Applying OM-AM to Analyze Digital Rights Management

(Extended Abstract)

Shouhuai Xu[1] and Ravi Sandhu[2]

[1] Department of Computer Science, University of Texas at San Antonio
shxu@cs.utsa.edu
[2] Department of Information and Software Engineering, George Mason University
sandhu@gmu.edu

Abstract. Digital Rights Management (DRM) is an important topic, and has no satisfactory solution yet. Failures of real-life systems suggest that the current solutions seem fundamentally flawed, and that pure technology-based solutions may not be sufficient. This observation encourages us to systematically re-examine the notion of digital rights management utilizing the OM-AM engineering principle. Our investigation leads to a new business and system model, which may substantially ease our reliance of DRM on technologies. Besides applying OM-AM to analyze DRM at a high level, we present a case study on protecting digital music. We introduce a notion called *executable license*, which is intended to encapsulate the adopted access control mechanisms, and thus may be independently interesting.

1 Introduction

Digital Rights Management (DRM) is an important topic, and has no satisfactory solution yet. Failures of real-life systems (e.g., [CWL+01,FAQ]) suggest that the current solutions seem fundamentally flawed, and that pure technology-based solutions may not be sufficient. This observation encourages us to systematically re-examine the notion of digital rights management. For this purpose, we utilize the OM-AM engineering principle [S00].

1.1 Our Contributions

The contributions of this paper are three-fold. First, given that there are currently no satisfactory solutions to DRM and that there is a lack of comprehensive investigation on the notion of digital rights management (see Section 5 for an overview of related work), we believe that a systematic re-examination can help broaden and deepen our understanding of DRM. Hopefully, this paper will trigger more investigations in this regard.

Second, our investigation does show that the OM-AM engineering principle, which was originally proposed for guiding the design of large-scale secure information systems, can be used to analyze DRM. Our investigation leads to a new business and system model, which somewhat reflects the successful credit card industry and the taxation system in the US. In particular, the new model gives people incentives to maintain a good reputation, as does the credit card system where one has incentives to maintain a good credit record, and the taxation system where most people have incentives to be honest.

Third, besides applying OM-AM to analyze DRM at a high level, we conduct a case study on protecting digital music. We introduce a notion called *executable license*, which is intended to encapsulate the adopted access control mechanisms, and thus may be independently interesting.

1.2 Organization of the paper

In Section 2 we briefly review the OM-AM engineering principle. In Section 3 we investigate the notion of digital rights management under the guideline of OM-AM. In Section 4 we conduct a case study on protecting digital music. We discuss related work in Section 5 and give our conclusions in Section 6.

2 OM-AM Engineering Principle

The OM-AM engineering principle was proposed by Sandhu [S00] for designing large-scale secure information systems, where OM-AM stands for *objective*, *model*, *architecture*, and *mechanism* layers in sequence. The objective and model (OM) layers articulate *what* the security objectives and trade-offs are, whereas the architecture and mechanism (AM) layers address *how* to meet these requirements. The hyphen in OM-AM emphasizes the shift from *what* to *how*.

Realizing that objective, model, architecture, and mechanism are highly overloaded words and mean different things to different communities, and that there may be some fuzziness in exactly where we draw the boundary between successive layers, OM-AM does not seek to give airtight meaning to these words but rather is an informal and intuitive engineering framework [S00]. This flexibility allows us to interpret OM-AM in the following way:

- a system achieving certain security objectives is viewed as a cube as depicted in Fig. 1;
- the model specifies which participants interact which participants (i.e., a bird's-eye view) for what purpose;
- the architecture specifies the layers of the system (e.g., the application layer and the infrastructure layer);
- the mechanisms are seamlessly incorporated into the system to implement the objectives according to the architecture.

