Information Systems International Conference (ISICO), 2 – 4 December 2013

Secure and Private Content Distribution in the DRM Environment

Antonius Cahya Prihandoko*, Hossein Ghodosi, Bruce Litow****
* School of Business (IT), James Cook University, Australia & Information System Dept., Jember University, Indonesia
** School of Business (IT), James Cook University (JCU), Townsville, QLD 4811, Australia

Keywords:

Digital Rights Management
Content Distribution System
Oblivious Transfer
Security
Privacy

ABSTRACT

Digital Rights Management (DRM) is required to provide balanced protection for both the content provider and the users in a content distribution system. The content provider demands secure content delivery so that only authorized users are able to access the content and use it properly. On the other hand, users require that their privacy be protected. However, most DRM systems tend to put greater emphasis on content providers' security and neglect users' privacy. This study aims to improve DRM by constructing a content distribution protocol that preserves the security of content provider and the privacy of users. To achieve this goal, we utilize the oblivious transfer (*OT*) concept. This concept allows a sender to securely send a set of information to a receiver in such a way that, at the end of the protocol, the receiver cannot learn more than he was supposed to learn, while the sender cannot determine what the receiver has learned. Assuming that tamper-proof device exists, the constructed protocol achieves perfect security for the content provider and privacy for the users. This oblivious content distribution ultimately enables DRM to be a privacy-aware protection system. The system does not merely focus on content providers' rights, but also seriously considers users' privacy protection.

Corresponding Author:

Antonius Cahya Prihandoko,
School of Business (IT), James Cook University
Townsville, QLD 4811, Australia.
Email: antonius.cahyaprihandoko@my.jcu.edu.au

1. INTRODUCTION

Secure content delivery is urgently required in digital content distribution systems. This form of content delivery aims to guarantee that only authorized users can access protected content. Digital Rights Management (DRM) is a popular approach to this security requirement. Under DRM protection, digital content is usually encrypted before it is delivered. Some methods, such as code obfuscation [1] and white-box cryptography [2-4], may also be applied to enhance security by modifying the implementation of the encryption algorithms. Users need to acquire an adequate license to decrypt and use the protected content properly.

Focusing on securing content delivery, the DRM systems often put a great emphasis on content providers' security and pay little attention to users' privacy. The systems usually collect users' personal data to allocate appropriate content usage rights to them. The users, however, lack information on how and when the content provider uses their data. This situation increasingly invades users' privacy and, thus, reduces users' satisfaction. Therefore, protecting users' privacy has to be seriously considered. DRM systems need to provide balanced protection for content providers and users [5]. The system must not merely focus on achieving security for content providers, but also on preserving privacy for users.

A typical DRM for content distribution consists of four parties (see Figure 1): content provider, distributor, clearing house, and consumer (user) [6]. First of all, the content provider encrypts the content for security purposes. The provider then passes the protected content to the distributor and the corresponding usage rules to the clearinghouse. The distributor makes the protected content available on a web server. A consumer can retrieve the content through the distribution channel and requests a license from the clearinghouse. The consumer has to register his profile, provide details of the purchased content, and then make a payment. After verifying the consumer's identity and other related information, and charging the consumer's account, the clearinghouse releases a license and delivers it to the consumer. The consumer decrypts and uses the content based on the rights described in the license.

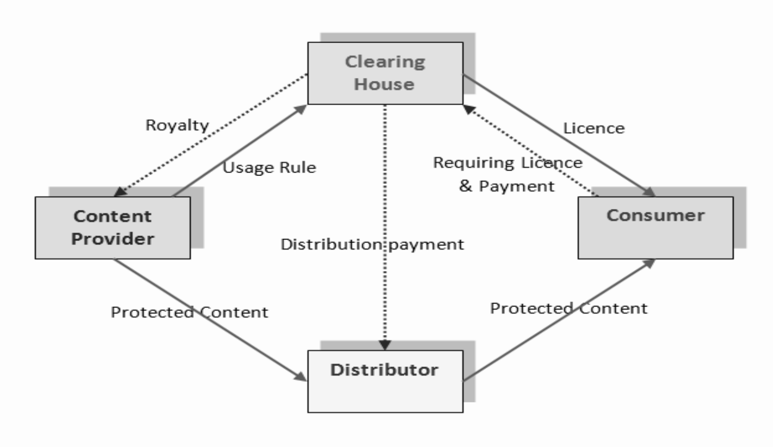


Figure 1. Typical DRM process for content distribution [6]

