Digital Forensics for Enterprise Rights Management Systems

Sebastian Schrittwieser
Vienna University of Technology
Vienna, Austria
sebastian.schrittwieser@tuwien.ac.at

Peter Kieseberg
SBA Research
Vienna, Austria
pkieseberg@sba-research.org

Edgar Weippl
SBA Research
Vienna, Austria
eweippl@sba-research.org

ABSTRACT

Digital forensics is the application of techniques to recover, reconstruct and analyze data from a computer or a similar system in order to gather digital evidence (e.g. on a suspicious employee or for law enforcement). Guidelines and standards for forensic investigations exist (e.g. NIST SP800-86), but do not cover Enterprise Rights Management (ERM), where data is usually encrypted and therefore inaccessible without knowing the cryptographic key. This paper explores forensic techniques for ERM systems and develops application specific guidelines for forensic investigations targeting Microsoft Active Directory Rights Management Services (RMS) and Adobe LiveCycle Rights Management. Moreover, we illustrate the important role of database forensics for investigations in ERM systems and finally show that with Microsoft's ERM solution no secure, centrally-managed revocation of specific documents in order to prevent digital forensics is feasible.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Access controls; H.3.4 [**Systems and Software**]: Information networks

General Terms

digital forensics, enterprise rights management, revocation, databases

1. INTRODUCTION

In today's enterprises, sensitive data and confidential documents are typically protected by access restriction systems provided by the operating system or a database (e.g. access control lists, ACL) and data encryption. However, on closer examination both of them have major limitations that could result in the disclosure of sensitive business data. Access restriction systems only provide an effective protection inside the running system or database. If an attacker has direct access to the filesystem outside these systems, access control mechanisms do not take effect anymore. Data encryption solves this problem, but does not guarantee persistent protection. When working on a document, the encryption has to be temporarily removed from the file. This process opens an attack vector as the decrypted file is loaded by the viewer application into the system's memory. Furthermore, data encryption protects the file as a whole, so there can only be full access or no access at all. Enterprises need to specify more precisely which operations should be permitted to whom on which document. Thus, usage control on sensitive information needs not only to be able to define who has access at all, but also enable detailed specification of the individual rights on the business data for the given users.

Another threat comes from inside the companies. The careless handling of sensitive data by authorized users exposes even more risks to the containing information. A 2012 report [23] by Verizon shows that about four per cent of data breaches in US companies are the result of untrustworthy or careless employees. As the report only covers reported incidents, it can be estimated that the number of unreported cases is much higher. Neither access control schemas nor data encryption are adequate solutions to mitigate that threat.

Enterprise Rights Management is the adoption of Digital Rights Management (DRM) technologies designed to protect and manage data in enterprises. In contrast to DRM, which may be considered as failed or at least lacks acceptance in its originally intended area of application as a copy-protection system, ERM is used in much more closed environments with different threat sources and different threat models. Enterprises can adopt these technologies without encountering most of the problems related to DRM (e.g. loss of control for the customer). Sensitive documents are encrypted and accompanied by a so-called license that contains a description of access rights and an encrypted document key. Viewer applications (ERM clients) allow access to protected documents while enforcing usage restrictions that are imposed by the license: the ERM client only decrypts the document in case the license is still valid and allows the access of the given document at the given time by the respective user. The possibility of implementing fine-grained access and usage control makes ERM the prime candidate for the protection of data in business environments.

iiWAS2012, 3-5 December, 2012, Bali, Indonesia.

However, the adoption of rights management systems in enterprises makes forensic investigations much more difficult and guidelines on how to mitigate resulting impacts on digital forensics do not exist. A forensic examination of computers and other devices aims to collect digital evidence related to criminal behavior. Similar to traditional forensics, an evidence has to be pure and reliable in order to be admissible in court and a clear chain of custody has to be maintained. The de-facto standard for digital forensics in the US, NIST SP800-86 [16], describes four basic phases of investigation, Collection, Examination, Analysis, and Reporting, none of them covering Enterprise Rights Management. In this paper, we evaluate existing ERM solutions and develop guidelines for forensic as well as anti-forensic techniques in ERM scenarios.

The main contributions of this paper are:

- We provide an evaluation of digital forensic capabilities in Enterprise Rights Management based on two real-life examples.
- We show that Microsoft's ERM cannot be used to centrally make all copies of one document unreadable without a major impact on the company's entire ERM.
- We demonstrate the tight binding of ERM forensics to database forensics and evaluate its impact. Even if information is no longer available in an ERM system, database forensics can still recover keys and the history of policy changes.

2. RELATED WORK

2.1 Enterprise Rights Management

Previous work on (Enterprise) Digital Rights Management range from theoretical approaches for security architectures in the late 1990s and early 2000s to detailed analysis of (E)DRM products available on the market. To the best of our knowledge, this paper is the first to evaluate the impact of Enterprise Rights Management on forensic investigations in a business environment.

A comprehensive taxonomy of rights management architectures was developed by Park et al. [20]. While the taxonomy includes architectures without a special client software, we consider the existence of a client that restricts the access to the digital information to be essential for an ERM environment.

Yu and Chiueh [24] describe a novel ERM approach where sensitive files remain on a secure fileserver and images of the content are streamed to the client instead - it is therefore called Display Only File Server (DOFS). In this architecture, forensic investigations have to focus primarily on the server as the client does not store any forensic material that can be used for evidence.

A work by Julie E. Cohen [10] discusses privacy issues in rights management systems and describes the two basic types that have to deal with different aspects of data privacy. Stand-alone systems do not validate against an authorization server and do not send data of the user to the server. The other type of rights management systems reports back to the server and therefore could track the activities of the users. From a forensic perspective, the latter type is more interesting because an activity log can contain useful information for a forensic investigation.

