A Secure and Mutual-Profitable DRM Interoperability Scheme

Sangho Lee
Dept. of CSE, POSTECH
Pohang, Republic of Korea
sangho2@postech.ac.kr

Heejin Park
LG Electronics Inc.
Seoul, Republic of Korea
parkhj84@lge.com

Jong Kim
Dept. of CSE, POSTECH
Pohang, Republic of Korea
jkim@postech.ac.kr

***Abstract*—In most cases, the use of digital contents on several devices is blocked by digital rights management (DRM) technology to protect the rights of digital content owners, which is called as the DRM's walled garden strategy. This strategy has raised many legal, economical, and ethical problems. DRM interoperability can complement this strategy. However, there is no agreeable systematic interoperability scheme between various DRM systems. This problem cannot be solved without the cooperation and participation of both DRM technology providers and content providers. Some previous attempts to solve the DRM interoperability problem have suggested that both providers need to open parts of their security properties, without the assurance of a beneficial outcome. They were therefore reticent about participating. In this paper, we propose a secure mutual-profitable DRM interoperability scheme which minimizes disclosure of the security properties of DRM technology providers and content providers while preserving their profits. We use a designated proxy re-encryption scheme to allow the providers to designate a proxy which re-encrypts their digital contents and a neutral format scheme to enable format-independent translations. Moreover, we allow the providers to manage and trace their digital contents, and to request additional fees for interoperability services. We describe detailed protocols and analyze the scheme. We also introduce a prototype implementation.**

***Keywords*-Digital Rights Management (DRM), Interoperability, Proxy Re-encryption**

I. Introduction

Digital rights management (DRM) was introduced to protect the copyright of digital contents in digital environments. Various DRM technologies are currently available [2], [3]. Most of them take the walled garden strategy [4] to protect the contents they provide and the profit they can get, even though it brings up many legal and ethical problems. One way to complement this strategy is DRM interoperability. Without DRM interoperability, consumers have to repeatedly purchase the same digital contents if they wish to use them on their heterogeneous devices. Consumers frequently criticize content providers because they are generally adopting non-interoperable DRM schemes [5]. On the other hand, a recent survey has shown that many consumers are willing to pay more money for contents with interoperability [6]. Therefore, DRM interoperability is required to increase activities in the digital market while protecting the digital copyright.

Several researchers have suggested schemes [7], [8], [9], [10] to solve the DRM interoperability problem. According to Koenen *et al.* [5], there are three possible approaches to interoperability in DRM systems: full format interoperability, connected interoperability, and configuration driven interoperability. Full format interoperability means that every DRM system shares the same security infrastructure, which is feasible by having a standard. However, due to many business reasons, the standard for DRM is still a long way to go. Because full format interoperability is difficult to acquire, an alternative approach which uses a neural format [9] for content translation has been proposed. Devices translate content to a neutral format when exporting it and then convert the received neutral format to their own DRM format while importing it. Some security weaknesses exist in this approach because content translations and license generations are performed by devices. Connected interoperability means that an external trusted entity manages interoperability services [8], [10]. This external trusted entity has to know all the security properties of the DRM technology providers such as encryption methods, content formats, and license formats. It may introduce security problems. Configuration driven interoperability means that a consumer's device can download heterogeneous DRM components as software to extend its functionality [7]. Because it is software-based, it has inherent security weaknesses.

Motivation and Research Goal. To solve the DRM interoperability problem, we need to encourage participation from both DRM technology providers and content providers. Nevertheless, previous studies on the DRM interoperability have considered how to fulfill consumers' needs while largely ignoring how to encourage the participation of DRM technology providers and content providers. Without their participation, DRM interoperability schemes are hard to achieve. Moreover, because the content providers want to make a profit with their digital contents even when it is not used on the original device, they want to trace the usage of their contents. Also, the technology providers are reluctant to disclose their security properties because they do not want to reveal their technology

The preliminary version of this paper was presented at [1]. This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-C1090-0902-0045) and WCU (World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-2008-000-10100-0):

for possible hacking. Previous work, however, would need the technology providers to open parts of their security properties and did not consider incorporating tracking and control features within a DRM interoperability scheme.

