Patrick Ausderau

Protection mechanisms for electronic books

Helsinki Metropolia University of Applied Sciences

Master of Engineering

Information Technology

Thesis

26/04/2014

Metropolia

Author(s)	Patrick Ausderau
Title	Protection mechanisms for electronic books
Number of Pages	46 pages + 3 appendices
Date	Saturday 26th April, 2014
Degree	Master of Engineering
Degree Programme	Information Technology
Specialisation option	Mobile Programming
Instructor(s)	Olli Alm, Project Manager Harri Airaksinen, Head of Degree Programme

The research for a new Digital Right Management (DRM) infrastructure for electronic books (e-book) is a task of the eReading work-package from the Next Media project, which goal is to to propose a protection system that could help limiting copyright infringement while allowing fair use of e-books.

This text reviews the existing technologies, like DRM, Watermark and Social DRM. It studies the upcoming Lightweight Content Protection standard (for EPUB). Apart from the technical aspect, the legal, educational and financial lewis to reduce copyright infringement are studied. Web crawler and its implication into finding infringing material on the internet is addition to this work.

The main outputs of this thesis are a deliverable (number 1.3.1.4) for the Next Media project, presentations, seminars and conferences to the Finnish National Library, Next Media project, Finnish Book Publishers Association and Finnish author unions and the participation to the redaction of the DRM recommendations for the Finnish Book Publishers Association members.

As a conclusion, this work recommends not to use DRM to protect e-books but to use either a lighter protection mechanism such as Watermark and Social DRM combination or to be DRM-free. This proposal is followed by two major Finnish publishers who abandon DRM for the lighter protection.

Keywords	Electronic book (e-book), EPUB, PDF, Digital Right Management (DRM), Social DRM, Watermark, Copyright, DRM-free

Licenses

I decided to publish my thesis work under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License because I strongly believe that you as reader deserve the freedom to copy, share and modify this work and if you do modify it, it is fair to give these same permissions to the others. I'm however a bit selfish and do not give you the full freedom with the Non Commercial clause, if you want to do business with my work, I would like to negotiate.

A copy in electronic form of this work can be found in github with the LaTeX source. The repository contains also the sides (under the Creative Commons BY-SA license) of the presentations done during this work.

The program written for this thesis, documented and listed herein is a free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. See `https://www.gnu.org/licenses/gpl.html`

I decided to publish the source code of the application written for this thesis under a free and open source licence. I choose the GPL version 3 for multiple reasons. The first and most important one is that I want to make sure that nobody will ever use that code to build a real DRM; I can not be part of system that will restrict someone else freedom. So with the GPL3, I'm sure that if you build a DRM system with my code, anyone will have the right to publish a modified version that will break such protection mechanism. Among the other reasons, one is about the philosophy of the free and open source software, if you do improve the code, make the community benefit from it. Also by ensuring that the source code can be studied, hopefully it could help people in discovering the upcoming W3C Web Cryptography API standard.

The application is running at `http://users.metropolia.fi/~patricka/wcrypt-book/genkey_and_export.html` and a copy of the source code is available in github. This application encrypt the public domain EPUB book Moby-Dick; or, The Whale by Herman Melville for the sole purpose of demonstration; the e-book originate from Project Gutenberg `http://www.gutenberg.org/ebooks/2701` where it is freely available.

Acknowledgement

I dedicate this work to my wife Laura for her support and unconditional love and to my nieces and nephews, as I wish they will grow-up in a world where freedom and privacy respect will still make sense.

All my considerations to Olli Alm and Harri Airaksinen, who believed in me when I was in doubt and pushed me toward the end of this work. Thank you for your comments and proof reading that helped me a lot.

A special thanks to Anna Galore, Marc Levy, Philippe Aigrain and Richard M. Stallman, as I strongly appreciated a personal answer from authors.

To the Finnish Book Publishers Association, I wish full success in selling DRM-free e-books. I hope they will go a step forward by removing Watermark and Social DRM in the future.

To Florence Bettschart, thanks a lot for your discussion about the work of the Swiss copyright governmental group, I wish you and the group a full success.

Thank you to Nick Van den Bleeken for the web crypto samples and PolyCrypt team. Without your work, I would not been able to test the maybe future Web Cryptographic API standard.

This work was written using LaTeX, served as a basis and benefit from the "Metropolia Thesis LaTeX Template" (CC-BY) with the help of Panu Leppäniemi and Patrik Luoto. Thank you two.