A work by Bradley and Josang [7] discusses five problems of today's rights management systems that might prevent success on the market. Further, they define features of Enterprise Rights Management frameworks that could solve these problems and introduce, based on this analysis, a novel approach for a rights management framework called Mesmerize.

In 2008 Pretschner et. al. [21] discussed the state-of-the-art and possible future aspects of the usage control mechanisms of selected DRM and ERM technologies, including Adobe LiveCycle. Furthermore, they presented a taxonomy for defining properties and capabilities inherent to existing enforcement mechanisms for usage control and enriched it by additional, theoretically possible, properties. Digital forensics, however, was not in the scope of their analysis.

Alawneh and Abbadi [5] discuss the danger of insider threats, especially focussing on content leakage. They especially focus on state-of-the-art schemes for mitigation of insider content leakage and the resulting loss of sensitive information. Furthermore, Morin [18] discusses the problem of users circumventing security measures that deprive them of the flexibility needed to do their work. This paper discusses a countermeasure, which could be used to mitigate the problem.

In 2010 Henriques et. al. [15] discussed the use of ERM technologies in order to protect CAD data containing lots of additional and sometimes historical information, e.g. model history, manufacturing information or used materials. Sinha [22] outlines the design and implementation of Data Paladin, an application-independent rights management system together with a discussion on limitations and mitigation strategies.

2.2 Digital Forensics for ERM

There are a number of publications on digital forensics and general standards (e.g. NIST SP800-86), however they do not cover guidelines for forensic investigations in Rights Management systems.

Some preliminary work describe the influence of data encryption on digital forensics. Casey and Stellatos [9] argue that the increasing adoption of strong encryption, in particular full disk encryption, forces investigators to change their best practices for digital forensics (e.g. acquisition: shutting down an evidentiary system prevents unintentional alterations of data, but may also prevent future access to the system).

Burmester and Mulholland [8] discuss the impact of Trusted Computing on digital forensics. On the one hand, Trusted Computing makes forensic investigations significantly harder and requires new practices (e.g. for data acquisition), but on the other hand, collected evidence is more compelling, due to new features such as digital signatures and tamper-proof logging.

Kessler [17] introduces anti-forensic methods that aim to prevent forensic investigations. While the author limits the legitimate uses of anti-forensics to the protection of privacy, the effective destruction of cryptographic keys to mitigate high costs associated with investigations demanded by public authorities, is another important use for the methods outlined in this work.

Arnab [6] discusses requirements for ERM suites with respect to their applicability in companies. The three most common products at that time, Microsoft RMS, Authentica Enterprise Digital Rights Management Solutions and Adobe LiveCycle are evaluated with respect to these requirements.

Even though, persistent protection is one of the key aspects in the evaluation, the possibility of digital forensics against ERM systems is not taken into account, the analysis is more focussed towards practical features like portability, integration with existing applications, transfer of rights and possible limitations.

2.3 Database Forensics

The possibilities of forensic investigations in ERM heavily depend on the rights management system's architecture. Databases are used by many rights management systems for managing licenses and cryptographic keys on the server. Thus, databases are a valuable source of information for forensic investigations.

Fruehwirt et al. [13] have shown that the InnoDB storage engine of MySQL does not physically delete data from the storage files, but instead flags them as deleted. Thus, recovery is possible as long as deleted rows have not been overwritten by new data. Furthermore, the authors developed forensic methods for automated extraction of deleted data from InnoDB tables.

A second starting point for database forensic investigations in InnoDB are the comprehensive log files. Fruewirth et al. [14] have shown the practicability of analyzing InnoDB log files for database forensic purposes. These files store log data that allows the recovery of deleted rows, deleted field data and even the rollback to previous versions of field data as long as the used areas of the log file have not been overwritten by other data. In many cases, a complete recovery of deleted data such as cryptographic keys is possible.

Fowler [12] have shown that deleted data can be recovered from Microsoft SQL Server in many ways:

- *Transaction Logs:* If the database uses the full recovery model, all transactions in the database are logged and data can be restored at any time.
- *Data Files:* Deleted records remain in the data files until they are overwritten by new data.
- *SQL Server Memory:* All data is temporary stored in the SQL Server Memory.

A work by Olivier [19] discusses the general differences between digital forensics in filesystems and databases.

3. PROBLEM DESCRIPTION

Guidelines and standards (e.g. NIST SP800-86) provide directions on how to gather evidence during a computer forensic investigation. They cover a broad range of domains (e.g. file systems and specific applications) and discuss the fundamental difficulties in extracting information from encrypted data. However, they do not provide guidelines for forensic investigations in Enterprise Rights Management environments.

The main difference between encrypted documents and ERM protected ones, is that for latter the location of the encryption key is known to the forensic investigator. It might not be accessible directly (e.g. because of organizational or technical reasons), but forensic techniques from other domains can be used to recovery keys from ERM systems in order to reenable access to protected documents. This paper aims at exploring ways of enabling digital forensic in ERM scenarios by borrowing techniques from other domains.

3.1 Forensic investigation of ERM protected documents

ERM protected documents are encrypted with a so-called content key. Access to the content is provided by the ERM system (e.g. the ERM client software requests the content key from the ERM licensing server). For digital forensics, the investigator requires access to the licensing server in order to be able to decrypt the content of ERM protected documents. General standards on how an investigator can access cryptographic keys on the server do not exist in today's guidelines for digital forensics. In this paper, we introduce guidelines for handling ERM protected files in Phases 1 (collection) and 2 (examination) defined by the de-facto standard for forensic investigations, the NIST SP800-86.