Most DRM systems make protected digital content available on their servers. This strategy preserves both security and privacy. On one hand, downloading protected content from the distributor's web does not seriously threaten the security of the content provider. The downloader cannot unlock the content, unless he receives the proper decryption key. On the other hand, in this stage, users can download the protected content he chooses while the provider cannot determine who is downloading which content. This mechanism clearly protects the users' privacy.

In contrast, acquiring a license for the clearinghouse creates a cause for concern over security and privacy. From the content providers' perspective, this mechanism may threaten to their security. If an eavesdropper steals licenses when a user requests them from the clearinghouse, revenue will be lost. From the users' perspective, the mechanism creates threat to their privacy. The personal information they submit to the clearinghouse is not guaranteed to be kept secret, as the clearinghouse may send the users' data and viewing detail to marketing agencies. The users expect that they have their privacy protected and are able to access digital content anonymously.

To overcome the problem, we construct a content distribution protocol by utilizing the oblivious transfer concept. Oblivious Transfer (OT) is a cryptographic protocol that allows two parties to privately exchange one or more secret messages. An OT protocol has to be set up in such a way that it will achieve security for the sender and privacy for the receiver [7]. The former means that the receiver will not be able to learn more than he was supposed to learn. The latter means that the sender will not know what the receiver has learned. The first OT protocol, introduced by Rabin [8], was intended to overcome the exchange of secrets (EOS) problem. This protocol enables a sender to deliver a message to a receiver in such a way that the receiver can access the message with probability $1/2$ and the sender will not know whether the message was received. Rabin's protocol was then generalized to the OT_1^2 [9] . In the OT_1^2 protocol, the sender has two secret messages and the receiver wishes to learn one of them. At the end of the protocol, the sender does not know which message was chosen while the receiver knows nothing of the unselected message. This scheme has been studied extensively and generalized to a wide variety of models including OT_1^N [10-12] and OT_K^N [13, 14]. To achieve an unconditional secrecy, a protocol may utilize a trusted initializer [15] to sends some information to both the sender and the receiver at the initialization step. Utilizing a trusted party, however, is unacceptable in the privacy preserving applications [16]. To omit the trusted party, Naor and Pinkas [16] proposed a distributed oblivious transfer (DOT) in which the task of the sender is distributed among several servers. The security of the DOT protocols has been intensively studied [17-20].

The efficiency of the system is also an important issue in the implementation of an OT protocol. OT is unlikely to be based on more efficient one-way functions or other private-key cryptographic primitives [21]. As a result, all known OT protocols needs public-key operations that are typically implemented using modular exponentiations, which are computationally intensive tasks. Our approach, described in section 2, requires an efficient computation.

The rest of paper is organized as follows. Section 2 provides our proposed oblivious distribution protocol and its implementation to improve DRM. Section 3 gives the security and privacy analysis of the implemented protocol. Finally, section 4 provides concluding remarks.

2. RESEARCH METHOD

To overcome the identified problem in the typical DRM systems, we do the following steps: (1) construct an oblivious content distribution protocol; (2) implement the protocol to improve the DRM model for content distribution; and (3) analyze the improved DRM model to show its security and privacy.

2.1. Oblivious Content Distribution Protocol

We propose an oblivious content distribution protocol that is more flexible and appropriate for DRM implementation. Our protocol utilizes tamper-proof devices. A tamper-proof device means any device that can be used only in a particular way, otherwise the device will be corrupted and its content will no longer be accessible. Utilizing tamper-proof devices in this protocol is less expensive. The device contains only two types of functions, `GetKey` and `GetContent`. `GetKey` function allows the user to ask for the key; that is, the input parameter to the `GetContent` function. `GetContent`, on the other hand, requires an authorized key to reveal the message stored in it. With this characteristic, the device can be mass produced at a low cost. Creating a single device containing all pairs of functions (`GetKey`,`GetContent`) may be reasonable and more efficient. However, for the sake of clarity in this sub section, we assume that one device contains a pair of functions (`GetKey`,`GetContent`).

