A Digital Rights Management System Based on Cloud

Franco Frattolillo
Department of Engineering, University of Sannio, Corso Garibaldi 107, Benevento, Italy
Corresponding author, e-mail: frattolillo@unisannio.it

Abstract

In the current Internet, digital entertainment contents, such as video or audio files, are easily accessible due to the new multimedia technologies and to broadband network connections. This cause is considerable economic loss to global media players since digital contents, once legitimately obtained, can be illegitimately shared through file sharing services on the Internet. Digital Rights Management (DRM) systems have been proposed to support the protection of copyrighted digital contents. Even though such systems have been widely adopted and promoted by global media players, they are based on proprietary mechanisms that usually work only in closed, monolithic environments. In this regard, systems based on watermarking technologies appear more suited to protect digital copyrighted content. This paper describes the implementation scheme of a DRM system able to ensure the copyright protection of digital content according to an innovative buyer-friendly watermarking protocol. The DRM system has been implemented by exploiting a cloud environment in order to improve the overall performance of the system. In particular, cloud behaves as a service infrastructural provider, since the content provider involved in the watermarking protocol uses cloud to speed up the watermark embedding process and to save storage and bandwidth costs needed to store and to deliver protected contents.

Keywords*: digital copyright protection, digital rights management, cloud environment*

1. Introduction

The rapid growth of the Internet has greatly increased the capabilities of reproducing and distributing digital contents at a very low cost and without loss of quality. Digital entertainment contents, such as video or audio files, have become easily accessible due to the reduction in cost of high-performance multimedia digital devices and to broadband network connections. However, digital contents, once obtained, can be illegitimately shared on the Internet by exploiting, for example, peer-to-peer file sharing services. As a consequence, copyright holders suffer considerable economic loss, and serious issues concerning digital copyright protection have to be addressed.

The problem of digital copyright protection involves two main entities with conflicting interests: web content vendors, usually called “content providers”, and consumers. The former aim at a wide dissemination of their copyright-protected contents which does not compromise the originality and creativity of their intellectual properties, receiving reasonable and proportionate remuneration for the use of such contents, selling as many products as possible, and reducing the costs of production and distribution. The latter think that anything that is on the Internet is of public domain and may be taken without permission from the creator/owner. They want to purchase digital content at the lowest possible price. Furthermore, they invoke the “fair use” on the purchased digital content, including the private copy, and require compliance with privacy. In particular, the “fair use” is invoked in order to prevent copyright owners from having the exclusive control over their creations than the copyright law intends, whereas the privacy problem is invoked in order to preserve the ownership and distribution of confidential data. Therefore, the scenario described above requires the adoption of specific legal measures, contractual mechanisms, and digital techniques in order both to actually protect copyrighted digital contents and to regulate “fair use” and minimize privacy conflicts in managing copyright protection.

Among the possible measures to support digital copyright protection, the adoption of web software platforms known as Digital Rights Management (DRM) systems is considered one of the possible approaches that can match the rising and conflicting expectations of authors, vendors, and web users [1, 2]. DRM systems exploit security technologies to solve the main

Received January 10, 2017; Revised March 28, 2017; Accepted April 13, 2017

problem of preventing who are not provided with valid license from illegally copying or gaining access to copyrighted digital content. They tend to protect the interests of content owners by maintaining a persistent control of the ownership over digital content distributed on the Internet [3].

DRM systems have become a primary investment of global media players involved in digital content production and distribution, such as Sony, Apple, or Microsoft, which have implemented proprietary and sophisticated DRM solutions to manage their protected contents. However, the experiences conducted by such players are mainly based on DRM systems that work only in closed, monolithic environments. They operate by packaging digital contents in proprietary data containers made accessible only by using proprietary trusted hardware/software.

On the contrary, web systems that employ watermarking based technologies [4] to protect copyrighted digital content distributed on the Internet do not need to exploit proprietary technologies, but they can be based on open solutions well-documented in the literature. Their core is represented by the “watermark insertion techniques” and by the “watermarking protocols” they adopt: the former define the way in which watermarking information or “fingerprints” are embedded into digital content [4], whereas the latter define the scheme of the interactions that have to take place among the entities involved in the processes of content protection and web-based distribution implemented by such systems [5].