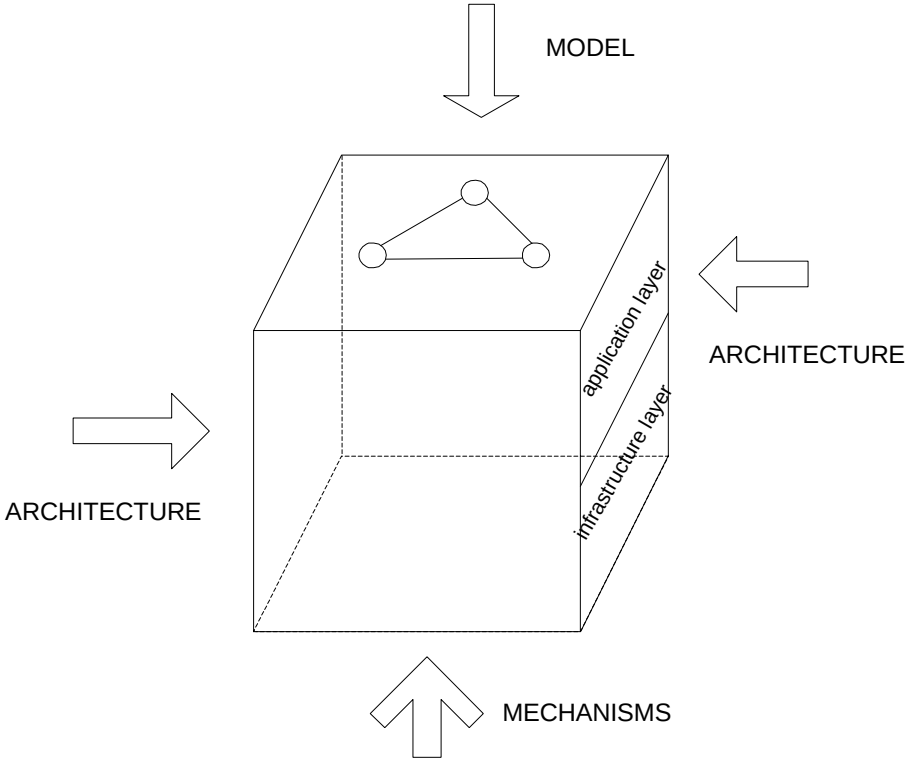

Fig. 1. The model, architecture, and mechanism in a system achieving certain objectives

3 Applying OM-AM to Analyze DRM

In this section we show how to apply OM-AM to analyze DRM. This section is organized in the order of objective, model, architecture, and mechanism.

3.1 Objective

We treat digital rights management as a business problem pursuing economic benefit while utilizing technical mechanisms. Denote by α the revenue for a given time period while deploying no new digital rights management system (e.g., the revenue of the current DVD industry in the next 10 year), β the revenue within the same time period while deploying a new digital rights management system (e.g. the revenue of the DVD industry in the next 10 year after deploying a new digital rights management system), δ the expense for deploying the new digital rights management system. Then, the digital rights management system makes sense only if $\alpha < \beta - \delta$.

3.2 Model

Inspired by the success of the credit card industry and the taxation system in the US, we propose the following DRM system model as shown in Fig. 2. This model aims at reflecting the following observations: In the credit card system an honest user can get increasingly better service, whereas in the taxation system, a user periodically contact the taxation authority. There may also be a punishment mechanism, which ensures that a dishonest user, once identified, will be punished. There are four categories of participants: users, digital rights owners, content distributors, and digital rights management authority.

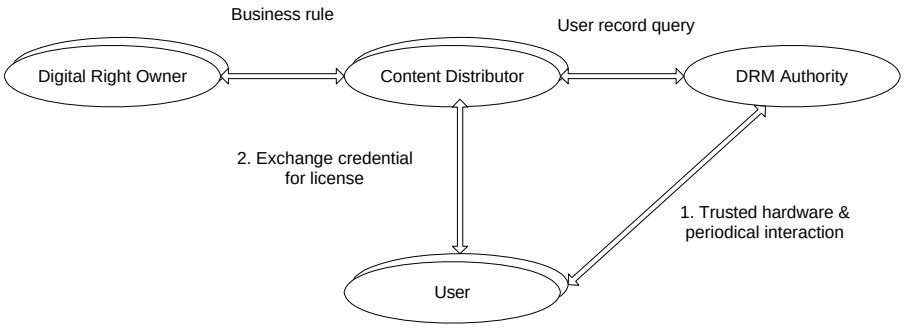

Fig. 2. The digital rights management system model

- Users: The users make use of digital content obtained from the digital content distributors. For this purpose, they may need to pay for the digital content. The use of digital content is done via certain trusted hardware obtained from the digital rights management authority (DRMA). The users need to periodically interact with the DRMA for checking the integrity of the trusted hardware.
- Digital rights owners: An owner does business with the content distributor, who actually distributes the content for their revenue. The concrete interaction is a pure business problem and thus beyond the scope of this paper. So, we simply assume that there is a set of well-defined business rules for content owners and distributors.
- Content distributors: They are the merchants who distribute the digital content. They interact with all of the other participants in the model: the digital rights owners, the users, and the DRMA. The interaction between a content distributor and a digital rights owner is based on some well-defined business rules. The interaction between a content distributor and a user is based on some flexible rules that give incentives to the users to be honest, as we will elaborate below. The interaction between a content distributor and the DRMA is to maintain transaction records for the users, as we will also elaborate later.
- DRM Authority (DRMA): This is a newly introduced participant and doesn't have to be a government agency. It manages the trusted hardwares it issued to the users, interacts with content distributors to maintain the user's transaction records to ensure that honest users will get cheaper and better services, refunds the honest users for their loyalty, etc. It bears resemblance to the credit card issuers or the tax authority in the real world. The DRMA may be hierarchical.