The protocol allows content provider to deliver contents to user in such a way that at the end of the protocol the user cannot access contents more than he is supposed to access and the content provider will not know which contents are accessed by the user. Suppose the content provider (say, Alice) provides N contents (e.g. movies), $(M_1, ..., M_N)$, and the user (say, Bob) wishes to access K, where $K < N$, of these contents. Alice has a secret code S to access the contents, and utilizes Shamir's secret sharing scheme [22], with the threshold parameter N-K, to share the secret. That is, she splits the secret into N pieces such that any set of at least N-K shares can reconstruct the secret.

The detail protocol is as follows. To share the secret and send the contents, Alice performs the following steps:

1. She secretly chooses random N-K-1 elements of Z_p, denoted $a_1, ..., a_{N-K-1}$ and forms the polynomial $f(x) = S + a_1 x^1 + ... + a_{N-K-1} x^{N-K-1}$. Note that p is a prime and $p > N$.
2. For $i = 1, ..., N$, she computes s_i, where $s_i = f(i) \bmod p$
3. She loads device d_i with s_i as the key value, and M_i as the content value.
4. She gives all devices to Bob.

After delivering the devices there is no subsequent communication between Alice and Bob. Bob can access K contents if he accepts sacrificing N-K contents that are not supposed to be accessed. This condition is applied with assumption that once a device is executed, it will be corrupted or will destroy itself. To obtain K contents, Bob performs the following steps (see also Figure 2 for a clear illustration).

1. For simplicity, assume that K contents Bob want to access are $M_1, ..., M_K$. Bob performs the `GetKey` function on the devices $d_{K+1},...,d_N$ (namely $GK_{K+1},...,GK_N$), to obtain N-K shares.
2. With the N-K shares, $s_{K+1},\ldots,s_N$, Bob can reconstruct the polynomial, e.g. using the Lagrange interpolation, and learn the secret S.
3. Using the access code S, Bob performances the `GetContent` function on devices $d_1, ..., d_K$ (namely $GC_1,...,GC_K$) to obtain the contents $M_1, ..., M_K$.

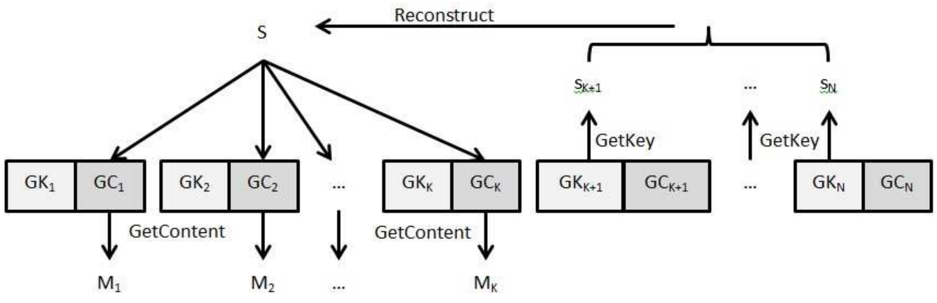


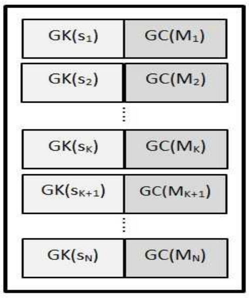
Figure 2. Process for obtaining K out of N contents.

The protocol described above can be modified to cover another need. For instance, instead of focusing on the *number_of_items* variable, the protocol can be pointed out to the *number_of_plays* variable (e.g. a customer wants to watch a movie for *K* times). The movie provider then sends the customer a package containing *N* pairs (`GetKey`,`GetContent`) with all `GetContent` functions associates with a movie of the same title.