In this paper, we propose a secure mutual-profitable scheme to address the DRM interoperability problem. The proposed scheme minimizes the disclosure of DRM technology and content providers' security properties by using designated proxy re-encryption and neutral format schemes [9]. The designated proxy re-encryption scheme allows a designated proxy to re-encrypt specific content without revealing the raw content, while the neutral format scheme allows for format-independent translations. Taban *et al.* [10] also used a proxy re-encryption scheme [11] for DRM interoperability. Their scheme, however, cannot designate a proxy to perform the re-encryption and also cannot specify the content to be re-encrypted. Therefore, if someone were able to obtain a re-encryption key from device A to device B, he/she could illegally re-encrypt and deliver all contents of the device A to the device B. In the proposed scheme, however, if someone obtained a re-encryption key, he/she could only be able to re-encrypt specific contents. Therefore, the proposed scheme is more secure than the Taban *et al.*'s scheme [10]. Moreover, in the proposed scheme, DRM technology and content providers are able to manage and trace DRM interoperability processes, and bill additional fees for DRM interoperability services. This is likely to encourage the providers to actively participate in the scheme to increase DRM interoperability.

Paper Organization. The rest of this paper is organized as follows. In Section II, we introduce preliminaries of this paper. In Section III, we discuss our system model. In Section IV, we describe our scheme and analyze it in Section V. In Section VI, we explain a prototype implementation of our scheme. Finally, we conclude this paper in Section VII.

II. Preliminaries

Bilinear Map. A map $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear map which has the following properties:

- G_1 and G_2 are groups of the same prime order q.
- For all $a, b \in \mathbb{Z}_q$ and $q, h \in G_1$, $e(g^a, h^b) = e(g, h)^{ab}$ is efficiently computable.
- The map is non-degenerate, i.e., if g generates G_1 and h generates G_1, then $e(g, h)$ generates G_2.

We set invertible functions $\psi_1 : \mathbb{Z}_q \rightarrow G_1$ and $\psi_2 : \mathbb{Z}_q \rightarrow G_2$.

Proxy Re-encryption. Proxy re-encryption allows a proxy to transform a ciphertext computed under A's public key into one that can be opened by B's secret key without any additional decryption. The temporary unidirectional proxy re-encryption scheme [11] is based on the ElGamal scheme operating over two groups G_1, G_2 of prime order q with a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. The system parameters are random generators $g \in G_1$ and $Z = e(g, g) \in G_2$.

Key Generation: User A's key pair is of the form $sk_A = a \in_R \mathbb{Z}_q^*$ and $pk_A = g^a \in G_1$.

Re-Encryption Key Generation: A delegates to B by publishing the re-encryption key $rk_{A \rightarrow B} = g^{b/a} \in G_1$, computed from B's public key.

First-Level Encryption: General public based encryption method is called a first-level encryption. To encrypt a message $m \in G_2$ under pk_A in such a way that it can only be decrypted by the holder of sk_A, $Z^{ak} = e(g^a, g^k)$ is computed where $k \in_R \mathbb{Z}_q^*$, and $c = (Z^{ak}, mZ^k)$ is the output.

Second-Level Encryption: This encryption is a preliminary encryption for proxy re-encryption. Therefore, second-level encryption should be performed first so that a proxy can perform the re-encryption. To encrypt a message $m \in G_2$ under pk_A in such a way that it can be decrypted by A and other delegatees, $c = (g^{ak}; mZ^k)$ is published.

Re-encryption: Anyone can change a second-level ciphertext for A into a first-level ciphertext for B with $rk_{A \rightarrow B} = g^{b/a}$. Using $c_a = (g^{ak}, mZ^k)$, $e(g^{ak}, g^{b/a}) = Z^{bk}$ is computed and $c_b = (Z^{bk}, mZ^k)$ is published.