3.3 Architecture

The architecture specifies how the participants interact. We adopt a two-layer architecture: the application layer and the infrastructure layer. All the interactions should be protected with appropriate mechanisms, which will be elaborated later.

Application Layer. At the application layer, a user purchases protected digital contents from content distributors. See Fig. 3.

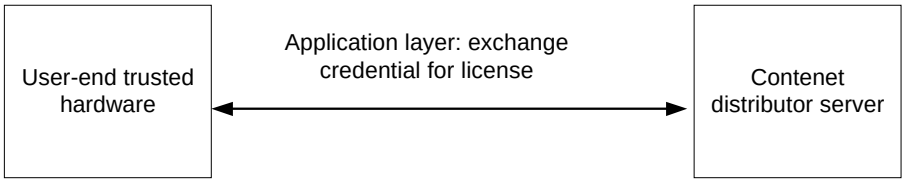

Fig. 3. The application layer

After obtaining protected digital content, the user makes use of content. The use of the protected digital content is done in some secure environment whereby it is guaranteed that the content will not be leaked. We will specify such a platform in our case study (see Section 4).

Infrastructure Layer. At the infrastructure layer, there are a set of protocols that facilitate the digital right management system. See Figure 4.

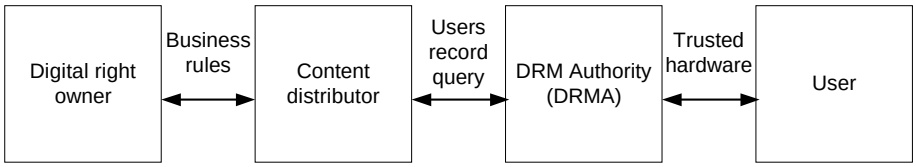

Fig. 4. The infrastructure layer

The protocol between a digital rights owner and a content distributor is the same as in the real world and beyond the scope of this paper, and thus omitted. The protocol between a content distributor and the DRMA is for retrieval of the users' reputation. The interactions between a user and the DRMA include an initialization process and periodic interactions. In the initialization process, a user registers at the DRMA (just like one applying for a credit card) and obtains a trusted hardware. The periodic interactions between a user and the DRMA are to maintain the integrity of the trusted hardware, based on which (and other factors) the DRMA can compute the user's reputation.

3.4 Mechanisms

In order to achieve the security objectives, we need to incorporate into the system a set of mechanisms of desirable properties. Besides well-known cryptographic mechanisms such as public key cryptosystems and digital signature schemes, we will need the following technical and non-technical mechanisms.

Trusted Hardware. Since tamper-resistant software is currently not available (see Section 5 for discussion), we adopt a trusted hardware (e.g., smart card) with cryptographic processing capability to build a trusted

runtime environment. We assume that the trusted hardware is immune to side-channel attacks such as those presented in [KJJ99]. We bear in mind, however, that it may not be possible to produce absolutely secure hardware, and therefore we may require a weaker property we call *tamper-evident*, which means that the DRMA can check (via either physical or logical methods) if a trusted hardware has been tampered with. Of course, such a process should disturb a user as little as possible. While physical checkup may be a matter of economic incentives, logical checkup does require the trusted hardware to have more advanced properties.

Access Control. Access control specifies what methods a user can operate on a given item of digital content. Theoretically, any access control mechanism can be utilized so that, for example, there is a *control set* specifying the policy and a *license* specifying the authorization to a user. In order to make a digital rights management system "access control policy neutral" (recall that the DRMA only provides trusted hardware for a runtime platform), we propose the concept of "executable licenses" that consist of the authorization (i.e., the traditional license) as well as the runtime system (i.e., the executable provided by the content distributor). An executable license is a program that can run on certain platforms and reflects the authorization to the user possessing the license. On the one hand, the executable licenses may be signed by the content distributor so that the licenses will not harm the trusted hardware. On the other hand, the trusted hardware may need to maintain the state information for the executable licenses if they are stateful (e.g., the license should count the number of times the digital content has been used).