2.2. Implementation to Improve DRM

To implement the constructed protocol in the DRM applications, we employ smart cards. A smart card contains an embedded microprocessor so that it can be used not only to store data, but also to process the data [23]. The microprocessor is also used for security purposes. Data are never directly available to the external applications as the microprocessor controls data handling and memory access according to a given set of conditions. In this implementation, a smart card is assumed to be a tamper-proof device and contains all pairs of functions (`GetKey`,`GetContent`).

Suppose the content provider provides *N* contents, M_1, ..., M_N. First of all, the content provider encrypts all contents using a secret key *S*. For a particular value *K*, $1 \leq K \leq N - 1$, *S* is split into *N* shares, $s_1,\ldots,s_N$, using Shamir's scheme with the threshold parameter *N* – *K*. The content provider then passes the protected contents to the distributor and the key's shares to the smart card (SC) manufacturer.



GK(s_1)	GC(M_1)
GK(s_2)	GC(M_2)
⋮	
GK(s_K)	GC(M_K)
GK(s_{K+1})	GC(M_{K+1})
⋮	
GK(s_N)	GC(M_N)

Figure 3. Smart card model; *GK* and *GC* stand for `GetKey` and `GetContent`, respectively.

The SC manufacturer creates smart cards and sends them to the distributor. The smart card model (see Figure 3) has the following characteristics. For a particular *K*, a smart card contains *N* pairs of functions (`GetKey(s`$_i$`)`,`GetContent(M`$_i$`)`), where $i = 1,2,\ldots,N$. Only one function can be executed from each pair. That is, executing the function `GetKey(s`$_i$`)` will disable the associated function `GetContent(M`$_i$`)` and, thus, will deny access to the associated content M_i. Conversely, executing the function `GetContent(M`$_i$`)` will disable `GetKey(s`$_i$`)`. In concrete terms, the smart card executes *N-K* `GetKey` functions associated with *N-K* unselected contents. The shares revealed by these functions are then combined to construct the key *S* that be used to unlock *K* selected contents.

A user can download the protected contents and purchases an appropriate smart card from the distributor's channel. To access the downloaded contents, the user's player must be connected to a compatible smart card reader. A *K*-valued smart card can be used to unlock *K* selected contents and denies access to *N-K* unselected contents.

3. RESULTS AND ANALYSIS

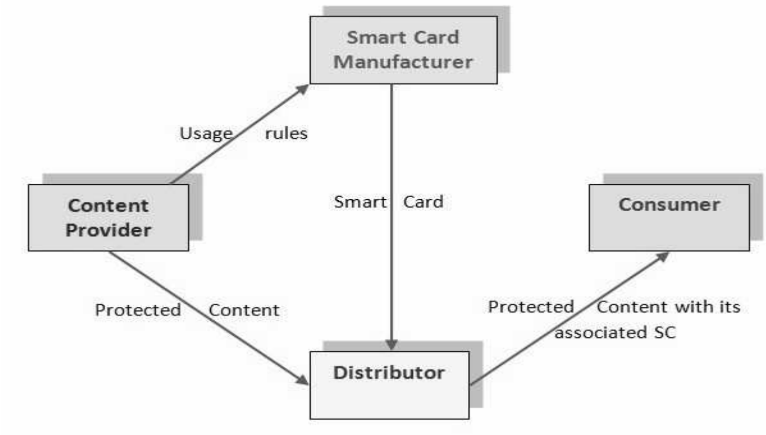


Figure 4. Improved DRM system

The improved DRM model for content distribution (see Figure 4) provides an efficient mechanism. Instead of a clearing house, the system employs a smart card manufacturer. Users obtain the content and the corresponding license (provided by an appropriate smart card) from one party, that is, the distributor. This mechanism makes the process more efficient. Furthermore, the improved system also achieves security and privacy for the content provider and the users, respectively. Analysis of both characteristics follows.