3.2 Anti-Forensics

Digital forensics can support companies in analyzing activities of suspicious employees and gather evidence against them. But also public authorities could demand data from forensic investigations and companies have to meet retention periods according to laws (e.g. the EU Directive 2006/43/EC as well as the Sarbanes-Oxley Act in the U.S.) that require business documents to be archived for a certain period of time. However, due to probably high expenses of forensic investigations, companies have a strong interest in techniques to delete out-dated enterprise data after expiration of retention periods and thus be able to prevent costly investigations demanded by public authorities in the future. Today's ERM environments do not provide adequate solutions to this problem. This paper discusses methods for effectively preventing forensic investigations of out-dated data. These activities are part of the research area of anti-forensics that aims to hinder forensic investigations (e.g. by hiding or destroying data).

4. METHODOLOGY

In this section, we discuss requirements for digital forensics as well as anti-forensics in ERM scenarios and introduce two solutions, Microsoft Rights Management Services and Adobe LiveCycle, based on which we evaluated forensic techniques. We chose these two systems because they not only are among today's most common used ERM systems, but also because of their differences regarding to usage license distribution. While RMS attaches an encrypted version of the license to the document, LiveCycle stores them in a server-side database.

Digital forensic investigations depend on the availability of data to analyze. In the context of ERM systems, access to the decryption keys is crucial. In general, without keys it is impossible to analyze the content of ERM protected files. As mentioned before, there are different types of license distribution methods. The license may either be attached to the content (the actual documents that should be protected) or delivered from the licensing server. Offline usage functionality is another type of distribution, where a special type of license is attached to the file, which does not require the viewer to authenticate against the server for a certain amount of time. In the area of anti-forensics, the possibility for destroying keys independently of limiting the access to other files in the ERM environment is crucial.

4.1 Microsoft Rights Management Services

Microsoft RMS [2] enables enterprises to control the usage of Microsoft Office documents. Users can request a so-

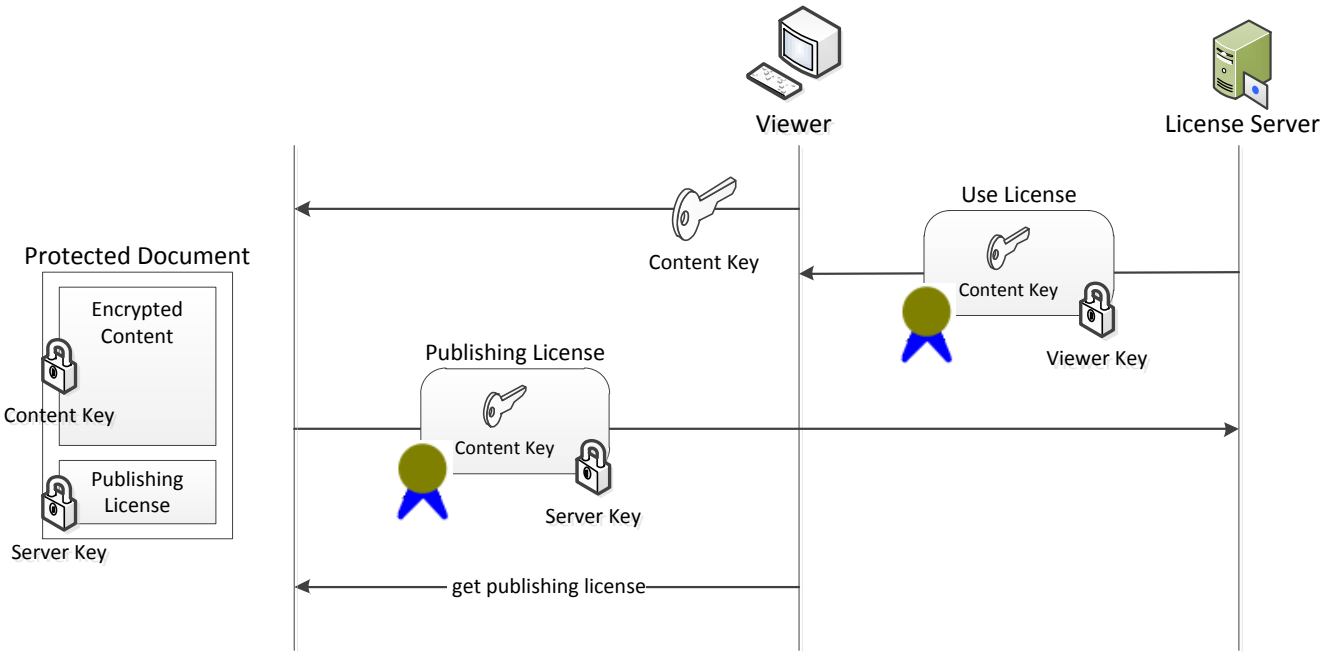

Figure 1: Architecture of RMS

called Publishing License that allows them to issue usage permissions for a file. Amongst others, it contains access and usage rights expressed in XrML [11] and a 128-bit AES key for content encryption (content key) that is encrypted with the public key of the licensing server. The Publishing License is attached to the document, thus the encrypted content key is stored inside the document. When a client wants to access the content, the Publishing License is sent to the server that decrypts the content key and returns it to the client (encrypted with the public key of the client). Figure 1 illustrates the architecture of RMS.

4.1.1 RMS Client software and Lockbox

The client software provides a set of APIs that can be called by RMS enabled applications. The so-called lockbox, a dynamic-link library (DLL-file), is the core of the RMS client software and performs all the encryption and decryption tasks for the client. It is created during the enrollment process of the client and is linked to the machine with a hash value (client ID) that is derived from hardware components of the client. According to Microsoft, the hashing algorithm is undocumented for security reasons, which seems much like the classical "security by obscurity" paradigm. The hash value calculation is not performed client-side, but by the RMS server. The client ID binds the lockbox to a specific machine and should make it more difficult to run it on another machine. The lockbox incorporates code obfuscation and anti-debugging methods (e.g. looks for debugging tools and the /Debug flag in the Windows boot.ini file) to prevent disclosure of the cryptographic key and other types of attacks on the RMS infrastructure. In addition, the lockbox checks the system clock to prevent manipulation (e.g. re-enabling expired licenses). Not only the clients have a lockbox, but also the RMS server for enabling trusted server software to access RMS protected files (e.g. antivirus scanner or indexing applications).