This paper presents the implementation scheme of a DRM system able to ensure the copyright protection of digital content according to the watermarking protocol presented in [6]. The design of the DRM system follows an experimental approach that is based on employing a cloud environment as a service infrastructural provider. In fact, such an approach enables the content provider involved in the watermarking protocol to use cloud to speed up the watermark embedding process and to save storage and bandwidth costs needed to store and to deliver protected contents. This makes it possible to solve the efficiency and scalability problem which usually affects the DRM systems that apply an “on-the-fly” protection to the distributed contents.

The paper is organized as follows. Section 2 describes the watermarking protocol that defines the transaction scheme by which the proposed DRM system can protect digital content. Section 3 describes the implementation scheme of the proposed DRM system. Section 4 concludes the work.

2. Watermarking Protocol

The watermarking protocol adopted by the proposed DRM system is documented in [6] and is defined as a “buyer-friendly” watermarking protocol. It involves the following web entities:

1. The *buyer* (*B*), who uses a common web browser to purchase the digital content distributed by a web content provider and to pay for the digital license.
2. The *content provider* (*CP*), which holds the digital rights of the distributed contents and wants to protect them by employing a DRM system.
3. The *registration authority* (RA), which is a trusted third party (TTP) that generates “tokens” and information to be used to unambiguously identify the buyer, the content provider, the purchased content, and the purchase transaction. It supports the easy participation of buyers in the protocol by playing a limited role only in the initial, registration phase of the protocol.

In Figure 1, a buyer *B* wishing to purchase digital content from a content provider *CP* contacts the registration authority *RA* and communicates his/her personal and payment credentials. *RA* generates a “nonce” *N* and a one-time public and private keys pair (pk_B, sk_B) linked to the *B*’s identity, the seller, the content, and the purchase transaction. Then, it generates further security tokens. Finally, it encrypts *N* with pk_B and signs all the generated information and tokens, which are returned to *B*.

B forwards the encrypted nonce, the key pk_B, and some of the received information to *CP*, which can thus use them to generate the watermark to be inserted into the chosen digital content directly in the encrypted domain, since the protocol is based on a privacy-homomorphic cryptosystem [7]. The watermark is the concatenation of the encrypted nonce *N* and of an encrypted watermark picked by *CP*. Then, *CP* sends the encrypted and watermarked content to *B*, who can decrypt it, thus obtaining the final watermarked content.

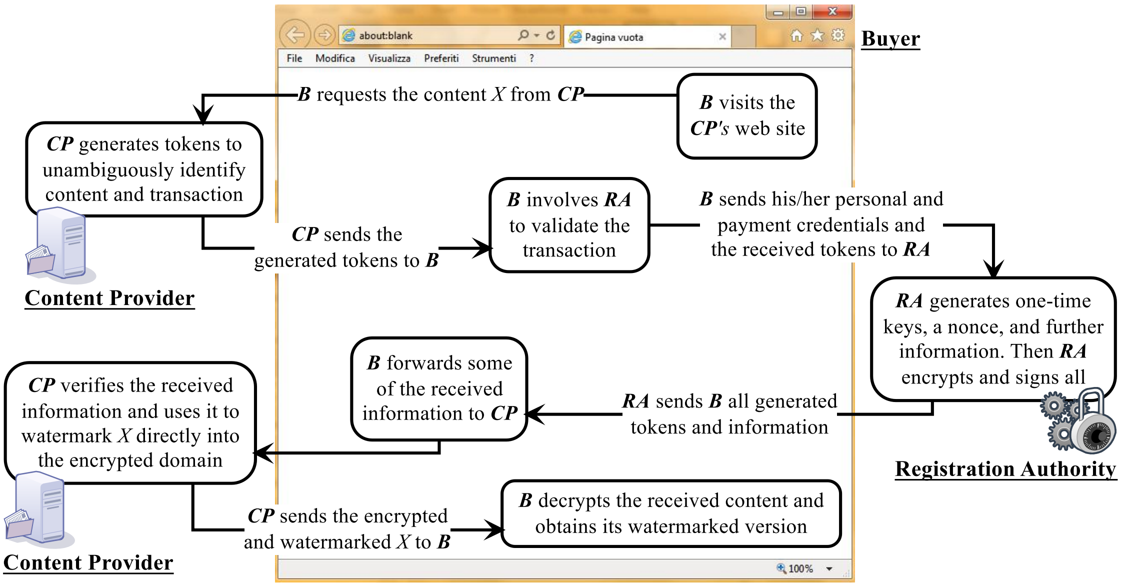
Buyer
B requests the content X from CP
B visits the CP's web site
CP generates tokens to unambiguously identify content and transaction
CP sends the generated tokens to B
B involves RA to validate the transaction
B sends his/her personal and payment credentials and the received tokens to RA
Content Provider
RA generates one-time keys, a nonce, and further information. Then RA encrypts and signs all
RA sends B all generated tokens and information
B forwards some of the received information to CP
CP verifies the received information and uses it to watermark X directly into the encrypted domain
Registration Authority
CP sends the encrypted and watermarked X to B
B decrypts the received content and obtains its watermarked version
Content Provider
about:blank
Pagina vuota
File Modifica Visualizza Preferiti Strumenti ?
100%