Decryption: To decrypt a first-level ciphertext $c_a = (\alpha, \beta)$ with $sk_A = a$, $m = \beta/(\alpha^{1/a})$ is computed and published.

Designated Proxy Re-encryption. Based on the temporary unidirectional proxy re-encryption [11], we propose a designated proxy re-encryption which allows message creators to designate a proxy to perform re-encryption.

Key Generation: A message creator C choose a key pair $sk_m = \mu \in_R \mathbb{Z}_q^*$ and $pk_m = g^\mu \in G_1$ for a message m.

Re-Encryption Key Generation: C computes a re-encryption key $rk_{\mu \rightarrow b} = g^{b/\mu} \in G_1$ which will be used to re-encrypt a message encrypted with a key μ to a key b.

First-Level Encryption: To encrypt a message $m \in G_2$ under $pk_A = g^a$ in such a way that it can only be decrypted by the holder of $sk_A = a$, $c_A = (Z^{ak} = e(g^a, g)^k, mZ^k)$ is computed and published where $k \in_R \mathbb{Z}_q^*$.

Second-Level Encryption: To encrypt a message $m \in G_2$ under pk_m in such a way that it can only be re-encrypted by the holder of $sk_\Pi = \pi \in \mathbb{Z}_q^*$, $Z^{\pi k} = e(g^\pi, g^k)$ is computed, and $c = (g^{\mu k}, mZ^{\pi k})$ is published.

Re-encryption: Only Π who has $sk_\Pi = \pi$ can change a second-level ciphertext of a message m into a first-level ciphertext for B with $rk_{\mu \rightarrow b}$. From $c = (g^{\mu k}, mZ^{\pi k})$, $Z^{b\pi k} = e(g^{\mu k}, g^{b/\mu})^\pi$ is computed and $c_B = (Z^{b\pi k}, mZ^{\pi k})$ is published.

Decryption: To decrypt a first-level ciphertext $c_B = (\alpha, \beta)$ with $sk_B = b$, $m = \beta/(\alpha^{1/b})$ is computed and published.

III. System Model and Requirements

In this section, we introduce a system model and the requirements of our scheme. The rest of this paper uses notations shown in Table I.

A. System Model

Our system comprised of six kinds of entities: DRM server, content provider, DRM interoperability server, DRM interoperability agent, device, and billing server (see Fig. 1).

TABLE I
NOTATIONS

Symbol	Meaning
ID_m	Identifier of content m
C_m	Normal format of content m
IC_m	Interoperable format of content m
lic	License
$rk_{\mu\to\alpha}$	Re-encryption key to re-encrypt a message encrypted with a key μ to a key α
$E_1(\mu; m)$	First-level encryption on a message m with a key μ
$E_2(\pi, \mu; m)$	Second-level encryption on a message m with a key μ designated to a holder of a key π
$SE(K_m; m)$	Symmetric key encryption on a message m with a key K_m

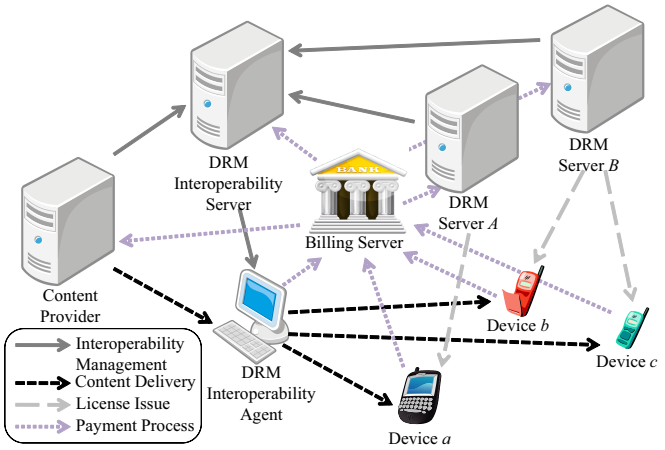


Fig. 1. System model

DRM Server (DS): DRM server is a DRM technology provider entity which manages DRM devices and issues licenses for contents.