Watermarking and Fingerprinting. Watermarks [CKLS97] and fingerprints [BS95] can be used to identify the copyright owner of a digital content. A robust watermarking or fingerprinting is hard to remove without significantly degrading the quality of the content unless, ideally, the adversary must know some cryptographically strong secret. In order to trace a pirate copy back to the illegal re-distributor, each copy of the digital content can be embedded with a different fingerprint that identifies the corresponding owner.

Non-technical Mechanisms. A non-technical mechanism (e.g., flexible-pricing) ensures that honest users will pay discounted prices, which gives them the incentives to buy genuine copies. Below we describe two typical scenarios.

- Denote by a the regular price of a digital content item, b the reputation of a user (maintained by the DRMA). Then the user can buy this content at the price of $f(a, b) < a$, where f is an appropriate function. Moreover, a user buying a fingerprinted version may get further discount so that she only needs to pay $f'(a, b) < f(a, b)$, where f' is another function.
- Suppose a group of n people want to buy a digital content item with a unique fingerprint. Denote by a the regular price of a digital content item, b_i the reputation of the i^{th} user. Then the actual price for the i^{th} user is $g(a; b_1, b_2, \cdots, b_n; i) < f'(a, b_i)$, where $1 \leq i \leq n$.

4 Protecting Music: A Case Study

In this section we present a case study on protecting digital music. In the specification below, we only discuss those components that cannot be directly inherited from the general analysis in Section 3. For simplicity, we assume that each user pays for digital content. This can be easily extended to the cases such as music rental.

4.1 Design Considerations

There are trade-offs that need to be made. First, what does the user-end platform looks like? There are two choices: either the user-end platform is a player being trusted hardware, or consists of a player as well as the trusted hardware. We adopt the latter because the trusted hardware can then be used for multiple purposes (e.g., protecting music, video), and because it also eases it eases the periodic integrity inspection process. Since a player communicates with the trusted hardware, a potential attack is the so-called man-in-the-middle attack where an attacker impersonates either the player or the trusted hardware. To block this attack, we assume that both the player and the trusted hardware have cryptographic processing capability and possess their own pair of public and private keys.

Second, should the user-end be allowed to verify watermarks? There are two strategies: (1) A song is embedded with a watermark indicating the copyright owner and a fingerprint indicating the user so that the watermark can be verified by the players, whereas the fingerprint can only be verified by the content distributor; (2) each copy of a song is embedded with a watermark or a fingerprint that the user-end can not verify. We adopt the latter strategy for simplicity.

4.2 Initialization

The initialization of the system includes:

1. A DRMA is established. It is responsible for issuing trusted hardware, maintaining a database for the transactions of the users, certifying the public keys of any relevant components in this system, and implementing an algorithm to compute a user' reputation.
2. A user needs to obtain a player (e.g., from a shop), and trusted hardware from the DRMA (e.g., after identifying herself to the DRMA). Suppose both the player and trusted hardware have a pair of public and private keys, $(pk_{player}, sk_{player})$ and (pk_{user}, sk_{user}), respectively. Both pk_{player} and pk_{user} are certified by DRMA, and neither sk_{player} nor sk_{user} is known to the user. Denote by $cert_{user} = SIG_{DRMA}(pk_{user})$ the certificate on pk_{user}, where SIG is the signature algorithm of a secure digital signature scheme.

4.3 Infrastructure Layer Protocols

The protocol between a digital rights owner and a content distributor is the same as in the real world, and is beyond the scope of this paper. A content distributor needs to contact the DRMA for two purposes: to send the information of the purchase transactions to the DRMA, and to retrieve the reputation of a user at a purchase transaction when the discounting strategy is adopted. A user needs to contact the DRMA for periodically inspecting the integrity of her trusted hardware.

4.4 Application Layer Protocols

We present two purchase protocols: one is based on the *discounting* strategy and the other is based on the *refunding* strategy. For the sake of simplicity, we assume that there is an appropriate payment mechanism available.

Purchase protocol based on discounting strategy. This protocol is executed by a user and a content distributor.