4.1.2 RMS enabled application

RMS enabled applications, such as Microsoft Office, offer the user interfaces to the RMS environment, while the actual encryption and decryption tasks are delegated to the RMS client software. An RMS enabled application has to be certified by Microsoft to ensure its trustworthiness and that it meets the RMS policies. When calling RMS client APIs, the application has to transmit the so-called application manifest, a Microsoft signed certificate, that contains a hash value of the application executable. This approach guarantees that only trusted applications can communicate with the RMS client software.

4.2 Adobe LiveCycle Rights Management

LiveCycle Rights Management [4] is an ERM solution by Adobe. It is part of the LiveCycle Enterprise Suite, Adobes SOA server software that provides a wide range of services such as content management. The Rights Management component of the LiveCycle software is primary used for the protection of PDF files, but also supports Flash, Microsoft Office and some CAD file formats.

The encryption algorithm that is used by LiveCycle Rights Management can be chosen by the user that creates a usage policy (no encryption, AES 128-bit, or AES 256-bit). Selecting the "no encryption" option results in a PDF file with compressed, but unencrypted text objects that can be read in cleartext after decompression. Therefore, the restriction of a file with disabled encryption does not offer any kind of protection. It is unclear, why this option is available when creating new policies. We consider this a major security risk. Due to the fact, that policies can be shared and reused, an attacker could create policies with disabled encryption and share it with other users. PDF files that are protected with

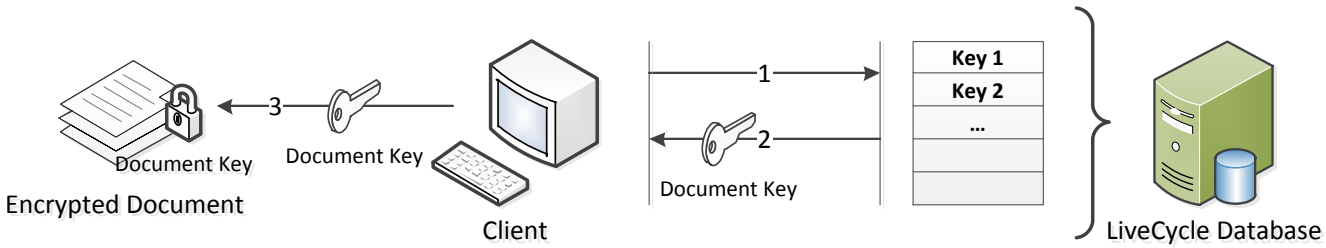

Figure 2: Architecture of LiveCycle

these policies would be readable as cleartext for unauthorized users by simply decompressing the (unencrypted) text objects.

LiveCycle stores the document keys inside the LiveCycle database on the server. When a client requests access to protected documents, the server returns the document key for that specific file to the client. The protected file does not store any policy or key data. Figure 2 illustrates the architecture of LiveCycle.

5. RESULTS

This section presents the results of our research on ERM forensics. We discuss characteristics of forensic investigations in ERM environments based on both Microsoft RMS and Adobe LiveCycle. These two systems have similar functionality, but different architectures and therefore approaches for investigations are different.

Encryption is one of the strongest methods against forensic investigations, because without knowing the cryptographic key, an investigator may not be able to access encrypted data. In ERM environments documents are encrypted too, however, the storage location of the key is published in the specifications and obtaining cryptographic keys from the licensing server is a valid way for forensic investigations in ERM environments in order to be able to decrypt and access the content of the document. Nevertheless, the architectures of the introduced ERM systems Microsoft RMS and Adobe LiveCycle differ and forensic investigations have to be based on different cryptographic keys and server modules (e.g. databases).

Another important aspect is offline use. Being able to access ERM protected documents without connecting to the licensing server is an essential requirement for rights management systems in business environments. If there exists a valid offline license, access to the content of the document is possible without validating against an authorization server because a copy of the required key is stored in the this license. Securely denying access to documents is only possible, if offline use is disabled. On the other hand, offline use reduces the complexity of forensic investigations on ERM protected files, because access to the content is possible without having access to cryptographic keys that are stored on the licensing server.

5.1 ERM forensics

A common use case for digital forensics in a company is the investigation of a suspicious employee's hard disk that contains ERM protected files. The investigator may want to access the content of these files in order to collect evidence against the employee.

5.1.1 *Rights Management Services*

RMS encrypts the content key with the public key of the server (Server Licensor Certificate) and attaches it to the Publishing License that is embedded in the document. Therefore, access to the private key of the server is required in order to be able to decrypt any content key and any document of the RMS environment accordingly. When an RMS client requests access to a document, the server creates a so-called *Use License* that is attached to the document and contains the content key (encrypted with the public key of the client). The refresh interval parameter of the Use License defines how long a client is allowed to access a document without re-authorizing against the licensing server (offline use).