Content Provider (CP): Content provider is an entity which owns contents and publishes them in a secure format. In our model, it can publish contents in two formats: a general format with content m for a device which has a secret key α,

$$C_m = (metadata, E_1(\alpha; K_m), SE(K_m; m)),$$

or an interoperable format with m for DIA which has a secret key π,

$$IC_m = (metadata, E_2(\pi, \mu; K_m), SE(K_m; m)).$$

DRM Interoperability Server (DIS): DRM interoperability server is the entity which manages overall DRM interoperability processes. When an interoperability service is initiated, it obtains a re-encryption key from CP and DS, and delivers it to a DRM interoperability agent (DIA).

DRM Interoperability Agent (DIA): DRM interoperability agent is an entity which translates IC_m to C_m. It converts the encrypted key K_m of IC_m without disclosure when a device requests IC_m. To do this, it requests a re-encryption key to DIS and performs re-encryption.

Device (D): Device is an entity which is used by a consumer. A consumer can request C_m from a DIA via his/her device. The device can convert C_m to its own format to use it.

Billing Server (BS): Billing server is an entity which manages the overall billing processes.

B. Requirements

Based on previous research [5], [12], [13], [14], [15], we introduce the following requirements for DRM interoperability schemes.

Persistent Protection: A DRM interoperability scheme has to guarantee the persistent protection of DRM contents. It means that irrespective of translation of DRM contents, the constraints that are imposed by DRM servers have to be enforced.

Security: A DRM interoperability scheme has to guarantee its security against several security attacks such as impersonation and replay attacks. Also, it needs to be protected against bogus DIAs.

Tracking the Translation of DRM Contents: A DRM interoperability scheme has to provide an ability to track the translation of DRM contents to prevent illegal translations by illegitimate entities.

Changing Rights during Translations: Because the policies of DRM technology providers and the functionality of their devices are different, it is difficult to apply the same license model to various DRM systems. Therefore, a DRM interoperability scheme has to allow changes of rights during translations.

Guaranteeing Content Originality: Content originality means that even if contents are converted, their ownership has to be linked with their original DRM server. When contents are re-distributed to other devices, a DRM interoperability scheme has to be able to guarantee the contents originality, i.e., the original DRM server has to be able to manage and trace the re-distribution process.

C. Assumptions

We have made the following assumptions for our scheme:

- Each entity, DS, CP, DIS, DIA, D, and BS, has a certificate for authentication and revocation.
- Entities create a secure channel using their certificates for secure communications between them, e.g., Transport Layer Security (TLS) [16].
- We only consider conceptual payment procedures. Other ideas such as a micro-payment scheme [17] may be integrated into our scheme for practical payment purposes.

IV. PROPOSED SCHEME

The DRM interoperability problem cannot be solved without the participation of the DRM technology providers and content providers. To encourage the participation of both providers, we have to minimize disclosure of their security

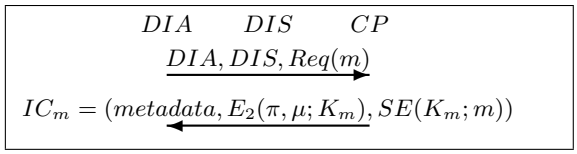
properties and assure it is of benefit to them. To minimize the disclosure of security properties, we use designated proxy re-encryption and neutral format schemes. The designated proxy re-encryption scheme ensures that only a designated DIA can re-encrypt ICs, while the neutral format scheme eliminates the need for the providers to open their security properties. Also, to ensure this approach is of benefit to them, we propose two protocols to manage the DRM interoperability processes. The first protocol is an acquisition protocol to acquire the IC. In this protocol, a consumer purchases IC_m from CP via his/her DIA and then stores them on his/her DIA. The second protocol is a transmission protocol to deliver the IC_m stored on a DIA to a device A. To deliver the IC_m to A, DIA has to re-encrypt the IC_m to C_m with a re-encryption key which is created by the DIS, CP, and DS_A. Then, to use the C_m, A has to purchase a corresponding license from the DS_A. This payment is distributed to the DIS, CP, and DS_A to ensure that the DRM technology providers and content providers benefit from in the DRM interoperability process.

