Digital Rights Management: Technology, Standards and Applications

October 2, 2012

Contents

Chapter 1

Personal DRM (PDRM) - A Self-Protecting Content Approach

Authors: Akla-Esso Tchao, Giovanna Di Marzo Serugendo, Jean-Henry Morin

1.1 Introduction

Long ago, mainframe computers were run by experts behind closed doors [26]. Documents leaving the mainframe would likely be printed on paper, or maybe stored on a magnetic tape, and both were controlled physically. Nowadays, almost everyone daily carries one or more personal computing devices with similar or even more advanced performances than the mainframe computers. With the development of the Internet, anyone can connect and share content with audiences they would otherwise never have had access to. It becomes easier than ever to publish and share content, whether personal or not, over the Internet or among personal devices, with the drawback that propagation and copies of this content is hard to control.

This ease of sharing content raises a major issue: the unauthorized redistribution, either

intentionally or not of content subject to governed use or policies (e.g., video, music, computer software, personal information, etc.). According to the 2010 IFPI report [12], one in five people across Europe's top markets are engaged in frequent unauthorized music-sharing, using file sharing networks, downloading from hosting sites, stream ripping, instant message sharing and downloading from forums and blogs.

Besides sharing copyrighted content, individual users are also faced with issues related to privacy, e.g. they may not want their picture, published on social medias among friends, to become available for companies to provide targeted advertisements, or for close friends to share the picture with friends of their own. Friendship is not transitive. Similarly, organizations handling sensitive or personal data, such as Home Offices, or HR departments are legally bound to take appropriate measures to handle such information. Such measures include for example not to divulge it to unauthorized people, to keep it for no longer than necessary, etc.

Digital rights management (DRM) has been initially introduced to allow copyright holders (usually organizations) to better control their rights. To protect copyrighted contents, DRM systems use either a hardware-based or a software-based solution. Most DRM systems proposed are often designed for large organizations, with little to no provision for people to use them for their own personal content [14]. In hardware-based solutions, a proprietary device is required to enforce the content owners' rights over their content. From an individual producer's perspective, building or buying a hard to fake proprietary device becomes prohibitively expensive. On the other hand, a software-based solution does not require any special hardware. However, the security of such DRM systems is intimately tied to the DRM client with which the protected content can be rendered, thus reducing the consumer's freedom. For instance, a content protected with Apple FairPlay DRM cannot be consumed with Windows Media Rights Management (WMRM) DRM client and vice-versa. For most critics, current DRM systems hamper consumers' ability to take advantage of the full promise of digital media [23] and the Internet.

To enable individuals to protect their rights over their personal content, we propose in this chapter a new software-based DRM system called *Personal DRM (PDRM)*. The

purpose of PDRM is to provide an easy to use and to deploy DRM system for personal content protection, while achieving the same or better individual rights protection compared to traditional DRM approaches. The main idea behind PDRM is to enable the content to protect *itself* against unauthorized access at an affordable cost. A detector, generated using an Artificial Immune System (AIS) technique, is attached to the content. The detector assesses the current context use and denies access to the content, when it recognizes unusual situations.

We intend to cover different situations such as: unauthorized content access (e.g. sensitive data lost cannot be read by third parties who may find it), legitimate content access in unauthorized situations (e.g. a bank employee not allowed to access customer's files from corporate laptop at home, in a foreign country, or when meeting other clients), authorized consumer trying to undertake unauthorized actions on the content (i.e. a Facebook's friend can view a picture but not send it to her own friends).

Compared with existing DRM systems, PDRM ensures flexibility, transparency and easy deployment. Contrary to existing DRM systems, a PDRM protected content consumer does not need to initially download a PDRM client, since the client software is bundled with the content. PDRM does not strive to provide 100% protection as full protection is not achievable [1], but rather it provides a system that is difficult for an attacker to break.

This chapter is organized as follows: Section 1.2 presents the context of DRM systems, sample scenarios in relation with personal content protection and the problem statement. Section 1.3 reviews previous and current efforts on personal content protection. Section 1.4 introduces the Artificial Immune System and focuses on the AIS mechanism used for designing the PDRM system. Section 1.5 presents the PDRM system itself. Section 1.6 discusses the proposed PDRM system comparing it with existing DRM systems in terms of adaptability, flexibility, security, accessibility and privacy using the scenarios described in Section 1.2. Section 1.7 provides conclusion and future work.

1.2 Context and problem statement

A typical DRM system is composed of a *content*, a *producer (also called owner)*, a *distributor* and a *consumer*. A content can be an image, a video, an audio, a computer software or simply a text. A producer creates the content or owns the rights on it. A content distributor is responsible for distributing the protected content to end users (e.g., Apple iTunes). The consumer then reads, plays or views the protected content using a DRM client. A content producer specifies the usage rules and rights associated to a content using a *Rights Expression Language (REL)* such as XrML [3] or ORDL [18]. Several rules can be attached to the same content. For example: a content owner specifies that a content can be consumed three times and from specific devices. He can also define an additional rule valid on any device for the same content specifying the time interval when that content can be consumed. The access information is in a *license*. It contains all the applicable rights, terms and conditions on the usage of the content. A license is bound to a protected content and may also be associated to one or more specific devices or legitimate consumers. Fig. 1.1 gives an overview of a typical DRM system. In this figure, a content producer packages the content (1) using DRM's content protection techniques such as encryption, watermarking, fingerprinting, etc. or a combination thereof. Then, the packaged content is posted on a content server owned by the distributor (2), and the associated license (access and usage rules provided by the content owner) on a key server (3). To get access to a content, a consumer requests a copy of the content to the distributor (4). Once in possession of the content, the consumer use a DRM client to view, play or read the content (i.e to actually use the content). In fact, the DRM client connects to the license service to request and get a license associated to the content (5), (6). Then it inspects the rules in the license to determine operations and actions allowed on the content, before actually allowing or denying action.