If offline use permissions are not available or expired, there are three possible ways to access protected content in a forensic investigation:

- The private key of the server decrypts the content key. Therefore, an investigator can retrieve the server's private key in order to be able to decrypt the document. This key is either stored unencrypted in the RMS configuration database (Table `dbo.DRMS_LicensorPrivateKey`) or in a secure hardware device, where key extraction is much more difficult to archive. A Use License that is attached to the document also contains the content key (encrypted with the public key of the client). However, the client's private key is protected by the lockbox and thus extraction is not feasibly possible.
- An RMS administrator can enable the so-called superuser that is allowed to access every document of the RMS environment.
- RMS enables trusted server software to access protected content (e.g. antivirus scanner or indexing applications). The company could develop a server application that can extract content from RMS protected documents for forensic investigations.

5.1.2 *LiveCycle Rights Management*

In contrast to RMS, LiveCycle based ERM systems maintain an individual document key for each protected file on the server. The LiveCycle database stores document keys in the Table `edclicenseentity`. Despite the fact that the encryption algorithm is known to be based on the Advanced Encryption Standard (AES), the exact procedure is not publicly known. However, there is a much simpler method to restore access the document within a LiveCycle client software. It is possible to manipulate a policy (e.g. extend the validity period of the policy) after the initial release of a document. LiveCycle policies are expressed in PDRL and stored in hexadecimal representation in the Table `ecdpoli-`

	RMS	LiveCycle
key	server's private key	document key
storage location	server: database (`dbo.DRMS_Licensor PrivateKey`) or secure hardware device	server: database (Table `edclicenseentity`)
decrypts	content key inside a Publishing License and therefore every document of the RMS environment	specific document

Table 1: Cryptographic keys.

`cyxmlentity`. An investigator can modify the policy directly in the database or in the LiveCycle web application.

5.2 Anti-forensics

If a company wants to prevent forensic investigations (e.g. to avoid costly investigations demanded by public authorities), access to the content key that decrypts the corresponding document has to be prevented. Although both, RMS and LiveCycle, can set validity periods for licenses as well as revoke them, further actions are required to securely deny access to protected documents in the presence of an investigator with access to the rights management server. The concept of revocation is used to prevent access to a specific document (e.g. because it is outdated), or to remove a specific client machine from the rights management system (e.g. because it is not trustful anymore). Thus, revocation withdraws access rights client-side, while the server has still access to the decryption key of the file. Table 1 shows the server-side cryptographic keys in RMS and LiveCycle that can provide access to encrypted documents. For secure access denial, these keys have to be destroyed. The following subsections explain secure access denial in RMS as well as LiveCycle and discuss the possible limitations.

5.2.1 *Rights Management Services*

In RMS, denying access to a specific protected document is not possible without replacing the server's private key, because a Publishing License that contains the encrypted content key is attached to every copy of the document and it is not feasible to delete them all. To be able to securely deny access to RMS protected content, the company has to destroy the key that decrypts the document key in the Publishing License and that is the private key of the server. However, without the server's private key the RMS system would not able to decrypt any other content key anymore, because they are all encrypted with the same server key. Every protected document of the RMS environment would be locked. Revocation and limited validity periods do not solve this problem, because the server's key can always decrypt the content key regardless of whether the license is valid. Therefore, denying access to a single document without affecting other documents is not possible in RMS. Figure 3 shows the limitations of RMS's revocation concept, where only re-encryption of all documents except from the revoked one with a new server key would would securely deny access to it.

5.2.2 *LiveCycle Rights Management*

In contrast to RMS, LiveCycle Rights Management stores content keys in a database on the server (Table `edclicenseentity`), where they are mapped to the encrypted documents. Deleting only these keys effectively prevents the decryption of the corresponding documents without affecting

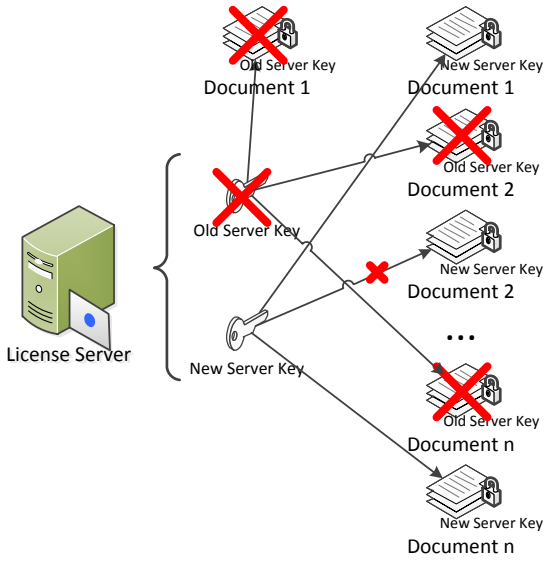

Figure 3: Limitations of permanent destruction of a document in RMS.

other content of the ERM environment. However, it has to be kept in mind that database forensic methods can recover deleted keys from the database, thus additional measures have to be taken.

Validity period settings and document revocation do not prevent digital forensics. The validity of a policy can be extended and revocation can be undone by manipulating the Table `ecdrevocation` of the LiveCycle database. Listings 1 and 2 show the differences between an active and an inactive revocation state for a document. `Rovokestate` is set from `1` to `2` when the document is revoked. `Revokestate` `3` indicates that the corresponding document was unrevoked. Note that the key is still stored with a revoked license. Figure 4 shows system-wide revocation in Adobe LiveCycle.