1. The user sends the content distributor the public key certificate $cert_{user}$ and the identity of the song she wants to purchase. Suppose the regular price of the song is a.
2. The content distributor retrieves the reputation corresponding to $cert_{user}$ by contacting the DRMA via a secure communication channel.
3. The DRMA returns the user's reputation b_{user} back to the content distributor via the secure communication channel.
4. The content distributor computes the discounted price $f(a, b_{user})$, where f is an appropriate function.
5. The user pays $f(a, b_{user})$ to the content distributor.
6. The content distributor sends back to the user ENC_{key}(watermarked song), ENC_{key1}(executable license), and $ENC_{pk_{user}}(key, key1)$, where $ENC_X()$ is the encryption algorithm of a secure symmetric key cryptosystem, $ENC_{pk_X}()$ is the encryption algorithm of a secure public key cryptosystem, and s is a cryptographically strong secret used by the content distributor to verify the watermark.
7. The content distributor sends the information about this transaction to the DRMA, which updates its database for this user.

Purchase protocol based on refunding strategy. This protocol is executed by a user and a content distributor.

1. The user sends the content distributor the public key certificate $cert_{user}$ and the identity of the song she wants to purchase. Suppose the regular price of the song is a.
2. The user pays a to the content distributor.
3. The content distributor sends back to the user ENC_{key}(watermarked song), ENC_{key1}(executable license), and $ENC_{pk_{user}}(key, key1)$, where $ENC_X()$ is the encryption algorithm of a secure symmetric key cryptosystem, $ENC_{pk_X}()$ is the encryption algorithm of a secure public key cryptosystem, and s is a cryptographically strong secret used by the content distributor to verify the watermark.
4. The content distributor sends the information about this transaction to the DRMA, which updates its database for this user. (Eventually, the DRMA should refund the user.)

Playback function. After obtaining an encrypted item of digital music from a content distributor, the user can playback the music on the platform consisting of a player as well as a trusted hardware. The watermarked music is encrypted under the key key and the executable license is encrypted under the key $key1$, both of the two keys are encrypted under the public key of the trusted hardware. The playback function has following steps (see also Fig. 5).

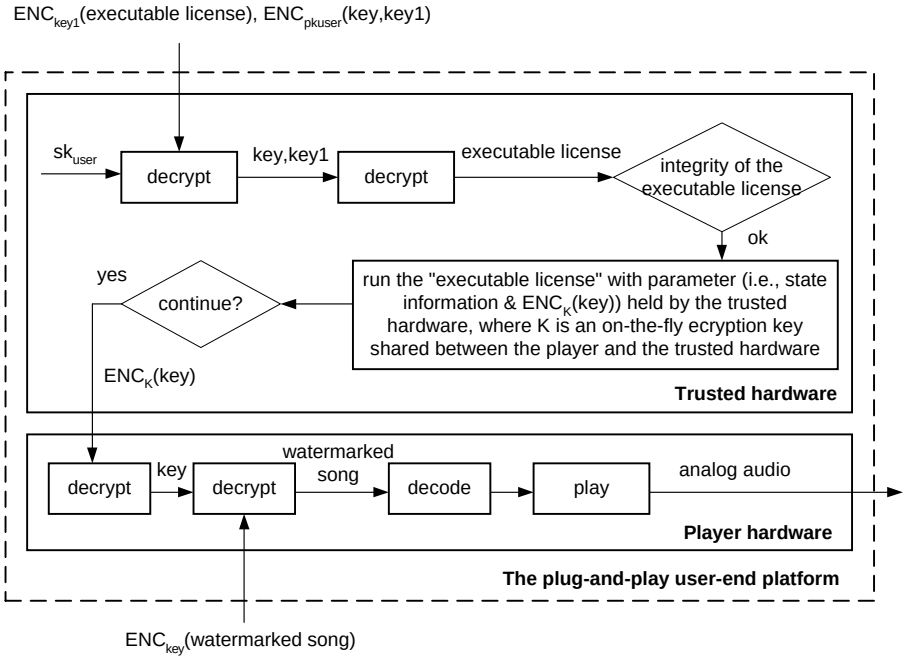

Fig. 5. The playback function

1. The input to the trusted hardware is ENC_{key1}(executable license) as well as $ENC_{pk_{user}}(key, key1)$. Suppose the player and the trusted hardware establish a key K using a secure mutual-authenticated key exchange protocol.
2. The trusted hardware decrypts ENC_{key1}(executable license) and then authenticates the plaintext "executable license" (e.g., its source and integrity). If it passes this verification, the trusted hardware runs the license by feeding it with parameters $ENC_K(key)$ that will be forwarded to the player, and possibly some state information corresponding to this license (e.g., the number of times this license has been executed).
3. The executable license decides if the request should be approved (e.g., based on the state information from the trusted hardware and the license itself). The execute license may need to ask the trusted hardware to update certain state information (e.g., increment the counter).