Some DRM systems require more components and services than the ones described above. For example, Windows Media DRM, Lightweight DRM, EMMS, Helix DRM, Aegis DRM use up to seven services (content service, license service, access service, tracking service, payment service, certification and authority service) [15].

Deploying such an infrastructure for personal content protection only is inappropriate as

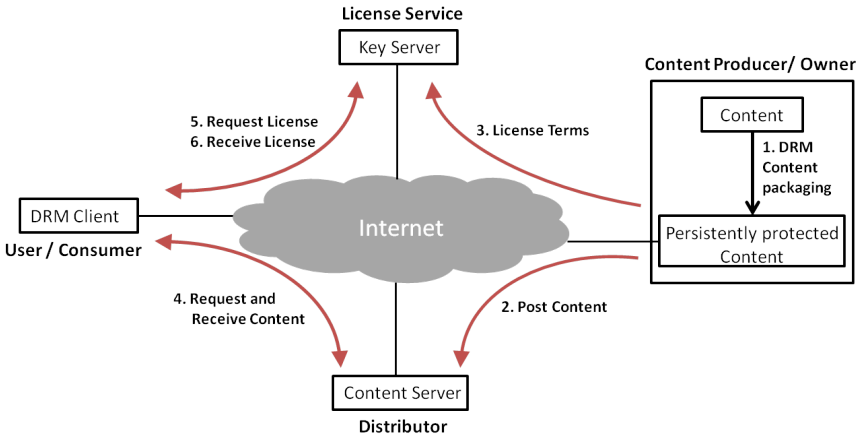

Figure 1.1: DRM system overview

it requires ordinary people to deploy a complex infrastructure for personal use.

1.2.1 Scenarios

In the context of personal content protection, we consider the following scenarios:

- Scenario 1: A person wants to share a private picture with family members and close friends, but doesn't want the information to be propagated to friends of friends or to others on the Internet.
- Scenario 2: A Blogger posts content relating to court cases on his blog and wants the content to "self-destroy" after a certain time according to policies such as the Right to Oblivion. Such policies specify the period beyond which certain categories of information should no longer be kept or used. They protect individuals from unreasonably long retention of data that could be harmful for their social or economic well being.
- Scenario 3: A software developer comes up with a fun game to share only with some colleagues, and may request a form of compensation any time people outside the circle of that team play the game.
- Scenario 4: A HR employee accesses files, containing employees records, from her desktop computer located within the corporate perimeters. The same employee is denied

access to these files from her company laptop at home.

Personal content protection for these scenarios are difficult to achieve with existing DRM systems. With current DRM systems, the content owner needs: to package the content using DRM techniques for content protection, make it available for download, provide a DRM client for consumers to access the content, generate a license, put in place a mechanism for the authentication and transmission of the license. He can also ask a third party to implement the management and the issuing of the license and the distribution of the protected content. All these procedures are much too complex and expensive to deploy for personal content protection.

In addition, many customers are reluctant to use DRM protected content since it reduces their flexibility over their content [23]. In current DRM systems, there is a lack of a fair balance between producers/distributors copyrights protection and consumers' needs because of the asymmetry of their respective interests [27]. Distributors use DRM systems to preserve their revenues, while customers are interested in flexibility and ease of use of the content they purchase or they create.

To honor individual rights over personal information, there is a need for a new DRM system that is easy to deploy allowing individuals to protect their own personal content and offering end users a fair and flexible use of the protected content.

Throughout this chapter, we will demonstrate using the scenarios described above, how the Personal DRM system, defined in this chapter, addresses these issues, while ensuring a persistent protection over personal content.

1.3 Related work

First we review three existing works related to self-protecting content: self-protecting document (SPD) [21, 20], self-protecting Container technology (DigiBox) [22] and post-release information privacy protection and violation framework [27].

The self-protecting document (SPD) [21, 20] prevents unauthorized and uncontrolled use and redistribution of a document, thereby protecting the rights of the content owners. It comprises an encrypted document as well as a secure set of permissions and the software necessary to process the document. SPD uses a secure polarization engine packaged with the content to act upon the document before it is stored or decrypted, so the document is never stored in the clear on the user's system. The generation of the polarization key and the rendering application happen behind a protecting shell. The polarization key is a combination of data elements taken from the user's system, such as elapsed time since the last keystroke, the processor's speed and serial number. Compared to PDRM, SPD requires a secure shell to operate and does not address the issue of flexibility of use of the SPD document. Indeed, since the polarization key is highly related to the user's device, the user can not consume the SPD document on any of his other devices.

The self-protecting Container technology (DigiBox) [22] provides a secure container to package information so that the information cannot be used except as provided by the rules and controls associated with the content. It allows rights management components to be integrated with content in highly flexible and configurable control structures. It uses encryption, digital signature and digital certificates to ensure data confidentiality and integrity. Contrarily to PDRM, DigiBox requires a prior deployment of tamper-resistant hardware. In fact, a secure environment, such as a secure processing unit (SPU) that contains a CPU, memory, program storage, and key storage in a single tamper-resistant hardware package is required for the container processing.

The post-release information privacy protection and violation detection framework [27] gives information owners more control over their released private information and provides effective ways to detect any violations of information privacy. It comprises a Self-destroying File (SDF) technique and Mutation Engine System (MES) technique. The SDF technique allows a subject's private information to be read only once. Then the file automatically executes an embedded program to destroy itself. The MES technique on the other hand allows a private file to be read more than once. MES ensures that the private file is encrypted immediately before the file is saved. Compared to PDRM, the proposed framework requires the consumer's computer system be equipped with a privacy-enhanced operation system

(PEOS), a stub program, a modified network interface card (NIC) and its reprogrammed driver file.

In addition to these approaches, other solutions were proposed to improve user flexibility in consuming copyright protected content.