6. DATABASE FORENSICS

The conceptual architecture of ERM systems affects the scope of forensic investigations as well as their prevention. The location of cryptographic keys, the support for offline usage of protected documents, server-side decryption of content for indexing and malware detection, and other aspects of the general architecture of the ERM system are key parameters for the feasibility of specific forensic methods within an ERM system. However, our research has shown, that the actual possibilities for forensic investigations also heavily depend on the technical implementation of these sys-

```
mysql> select * from edcrevokationentity;
+--------------------------------------+----------------+--------------------------------------+----------+---------------+--------------+-----------+...
| id                                   | sequencenumber | licenseid                            | reasonid | revokemessage | revokestate | revokeurl |
+--------------------------------------+----------------+--------------------------------------+----------+---------------+--------------+-----------+...
| A1440F59-F911-082F-D983-0000BD120003 |       16777239 | A1441D8E-D612-01AB-E379-0000EF2312AB |        1 | NULL          |           2 |           |
+--------------------------------------+----------------+--------------------------------------+----------+---------------+--------------+-----------+...
```

Listing 1: Revoked document in LiveCycle

```
mysql> select * from edcrevokationentity;
+--------------------------------------+----------------+--------------------------------------+----------+---------------+--------------+-----------+...
| id                                   | sequencenumber | licenseid                            | reasonid | revokemessage | revokestate | revokeurl |
+--------------------------------------+----------------+--------------------------------------+----------+---------------+--------------+-----------+...
| A1440F59-F911-082F-D983-0000BD120003 |       16777240 | A1441D8E-D612-01AB-E379-0000EF2312AB |        0 | NULL          |           3 | NULL      |
+--------------------------------------+----------------+--------------------------------------+----------+---------------+--------------+-----------+...
```

Listing 2: Unrevoked document in LiveCycle

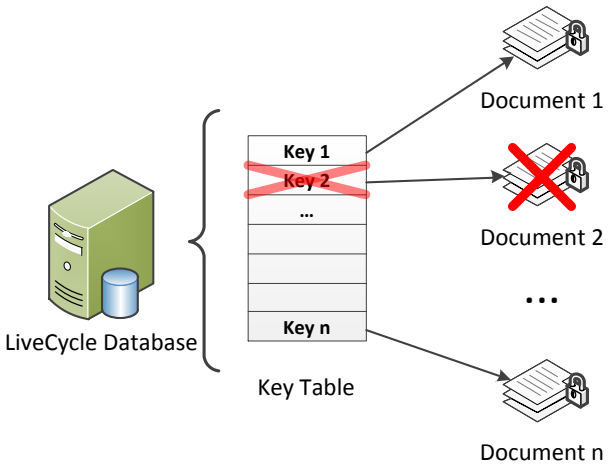

Figure 4: Permanent destruction of a document in LiveCycle.

```
0100B20 20202020 20202020 203C4E41 4D453E4D          <NAME>M
0100B30 61636869 6E653C2F 4E414D45 3E202020 achine</NAME>
0100B40 20202020 20203C2F 4F424A45 43543E20       </OBJECT>
0100B50 20202020 20202020 3C454E41 424C494E         <ENABLIN
0100B60 47424954 53207479 70653D22 7365616C GBITS type="seal
0100B70 65642D6B 6579223E 20202020 20202020 ed-key">
0100B80 2020203C 56414C55 4520656E 636F6469    <VALUE encodi
0100B90 6E673D22 62617365 36342220 73697A65 ng="base64" size
0100BA0 3D223631 3434223E 6C785A33 41637338 ="6144">lxZ3Acs8
0100BB0 2F69666B 6C557661 5547594B 656B614D /ifklUvaUGYKekaM
0100BC0 73663035 4A78776D 696D4C6C 57793374 sf05JxwmimLlWy3t
0100BD0 73584D41 66787276 47762B57 37413945 sXMAfxrvGv+W7A9E
0100BE0 74497554 6372334A 344D6B61 6F346B4C tIuTcr3J4Mkao4kL
```

Figure 5: Excerpt from an InnoDB data file containing a deleted rights management license (including a cryptographic key).

tems. Both RMS and LiveCycle Rights Management use databases for storing cryptographic keys. RMS is built on a Microsoft SQL Server platform, LiveCycle can use many different database management systems. In the following, we show, based on the example of InnoDB, how database forensic techniques can be used to recover ERM related data from databases.

6.1 MySQL (InnoDB storage engine)

The default configuration of LiveCycle stores policies as well as content keys in a MySQL database with InnoDB storage engine [1]. As shown by Fruehwirt et al. [13] InnoDB does not physically delete data from the storage files, but instead flags them as deleted. Therefore, recovery is possible as long as deleted rows have not been overwritten by new data.

We evaluated the persistence of specific records in an InnoDB data file and reasoned that InnoDB does not replace deleted rows very quickly. In our evaluation, records even remained in the data files after the creation of millions of new records. Of course, the stability of data files heavily depends on the structure of the table and InnoDB also performs table optimization that removes deleted records and defragments the content of the database storage file, but our evaluation still showed that cryptographic keys can be found in data files of databases even after removal from the database management system. Figure 5 shows an excerpt from an InnoDB data file containing a previously deleted license. In a forensic investigation, even a revoked key could be easily extracted from the data file.

Not only the data files of InnoDB are a valuable source for forensic investigations, but also the log files. Fruewirth et al. [14] have shown the practicability of analyzing InnoDB log files for database forensic purposes. The InnoDB redo logs store a chronologic history of SQL statements that were executed on the database. Thus, it is possible to recover deleted records and field data, and even rollback to previous versions of field data as long as the used areas of the file have not been overwritten by other data. Log files are created with a fixed file size and unused space is padded with zero bytes. New log data is written to these empty areas until the end of the file is reached. Only then the oldest log data is overwritten by new one. In many cases, a complete recovery of deleted keys is possible.

