International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163
Issue 3, Volume 2 (March 2015) www.ijirae.com

DIGITAL RIGHTS MANAGEMENT SYSTEM FOR MULTIMEDIA FILES

Saiprasad Dhumal *
VJTI, Mumbai.

Prof. K.K. Joshi
VJTI, Mumbai

Prof Sowmiya Raksha
VJTI, Mumbai.

Abstract— piracy of digital content is a one of the challenge faced by content publishers and computer code vendors these days. Digital piracy is unauthorized repetition and/or distribution of computer code, music or films. Pirated software package hurts everybody from software package developers to any other digital content publisher. Moreover, the black-market duplication and distribution of software package contains a significant impact on the economy. Piracy exists within the domain of music, videos, games and software package. Completely different solutions exist for music and video antipiracy and software, games antipiracy. The DRM framework uses cryptographic techniques and supports protection of digital content viz., audio files by implementing user rights like read, copy, play or print as applicable. The look may be extended to protection to varied file formats with a DRM license which will be upgraded for added rights or be renewed to induce an extended validity. The DRM framework additionally accommodates offline use of protected content by a one-time (initial) setup and a user license hold on regionally. Finally, the paper analyzes the look for DRM's crucial necessities like security, flexibility, efficiency and interoperability.

Keywords— music, videos, games and software , Digital rights management; content protection; cryptography; software security; software anti-piracy

I. INTRODUCTION

In this digital era, information sharing and unauthorized distribution of digital content has grown multifold. The new wave of social networking sites augmented by handheld gadgets like smart phones, have added to the complexity. Content publishers and software vendors find it increasingly difficult to protect their work and copyrights. E-books and multimedia files are freely distributed over internet and software cracks are created to bypass vendor's restrictions. Software vendors are forced to adapt strong anti-piracy enforcement strategies and technologies to prevent such risks and check intellectual property (IP) infringements. Digital rights management (DRM), as a collection of access control technologies can be used by copyright holders, publishers and hardware manufacturers to limit the usage of digital content and devices. Digital content can be in the form of documents, e-books, audio, video and games. Digital rights management controls the access to sensitive content by including information about the user rights of the content in the form of a license. Such rights include information on the duration of the file to be accessed and permissions to read or print the content. A user license to that effect is issued to the client for consumption of the content.

On the other hand, software licensing solutions are used to control or dictate permissions related to software. A software license may include a time limit for the use of software. Various control strategies are instilled to hinder unauthorized duplication and use of software. One such approach is to provide a hardware dongle for software interlock. Majority of software distributed in CD-ROM requires the user to enter a key number or customer identification number before installing the software. Yet another approach requires registering the software with the manufacturer to obtain an operational code necessary to install the software. Some software licensing solutions require the end user to activate her subscription on internet or phone that sets the validity date on the server hosted by the vendor. The installed client software communicates periodically with the server to ensure that the subscription is valid. These solutions are designed for legitimate use of software. However, reverse engineering of software libraries facilitate the attackers to acquire sensitive data by disassembling and analyzing the design of system components. This causes substantial loss to software vendors from copyright and IP infringements. Most of the reverse engineering techniques are used to understand software architecture, circumvent software restrictions and access source code.

II. BACKGROUND

A survey on DRM technologies is presented. Moreover, a range of DRM solutions are available in the market such as Microsoft Active Directory Rights Management Services (ADRMS) [4], ADOBE Content server [5] and IBM's WebGuard [6] and DRM-JVM [7]. ADRMS primarily protects Microsoft Office documents using DRM licenses and certificates combined with encryption techniques. The encrypted content is decrypted by the end user and consumed. For consumption, the end user should have connectivity to ADRMS server to obtain a DRM license. Adobe's Content Server is designed to protect PDF documents and e-books on a wide variety of platforms and handheld devices. IBM's WebGuard is primarily designed to protect web content like html, image and audio content. DRM-JVM is a DRM enabled Java virtual machine. DRMJVM implementation is based on two components: the DRM Security Manager and the DRM Protocol Handler. These components working together are responsible for enforcing the rights acquired by the user. If the JVM refuses the installation of the DRM Security Manager, access to protected content is denied. Typically, DRM solutions, with some exceptions such as DRM-JVM, do not offer any default protection to software applications.