Lightweight Digital Rights Management (LWDRM) [17] enable users to freely consume a legally acquire DRM content on any of their devices. It uses two files formats: local media file (LMF) format and signed media file (SMF) format. LMF format binds a media file by hardware-driven keys to the very end-user device on which the file was downloaded. To transfer the content to another device the user must transform the file into a SMF format by signing the content. Compared to PDRM, LWDRM introduces user signatures into the media files which may be used to trace back the user behavior. To solve this problem a separation-of-duty approach to privacy via a neutral certification authority protection for LWDRM was proposed [9]. However, in terms of flexibility of deployment for personal use, PDRM system in contrary to LWDRM doesn't require any third party service like certificate authority nor a public key infrastructure. Other works to leverage users flexibility to consume legally acquired DRM contents in a home were proposed and were based around the concept of Authorized Domain (AD) [10, 19] (i.e. set of devices that are owned by a single household). However, as for the LWDRM system, the proposed architectures are not appropriate in terms of facility of deployment, for personal content protection.

Another approach to ensure user flexibility is Exception Management in DRM environments [16]. This approach addresses the issue of usability, or lack thereof, of DRM systems. The model allows users to claim temporary usage rights (short lived licenses) based on credentials they provide as evidence for logging and monitoring. Although Exception management in DRM environments offers more flexibility in terms of usability, it suffers the same limitations of traditional DRM systems.

Finally, Studet et al. [25] propose a Mobile User Location-specific Encryption (MULE) to encrypt user-specified sensitive files on their laptop. MULE uses a location-specific information from a trusted location to automatically derive a decryption key and allow access to the sensitive files. MULE requires that laptops are equipped with trusted platform mod-

ules (TPMs), and a pre-installed Trusted Location Device (TLD). Although MULE is not directly related to digital rights management, the fact that with MULE no user effort is required in order to ensure the protection of sensitive files in trusted location make it a suitable candidate to be considered for personal content protection. The main drawback of MULE compared to PDRM resides in its hardware requirements.

1.4 The Artificial Immune System

The human immune system protects our body against infections caused by *pathogens* like viruses, bacteria and parasites. The immune system actually recognizes *antigens*, a protein attached to the surface of a pathogen, that triggers an antibody response. The immune system is composed of diverse sets of cells (*B-cells*, maturing in the bone marrow, trigger *antibodies* that bind and mark antigens, and *T-cells*, maturing in the thymus, that actively destroy pathogens), molecules and organs that work together to protect the organism. The immune system adapts to molecular patterns previously seen, learns and constitutes a memory of known patterns. The powerful information processing capabilities of the immune system, such as feature extraction, pattern recognition, learning, memory, and its distributed nature provides a rich metaphor that can be used in computer science to solve problems.

The Artificial Immune System (AIS) is an adaptive system inspired by theoretical immunology and observed immune functions and models, aiming at solving problems [4]. Some metaphors extracted from the human immune system applied to AIS include:

- **The immune network theory** - The Jerne [13] immune network theory views the immune system as a dynamic system, where cells and molecules of the immune system communicate and recognize each other. Communication takes the form of stimulation and inhibition between antibodies - some antibodies playing the role of antigens for other antibodies - thus regulating the level of antibodies. Patterns of interactions among cells govern the behavior of the immune system. The arrival of antigens cause perturbations among these interactions and trigger appropriate responses from the immune system.

- **The negative selection mechanism** - This theory is used to explain the ability of the immune system to differentiate between the cells of the organism known as *self* cells, and the foreign elements that can cause disease known as *non-self*. It is based on the negative selection of *T-cells* within the thymus. The next section will describe this mechanism in more details.
- **The clonal selection principle** - This principle explains how the immune system learns to cope with antigens. According to this theory, cells that successfully recognize antigens proliferate at the expenses of other cells that do not play any specific role and who are subsequently eliminated.

Among these metaphors, the negative selection mechanism is commonly used in research for applications such as virus detection [8], network intrusion detection [11] or hardware fault-tolerant systems [2]. In a similar way, we will use the negative selection algorithm mechanism for the PDRM.

1.4.1 The negative selection mechanism

The negative selection mechanism is the ability of the immune system to differentiate between the cells of the organism known as *self* cells, and the foreign elements that can cause disease known as *non-self*. Forrest et al. [8] propose a negative selection algorithm for AIS in the framework of network intrusion detection, that is inspired by the mechanism used by the immune system to train the *T-cells* to recognize *non-self* from *self* cells. The algorithm has two phases: a *censoring phase*, and a *monitoring phase*. In addition there is also a *co-stimulation* mechanism.

1. **Censoring phase:** In this phase, a set S of *self* patterns is to be protected. These *self* patterns are usually encoded in the form of binary strings. The set of *self* patterns is built over time through observation, without prior knowledge. For example, in a network intrusion detection application [11], the set of *self* patterns encodes all normally observed and acceptable connections, both within the local area network and those connecting the local area network from the outside world.

Randomly generated binary strings are initially placed in a set R_0. The strings in the set R_0 are then tested against the ones in the *self* set S for *matching*. If a string from R_0 matches at least one string of S, above a certain *threshold*, then this string is recognized as *self* pattern and has to be eliminated (negative selection); otherwise the string is introduced into the detector set R. For example, two binary strings match if they have more than a given number of bits in common.

In summary, a detector set is constituted of binary strings that *do not match* any string in the set of *self* strings according to the threshold based rule specified. Binary strings in the detector set R that do not match any string in the *self* set during the censoring phase are considered as mature. A mature detector has a finite lifespan, after which it is replaced by a new (immature) detector [7].

Fig. 1.2 summarizes the censoring phase of the negative selection algorithm presented in [8].

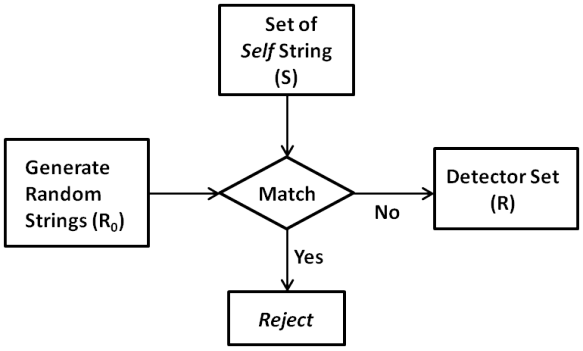

Figure 1.2: Generation of a valid detector set (Censoring) - [Taken from [8]]

After the set of detectors is generated, the algorithm enters a monitoring phase.