Moreover, other scenarios for the application of database forensics for ERM are imaginable, too. Database forensics can reveal details about policy modifications. Policies in LiveCycle can change, even after the initial release of a document. However, LiveCycle only shows the current version of a policy. An investigator could restore previous versions of a policy by extracting them from InnoDB log or data files, and reveal who had access to protected documents in the past. For example, after the exposure of sensitive data, a suspi-

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <Policy PolicyCreationTime="2010-02-12T14:21:53.422+00:00" PolicyDescription="" PolicyID="1281F032-F911-102C-A993-0000AC100103" PolicyInstanceVersion="1"
          PolicyName="p1" PolicySchemaVersion="1.0" PolicyType="2" xmlns="http://www.adobe.com/schema/1.0/pdrl">
    <AuditSettings isTracked="true"/><PolicyEntry>
    <ns1:Permission Access="ALLOW" PermissionName="com.adobe.aps.onlineOpen" xmlns="http://www.adobe.com/schema/1.0/pdrl-ex" xmlns:ns1="http://www.adobe.com/schema
          /1.0/pdrl"/>
    <ns2:Permission Access="ALLOW" PermissionName="com.adobe.aps.pdf.copy" xmlns="http://www.adobe.com/schema/1.0/pdrl-ex" xmlns:ns2="http://www.adobe.com/schema/1.0/
          pdrl"/>
    <ns3:Permission Access="ALLOW" PermissionName="com.adobe.aps.revoke"
    [...]
```

Listing 3: Old Version of a LiveCycle Policy

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <Policy PolicyCreationTime="2010-02-12T14:21:53.422+00:00" PolicyDescription="" PolicyID="1281F032-F911-102C-A993-0000AC100103" PolicyInstanceVersion="2"
          PolicyName="p1" PolicySchemaVersion="1.0" PolicyType="2" xmlns="http://www.adobe.com/schema/1.0/pdrl">
    <AuditSettings isTracked="true"/><PolicyEntry>
    <ns1:Permission Access="ALLOW" PermissionName="com.adobe.aps.onlineOpen" xmlns="http://www.adobe.com/schema/1.0/pdrl-ex" xmlns:ns1="http://www.adobe.com/schema
          /1.0/pdrl"/>
    <ns2:Permission Access="ALLOW" PermissionName="com.adobe.aps.revoke"
    [...]
```

Listing 4: New Version of a LiveCycle Policy

cious employee could deny ever having had access to that document. The current version of the policy may prove this claim, however, the policy could have changed. Database forensics may reveal that in the past the employee had access to the document and therefore had the potential opportunity for the security breach. Listings 3 and 4 show the beginning of two versions of a LiveCycle policy. The value of `PolicyInstanceVersion` brings them in a chronological order. The second policy (`PolicyInstanceVersion 2`) updated the first one (`PolicyInstanceVersion 1`) in the way that the copy-flag was removed. This example shows that even if the ERM system does not maintain logging information or someone has deleted the log files, forensic investigations in the database system can help to restore previous versions of ERM storage data in order to either prove or deny some evidence.

6.2 MS SQL Server

The logging database of RMS records every client request (successful or not), containing among others the username, the IP address of the client machine, and a timestamp. Thus, it is an important source of information for forensic investigations. If RMS logging data was deleted, database forensics can be used to restore it (e.g. from the Transaction Logs of the MS SQL Server).

7. GUIDELINES FOR ERM FORENSICS

This section on the one hand describes our proposed application-specific guidelines for forensic investigations in ERM environments and on the other hand shows methods for denying access to out-dated documents in RMS and LiveCycle Rights Management (anti forensics).

7.1 Towards Forensic Investigations

1. Follow general guidelines for forensic investigations (e.g. NIST SP800-86) and create a snapshot of the filesystem to preserve its state.

2. Determine the deployed ERM system by examining protected documents. RMS attaches a Publishing License to the file that is expressed in XrML, an XML based Rights Expression Language. Listing 5 shows a typical beginning of an RMS Publishing License.

```
    <?xml version="1.0"?><XrML version="1.2" xmlns=""><BODY type="
          Microsoft Rights Label" version="3.0"><ISSUEDTIME>2010-01-29
          T00:01</ISSUEDTIME><DESCRIPTOR><OBJECT><ID type="MS-GUID">{
          C2572064-A6D7-4BFF-B088-F0A426412D96}</ID>
    [...]
```

Listing 5: RMS Publishing License

LiveCycle adds an encryption dictionary with characteristic parameters to protected PDF files. The absence of a user and an owner password (PDF options /O and /U in the encryption dictionary) indicates the protection of the file with a client/server based ERM system (e.g. LiveCycle Rights Management) instead of the client-only PDF restriction system of the PDF standard [3]. Listing 6 shows the beginning of the encryption dictionary of a LiveCycle protected PDF file.

```
25 0 obj
<</CF<</DefaultCryptFilter<</CFM/AESV2/Length 128>>>>/EDCData(
      TVVhckZXcWhlSGdVOE1maUN6eH
[...]
endobj
```

Listing 6: Encryption dictionary of PDF

3. Check the existence of offline use licenses by trying to open the protected document in the ERM client application.

4. If a valid license exists, access to protected content does not depend on cryptographic keys that are stored on the server. The investigator can access the documents without collecting any additional keys.

5. If an offline licenses does not exist:
   Rights Management Services: Get access to the RMS server system and use the server key to decrypt the document key, which is stored in the publishing license of the protected document. Alternatively, the superuser can be activated or an RMS server application can be used to decrypt the content.
   LiveCycle Rights Management: Get access to the LiveCycle server. If the protected file is still registered with the LiveCycle server, create a valid policy for the protected document in the database in order to get access to the content within a LiveCycle client application

(e.g. Adobe Reader). It no policy exists anymore, use database forensic techniques to recover it (see Section 2.3).

7.2 Secure Access Denying

1. Determine the deployed ERM system, the location of decryption keys and offline licenses. Secure access denying is only possible with disabled offline use as it causes some latency until server-side measures take effect on the client.