Figure 1. Watermarking protocol implemented by the proposed DRM system

3. Architecture of the DRM System

The watermark protection applied to content by means of the protocol described above unequivocally links the buyer, the content provider, and the purchase transaction. As a consequence, if the embedded watermark can survive manipulations and attacks, it can be exploited to encode the credentials of both the seller and the user who buys the content, thus enabling each buyer to obtain a unique and personalized copy of the purchased content. This can actually deter web users from illegally sharing content, since the applied protection makes it possible to establish if a watermarked content found in a suspicious location has been legitimately purchased by a buyer and then illegally shared on the Internet [5, 6]. However, this also means that watermark protections have to be applied “on-the-fly”, since each buyer has to receive a personalized copy of the purchased content. Therefore, both the computational burden due to watermark embedding and the required bandwidth due to content distribution grow linearly with the number of buyers. This makes it necessary to have a huge amount of resources, and poses a problem of scalability.

To solve such a problem, many solutions have been proposed in the literature. Some of them are based on relying the protection tasks on buyers. These solutions are mostly based on client-side watermark embedding techniques [8, 9]. However, recent advances in networking technologies and the increasing demand for computing resources have motivated many content providers to outsource their storage and computing needs by exploiting a new economics and computing model commonly referred to as “cloud computing” [10, 11].

Cloud computing enables content providers to move their copyrighted digital contents to cloud service providers (CSPs for short), thus relieving content providers of the costs of building and maintaining private storage and computing infrastructures. Such an outsourcing approach makes it possible to give a response to the scalability problem as well as provides several benefits, such as flexibility, availability, and reliability, at a relatively low cost: flexibility is ensured since a content provider can choose to pay a service provider as a function of its needs; availability depends on the capability of buyers to access copyrighted digital content from anywhere and at any time; reliability is guaranteed since content providers have not to worry about backups.

Based on the considerations reported above, the proposed DRM system has been implemented by exploiting a CSP as an active platform-level service. In particular, the CSP behaves as a platform able to implement the protection services provided by the content provider directly. Thus, an efficient, effective, and secure DRM system can be made available to content providers, in which a great computing power together with a very economical “pay-asyou-go" business model can be exploited according to the protection scheme defined by the watermarking protocol described in Section 2.

