Ain Shams Engineering Journal (2014) 5, 703–720

Ain Shams University

Ain Shams Engineering Journal

www.elsevier.com/locate/asej
www.sciencedirect.com

ELECTRICAL ENGINEERING

A comprehensive Software Copy Protection and Digital Rights Management platform

Ayman Mohammad Bahaa-Eldin *, Mohamed A.A. Sobh

Computer and Systems Engineering Department, Ain Shams University, Cairo, Egypt

Received 18 January 2014; revised 2 March 2014; accepted 17 March 2014
Available online 14 April 2014

KEYWORDS

Software Copy Protection;
Hacking and cracking;
Digital Rights Management;
Software license and grants;
XrML

Abstract This article proposes a Powerful and Flexible System for Software Copy Protection (SCP) and Digital Rights Management (DRM) based on Public Key Infrastructure (PKI) standards. Software protection is achieved through a multi-phase methodology with both static and dynamic processing of the executable file. The system defeats most of the attacks and cracking techniques and makes sure that the protected software is never in a flat form, with a suitable portion of it always being encrypted during execution. A novel performance-tuning algorithm is proposed to lower the overhead of the protection process to its minimum depending on the software dynamic execution behavior. All system calls to access resources and objects such as files, and input/output devices are intercepted and encapsulated with secure rights management code to enforce the required license model. The system can be integrated with hardware authentication techniques (like dongles), and to Internet based activation and DRM servers over the cloud. The system is flexible to apply any model of licensing including state-based license such as expiration dates and number of trials. The usage of a standard markup language (XrML) to describe the license makes it easier to apply new licensing operations like re-sale and content rental.

1. Introduction

Software industry is highly affected by piracy. Illegal copying, tampering, and other non-legitimate distribution of digital products are increasingly widespread. This reduces the profits of software industry, hinders the software ecosystem, and causes huge losses [1]. Software Copy Protection (SCP) and Digital Rights Management (DRM) are both needed to protect the software and the digital contents distributed along with the software or in a stand-alone fashion.

The term Digital Rights management (DRM) broadly refers to a set of policies, techniques and tools that guide the proper use of digital content [2]. Fig. 1 shows a high-level view of the flow of protected digital content from its producer to the consumer. The entities defined in that figure are [2]:

- The content creator is mainly concerned with the core data/ information that goes into the content. This could be viewed as raw content, which needs to be processed further with respect to adhering to certain formats, the suitable

* Corresponding author. Tel.: +20 (11) 11 555750.
E-mail address: ayman.bahaa@eng.asu.edu.eg (A.M. Bahaa-Eldin).
Peer review under responsibility of Ain Shams University.

ELSEVIER Production and hosting by Elsevier

2090-4479 © 2014 Production and hosting by Elsevier B.V. on behalf of Ain Shams University.
http://dx.doi.org/10.1016/j.asej.2014.03.001

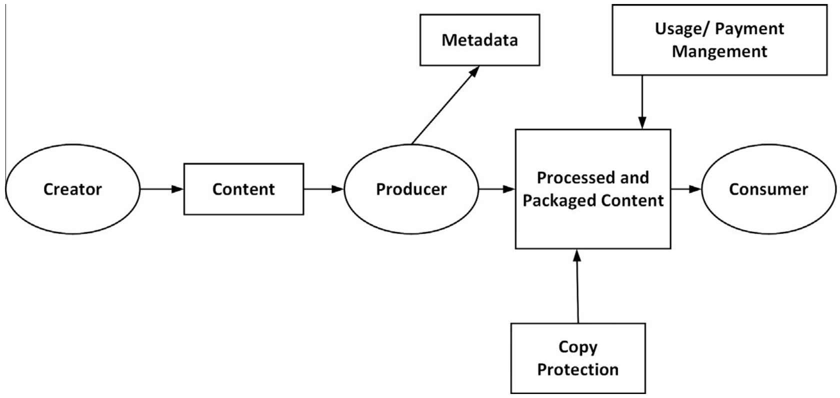

Figure 1 Overview of protected contents/software flow of control.

integration of different kinds of media, quality enhancement, additions of possible special effects, and derivation and addition of metadata (information about the data).
- The producer of the content performs the necessary processing and generates the packaged content. The packaged content is in a form that is suitable for consumption and for the tracking and management of content usage.
- The consumer is the ultimate user of the content.

The SCP is used to prevent reverse engineering and cracking techniques and to enforce the provided license rights of the user. DRM is employed to indicate what the end user is illegible to do with these contents. New DRMs even deals with rights like re-selling of the contents and re-use them within a new-produced contents [3]. If the protected contents are provided alone, they require special software players or viewers that are able to deal with DRM information. Usually those players are provided by the operating system providers like Microsoft and Apple. If the protected contents are provided within protected software, the embedded protection layer inside the executable files is responsible on processing the DRM information and controls the protected software behavior.

The executable file protection may involve some development (Software Development Kit (SDK) Protection Mode). Alternatively, it can be performed automatically (Shell Protection Mode). SDK protection requires integration of protection libraries inside the program code. The final program should be designed to be dependent on the protection libraries. Shell protection is applied to the software executable (object) file after compilation. It does not require the developer interaction. It intercepts the system calls, encrypts the code section and injects the protection layer. Sometimes it is required to apply both techniques for better or customized security.

The digital contents protection requires data encryption using stream or block oriented techniques. The encryption keys are stored in a secure hardware or software store. The protection layer is responsible for the decryption after successful authentication of the originality of the software and/or user rights.

The protection license may vary from trivial to sophisticated license. Trivial license is used to provide run/no-run situation after making the authentication. Sophisticated license may divide the software to modules and control each module separately. It also allows the vendors to define an expiration condition, control the authenticate periodicity, and use multiple authentication techniques. Some protection solutions use custom data structure to describe the license, other uses standard, readable language like XML or XrML.

Software authentication is performed to determine whether the client is the license owner or not. Authentication process involves hardware checking and security key extraction. Some authentication techniques extract the Machine-ID to identify the client; other techniques uses special type of CD/DVD disks, other techniques uses special security tokens like Dongles, Smart Tokens and Smart Cards. The security key extraction may use symmetric or asymmetric (PKI) techniques to unwrap the keys or simply read the keys from hidden or secrete hardware store.

2. Previous work on software protection and licensing

2.1. Overview

Software piracy is considered a major problem threatening the Software industry [4]. In 2007, Business Software Alliance [5] published a result stating that the weighted average of software piracy rate (by country) is around 60% of the total software industry [6]. The rise of software piracy led to the extensive work in the research and development of different Software Copy Protection techniques. The Software Copy Protection is established by different layers. The layers are Code Protection layer and Licensing and DRM layer.

Code protection helps in protecting Intellectual property of software vendors by providing methods against reverse engineering and cracking. Code obfuscation for interpreted languages like Java and C# is one of the popular methods for establishing the protection. Code obfuscation uses Random Number Generator [7] to change the original code to unreadable one. Later on, another technique [8] proposes stronger obfuscation using cryptography. For the compiled languages, which produce standard executable files, the protection is applied by dynamic injection and code redirection, hashing functions [9] or by remote code distribution [10]. Code protection requires an enforcement security layer to stop debuggers and to prevent the reverse engineering or memory spying [11,12]. Code protection strength up the executable files against cracking, however it is not enough to prevent the software distribution. SCP should be combined with certain hardware to avoid environment duplication (ex: CD, Dongles, Machine) [13–16].