2. **Monitoring phase:** In this phase, the detector set is put to work. Binary strings of a set S^* are matched against the elements of the detector set R. In the case of network intrusion detection, S^* represents current network activities. If recognition occurs between a string in the set S^* and a string from the detector set, then a *non-self* pattern (string) is detected. Forrest et al. [11] completed the negative selection mechanism with a memory phase. Indeed, similarly to the human immune system that learns to detect new pathogens, and retains the ability to recognize previously seen pathogens through immune memory, the detector set, also learns to detect new *non-*

self patterns and keeps in memory *non-self* patterns already detected. For example, in the network intrusion detection, the detector learns to recognize new intrusions and remembers the signatures of previous attacks. Fig. 1.3 summarizes the monitoring phase of the negative selection algorithm described in [8].

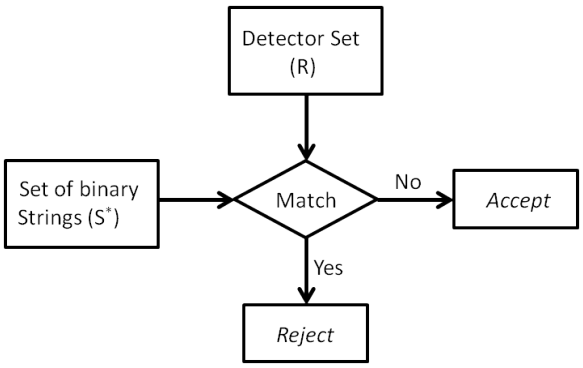

Figure 1.3: Monitoring protected strings for changes - [Adapted from [8]]

3. **Co-stimulation mechanism:** Forrest et al. [11] implemented the detector ability to learn, using a mechanism similar to the *co-stimulation* of B-cells by T-cells. They proposed that the co-stimulation signal is provided by a human observer. In fact, if a mature detector doesn't receive any co-stimulation signal within a certain period of time, it will die. However, detectors that have detected an anomaly and received confirmation from the human operator, enter a competition and the best-matching detector becomes a *memory* detector. This allows the system to adapt to incomplete or evolving definitions of *self*.

The Personal DRM system proposed in this chapter exploits the negative selection mechanism together with the co-stimulation adaptation feature of the immune system to ensure the self-protection of a personal content.

1.5 Personal DRM (PDRM) system

Fig. 1.4, gives an overview of a PDRM system. A PDRM user requests and gets (2) a PDRM protected content (1) from the producer. There is no need to download a PDRM client or

to contact a third party service to get a license like in current DRM systems but rather, the actual content is packaged into an active autonomous software agent, carrying its own access and usage rules.

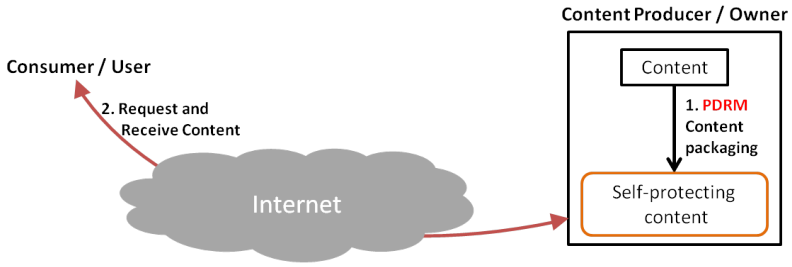

Figure 1.4: PDRM system overview

Scenarios. In Scenario 1, described in Section 1.2.1, a person who wants to share her private picture with family members and close friends packages the picture using PDRM. The packaging incorporates, in the form of a software component, the usage rules associated to the picture (i.e. denying access to non family members or non close friends). There is no additional task required by the consumers to view the picture. If a family member forwards the picture to a friend, the latter will not be able to view the picture. Indeed the self-protecting content will detect that the friend is not part of the set of people authorized to access the content. In Scenario 2, the packaging integrates the situation of usages and accesses breaking the policy under which the content is subject to. Scenario 3 is similar to Scenario 1, the content is enhanced with the specific access rules for people outside the circle of colleagues. Finally, in Scenario 4, the packaging incorporates the notion of location and device for specifying legitimate accesses.

1.5.1 PDRM content packaging

A PDRM protected content consists of a *detector set* R, the *scrambled content*, a *protected scrambling key* and a *PDRM player*.

We separate authorized *accesses* to the content from authorized *actions* (read *only*, execute *only*, forward, etc.) over the content. Authorized accesses are monitored by the detector

set, while authorized actions are monitored by the PDRM player included in the PDRM package.

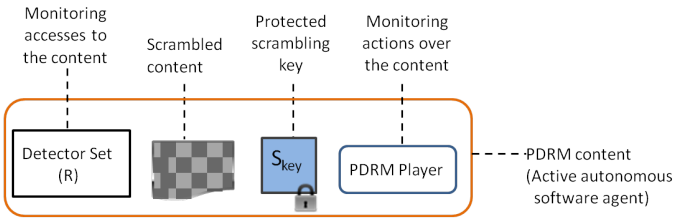

Figure 1.5: A PDRM protected content

1. **The detector set**

   The detector set is generated according to the negative selection algorithm described in Section 1.4. According to this algorithm, we need to define first what is *self*. In the context of personal content protection, we define *self* as any authorized access to the content and *non-self* being the opposite. Authorized access to the content includes for example:

   - Authorized group of consumers
   - Authorized device from which one can play the content
   - Trusted locations
   - Access within the boundaries of a specific policy (e.g. law regulation, etc.).

   Therefore, *self* is the set of authorized accesses generated according to an access policy defined by the content owner.