A. Acquisition Protocol

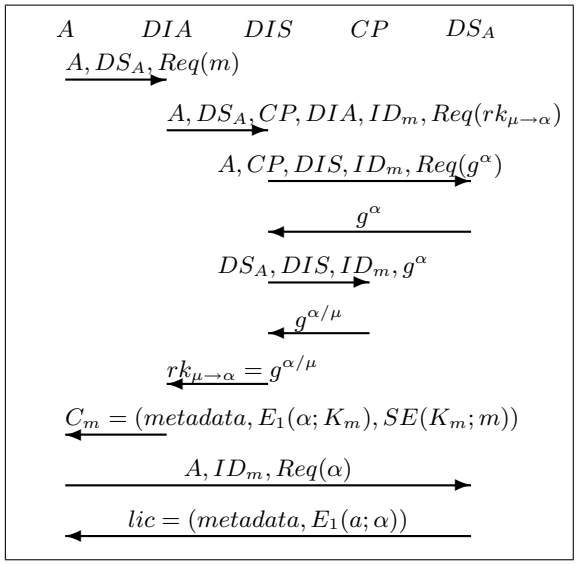
In the acquisition protocol, a consumer buys IC_m from the CP through his/her DIA. Along with content m's information and payment information, the DIA sends its own information which includes its public key g^{π} and its server DIS's information to the CP. The CP verifies the payment information and the information from the DIA and DIS. Then, to create IC_m for the DIA, CP encrypts m with a symmetric secret key K_m, and then performs two-level encryption on K_m with an asymmetric secret key μ, DIA's public key g^{π}, and a randomly selected asymmetric secret key k_1 as $E_2(\pi, \mu; K_m) = (g^{\mu k_1}, \psi_2(K_m) \cdot Z^{\pi k_1})$. The created $IC_m = (metadata, E_2(\pi, \mu; K_m), SE(K_m; m))$ is then stored on the DIA for further transmissions (see Fig. 2a).

B. Transmission Protocol

Assume that a consumer wants to play m which is stored on a DIA with his/her device A. A sends a request for m to the DIA along with its information and its server DS_A's information. If the information of m, A, and DS_A is valid, DIA requests a re-encryption key $rk_{\mu \to \alpha}$ from the DIS along with its information and information about m, A, DS_A, and CP. The DIS checks the validity of the information about the DIA and CP, and then sends a request for g^{α} to DS_A along with its information and information about m, A, and CP. When the received information is valid, DS_A randomly creates an asymmetric secret key α and sends g^{α} to DIS. DS_A stores information of m, A, and α to issue licenses later. The DIS sends g^{α} to the CP along with its information and information about m and DS_A. The CP verifies the received information and then returns $rk_{\mu \to \alpha} = g^{\alpha/\mu}$ to the DIS. DIS sends $rk_{\mu \to \alpha}$ to DIA. Then, DIA re-encrypts $E_2(\pi, \mu; K_m)$ as $E_1(\alpha; K_m) = (Z^{\alpha \pi k_1}, \psi_2(K_m) \cdot Z^{\pi k_1})$ and sends $C_m = (metadata, E_1(\alpha; K_m), SE(K_m; m))$ to A. To decrypt C_m, A sends a request for the secret key α to DS_A along with information about itself and m. The DS_A verifies the received information and computes $E_1(a; \alpha) = (Z^{ak_2}, \psi_2(\alpha) \cdot Z^{ak_2})$. Then, the DS_A sends a license $lic = (metadata, E_1(a; \alpha))$ to A (see Fig. 2b).



(a) Acquisition protocol



(b) Transmission protocol

Fig. 2. Flow of acquisition and transmission protocols

C. Content Usage

Because the C_m that was translated from the IC_m includes a neutral format [9] of content m, a device A has to decrypt the C_m with a secret key α in the lic, and then transforms m to its own format to use it. To avoid repeated transforming, a device can store the transformed m in a secure storage if it is available, e.g., Trusted Platform Module (TPM) [18].