Coding and documenting was a pleasure with the help of `devdocs.io`, `php.net`, `stackoverflow.com` and `tex.stackexchange.com` communities. Thanks to them.

Contents

Appendices

Appendix 1 Web Crawler

Appendix 2 Use Cases

Appendix 3 Source Code

1 Introduction

From the Universal Declaration of Human Rights:

> (1) Everyone has the right freely to participate in the cultural life of the community, to enjoy the arts and to share in scientific advancement and its benefits.
> (2) Everyone has the right to the protection of the moral and material interests resulting from any scientific, literary or artistic production of which he is the author. [1, article 27]

Since the computer era and with the universal adoption of the internet, it became easy to copy and share high value works without loss of quality, making possible for everyone to participate to the dissemination of the culture. Unfortunately, a lot of file sharing is done without respecting the copyright, such as without author consent and thus make the authors and publishers fear for their moral and material interests by publishing their works on-line.

This document describes different control strategies and the laws associated to them. In security and access control domain [2], the protection and identification mechanisms can be divided into two categories:

- **A priori** where technologies are used to impose how the digital content can be consumed. The **Digital Right Management (DRM)** and the **Password Protection** belong to this category.
- **A posteriori** where technologies are used to identify a file and retrieve its owner; but do not impose restriction on usage. **Watermarking**, **Fingerprinting** and **Social DRM** belong to this category.

The third option is to have **no technical protection** which only rely on the copyright law or **licence**.

The practical work tests the upcoming W3C[1] Web Cryptography API[2] standard. It tries a possible use case for protecting an e-book while answering one of the research question for a DRM system with better interoperability.

[1] The World Wide Web Consortium (W3C) is an international community that develops open standards to ensure the long-term growth of the Web (from: `http://www.w3.org`).

[2] Application Programming Interface

This research focuses on the protection of **electronic book (e-book)**. Depending on the sources, the definition of e-book refers to the file as the container for the texts, images, formatting, etc. in a computer readable format, or the software needed to display the book on computer screen, or the specialized reader device. In this text, when not specified, the term e-book will refer to the file.

To be Finally mentioned, this work[3] was mandated by the eReading work-package of the TEKES funded Next Media[4] TIVI research program project. It was motivated with the postulate that the current DRM systems "restrict the fair use that allows books to be read or to be borrowed by anybody"[5]. It opened research questions such as, if the printed world could be imitated where "a book can be read by anybody, but only one user at a time" or if "the need for authentication" could be minimised, etc.

[3] The title of the task originally was "New DRM infrastructure for books"

[4] `http://www.nextmedia.fi`

[5] This document is confidential to consortium members. No public source available. The short quotes with courtesy of my thesis instructor.

2 Protection and Identification

2.1 Digital Rights Management (DRM) and Password Protection

A definition of Digital Rights Management (DRM) given by Satish [3, p. 4] is “a technology that restricts usage and access to the content” that gives to the author a way “to secure and distribute [its work] only to the people whom it is meant for”.

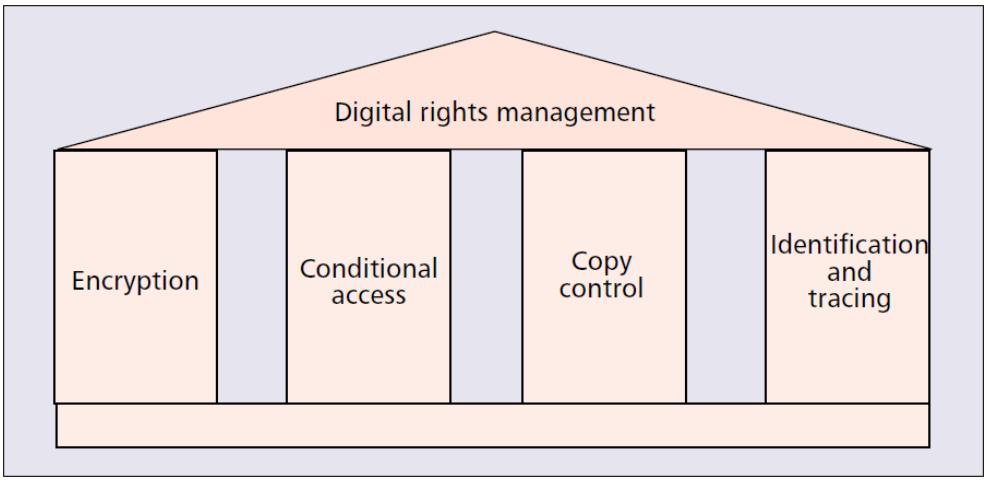

Figure 1: The DRM pillar model (Copied from Hartung and Ramme (2000) [4]).