   Scenarios. In Scenario 1, where the person wants to share her personal picture only with her family members and close friends, *self* represents the set of family members and close friends. Alternatively, it can also encode the concept of social links between the owner and her family and friends. In Scenario 2, where the blogger wants his personal content to "self-destroy" when the policies related to the content change, *self* represents a series of situations captured by the set of policies applicable to the content (i.e. age of the content or number of years after which the content should expire). In Scenario 3, where the software developer wants to share her fun game only with

her team colleagues, *self* represents the set of team colleagues, or the work relation between the software developer and her colleagues. Finally, in Scenario 4, *self* encodes the authorized combinations of locations and devices from which access is possible.

Fig. 1.6 shows different definitions of *self* for the four scenarios described in Section 1.2.1.

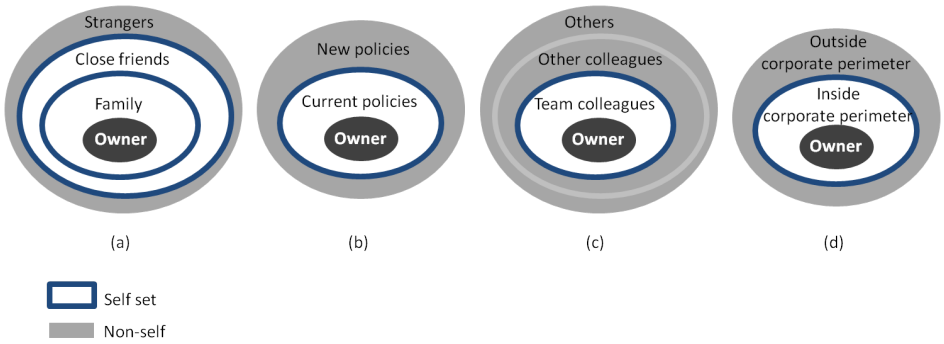

Figure 1.6: Definition of *self* in PDRM

2. **Detector generation**

Initial phase - The initial repertoire of *self* set is provided by the content owner according to the access policy specified for the content. This requires a content owner effort. We introduce this phase in order to reduce the time necessary to accumulate the *self* patterns through learning and observation only.

Learning phase - The learning phase follows the initial phase and depends on the kind of protection intended by the content owner.

Scenarios. For each scenario described in Section 1.2.1, the learning phase is as follows. In Scenario 1, the PDRM system will learn over time from the content owner social network (Facebook, Google+, etc.), instant messaging (Skype, Yahoo, Microsoft Messenger, etc.) or mail box contacts. The PDRM system will build automatically over time the different levels of the content owner relationships (Fig. 1.6). In Scenario 2, the learning phase will consist of gathering policies related to the content, the associated countries in which the regulation is applicable, the dates, and any other relevant information related to the applicability of the policy. In Scenario 3, the PDRM system will learn over time from the developer company organization (e.g., Human resources

charts, company directories, etc.). Finally, in Scenario 4, the PDRM System will learn from the security system of the organization.

The *self* patterns learned in this phase are added on top of the initial repertoire of *self* set. Depending on the scenario envisioned, the learning phase may suffice to build a complete description of *self*. However, in certain scenarios, continuous learning may be required.

Once the learning phase is over and the different circles are constituted (Fig. 1.6), unique data elements are associated to each object in the levels corresponding to *self*. These unique data elements can be for example the location information (e.g., IP address, GPS), devices informations (e.g., device hardware serial number, Mac address, etc.), employee and employee machine identifier, policy regulation reference, policy applicability context, etc. Using these informations, a set of binary strings is generated. It consists of the valid concatenations (or as in Scenario 4 valid combinations) of the unique identifiers associated to each element in the *self* circle.

Maturation phase - Together with a randomly generated set of strings, the *self* strings are used as an input for the generation of the detector set (Fig. 1.7). Any randomly generated binary string that does not match any *self* string up to a certain threshold will enter the detector set. The threshold selection is based on a partial matching. A partial matching is used because a perfect detection will require as many detectors as the number of *non-self* strings, and this can be huge depending on the length of the *self* pattern. For example: a set of l *self* strings of m bits will require up to $2^m - l$ *non-self* strings. This increases exponentially when the *self* string length increases. Therefore, with partial matching a single detector in the detector set can match a larger subset of *non-self*, and so fewer detectors are needed to cover the whole set of *non-self* strings. However, as the subset of the *non-self* strings that can be detected increases, the ability to make precise discrimination decreases, diminishing the detection ability. The main challenge here lies in generating a detector set R that matches as many of the *non-self* strings as possible, without matching any of the *self* strings. In other words, this consists in defining the probability that a random *non-self* string will not be matched by any of the detectors in the detector set R (i.e. defining the false negative rate). Some methods were proposed in [6] and will be analyzed in future work for the implementation

of the PDRM system.

3. **The scrambled content**

   The PDRM content is scrambled using a scrambling key (S_{key}). The scrambling algorithm is randomly selected among a set of scrambling algorithms. This algorithm can be part of a large family of related crypto-algorithms [24]. The selected algorithm is identified by its slot (*slot*). The purpose of having different scrambling algorithms is to minimize the *break once, break everywhere* problem [1]. In fact, each PDRM protected content would require its own attack. The overall system would survive repeated attacks, and content would only leak out slowly over time. In addition, there is a freedom in manipulating the scrambling keys, making it potentially more difficult for an attacker to recover these keys as compared to the cryptographic keys [24].

4. **The protected scrambling key**

   The scrambling key S_{key} is protected using a standard encryption algorithm (e.g., AES).

5. **The PDRM player**

   The role of the PDRM player is to prevent unauthorized operations such as print, copy, print screen, etc. over the content once access is granted to it. By bundling the player with the protected content, we can easily customize the player software. Different flavors of the player can be imagined for each protection policy.

   Scenarios. In Scenario 1, the PDRM player simply denies actions such as viewing, opening or reading the content for an unauthorized consumer. In Scenario 2, the PDRM player allows the content to be viewed if the policy still allows it, then destroys the content once the time has expired. In Scenario 3, the PDRM player will ask a token of compensation before allowing the consumer to play the game if the player is not a colleague. Finally, in Scenario 4, the PDRM player will allow or not the HR employee to read, modify the files, depending on whether or not the employee is in an authorized location with the correct device and access is granted or not to the content.