4. The player decrypts $ENC_K(key)$ to get key whereby it can playback the watermarked music.

Remark. The key key used to encrypt the music is indirectly sent to the player via the trusted hardware. While it can be directly encrypted under the public key of the player, this solution has the disadvantage that a user has to bind herself to a single player, which may be over-restrictive.

4.5 Putting Things Together

Putting things together, we get a system based on discounting strategy in Fig. 6, and a system based on refunding strategy in Fig. 7.

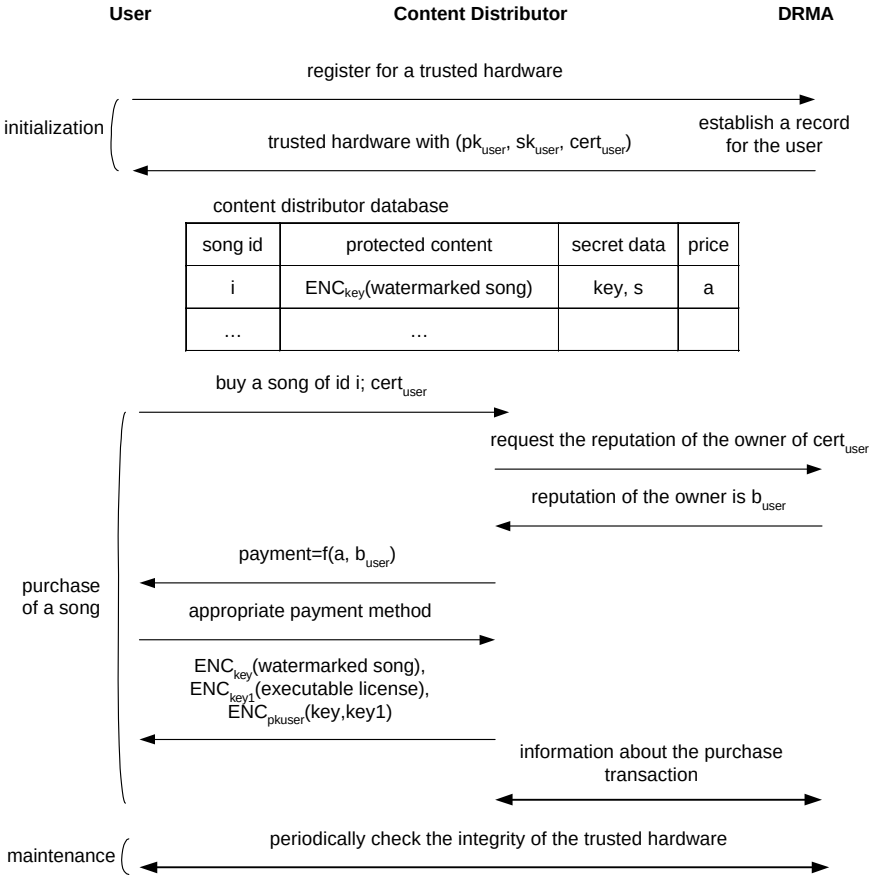

Fig. 6. A system based on discounting strategy

5 Related Work

We discuss related work in the order of business model, system model, architecture, and mechanisms. There have been some investigations on re-examining the concept of DRM [Be01,Bu01,C02]. In particular, it has been pointed out that an appropriate business model may be the key to the success of a DRM system [LSM97,FAQ]. For example, a sensible business model that combines pricing, ease of use, and legal prohibition in a way that minimizes the incentives for consumers to deal with pirates. This indeed inspired us to conduct a systematic re-examination of the notion of digital rights management. Although the above cited

