarks of AES and SHA-512 are 84 MiB/Second and 99 MiB/Second, respectively. It is obvious in Table 1 that our scheme is more efficient than others.

Table 1. The comparisons of smart card based DRM systems

DRM System	Sun *et al*. [6]	Lee *et al*. [7]	Our scheme
Involved Roles	Content Provider, Content Server, License Server, Consumer	Content Provider, Content Distributor, License Server, Consumer	Content Provider, Content Distributor, License Server, Consumer
Role Unknowing Content Key	Content Provider	Content Distributor	Content Distributor
Secret Data Stored in Smart Card	Two secret Keys	One secret Key	One secret Key
Content Protection	Yes	Yes	Yes
User Privacy	Yes	No	Yes
Providing Portability	Yes	Yes	Yes
Providing Superdistribution	Yes	Yes	Yes
Computational Cost Needed in Play Phase	$3T_{Sym}$[1] + $2T_H$[2]	$2T_{Sym} + 2T_{XOR}$[3]	$1T_{Sym} + 1T_H + 1T_{XOR}$

1: The cost required to perform a symmetric-key en/decryption operation
2: The cost required to perform a one-way hash function operation
3: The cost required to perform a XOR operation

4.3 Efficiency Analysis

The validity of the license can be verified with the public key of the license server. From the computational capability of the smart card, time-consuming signature verification brings a heavy burden. In the proposed scheme, yet it is not necessary to perform the signature verification on the license all of the time. According to **Propositions 1-4**, it is believe that only the valid license will lead to a correct content key derived by the smart card and vice versa. If the content key is correct, the validity of the signature on the license is convinced. Hence, the smart card can save the cost of signature verification to speed up the performance of this proposed scheme. Furthermore, according to Table 1, it is clear that the computational cost of our scheme is more lightweighted than other related DRM systems.

In sum, the content protection and the user privacy can be achieved in our proposed DRM scheme. Besides, the proposed scheme preserves not only practical DRM architecture but also efficiency property.

5 Conclusion

This paper presents a new smart card based DRM scheme. In the proposed scheme, the consumer is permitted to anonymously purchase his/her desired contents and render them anywhere and anytime. Furthermore, the scheme is more efficient than other smart card based DRM systems. Duo to the implementation based on a typical DRM architecture, the proposed scheme can be applied into the real world smoothly.

References

[1] Microsoft Media Digital Rights Management, Available: http://www.microsoft.com/windows/windowsmedia/forpros/drm/default.mspx, 2007.

[2] Apple iTunes, Available: http://www.apple.com/itunes/, 2007.

[3] D. Bradbury, "Decoding Digital Rights Management," *Computers & Security*, Vol. 26, No. 1, pp. 31-33, Feb. 2007.

[4] E.W. Felten and J. A. Halderman, "Digital Rights Management, Spyware, and Security," *IEEE Security & Privacy Magazine*, Vol. 4, No. 1, pp. 18-23, Jan.-Feb. 2006.

[5] Q. Liu, R. Safavi-Naini, and N. P. Sheppard, "Digital Rights Management for Content Distribution," in *Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003*, Adelaide, Australia, Vol. 21, pp. 49-58, 2003.

[6] H. M. Sun, C. F. Hung, and C. M. Chen, "An Improved Digital Rights Management System Based on Smart Cards," in *Proceedings of the 2007 Inaugural IEEE International Conference on Digital EcoSystems and Technologies*, Cairns, Australia, pp. 308-313, Feb. 2007.

[7] W. B. Lee, W. J. Wu, and C. Y. Chang, "A Portable DRM Scheme Using Smart Cards," *Journal of Organizational Computing and Electronic Commerce*, Vol. 17, No. 3, pp. 247-258, 2007.

[8] C. Conrado, F. Kamperman, C. J. Schrijen, and W. Jonker, "Privacy in an Identity-based DRM System," in *Proceedings of the 14th IEEE International Workshop on Database and Expert Systems Applications*, pp. 389-295, 2003.

[9] FIPS PUB 197, *Specification for the Advanced Encryption Standard (AES)*, Nov. 2001.

[10] National Institute of Standards and Technology, *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, Technology Administration, U.S. Department of Commerce, May 2004.

[11] National Institute of Standards and Technology, *Federal Information Processing Standard (FIPS) PUBS 180-2, Secure Hash Standard*, Available: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf, Aug. 2002.

[12] M. T. Maat, "The Economics of E-Cash," *IEEE Spectrum*, Vol. 34, No. 2, pp. 68-73, Feb. 1997.

[13] B. Schneier, *Applied Cryptography*, 2nd Edition, John Wiley and Sons, New York, 1996.

[14] W. Dai, "Crypto++ 5.5 Benchmarks," Available: http://www.eskimo.com/~weidai/benchmarks.html, Dec. 2007.