Fig. 1.7 summarizes the PDRM content packaging described above.

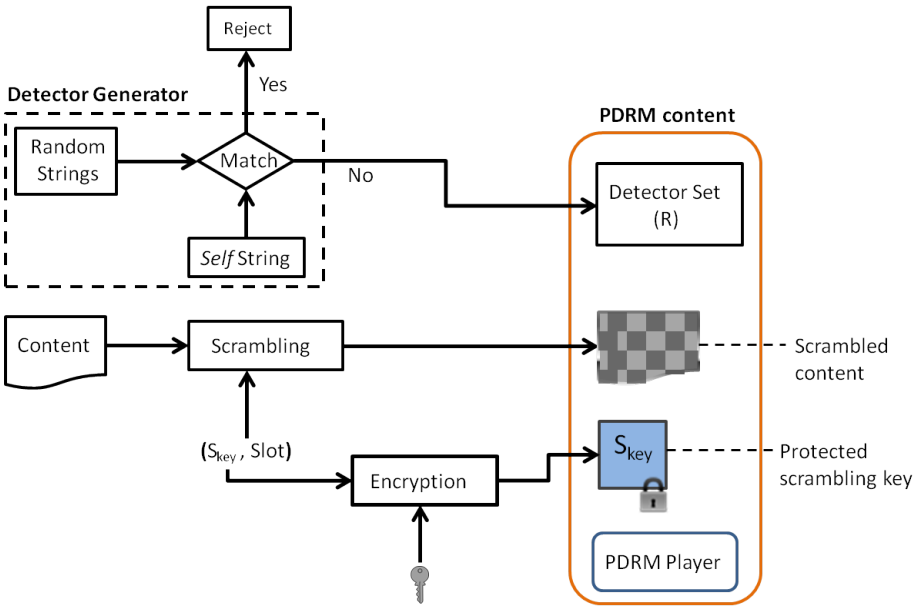

Figure 1.7: PDRM content packaging

1.5.2 Access monitoring and the detector life cycle

Whenever a user wants to access a PDRM protected content, the current access context informations are used to generate a binary string, which is passed to the detector. If the binary string is not detected, this means the access to the content is within the boundaries specified by the owner. In this case, the detector generates the key necessary to decrypt the content scrambling key, and the slot corresponding to the selected scrambling algorithm. Then, the decrypted scrambling key (S_{key}) along with the slot ($slot$) are used together to descramble the content. Once descrambled, the content is passed to the embedded player for opening or viewing. In case the binary string is detected, meaning the content is accessed outside the boundaries specified by the content owner, access to the content is denied. Fig. 1.8 shows the personal content recovering process in PDRM.

In some scenarios, the description of *self* can change over time. For example: In Scenario 1, the content owner can add to or remove friends from her circle of close friends. In Scenario 2, the blogger can add new policies to her content or the policy itself can change (e.g. expiration time of the content is shortened). In Scenario 3, the developer team's colleague can leave the company. In Scenario 4, the security policies of the organization can change.

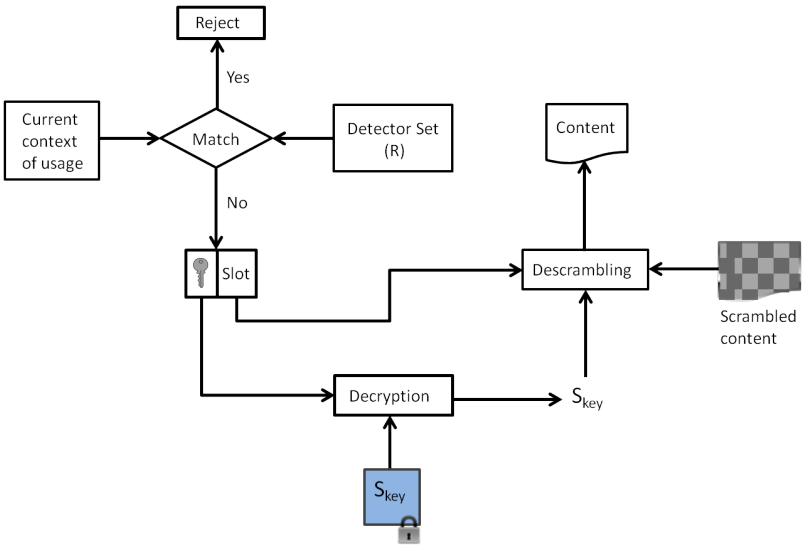

Figure 1.8: PDRM protected content access process

To address these legitimate changes to the description of *self*, we propose two approaches: First, by using the co-stimulation mechanism described in Section 1.4, and second, by a periodic update of the detector set by the content owner. These two approaches are described as follows:

Co-stimulation - When a binary string s encoding a current context of usage is detected, the PDRM system checks if this string s has already been detected or not in the past. If not, a feedback is requested to the content owner to decide whether the string s is *self* or not. If s is actually *non-self*, the string s is added to the memory of detected *non-self* patterns and access to the content is denied. Otherwise, the access to the content is granted.

Detector update - The content owner can periodically update the detector set according to changes in the description of *self*. This can happen when the content owner changes access policies or new users and devices are added/removed to/from the set of authorized users and devices. For example when a binary string representing a new device of the user is detected by the detector, the content owner can update the detector set after receiving the co-stimulation request from the PDRM system and checking that the device really belongs to the authorized user.

Fig. 1.9 summarizes the detector set evolution over time and Fig. 1.10 summarizes the detector life cycle in PDRM system.