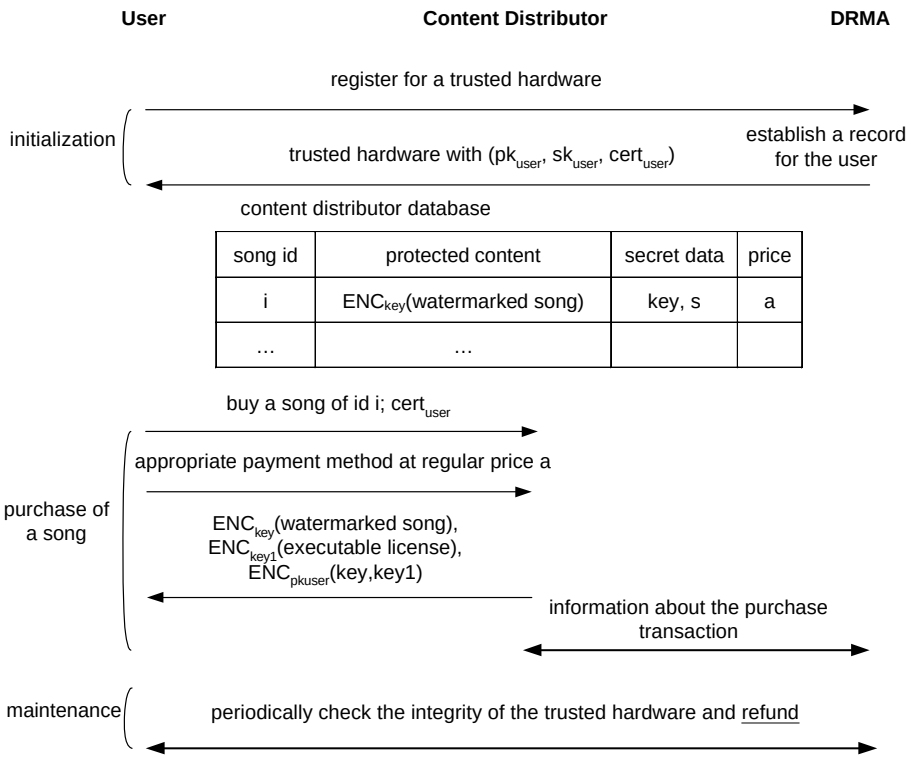

Fig. 7. A system based on refunding strategy

works seemingly draw their observations from the failures of real-life DRM systems, none of them did a systematic investigation as ours. Our business model, nevertheless, corresponds to the "super-distribution" in the taxonomy of [Bu01].

In our system model, it is assumed that the DRMA can periodically inspect, physically or logically, the integrity of the trusted hardwares it issued to the users. This is reminiscent of the strategy adopted to protect satellite video services such as SKY TV [M96]: the system designers have been able to issue electronic countermeasures to make the security measures renewable. An important difference between the protection of satellite TV and digital music is that the decryption function of the trusted hardware in a digital music system cannot be completely renewed, since a user may always need to play the songs she bought earlier. Our system model is also somewhat correlated to the software aging model [JR01]. Finally, it is argued in [L02] that trustworthy computing devices, robust trust management engines, and a general-purpose rights expression/authorization language are three indispensable components in a robust DRM system.

In the regard of system architecture, our system is reminiscent of [KYS01], which however focused on the mechanisms such as the individualization of secure application component (in our terms, trusted hardware).

There have been considerable literature in investigating DRM mechanisms. Traitor tracing (see, e.g., [KY02,DFKY03]) is useful in protecting digital contents distributed over a broadcast channel. The content provider gives each subscriber a decoder that contains a secret decryption key whereby the subscriber can decrypt the protected content, while the provider can recover the identity of some of the traitors who participated in the construction of a pirate decoder. Digital signets [DLN97] ensure that a pirate has either to reveal some sensitive information (such as a credit card number), or to use a channel of essentially the same bandwidth as the original distribution channel. We utilized trusted hardware, given the current inadequate security guarantee of software-based protection techniques (see, e.g., [CT00]). Theoretical analysis showed that software protection using trusted hardware is possible [GO96], whereas pure software-based obfuscation (i.e., without using trusted hardware) is impossible in the (natural but very strong) black-box model [BGI+01]. Nevertheless, as it has been noted in [BGI+01], it may still be possible to achieve software

obfuscation in a weaker sense (see, e.g., [CA01,HMST01,CEJ+02a,CEJ+02b]). Finally, [H02] showed that the current practice – introducing deliberate data errors into discs during manufacturing to cause incompatibility with PCs without affecting ordinary CD players – is harmful to legitimate CD owners.

6 Conclusion and Future Work

We applied the OM-AM engineering principle to analyze digital rights management. This leads to a new business and system model for digital rights management system, and a concept called "executable license" that can be used to encapsulate access control policies. We also conducted a case study for the protection of digital music. Our investigation suggests interesting research problems.

- How to concretize the pricing mechanisms?
- How to minimize the affect of our DRM system on users' privacy?

References

[BGI+01] B. Barak, O. Goldreich, R. Impagliazzo et al. On the (Im)possibility of Obfuscating Programs. In the Proceedings of Crypto'2001.