2. *Rights Management Services:* RMS uses one single key (server's key) for content key encryption. It is not possible to deny access to a single document without locking other files too or re-encrypting everything with a new server key.
   LiveCycle Rights Management: In LiveCycle document keys can be found in the database on the server (Table `edclicenseentity`) where they are mapped to the encrypted documents. Deleting these keys effectively prevents the decryption of the corresponding documents. However, counter measures against database forensic methods have to be applied. This includes the analysis of data as well as log files of the database.

8. CONCLUSION

In this paper, we have discussed requirements and approaches for forensic investigations of ERM protected documents. Today's ERM products have very different architectures and therefore *guidelines to forensics* have to be developed for each ERM system separately. In general, there are three ways of accessing ERM protected content. First, forensic investigations can be based on offline licenses that allow an investigator to access encrypted content without obtaining cryptographic keys from the licensing server. However, valid offline licenses are not always available. Second, the decryption of ERM protected content can also be achieved by obtaining the required keys from the server, either the content keys (LiveCycle Rights Management) or the private key of the server (RMS). Third, in LiveCycle policies are not fixed when they are linked to a document and can be modified to restore access to encrypted content.

Key destruction to avoid costly investigations demanded by public authorities can by done in Adobe LiveCylce Rights Management by deleting content keys from the central LiveCycle database on the server while considering possible traces in database files. In RMS, however, it is not possible to deny access to a file independently from other RMS protected content, as a server-key encrypted version of the document key is attached to the protected file. The destruction of the server's private key denies access to all documents that are protected by that RMS environment and is therefore not feasible.

Databases are heavily used by ERM systems to store keys and other data that is relevant in forensic investigations. In this paper, we have shown that the data and log files of the InnoDB storage engine of MySQL store sufficient data to reveal older versions of LiveCycle policies and even allow the recovery of deleted cryptographic keys. Thus, forensic techniques in the context of Enterprise Rights Management systems can be tremendously enhanced by database forensic methods.

In future research, more efficient methods for reliable key destruction in ERM environments should be developed. This might require a radical redesign of today's ERM architectures as they were not developed with this requirement in mind and thus lack secure key destruction workflows.

Acknowledgments

The research was funded by COMET K1 and grant 826461 (FIT-IT) by the FFG - Austrian Research Promotion Agency.

9. REFERENCES

[1] MySQL 5.1 reference manual. 2007.
[2] *Microsoft TechNet: Windows Rights Management Services (RMS)*, 2009.
[3] Adobe Systems Inc. *PDF Reference: Version 1.7*. 6th edition, 2006.
[4] Adobe Systems Inc. *LiveCycle ES Overview*. 2008.
[5] M. Alawneh and I. Abbadi. Defining and analyzing insiders and their threats in organizations. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, pages 785–794. IEEE, 2011.
[6] A. Arnab and A. Hutchison. Requirement analysis of enterprise drm systems. *Information Security South Africa*, 2005.
[7] D. Bradley and A. Josang. Mesmerize: an open framework for enterprise security management. In *Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation-Volume 32*, pages 37–42. Australian Computer Society, Inc. Darlinghurst, Australia, Australia, 2004.
[8] M. Burmester and J. Mulholland. The advent of trusted computing: implications for digital forensics. In *Proceedings of the 2006 ACM symposium on Applied computing*, page 287. ACM, 2006.
[9] E. Casey and G. Stellatos. The impact of full disk encryption on digital forensics. 2008.
[10] J. Cohen. DRM and Privacy. *Communications of the ACM*, 46(4):46–49, 2003.
[11] I. ContentGuard. eXtensible rights Markup Language (XrML) 2.0 Specification. 2001.
[12] K. Fowler. SQL Server Forensic Analysis. 2008.
[13] P. Fruehwirt, M. Huber, M. Mulazzani, and E. R. Weippl. InnoDB Database Forensics. 2010.
[14] P. Fruehwirt, P. Kieseberg, S. Schrittwieser, M. Huber, and E. Weippl. Innodb database forensics: Reconstructing data manipulation queries from redo logs. In *The Fifth International Workshop on Digital Forensics*, 8 2012.
[15] J. R. Henriques and R. A. andMarco Grimm. Analysis of enterprise rights management solutions for cad data according to the requirements of the automotive industry and a proposal to increase the erm security level. In *Proceeding of ASME 2010 International Mechanical Engineering Congress & Exposition*, Nov 2010.
[16] T. G. H. D. Karen Kent, Suzanne Chevalier. *Guide to Integrating Forensic Techniques into Incident*

Response. National Institute of Standards and Technology, August 2006.

[17] G. Kessler. Anti-Forensics and the Digital Investigator. In *Proceedings of The 5 th Australian Digital Forensics Conference*, page 1. Citeseer, 2007.

[18] J. 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, 2009.

[19] M. Olivier. On metadata context in database forensics. *Digital Investigation*, 5(3-4):115–123, 2009.

[20] J. Park, R. Sandhu, and J. Schifalacqua. Security architectures for controlled digital informationdissemination. In *Computer Security Applications, 2000. ACSAC'00. 16th Annual Conference*, pages 224–233, 2000.

[21] A. Pretschner, M. Hilty, F. Schutz, C. Schaefer, and T. Walter. Usage control enforcement: Present and future. *Security & Privacy, IEEE*, 6(4):44–53, 2008.

[22] S. Sinha. *Data Paladin: An Application Independent Rights Management System*. PhD thesis, Citeseer, 2008.

[23] Verizon RISK Team. Verizon 2012 data breach investigations report. Technical report, 2012.

[24] Y. Yu and T. Chiueh. Enterprise digital rights management: Solutions against information theft by insiders. *Relatório de experiência professional, Experimental Computer Systems Lab, Department of Computer Science of the State University of New York (SUNY), Setembro de*, pages 2003–01, 2004.