At large, as shown in figure 1, the DRM groups many technologies that will typically:

- **Copy control**: Authenticate and identify the user and/or its devices to make sure the content is accessed and consumed by authorized person.
- **Encryption**: Protect and encrypt the content to avoid third party access.
- **Identification and tracing**: Make each copy unique in order to identify them and track their usage.
- **Conditional access**: Define and enforce the license such as if the user can copy the text, print it, limit the number of device that can access it, set the access duration (e.g. for library lending), if the copy can be shared and how many times, etc.

A typical DRM content protection process is presented in figure 2. On the provider side, once the user purchases content, a key is associated to the user identity. The key will be used to sign the file to make the copy unique and protect the content with encryption

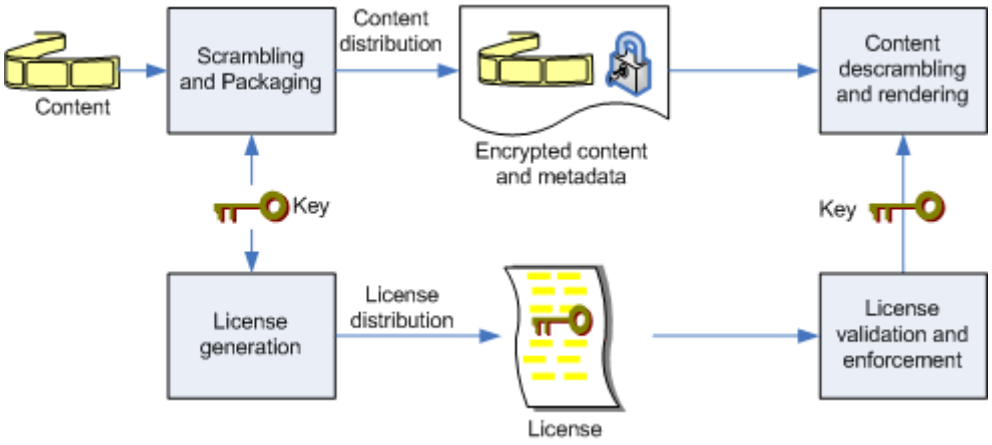

Figure 2: A generic DRM system (Copied from Boštjan Marušič et al. (2005) [5]).

(figure 2: *Scrambling and Packaging*). In parallel, the usage contract (*License generation*) is issued. After these steps, the encrypted content and the license are delivered to the user. On the user device, if the reading application is legitimate and will satisfy and enforce the license (*License validation and enforcement*), the key is used to decrypt (*Content descrambling*) the file and the content is now displayed to the user screen (*Content rendering*).

Even with strong encryption and full control on the software and hardware, there will always be ways to break any technical measure. To hamper software cracking to be implemented, distributed and used for protection removal, new articles have been added in the international copyright treaty to make cracking technologies illegal.

Daniels [6] use the term "**hard DRM**" to describe the DRM system that restricts physical access and usage of the file. Rosenblatt [7] describes it as "**heavyweight DRM**". With that definition, he introduces the "**lightweight DRM**" concept which is derived from the password-based encryption. With the **password protection** approach, there is no need to call back a distant server for authentication since the protection is embedded within the file. The key to decrypt the file is a password that the user has to enter before he can access the content. Like with hard DRM, this technique can also manage the authorization such as restricting printing or limiting copy of content. Usually, as there is no communication with a authentication server, this technology will not block concurrent access nor limit the number of devices which host copies of the file.

2.2 Social DRM, Watermark and Fingerprint

Watermarking and **fingerprinting** are technologies used to identify a file. They originate from the information hiding techniques as presented in figure 4. Petitcolas et al. [8] classify them as robust copyright marking if it is infeasible to remove them or if removing them will destroy the file at the same time. Otherwise, they classify them as fragile if the copyright marking is destroyed when the file is modified.