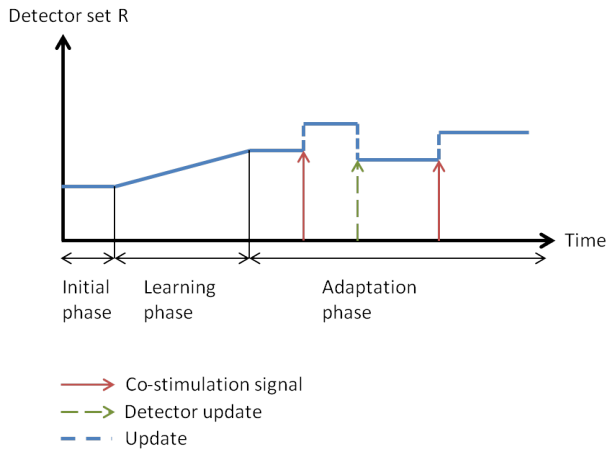

Figure 1.9: Detector set evolution over time

1.6 Discussion

Next, our PDRM system is evaluated along the following metrics: adaptability, flexibility, accessibility, security and privacy.

- **Adaptability:** It occurs along different lines. First, a detector set dynamically adapts to changing situations, without any specific instruction from the content owner. Indeed, let's consider Scenario 1, where the content owner shares pictures only with family members and close friends. The detector set encodes the fact that people not directly linked with the content owner are not granted access to the picture. Therefore, if a close friend, who was previously granted access to the content owner's pictures, suddenly loses the social link to the content owner (i.e. he is no longer listed as a friend through social media such as Faceboook), the detector set will naturally deny access to that person. Learning, co-stimulation and periodic update are additional mechanisms favoring adaptation.

- **Flexibility:** From the user's point of view, PDRM allows a flexible use of the PDRM protected content in case of time shifting or space shifting. Since the access rules (ensured by the detector set) are included in the PDRM package along with the usage rules (ensured by the content player), a PDRM protected content can be stored for later use. In addition, the adaptivity of the detector set enables the PDRM protected content

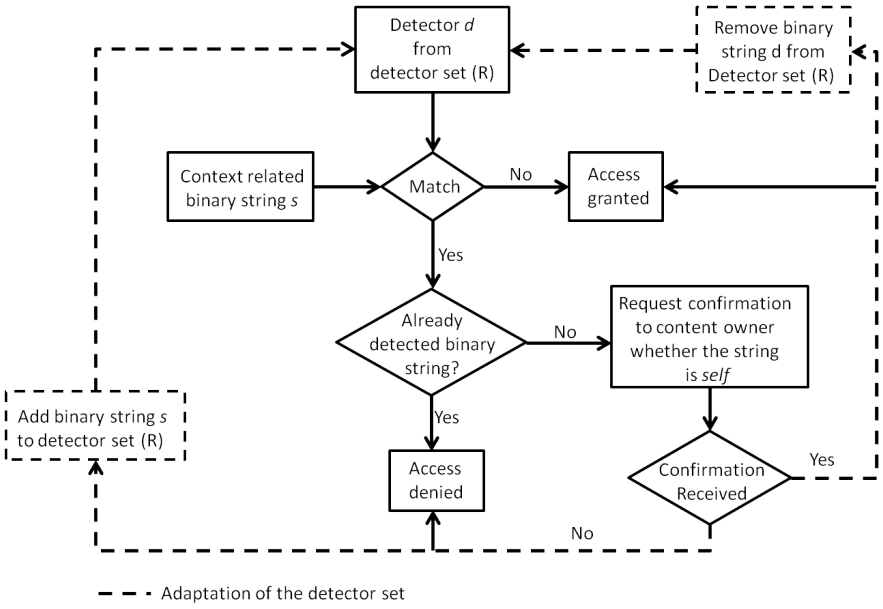

Figure 1.10: PDRM detector life cycle (Detector adaptation through co-stimulation)

to be consumed interchangeably on any device of the consumer and at any location. For example, when a legitimate user wants to access a PDRM protected content with a new device, the co-stimulation mechanism described in Section 1.5 will trigger an update of the detector set that takes into account the new device. Moreover, a PDRM protected content can be shared among authorized users without any prior requirements such as pre-installation of a software or a hardware. For example, in Scenario 1, once the picture is packaged along with the detector set (allowing access to family members and close friends), and a player (running on platforms identified during the learning phase), the picture is shared and consumed on any of the family members or close friends device. This happens without any intervention of the content owner. In Scenario 2, the content can also be consumed on any device at any location provided that the detector identifies any modification in the policies applicable to the content.

From the content owner point of view, the PDRM system doesn't require any pre-installation of a third party services or hardware. However, to shorten the learning phase, necessary to build the *self* set, the PDRM system requires some effort from the content owner for the generation of the initial repertoire of the *self* set. The content owner may also explicitly request the co-stimulation signal depending on the changes

occurring in her environment.

- **Accessibility:** Contrarily to current DRM systems, the PDRM system does not involve any permanent link or dependency to the content owner or to any service. An Internet connection is required from time to time only, for updating the detector set or for the co-stimulation signal.

- **Security:** In terms of security, the main threat for a software based DRM is reverse engineering. The PRDM system is designed in a way to make it hard for a deliberate attacker to break one instance of a PRDM protected content and hence to break all instances. This is achieved through the uniqueness of the detector set attached to each PDRM protected content. Therefore, if one detector is compromised, it doesn't affect other contents.

  In addition, having different sets of scrambling algorithms, and randomly selecting one from a list of algorithms to scramble the content makes the life of an attacker more difficult. Indeed, even if an attacker is able to break one particular piece of content, breaking another piece of content destined for the same user is still a challenging task. Since the scrambling algorithm is proprietary, recovering the scrambling keys without recovering the algorithm is insufficient to recover the content. Moreover, there is additional freedom in manipulating the scrambling keys, making it potentially more difficult for an attacker to recover these keys as compared to the cryptographic keys.