D. Billing Scenario

To encourage the participation of the DRM technology providers and content providers in solving the DRM interoperability problem, we have to ensure that this scheme is of benefit to them. We introduce two kinds of payment: P_o and P_t. P_o is the price of content including an interoperability approval fee, and P_t is the price of transmission. We can classify the billing scenario into two cases: on-demand payment and pre-payment.

On-demand Payment: In the acquisition protocol, a consumer pays P_o to the BS when purchasing IC. Then, in the transmission protocol, the DIA requests a re-encryption key from the DIS. Before it gives the re-encryption key to the DIA, the DIS asks the content is interoperable with the DS_A and whether P_t has been paid to the BS. After verifying the payment of P_t, the BS generates a random

number $R = R_1||R_2$ and then creates the payment data:

$$PaymentData = (H(R)||DIA||DIS||CP||DS_A||ID_m).$$

The payment data is stored in the BS as evidence for P_t. The BS sends this random number $R = R_1||R_2$ to DIS. Next, the DIS transfers R_1 to the CP and R_2 to the DS_A with re-encryption key request messages. The subsequent billing scenario starts after the device A obtains a corresponding license from the DS_A. The DS_A, which issues a new license, requests its profit from the BS. At this time, CP and DS_A transmit R_1 and R_2 as evidence of completed content transmission to the BS. Then, the BS compares a hash value of $R = R_1||R_2$ with the payment data. If they are same, then BS pays P_t to CP, DS_A, and DIS as the ratio of p, q, and r ($p+q+r = 1$).

Pre-payment: The payment certificate is purchased in advance for proof of payment in content transmissions. Initially, consumers purchase the following payment certificate from the BS through the DIA.

$$PaymentCert = (H(R)||DIA||\#transmissions)$$

When the DIA requests a re-encryption key from the DIS, the BS examines the DIA's payment certificate. At this stage, the BS compares the DIA's payment certificate with the certificate it stores. If they are same, it reduces the number of transmissions by 1 and then creates payment data for this transmission. The remainder is the same as in the on-demand payment situation.

V. Analysis

We analyze our scheme according to the requirements in Section III.

Persistent Protection: In our scheme, when contents are translated and delivered to a device, the content encryption key is re-encrypted with a secret key α which is selected by the DS of that device. Thus, each device has to obtain a corresponding license from its DS to know α. Therefore, persistent protection is guaranteed.

Security: We analyze the security of our scheme. First, no attacker can impersonate a legal entity because each entity has a certificate for authentication. Second, the DIA cannot obtain the content encryption key of IC because that key is encrypted with a secret key μ which is selected by CP. Also, that key will not be revealed during re-encryption. Third, a device cannot obtain the raw content of IC until it receives a corresponding license because the content encryption key of the IC is encrypted with a secret key α which is selected by its DS. Fourth, a bogus DIA cannot give translated IC to other devices because it is encrypted with a secret key α which is selected by DS. Devices of other DSes cannot obtain α. Also, other devices of the same DS cannot obtain α because that DS will not give licenses to devices that did not purchase that IC.

Tracking the Translation of DRM Contents: In our scheme, DIA has to receive a re-encryption key from CP and DS with every transmission. Otherwise, it cannot re-encrypt IC. By using this, the CP can trace translations of its contents.

TABLE II
Comparison on the running time between functions of PRE2 and DPRE (for a 160-bit group with an Intel Pentium 4 3.0 GHz CPU)

Functions	Running time (ms)		Overhead (%)
	PRE2	DPRE	
gen_params()	307.9	-	-
keygen()	61.92	-	-
level1_encrypt()	9.73	-	-
level2_encrypt()	18.82	53.30	283.2
delegate()	33.57	-	-
reencrypt()	32.09	47.68	148.6
decrypt()	9.52	-	-

Changing Rights during Translations: In our scheme, the rights of the re-distributed contents can be changed because DS issues new licenses at the end of the translations. Therefore, our scheme supports changes of rights during translations.