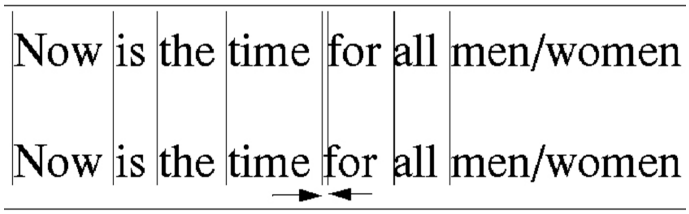

Figure 3: Word-shift watermark example (Copied from Insup Lee (2001) [9]).

Watermark consists of data representing the user that are inserted and hidden inside the file bytes. For example, they can be imperceptible changes to the content such as spaces and invisible characters at the end of the chapter, modified pixels of the font, changes in the line/text spacing like in figure 3. One important aspect is the number of different watermark techniques used, so the cracker can never be sure to have defeated them all.

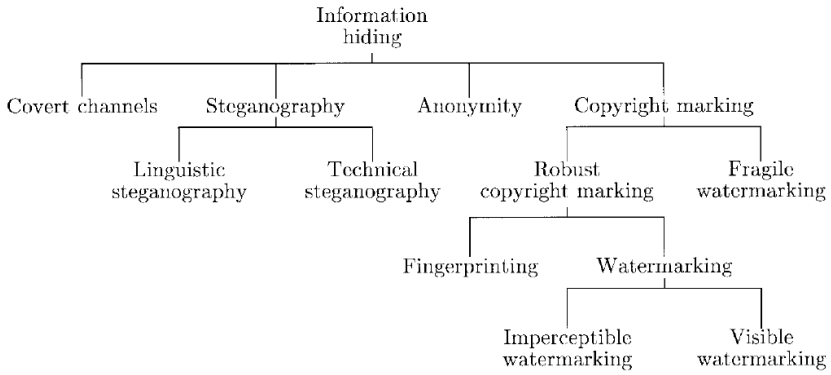

Figure 4: A classification of information-hiding techniques (Copied from Petitcolas et al. (1999) [8]).

Fingerprint is a technique only to identify a copy of a file by generating a cryptographic hash. With Watermark, these techniques can be used for a posteriori protection. Because identification does not control the permission, the user can do whatever he wants with the file. The idea is, if illegal content appears for example on a peer-to-peer network, to be able to find the user who did it.

A similar approach to watermarking is **social DRM**. The difference is that the information

about the user is not hidden but visible. It can be for example part of meta-data or simply visible in plain text e.g. in the footer of every page. In figure 5, the PackaDRM[1] presents the user information in the 'terms and conditions' section of the e-book. The idea is to let the user do what he want but putting social pressure on him not to do anything illegal.