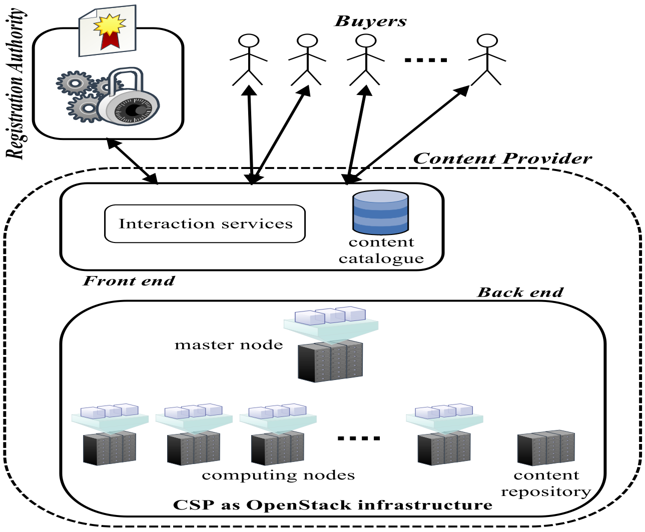

Figure 2. The architecture of the proposed DRM system

Figure 2 shows the architecture of the proposed DRM system. As depicted in the figure, the content provider that adheres to the proposed DRM system can be arranged according to a two levels structure made of a *front end* and a *back end*: the former is directly implemented by the content provider and includes the services needed to manage the interaction with buyers and the registration authority, whereas the latter is implemented by the CSP on behalf of the content provider and includes the processes and computing and storage resources needed to apply watermark protections to distributed contents. In particular, the CSP runs the OpenStack cloud operating system [12] and is exploited in the form of "Infrastructure as a Service" (IaaS). It executes a set of processes which interact through the functions belonging to the API of OpenStack according to what is stated by the adopted watermarking protocol. To achieve such a goal, the CSP adopts a computing model based on the presence of a *master node* and a set of *computing nodes*. In particular, the *master node* represents the interface of the back end towards the front end, and coordinates the computing action implemented by the computing nodes provided by the CSP.

3.1. Front End of the Content Provider

Figure 3 shows the activities and interactions implemented by the content provider according to the proposed DRM system. Focusing on the front end, the interaction among the buyer *B*, the content provider *CP*, and the Registration Authority *RA* is implemented by means of dynamically built HTML5 pages and JavaScripts. Such pages contain the tokens needed to unambiguously identify the content distributed by *CP* and the purchase transaction. JavaScripts are used to manage the communication and interaction that take place between *B* and *CP* in the negotiation phase of the watermarking protocol. They are also used to manage: (1) the interaction between *B* and *RA* during the process that takes charge of verifying the *B*'s identity together with his/her payment credentials; (2) the subsequent phase by which *RA* generates the security tokens that have to be returned to *B* and then forwarded to *CP*. Therefore, a buyer wanting to purchase a digital content provided by a *CP* has only to be provided with a common web browser by which he/she can display HTML5 pages and run JavaScripts.