Guaranteeing Content Originality: In our scheme, only CP can create second-level encrypted contents IC. As an IC is translated by DIA, it is changed to a first-level encrypted form which cannot be translated to other forms. Therefore, the content originality is guaranteed because only the CP can allow re-distribution of its contents.

VI. Prototype Implementation

We implement a prototype using the proxy re-cryptography library [19] in a Linux system. The proxy re-cryptography library uses the Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL) [20]. It has two algorithms, PRE1 and PRE2, of Ateniese *et al.*'s [11]. The PRE2 algorithm is the algorithm which was introduced in Section II; thus, we implement the designated proxy re-encryption (DPRE) algorithm by modifying it. The DPRE algorithm is comprised of seven functions:

- gen_params(): generate domain parameters
- keygen(): generate a public/private key pair
- level1_encrypt(): perform first-level encryption
- level2_encrypt(): perform second-level encryption
- delegate(): generate a re-encryption key
- reencrypt(): re-encrypt a second-level encrypted message
- decrypt(): decrypt an encrypted message

The gen_params(), keygen(), level1_encrypt(), delegate(), and decrypt() functions of the DPRE are same for each of the PRE2. The level2_encrypt() and reencrypt() functions are modified to use the public/private key pair of the DIA. The level2_encrypt() function of DPRE is about 2.8 times slower than the PRE2 because of the additional bilinear map operation and the reencrypt() function of DPRE is about 1.5 times slower than the PRE2 because of the additional exponentiation (see Table II). This overhead is not a big problem because these two functions are used by servers.

We also implement four simple programs that represent the entities of our system model: DPRE_CP for CP, DPRE_DS

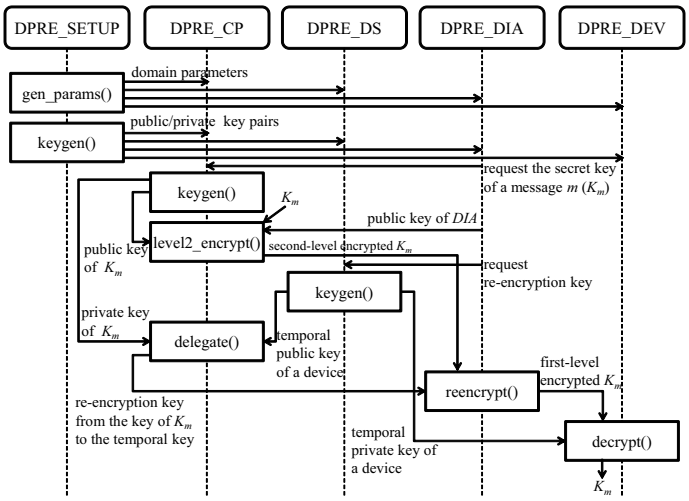


Fig. 3. Functional entities and their interactions

for DS, DPRE_DIA for DIA and DIS, and DPRE_DEV for D. In addition, we implement DPRE_SETUP which generates domain parameters and public/private key pairs of the above programs. The interactions between these programs are as follows (see Fig. 3).

1) The DPRE_SETUP generates domain parameters and public/private key pairs of each program. It then sends them to each program.
2) When the DPRE_DIA requests the secret key of a message m (K_m), the DPRE_CP generates a public/private key pair for K_m, and then performs second-level encryption on K_m with the public keys of K_m and DPRE_DIA. It sends the result to the DPRE_DIA.
3) After the DPRE_DIA receives the second-level encrypted K_m, it requests a re-encryption key to the DPRE_DS. The DPRE_DS generates a temporal public/private key pair of a device, and then sends the temporal public key to the DPRE_CP and the temporal private key to the DPRE_DEV. The DPRE_CP generates the re-encryption key using the temporal public key and the private key of K_m, and then sends it to the DPRE_DIA.
4) After the DPRE_DIA receives the re-encryption key, it performs re-encryption on the second-level encrypted K_m to generate the first-level encrypted K_m. It sends the result to the DPRE_DEV.
5) The DPRE_DEV decrypts the first-level encrypted K_m with the temporal private key.