[Be01] S. Bechtold. From Copyright to Information Law – Implications of Digital Rights Management. In the Proceedings of Digital Right Management Workshop 2001.

[BS95] D. Boneh and J. Shaw. Collusion-Secure Fingerprinting for Digital Data. In the Proceedings of Crypto'95.

[Bu01] W. Buhse. Implications of Digital Rights Management for Online Music – A Business Perspective. In the Proceedings of Digital Rights Management Workshop 2001.

[C02] J. Camp. DRM Doesn't Really Mean Digital Copyright Management. In the Proceedings of ACM Conference on Computer and Communcations Security 2002.

[CA01] H. Chang and M. Atallah. Protecting Software Code by Guards. In the Proceedings of Digital Rights Management Workshop 2001.

[CEJ+02a] S. Chow, P. Eisen, H. Johnson, and P. van Oorschot. White-Box Cryptography and an AES Implementation. In the Proceedings of Selected Areas in Cryptography 2002.

[CEJ+02b] S.Chow, P. Eisen, H. Johnson, and P.C. van Oorschot. A White-Box DES Implementation for DRM Applications. In the Proceedings of Digital Rights Management Workshop 2002.

[CKLS97] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure Spread Spectrum Watermarking for Multimedia. IEEE Trans. on Image Processing, vol. 6, no. 12, 1997, pp 1673-1687.

[CT00] C. Collberg and C. Thomborson. Watermarking, Tamer-proofing, and Obfuscation. Technical Report TR00-03, Department of Computer Science, University of Arizona, 2000.

[CWL+01] S. Craver, M. Wu, B. Liu et al. Reading between the Lines: Lessons from the SDMI Challenge. In the Proceedings of Usenix Security'01.

[DFKY03] Y. Dodis, N. Fazio, A. Kiayias, and M. Yung. Scalable Public-Key Traitor Tracing. In the Proceedings of ACM Principle of Distributed Computing 2003, to appear.

[DLN97] C. Dwork, J. Lotspiech, and M. Naor. Digital Signets: Self-Enforcing Protection of Digital Information. In the Proceedings of ACM STOC'97.

[FAQ] SDMI Challenge FAQ, http://www.cs.princeton.edu/sip/sdmi/faq.html.

[GO96] O. Goldreich and R. Ostrovsky. Software Protection and Simulation on Oblivious RAMs. Journal of the ACM, vol. 43, no. 3, 1996, pp 431-473.

[H02] J. Halderman. Evaluating New Copy-Prevention Techniques for Audio CDs. In the Proceedings of Digital Rights Management Workshop 2002.

[HJJY00] J. Hastad, J. Jonsson, A. Juels, and M. Yung. Funkspiel Schemes: An Alternative to Conventional Tamper Resistance. In the Proceedings of ACM Conference on Computer and Communications Security 2000.

[HMST01] B. Horne, L. Matheson, C. Sheehan, and R. Tarjan. Dynamic Self-Checking Techniques for Improved Tamper Resistance. In the Proceedings of Digital Rights Management Workshop 2001.

[JR01] M. Jakobsson and M. Reiter. Discouraging Software Piracy using Software Aging. In the Proceedings of Digital Rights Management Workshop 2001.

[KY02] A. Kiayias and M. Yung. Breaking and Repairing Asymmetric Public-key Traitor Tracing. In the Proceedings of Digital Rights Management Workshop 2002.

[KYS01] D. Kravitz, K. Yeoh, and N. So. Secure Open Systems for Protecting Privacy and Digital Services. In the Proceedings of Digital Rights Management Workshop 2001.

[LSM97] J. Lacy, J. Snyder, and D. Maher. Music on the Internet and the Intellectual Property Protection Problem. In the Proceedings of IEEE International Symposium on Indurstrial Electronics, 1997.

[L02] B. LaMacchia. Key Challenges in DRM: An Industry Respective. In the Proceedings of Digital Rights Management Workshop 2002.

[M96] J. McCormac. European Scrabmbling Systems, Circuits, Tactics, and Techniques. Waterford University Press, 1996.

[KJJ99] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In the Proceedings of Crypto'99.

[S00] R. Sandhu. Engineering Authority and Trust in Cyberspace: The OM-AM and RBAC Way. In the Proceedings of ACM Workshop on Role Based Access Control 2000.

[SV01] W. Shapiro and R. Vingralek. How to Manage Persistent State in DRM Systems. STAR-TR-01-06, 2001, InterTrust Inc.