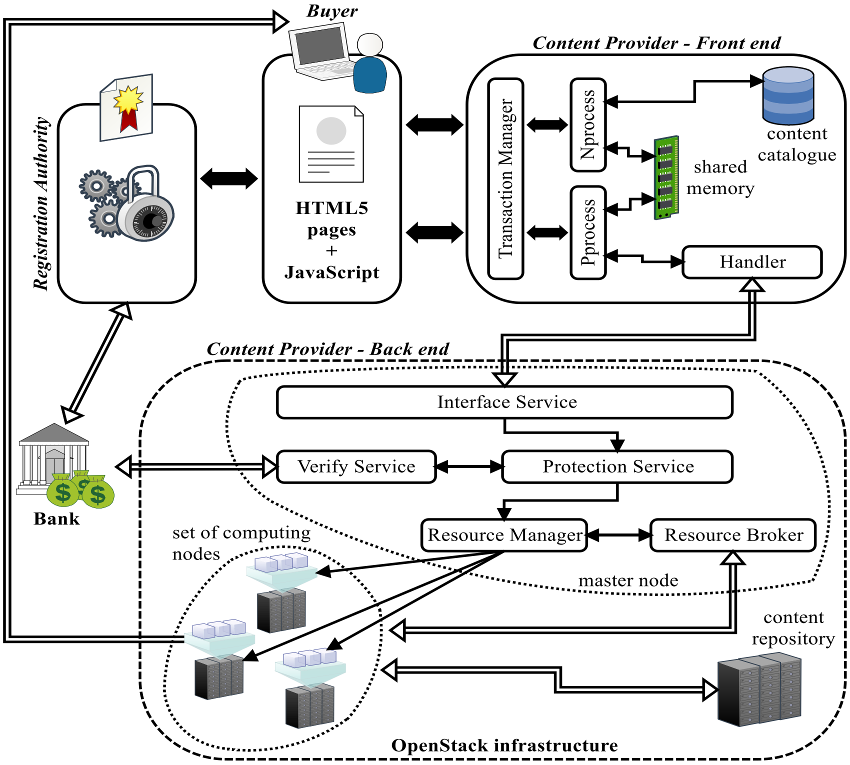

Figure 3. Activities and interactions within the proposed DRM system

The *Transaction Manager* is implemented by *CP* and interacts with *B* to generate and manage the HTML5 pages to be returned to him/her in the preliminary part of the purchase transaction. It publishes the content catalogue of *CP* and receives the purchase request from *B*. Then, it interacts with *Nprocess*, which generates the tokens that link *B* and *CP* to the chosen content and the current transaction. The tokens are saved by *Nprocess* in a specific, *shared memory*, and are then returned to *B* through the *Transaction Manager*.

B receives the tokens from *CP* and passes them on to *RA*. *RA* takes charge of generating one-time keys, a nonce, and further information, which are then returned to *B*.

B forwards the received information to *CP*, which manages it by means of the *Transaction Manager*. This module also involves *Pprocess*, which accesses the shared memory in order to retrieve the information stored by *Nprocess* in the previous step of the purchase transaction. If the retrieved data turn out to be correct, *Pprocess* uses the *Handler* to start the protection process, which is implemented by the back end of the DRM system that is the cloud platform. The *Handler* receives all the data needed to protect the digital content chosen by *B* and passed them on to the *Interface Service*, which represents the entry point to the back end.

3.2. Back End of the Content Provider

As reported in Section 3, the back end of the infrastructure implemented by *CP* includes a master node and a set of computing nodes. Figure 4 shows the implementation scheme of the master node. Such a node has the task of managing the back end of the DRM system. It implements the *Interface Service*, which takes charge of creating a job for the content to protect on the basis of the security information received by the front end services.

The job is then passed on to the *Protection Service*, which implements two main tasks: the former verifies the *B*'s credentials, that are personal and payment credentials, whereas the latter applies the protection to the content. In particular, the *Protection Service* exploits two services to perform these two tasks: the *Verify Service* and the *Resource Manager*.

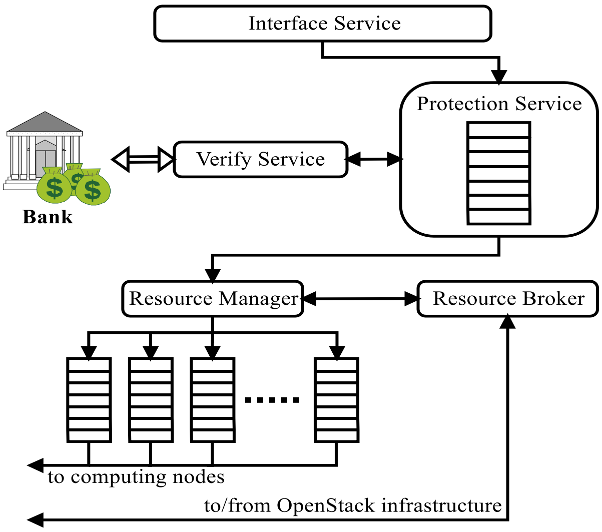

Figure 4. The scheme of the master node

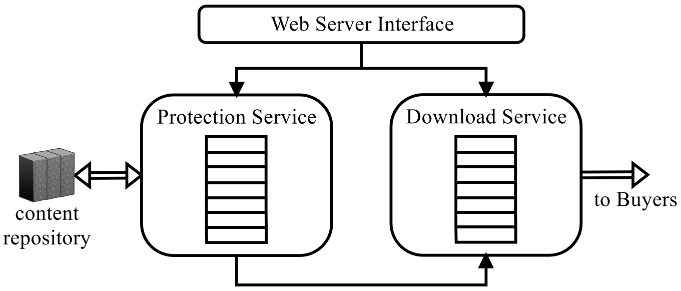

Figure 5. The scheme of a computing node

The *Verify Service* manages the payment phase, whereas the *Protection Service* has to complete the protection phase, if the payment phase has succeeded. In particular, the *Protection Service* sends the *Resource Manager* the job including the reference to the content to protect together with the information needed to apply the protection.