Software vendors combine code obfuscation techniques along with software license solutions to protect their products from reverse engineering, tampering and exploitation. Software guards, encoding techniques, and watermarking techniques are also used to hide and track source code. The limitations of these solutions are as follows.

- *Code obfuscators can resist reverse engineering techniques to some extent but cannot offer a foolproof protection*
- *Password based protection techniques are not robust and often come up with shortcomings such as sharing of passwords or reuse of serial numbers.*
- *Watermarking solutions can act as deterrents but cannot actively prevent misuse of software.*

Further, in the case of evaluation software, a serial number is provided by the software vendor to activate the product. The software generally stores the activation date in registry and checks the evaluation period based on that. This protection is easily overcome by evaluators by clearing the registry entities or resetting the system clock. Moreover the same serial number is used on different machines to get access to multiple installations. In some scenarios, the evaluation period needs to be extended legitimately. For this purpose, the user needs to request a separate license. However, the vendor has no control on the license already issued, in the event of an evaluator violating the licensing terms. When users change roles or leave the organization, some user rights need to be revoked and new rights need to be granted. But, in most of the DRM solutions revocation of rights is cumbersome and requires a new license to enforce it. The framework discussed in the paper is robust and targets to overcome the limitations stated above. The framework offers:

- *A common framework to protect digital objects and software libraries*
- *Extensibility of DRM framework to protect multiple digital formats and support for variety of clients on different platforms*
- *Protection of software libraries from reverse engineering*
- *DRM controlled software execution and implementation of software vendor's restrictions and constraints*
- *Extensibility of design to support other types of licensing model viz., subscription, rentals, trial licenses, one-time use software and evaluation copies like try-before-you-buy or try only.*
- *Control of trial versions by granting additional rights on need basis and even revocation of existing rights*
- *Support for online and offline models, so that appropriate type of delivery of content and licensing methods can be used*
- *Support for offline DRM users with one-time activation or registration.*

III. DRM FRAMEWORK

The proposed DRM design consists of a DRM server that is responsible for managing user rights and a DRM client to enforce user rights. The DRM client can be in online or offline mode. The DRM administrators can add, delete and edit the users in the DRM. The users classified as publisher have the right to publish their content and assign rights to others. Consumers are end users of the content.

To solve the problem of unauthorized copying and limiting the access to the rightful individuals, the digital content will be initially encrypted by the publisher. The publisher who owns the distribution rights (or the content itself) sets the user's rights. These rights include permissions such as print, view, execute, play and constraints like time limit or number of views. The encrypted content is distributed to users for consumption. We now discuss two modes of operation for DRM in terms of distribution and use of content.

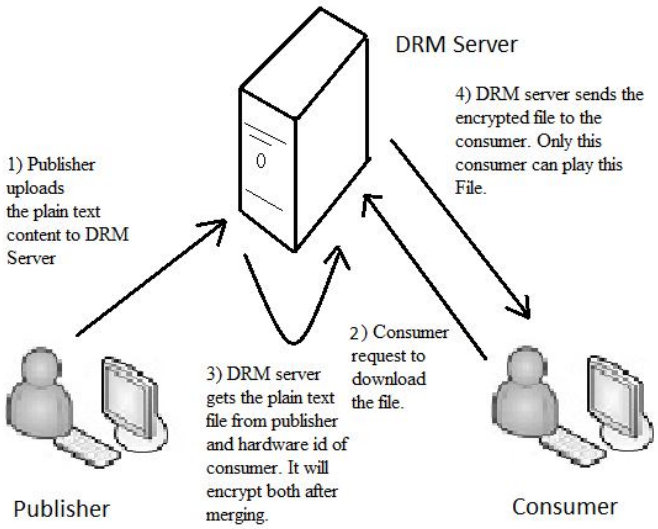

Figure 1. DRM architecture

In this model, the DRM client has to connect to the DRM server to consume the content (Fig. 1). Once connected, the end user authenticates to the server and gets a license through a secure session (https). The license that includes the secret key (for content decryption) along with information on rights is transferred to the client during the consumption of the content.

The DRM client decrypts the content in the memory and enforces the rights specified in the license. In this case, neither the decrypted content nor the DRM license is locally stored. Further, the rights granted by the publisher, for the user, are fetched from the server and enforced. By changing the rights information on the server the publisher can grant new rights or extend existing privileges to the user. This online model provides the greatest flexibility when assigning rights to any combination of users and content. Offline DRM model is applicable to the scenarios where the DRM client has limited connectivity to DRM server. In this case, the client can choose to work offline after initial connectivity to the DRM server. The DRM server establishes the trust among the client's workstation, end user. We now explain the four phases in offline DRM model. Fig. 2 describes the entities and phases involved.