Analysis of Security

In the proposed protocol, the shares of the secret key and the function for accessing content are stored in a smart card which is assumed to be a tamper-proof device. The user cannot access content without obtaining the secret key. The key, however, is split into several pieces of shares and distributed among the pairs of functions (`GetKey`,`GetContent`) inside the device using Shamir's secret sharing scheme [22]. This scheme is secure because knowing less than a predetermined number of shares gives the user no way to reconstruct the secret. As a result, the user can only obtain the secret key if (and only if) he sacrifices all contents that he is not supposed to access. This means that the user is not able to access anything other than the contents that are supposed to be accessed. Additionally, the smart card is only allocated to the user who has made the payment for it. A particular smart card allows the user to access a limited number of contents as detemined in it. Therefore, the proposed protocol achieves perfect security for the content provider.

Analysis of Privacy

In the proposed protocol, there is no interaction between content provider and user after the content provider gives all devices to the user. There is no way for the content provider to determine which devices the user has used. As all pairs of functions (`GetKey`,`GetContent`) are corrupted at the end of the protocol, the content provider has no knowledge about which content that has been accessed by the user. Additionally, in the protocol implementation, to unlock the content, a user does not need to provide his personal data for the license. Instead, he purchases the corresponding smart card anonymously. The content and its associated smart card will not be connected to the user's identity. Therefore, the user‘s privacy is protected.

Advance Implementation

In the oblivious content distribution scenario described above, a user can decrypt a set of contents no more than he was supposed to access. However, once the content has been decrypted, the user can play it without limit. If the restriction of the number of plays is also considered in a business scheme, then an extra variable must be added to the content distribution protocol.

The proposed scenario can be enlarged to cover more variables of the usage rules. That is, we can combine the variables *number_of_items* and *number_of_plays* in one scheme. For example, a user may purchase 5 items, namely content M_1,M_2,M_3,M_4,M_5, and 20 plays. In this case, the user can play all items, but no more than 20 times overall. He may play M_1 for 3 times, M_2 for 4 times, M_3 for 7 times, M_4 for 4 times and M_5 twice. However, he cannot play M_2 for 10 times and M_5 for 11 times. This advanced scenario provides flexible content distribution that still preserves security and privacy.

4. CONCLUSION

We construct an oblivious content distribution protocol. In this protocol, the secret key is split into several shares to enhance the security of the distributed content. Without adequate shares, it is impossible to reconstruct the secret key. The protocol is then implemented to advantage DRM. This implementation makes use of smart cards that can be used not only to store data, but also to process the data independently. Assuming that tamper-proof device exists, the mechanism achieves security for the content provider and preserves privacy for the users.

If more restrictions of the content usage rules are applied, the proposed protocol can also be enlarged to cover more variables. Despite providing flexibility, the system still preserves security and privacy. These characteristics are important to make the improved DRM a privacy-aware rights protection system. The system does not merely focus on achieving security for the content provider, but also on preserving privacy for users.

REFERENCES

[1] B. Barak, *et al.*, "On the (Im)possibility of Obfuscating Program," in *Advance in Cryptology - CRYPTO 2001: 21st Annual International Cryptology Conference*, Santa Barabara, California, USA, 2001, pp. 1-18.

[2] S. Chow, *et al.*, "A White-Box DES Implementation for DRM Applications," presented at the DRM 2002, 2003.

[3] S. Chow, *et al.*, "White-Box Cryptography and an AES Implementation," presented at the SAC 2002, 2003.

[4] B. Wyseur, "White-Box Cryptography: Hiding Keys in Software", *MISC HS 5 Magazine*, 65-72, 2012.

[5] A. C. Prihandoko, *et al.*, "DRM's Rights Protection Capability: A Review," in *The First International Conference on Computational Science and Information Management*, Medan, Indonesia, 2012, pp. 12-17.

[6] Q. Liu, *et al.*, "Digital Rights Management for Content Distribution," presented at the Australian Information Security Workshop on ACSW Frontiers'03, 2003.

[7] H. Ghodosi, "A General Model for Oblivious Transfer," in *the Sixth International Workshop for Applied PKC*, Perth, Australia, 2007, pp. 79-87.

[8] M. O. Rabin, "How to Exchange Secrets with Oblivious Transfer," Aiken Computation Lab, Harvard University, Technical Report TR-81, 1981.