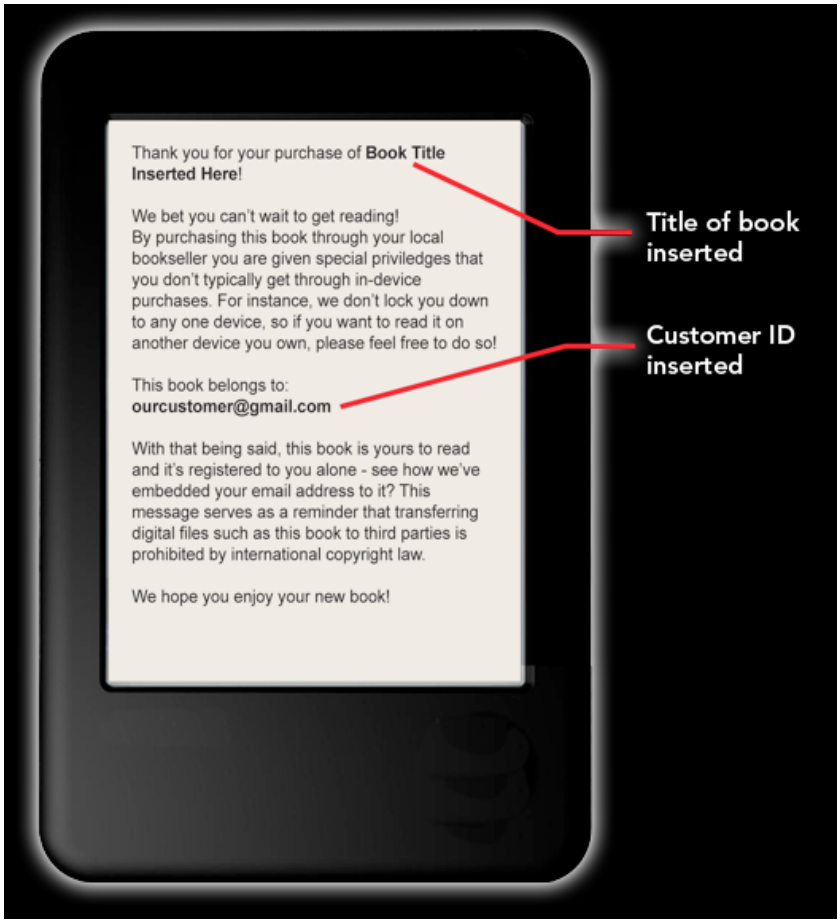

Figure 5: Social DRM (Copied from Kevin Franco (2012) [10]).

The watermarking and all other identification techniques can and are often used in combination with the DRM. For example, Rosenblatt [11] specifies that the Lightweight DRM "is intended to be complementary to watermarking". Hartung and Ramme [4] state "watermarking […] is only useful as a system component, with the most important application being DRM and copyright protection in general". Daniels [12] refers to the combination of social DRM and watermark as "**soft DRM**" because it does not restrict usage unlike DRM and still help in reducing copyright infringement, so "better than no DRM at all".

The task to find illegal copies of the work, whether it is protected or not, is usually devoted to crawlers as described in appendix 1.

[1] Packaged Digital Rights Messaging or PackaDRM™ (`http://www.enthrill.com/pubtech`)

2.3 Law

In order to have effective protection and to make the crack illegal, the World Intellectual Property Organization[2] (WIPO) has added measures to protect DRM in its WIPO Copyright Treaty (WCT) of 1996 [13, articles 11 and 12]. These articles are enacted in the European Union (EU) in the DIRECTIVE 2001/29/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 22 May 2001 on the harmonisation of certain aspects of copyright and related rights in the information society [14, chapter III] implemented in Finland under the COPYRIGHT LEGISLATION of 2010 [15, chapter 5a]. Under the 89 WCT contracting parties such as China, Japan, Canada, Russian Federation, let mention the United States of America who enacted this treaty under the Digital Millennium Copyright Act of 1998 [16, section 103]. Notice also that countries such as India or Brazil are not signatories of the WCT.

In the WCT [13, article 11], EU DIRECTIVE 2001/29/EC [14, article 6] and Finnish COPYRIGHT ACT [15, sections 50a and 50b], removing, circumventing any effective technological measures (such as hard and lightweight DRM), providing or producing tools to remove or circumvent it are prohibited. Finnish COPYRIGHT ACT specifies that anyone who circumvent DRM, produce or distribute tool or device for circumventing a technological measure “shall be sentenced […] to a fine for a violation of a technological measure” [15, section 56e] and “shall be obliged to pay the author damages for any loss, mental suffering or other detriment caused by the crime” [15, section 57(3)].

The WCT [13, article 12], EU DIRECTIVE 2001/29/EC [14, article 7] and Finnish COPYRIGHT ACT [15, section 50d] make illegal to remove or alter right management information. They also make illegal to distribute, import for distribution, broadcast, communicate or make available to the public a work from which electronic rights-management information has been removed or altered. The Finnish COPYRIGHT ACT [15, chapter 7, sections 56f and 57(3)] specifies the same type of punishment as for circumventing a technological measure.

The the WCT [13, article 12(2)] defines right management information as “information

[2] The WIPO is an United Nations (UN) agency established in 1967 which is responsible for the use of intellectual property (such as copyright, patents, trademarks, etc.) and count 185 member states (from: `http://www.wipo.int/about-wipo/en/`).

which identifies the work, the author of the work, the owner of any right in the work, or information about the terms and conditions of use of the work, and any numbers or codes that represent such information". To make sure that the social DRM belong to that definition, part of the social DRM should be placed in under the copyright notice or under the terms and conditions section of the e-book. Another solution could also be to have a book ID (which identifies the work) in different places in the e-book and to store that information with user purchase detail in reseller database.

According to Rosenblatt [17], watermarking does not qualify as right management information. Even if the terms and conditions would state that removing the watermark is forbidden, "it is possible that copyright law may prevail over such terms; this is a legal gray area". Another problem with watermark that he raises is that "a lightweight DRM that is susceptible to one-click crack has more protection than, say, a watermark removal tool" [7]. A watermark removal tool is legal because the watermark cannot be considered as an effective technological measure, while a DRM removal tool is illegal. This incomplete law protection could be one more argument to use watermark only in combination with other protection techniques or as Hartung and Ramme state "watermarking is not a standalone technology"[4].