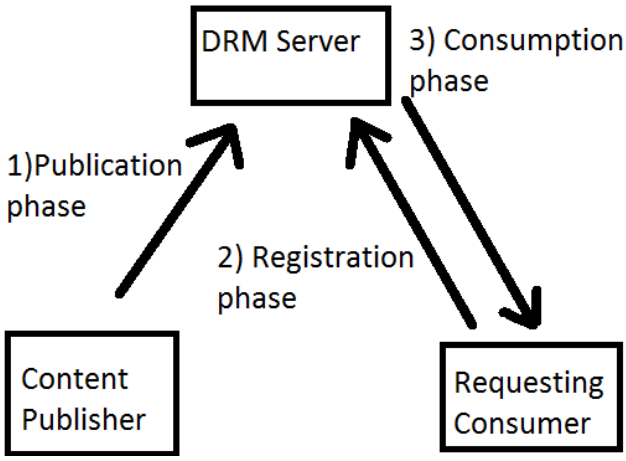

Figure 2. DRM entities and phases

The entities involved are:

- *End user, who legally obtains the software or digital content for consumption*
- *Publisher, who distributes the content and is responsible for managing user rights.*
- *DRM client, which authenticates to DRM server and requests access to specific content or digital object.*
- *DRM server, which holds the content decryption keys and rights information and distributes them to clients in the form of a license Notations.*

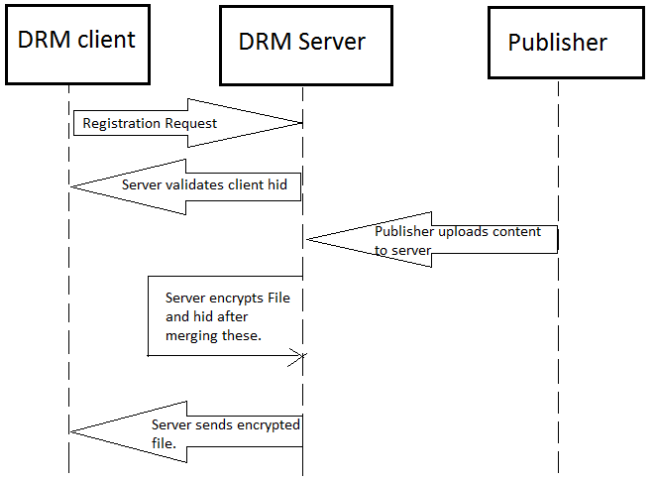

Figure 3. Sequence diagram of DRM activities

In Fig. 3, we have illustrated the various entities involved and the sequence of interactions between them. An offline DRM model uses asymmetric and symmetric cryptographic combination of keys and DRM licenses. A major difference between offline and online model is in the license format and its delivery mechanism. In offline model, the content decryption key in the license is wrapped with consumer's public key and this license can be stored locally.

Offline DRM licenses are also bound to specific users and machines. The proposed model consists of four phases, viz., Registration, Publication and Consumption. A detail description of each of the phases is as follows.

a) Registration phase

An existing DRM (online) user needs to accomplish certificate registration to work in offline mode. In this phase, the user acquires machine certificate and a user certificate to establish a trust between DRM server and the user's workstation.

Step 1. User U authenticates to server using Uid and Upwd and requests machine certificate (by a signed applet or a trusted application such as the DRM client itself) from DRM server.

Step 2.Server validates the hid of the client by sending a sample program and verifying its output.

b) Publication phase

In this phase, the publisher uploads the content to the DRM server and encrypts it with a random symmetric key known as content key. The content key is further protected and stored in DRM server's database. When user requests access to a specific content, the content key and rights information are transferred to the user in the form of a tamperproof DRM license.

Publisher P uploads a digital content C, and encrypts it using the encryption function E() and a random symmetric key k to generate EC.

c) Consumption phase

In this phase, the DRM client enforces the user rights for consumption of the content. A single DRM client can render multiple file formats based on file type.

Step 1. To work in offline mode, user needs to request the license file UL from the DRM server. UL can also be sent bythe publisher.