VII. Conclusion

In this paper, we proposed a secure mutual-profitable interoperable DRM scheme which guarantees the needs and requirements of both providers and consumers. Our scheme uses designated proxy re-encryption and neutral format schemes to minimize the disclosure of security properties of DRM technology providers and content providers, and suggests a billing scenario to encourage the participation of both providers to solve the DRM interoperability problem. Therefore, our scheme meets the needs of consumers and providers, and allows for effective interoperable DRM systems.

References

[1] H. Park, S. Lee, and J. Kim, "Rights-preserving interoperability protocol in DRM," in *Proceedings of Conference on Information Security and Cryptology - Winter 2008 (CISC-W08)*, 2008, pp. 141–144, text in Korean.

[2] S. Michiels, W. Joosen, E. Truyen, and K. Verslype, "Digital rights management—a survey of existing technologies," Department of Computer Science, Katholieke Universiteit Leuven, Tech. Rep., November 2005.

[3] B. Rosenblatt, B. Trippe, and S. Mooney, *Digital Rights Management—Business and Technology*. New York: M&T Books, 2002.

[4] N. W. Netanel, "Temptations of the walled garden: Digital rights management and mobile phone carriers," *Journal on Telecommunications and High Technology Law*, vol. 6, 2007.

[5] R. H. Koenen, J. Lacy, M. Mackay, and S. Mitchell, "The long march to interoperable digital rights management," *Proceedings of IEEE*, vol. 92, no. 6, pp. 883–897, June 2004.

[6] N. Dufft, A. Stiehler, D. Vogeley, and T. Wichmann, "Digital music usage and DRM—results from an european consumer survey," May 2005, http://www.indicare.org/survey.

[7] ISO/IEC 14496-13, "Information technology: Generic coding of moving pictures and associated audio information, part 2: IPMP on MPEG-2 systems," 2002.

[8] D. W. Kravitz and T. S. Messerges, "Achieving media portability through local content translation and end-to-end rights management," in *Proceedings of the 5th ACM workshop on Digital Rights Management*, 2005, pp. 27–36.

[9] D.-W. Nam, J.-S. Lee, and J.-H. Kim, "Interlock system for DRM interoperability of streaming contents," in *Proceedings of IEEE International Symposium on Consumer Electronics 2007 (ISCE 2007)*, 2007.

[10] G. Taban, A. A. Cárdenas, and V. D. Gligor, "Towards a secure and interoperable DRM architecture," in *Proceedings of the 6th ACM workshop on Digital Rights Management*, 2006, pp. 69–78.

[11] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 1, pp. 1–30, 2006.

[12] A. Arnab and A. Hutchison, "Digital rights management—an overview of current challenges and solutions," in *Proceedings of Information Security South Africa (ISSA)*, 2004.

[13] F. Bartolini, V. Cappellini, A. Piva, and A. Fringuelli, "Electronic copyright management systems: Requirements, players and technologies," in *Proceedings of the 10th International Workshop on Database and Expert System Applications*, 1999, pp. 896–898.

[14] D. Mulligan, J. Han, and A. J. Burstein, "How DRM-based content delivery system disrupt expectations of "personal use"," in *Proceedings of the 3rd ACM workshop on Digital Rights Management*, 2003, pp. 77–89.

[15] J. Park, R. Sandhu, and J. Schifalacqua, "Security architectures for controlled digital information dissemination," in *Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC'00)*, 2000, pp. 224–233.

[16] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246 (Proposed Standard), Internet Engineering Task Force, Aug. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5246.txt

[17] M.-S. Hwang, I.-C. Lin, and L.-H. Li, "Simple micro-payment scheme," *Journal of Systems and Software*, vol. 55, no. 3, pp. 221–229, January 2001.

[18] Trusted Computing Group, http://www.trustedcomputinggroup.org/.

[19] The JHU-MIT Proxy Re-cryptography Library, http://spar.isi.jhu.edu/prl/.

[20] Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL), http://www.shamus.ie/.