  The fact that the protection of a content is mainly based on the detector set has some disadvantages. First, the detector generation can be computationally expensive. In fact, as mentioned previously, the size of the detector set can increase exponentially with the size of the *self* set. This complexity can be seen also as an advantage, since it would be difficult for an attacker to simply alter the detector set so that a *non-self* situation could not be detected. Second, to reduce the complexity of generating the detector set, detectors matching a larger subset of *non-self* are needed. However, the larger the subset of *non-self* detected by a single detector, the more the precision of the detection decreases. This situation can introduce discrimination errors under the form of *holes*. A hole is a *non-self* string for which no valid detector can be generated [5, 6]. These holes can be exploited by an attacker. However using solutions such as

mutli-representation of the detectors [7], we can reduce the overall number of holes.

- **Privacy:** No user identity is associated to a PDRM package. The detector set included in a PRDM package comprises the negative part of the intended user's identity. Therefore, the user's identity cannot be deduced from the detector set, hence, avoiding any tracking of a user's habit.

1.7 Conclusion and future work

This chapter proposes the design of Personal DRM system, targeting personal (as opposed to centralized, proprietary) content. The PDRM approach is inspired by the human immune system and uses a negative selection mechanism to encode unauthorized situations of usage and access of the content. Future work includes: actual modeling of *self* strings in the different Scenarios discussed above; determination of authorized consumer based on social links gathered from social media such as Facebook; actual implementation of a PRDM player and a software for building the detector set; specification of a measure of distance for the matching of strings encoding current activities against strings in the detector set; and works tackling the problem of finding an optimum algorithm for the generation of the detector set that matches as many of the *non-self* strings, without matching any of the *self* string.

References

[1] P. Biddle, P. England, M. Peinado, and B. Willman. The darknet and the future of content distribution. In *ACM Workshop on Digital Rights Management*. ACM, 2002.

[2] D. Bradley and A. Tyrrell. A hardware immune system for benchmark state machine error detection. In *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02*, volume 1, pages 813–818. IEEE Computer Society Press, 2002.

[3] ContentGuard. Xrml - the digital rights language for trusted content and services. `http://www.xrml.org/index.asp`.

[4] L. N. De Castro and J. Timmis. *Artificial immune systems: a new computational intelligence approach*. Springer-Verlag, 2002.

[5] P. D'haeseleer. An immunological approach to change detection: Theoretical results. In *Proceedings of the 9th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press, 1996.

[6] P. D'haeseleer, S. Forrest, and P. Helman. An immunological approach to change detection: Algorithms, analysis and implications. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 1996.

[7] S. Forrest and S. Hofmeyr. Engineering an immune system. *Graft*, 4(5):5–9, 2001.

[8] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri. Self-nonself discrimination in a computer. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, pages 202–212. IEEE Computer Society Press, 1994.

[9] R. Grimm and P. Aichroth. Privacy protection for signed media files: a separation-of-duty approach to the lightweight DRM (LWDRM) system. In *Proceedings of the 2004 workshop on Multimedia and security*, (MM&Sec '04), pages 93–99, New York, NY, USA, 2004. ACM.

[10] S. Heuvel, W. Jonker, F. Kamperman, and P. J. Lenoir. Secure content management in authorised domains. In *International Broadcasting Convention, IBC*, pages 467–474, Amsterdam, The Netherlands, 2002.

[11] S. Hofmeyr and S. Forrest. Architecture for an artificial immune system. *Evolutionary Computation*, 7(1):1289–1296, 2000.

[12] IFPI. Ifpi digital music report 2010. `http://www.ifpi.org/content/library/dmr2010.pdf`.

[13] N. Jerne. Towards a network theory of the immune system. *Annales Institut Pasteur (Immunology)*, 125C:373–389, 1974.

[14] W. Ku and C. H Chi. Survey on the technological aspects of digital rights management. In *Information Security Conference (ISC'2004)*, volume 3225 of *LNCS*, pages 391–403, 2004.

[15] S. Michiels, K. Verslype, W. Joosen, and B. De Decker. Towards a software architecture for DRM. In *Proceedings of the 5th ACM workshop on Digital rights management*, DRM '05, pages 65–74, New York, NY, USA, 2005. ACM.

[16] J.H. Morin. Exception based enterprise rights management: Towards a paradigm shift in information security and policy management. *International Journal On Advances in Systems and Measurements*, 1(1):40–49, 2008.

[17] C. Neubauer, F. Siebenhaar, and K. Brandenburg. Technical aspects of digital rights management systems. In *Audio Engineering Society Convention 113*, Oct. 2002.

[18] ODRL Initiative. Odrl initiative - an open policy language for the digital commons. `http://odrl.net/prev.html`.

[19] B. C. Popescu, B. Crispo, A. S. Tanenbaum, and F.L.A.J. Kamperman. A DRM security architecture for home networks. In *Proceedings of the 4th ACM workshop on Digital Rights Management*, pages 1–10. ACM, 2004.

[20] P. Ram, T. T. Ta, and X. Wang. *Self-protecting documents.* Google Patents, February 2003. US Patent 6,519,700.

[21] P. Ram, T. T. Ta, and X. Wang. *Self-protecting documents.* August 2005. EP Patent 0,999,488.

[22] O. Sibert, D. Bernstein, and D. Van Wie. Digibox: A self-protecting container for information commerce. In *Proceedings of the first USENIX workshop on electronic commerce*, pages 1–13, 1995.

[23] D. Sohn. Understanding DRM. *Queue*, 5(7):32–39, November 2007.

[24] M. Stamp. Digital rights management: the technology behind the hype. *Journal of Electronic Commerce Research*, 4(3):102–112, 2003.

[25] A. Studer and A. Perrig. Mobile user location-specific encryption (MULE): using your office as your password. In *Proceedings of the third ACM conference on Wireless network security (WiSec'10)*, pages 151–162. ACM, 2010.

[26] M. Weiser and J. S. Brown. The coming age of calm technology. In *Beyond calculation*, pages 75–85. Copernicus, New York, NY, USA, 1997.

[27] Y. Zuo and T. O'Keefe. Post-release information privacy protection: A framework and next-generation privacy-enhanced operating system. *Information Systems Frontiers*, 9(5):451–467, Nov. 2007.