Step 2. Server generates the required license file UL to access encrypted content EC. The license file UL contains contentkey encrypted with user's public key Upub.

Step 3. DRM client stores the encrypted file EC and corresponding license UL in its local database.

Step 4. When user tries to access a file from her local DRM client, the Mid from the machine certificate Mcert is matchedwith current machine ID. The integrity of the corresponding license file UL is also verified.

Step 5. The DRM client decrypts the encrypted content EC, and renders the content after applying the rights.

IV. **ANALYSIS**

In this section, we analyze our architecture of the DRM requirements like flexibility, efficiency, interoperability and security based on our observations.

A. Flexibility

Content format: It is possible to protect multiple formats (theoretically any type of content) with the DRM server. Such formats include Text, PDF, image, audio, etc. Assignment of rights is flexible. Permissions and constraints can be enforced at granular level, for selected user, on selected content. Revocation of rights is also simple. Authorized consumers can consume the content on any machine using the DRM client. The framework is portable to multiple platforms.

B. Robustness

The architecture is robust to support load balancing and clustering. In case of any failure at primary
server, a backup server can take over. The user load can also be distributed on to multiple DRM servers.

C. Complexity

The model is fairly simple and can be easily implemented. In offline model, key and certificate management is the overhead for the advantage of offline consumption of content.

D. Availability

The protected content is available to all the legitimate users from the server. If a license file is lost, a backup copy can be obtained. If there are two publishers say A and B, publisher A cannot grant any rights on the content published by B. The same applies to generation of DRM licenses. In Multi-user environment, users need to authenticate to DRM server for fetching licenses. In offline model, the licenses are bound to specific users on specific machines. Copying license files or sealed objects from one machine to other cannot break the DRM protection.

E. Security

All the communications from DRM client to server happen over SSL/https. Every time the content needs to be consumed, user authentication to server over https is needed. Thus eavesdropping and man-in-the middle type of attacks are not possible. In the event of a tampered license, the DRM client will not accept the license file. The license file carries the DRM server's digital signature to facilitate the detection of tampering. All the cryptographic keys are adequately protected using access controls or using key wrapping. For additional protection to cryptographic keys, white-box cryptographic techniques can be used so that the attackers do not have explicit access to the keys during decryption. Deployment of an IDS/IPS and monitoring the DRM server logs can help for prevention of such attacks. D. Interoperability Rights Expression.

V. CONCLUSIONS

In this paper, we presented a DRM framework to protect digital multimedia content and which can be extended to software applications. The framework can also be extended to support multiple content formats. We found it flexible and efficient to manage user rights, reasonably secure in design and interoperable with the open or third-party clients. Future scope of this work includes the investigation on extending this framework to a cloud environment.

REFERENCES

[1] Privacy-Preserving Digital Rights Management based 978-1-4799-3223-8/14/$31.00 ©2014 IEEE
[2] A Robust Approach to Prevent Software Piracy 978-1-4673-0455-9/12/$31.00 ©2012 IEEE
[3] A P2P Cultural Multimedia Network – Maximizing Cultural Dissemination and supporting Copyright Protection and Management, ©2014 IEEE
[4] ADRMS: Active Directory Digital Rights Management Overview. Available: http://msdn.microsoft.com/enus/library/cc530389(v=vs.85).aspx.
[5] Adobe Content Server, In http://www.adobe.com/products/
[6] WebGuard: A System for Web Content Protection. Available: http://domino.watson.ibm.com/comm/research_projects.nsf/pages/labasec.WebGuard.htm
[7] DRM-JVM: Digital-Rights-Management-Enabled Java Virtual Machine. Available: http://domino.watson.ibm.com/comm/research_projects.nsf/pages/labasec.DRM-JVM.html
[8] New Method of Hardware Encryption against Piracy 978-0-7695-3600-2/09 $25.00 © 2009 IEEE
[9] Enterprise Digital Rights Management System based on Smart Card. 978-1-61284-842-6/11/$26.00 ©2011 IEEE
[10] A DRM Framework Towards Preventing Digital Piracy. 978-1-4577-2155-7/11/$26.00 c_2011 IEEE
[11] Unifying Broadcast Encryption and Traitor Tracing for Content Protection. 1063-9527/09 $26.00 © 2009 IEEE
[12] Annoyance Maximization for Digital Cinema Anti-piracy Applications 978-0-7695-3959-1/09 $26.00 © 2009 IEEE