The *Resource Manager* manages the pool of computing nodes supplied by the OpenStack structure: when it receives a job from the *Protection Service*, it takes charge of choosing the less loaded computing node among those ones available. If a node is available, the *Resource Manager* pushes the job into the queue associated to the node. However, if all available nodes are overloaded, the *Resource Manager* contacts the *Resource Broker*, whose task consists in demanding new computing nodes from the OpenStack infrastructure. Once obtained, the new nodes are associated by the *Resource Broker* to the pool of available computing nodes, and can be exploited by the *Resource Manager*.

After this phase, each computing node, whose scheme is depicted in Figure 5, can retrieve a job from its queue managed by the *Resource Manager* and start the actual protection process. This task is accomplished by the *Web Server Interface* running on the node. This interface manages two processes: the *Protection Service* and the *Download Service*. The former embeds the watermark into the content retrieved from the content repository, and inserts the protected content into the queue managed by the *Download Service*. The latter notifies the availability of the protected content to the legitimate buyer *B*, who can thus download the purchased and protected content from *CP*.

4. Conclusion

Digital copyright protection is a relevant problem of the current Internet because modern multimedia and interconnection technologies have greatly increased the capabilities of illegally reproducing and distributing digital content without loss of quality.

In this paper, the implementation scheme of a DRM system is presented. The DRM system supports the copyright protection of digital content applied according to an innovative buyer-friendly watermarking protocol suited for the web context. The system adopts a new design approach that is based on a cloud environment which exploits OpenStack to implement a service infrastructural provider. In this regard, the content provider that adheres to the proposes DRM system can take advantage of the cloud infrastructure to speed up the watermark embedding process and to save storage, computing, and bandwidth costs needed to protect, store, and deliver protected contents. The result is a modular and very scalable DRM system, which can dynamically exploit the computing power made available by the cloud environment within a security context.

References

[1] Zhang Z, Pei Q, Ma J, Yang L. Security and trust in digital rights management: A survey. *Int. Journal of Network Security*. 2009; 9(3): 247-263.

[2] Frattolillo F, Landolfi F. *A Cluster Grids Based Platform for Digital Copyright Protection*. Proc. of the 12th IEEE Int. Symp. on Web Systems Evolution. Timisoara, Romania. 2010: 83-87.

[3] Ku W, Chi CH. *Survey on the technological aspects of digital rights management*. Proc. of the 7th Int. Information Security Conference. Palo Alto, CA, USA. 2004: 391-403.

[4] Cox I, Miller M, Bloom J, Fridrich J, Kalker T. Digital Watermarking and Steganography. Burlington, MA, USA. Morgan Kaufmann. 2007.

[5] Frattolillo F. Watermarking protocols: Problems, challenges and a possible solution. *The Computer Journal*. 2015; 58(4): 944-960.

[6] Frattolillo F. A buyer–friendly and mediated watermarking protocol for web context. *ACM Transactions on the Web*. 2016; 10(2): 9.

[7] Fontaine C, Galand F. A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security*. 2007; 2007.

[8] Katzenbeisser S, Lemma A, Celik MU, van der Veen M, Maas M. A buyer-seller watermarking protocol based on secure embedding. *IEEE Trans. Inf. Forensics Security*. 2008; 3(4): 783-786.

[9] Bianchi T, Piva A. TTP-free asymmetric fingerprinting based on client side embedding. *IEEE Trans. Inf. Forensics Security*. 2014; 9(10): 1557-1568.

[10] Rittinghouse JW, Ransome JF. Cloud Computing: Implementation, Management, and Security. Boca Raton, FL, USA. CRC Press. 2009.

[11] Abbadi IM. Cloud Management and Security. Hoboken, NJ, USA. Wiley. 2014.

[12] Fifield T, Fleming D, Gentle A, Hochstein L, Proulx J, Toews E, Topjian J. OpenStack Operations Guide: Set Up and Manage Your OpenStack Cloud. Sebastopol, CA, USA. O’Reilly. 2014.