[9] S. Even, *et al.*, "A Randomized Protocol for Signing Contracts," *Communications of the ACM*, vol. 28, pp. 637-647, 1985.

[10] M. Naor and B. Pinkas, "Oblivious Transfer and Polynomial Evaluation," in *Thirty-first Annual ACM Symposium on Theory of Computing*, Atlanta, Georgia, USA, 1999, pp. 245-254.

[11] W.-G. Tzeng, "Efficient 1-Out-n Oblivious Transfer Schemes," in *PKC 2002*, 2002, pp. 159-171.

[12] W.-G. Tzeng, "Efficient 1-Out-of-n Oblivious Transfer Schemes with Universally Usable Parameters," *IEEE Transactions on Computers*, vol. 53, pp. 232-240, 2004.

[13] M. Naor and B. Pinkas, "Oblivious Transfer with Adaptive Queries," in *CRYPTO'99*, 1999, pp. 573-590.

[14] C.-K. Chu and W.-G. Tzeng, "Efficient k-Out-of-n Oblivious Transfer Schemes with Adaptive and Non-adaptive Queries," in *PKC 2005*, 2005, pp. 172-183.

[15] R. L. Rivest, "Unconditionally Secure Commitment and Oblivious Transfer Schemes Using Private Channels and a Trusted Initializer," 1999.

[16] M. Naor and B. Pinkas, "Distributed Oblivious Transfer," in *ASIACRYPT 2000*, 2000, pp. 205-219.

[17] C. L. F. Corniaux and H. Ghodosi, "An Information-Theoretically Secure Threshold Distributed Oblivious Transfer Protocol," in *Information Security and Cryptology - ICISC 2012*, 2013, pp. 184-201.

[18] C. L. F. Corniaux and H. Ghodosi, "A Verifiable 1-out-of-n Distributed Oblivious Transfer Protocol," Cryptology ePrint Archive, Report 2013/063, 2013.

[19] H. Ghodosi, "Analysis of an Unconditionally Secure Distributed Oblivious Transfer," *Journal of Cryptology*, vol. 2013, pp. 75-79, 2013.

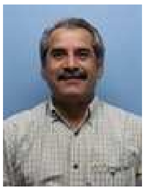
[20] C. Blundo, *et al.*, "On Unconditionally Secure Distributed Oblivious Transfer," *Journal of Cryptology*, vol. 20, pp. 323-373, 2007.

[21] M. Naor and B. Pinkas, "Efficient Oblivious Transfer Protocols", *SODA'01*, 2001.

[22] A. Shamir, "How to Share a Secret," *Communications of the ACM*, vol. 22, pp. 612-613, 1979.

[23] Z. Chen, "Java Card Technology for Smart Cards: Architecture and Programmer's Guide", *e-book*, 2000.

BIOGRAPHY OF AUTHORS

	Antonius Cahya Prihandoko is a lecturer at the Information System Dept., University of Jember, Indonesia. He received his Bachelor degree in Mathematics Education from the University of Jember and Master of Applied Science in Computer Science from JCU, Australia, in 1992 and 1999, respectively. Currently, he is pursuing a PhD in Information Technology at the School of Business (IT), JCU. His research topic is Rights Protection of Digital Content in the Digital Rights Management (DRM) Environment.
	**Dr Hossein Ghodosi** is a senior lecturer in Information Technology at the School of Business, JCU Australia. He received his Bachelor degree in Mathematics and Computer Science from Tehran University, Iran, in 1975; and his MSc and PhD degree both in Computer Science from University of Wollongong, Australia, in 1994 and 1998, respectively. His primary research area is Cryptography. More precisely, Society-Oriented and/or Threshold Cryptography. Currently his principle research themes are: Multi-Party Computations, Oblivious Transfers, and Secret Sharing Schemes.
	Associate Professor Bruce Litow is an adjunct staff at the Department of IT, School of Business, JCU. His research interest is in computational complexity and how it impinges on numerous application fields. Currently his principal research themes are: (1) how chinese remainder representation is connected to combinatorial optimization problems, (2) parallel complexity of integer GCD, (3) exact computations with algebraic numbers.