Without technical protection, any work in electronic form still benefit from the copyright law. The author will continue to profit from the exclusive rights such as the protection for the usage and distribution of its work. With the ease of copying, modifying and sharing electronic content compare to the physical world, instead of fighting against the users, the author can surrender some of its rights for example allow non-commercial sharing and remixing. This waiving could be done for example by using a copyright license such as one of the Creative Commons[3] licenses.

[3] http://creativecommons.org/

3 Electronic Book (e-book)

According to Oxford Dictionaries [18], an **electronic book (e-book)** is “an electronic version of a printed book which can be read on a computer or a specifically designed handheld device”. Unfortunately, this definition doesn't consider that some e-books do exist without theirs printed equivalent[1]. The Cambridge Dictionaries [19] makes it clearer by defining it as “a book that is published in electronic form”. They also add a second definition where the e-book term can be used for the hardware specific “electronic device with a screen which allows you to read an electronic book”. Marshall [20, section 2.4] completes the definition by including also the software/application needed to display the book on screen.

About the question of whether or not the physical books should have their electronic equivalent, Marc Levy (a best-selling French author) answered:

> [Speaker:] You, the e-book scares you or seduces you?
> [Marc Levy:] […] I think the e-book is a great opportunity for the book. Nobody can prevent the emergence of the electronic tablets [device]. If on the electronic tablet there is only video games, manga and television films, it means that on the entertainment platform there will be no books and within 20 years nobody will read any more.[2] [21, from 13:45 to 14:28]

3.1 E-book File Formats and Protection

For e-book there are different file formats available with their qualities and limitations.

The most basic file format is the plain text which has the advantage of being universal which means that it can be read on any operating system (OS), even in command line environment. They also can be compressed with an external tool for transport. This format was designed to display text only and do not offer formatting such as bold or italic text, makes impossible to insert images or interactive links, etc. And finally it offers no support for content protection; with administrator privilege, the end user can easily copy the full

[1] such as Mile 81 from Stephen King `http://www.stephenking.com/promo/mile_81/`.
[2] My translation

text and modify it.

In 2008, after seventeen years of existence, the portable document format (PDF) became a standard. It is a popular format on the internet for document exchange. For e-book, according to a 2012 French survey [22, p. 19], it is the preferred format for 53% of the readers. By being a standard, every operating system can have application to display and print files in this format. PDF format was designed to be page oriented where the electronic version is the same as its print equivalent (this is sometimes seen as a problem, especially on small screen) but supports also re-flow feature. At the protection level, the author of the file can allow or deny the printing, copying of content, page extraction, etc. and the file can be secured by password or be electronically signed [23].

Another open standard file format widely use for e-book is EPUB (short for electronic publication) standardized in 2007 which is the successor of the Open eBook (OEB) format of 1999 [24]. As of 2012 its latest version is EPUB 3.

EPUB format uses web standards:

- Hyper Text Markup Language (XHTML (in EPUB 2) or HTML 5 (in EPUB 3)) files representing the text and structure.
- Cascading Style Sheet (CSS) for the formatting.
- eXtensible Markup Language (XML) for the navigation.
- Additionally images, sounds, videos, etc.

all together compressed into the EPUB file. This native compression is advantageous for the device disk usage and for faster transmission. The other big difference with the PDF is the use of the dynamic layout and pagination of the HTML so the text will be adapted on the fly to the display area and the user preferred font size [25].

The EPUB file format offers a protection layer but “does not specify a required format for DRM” [26, section 2.5.5]; so the choice is in the hands of the vendor. As example, the Adobe DRM can be use to protect EPUB files. Actually, the International Digital Publishing Forum (IDPF)[3] is proposing “requirements for a potential content protection scheme for EPUB” [27], the lightweight DRM.

In 2000 Mobipocket.com developed its own e-book file format as part of their MobiPocket

[3] IDPF is the organization responsible to maintain the EPUB standard.

Reader application. This format, like ePub, is based on the Open eBook (OEB) specifications. The file can be natively secured with their solution but also left unencrypted. Mobipocket became part of Amazon in 2005 [28].

After Amazon acquisition, they build their own protection on top of the Mobipocket format (with a .azw extension). This has made the future of the protected .mobi e-book unclear [29] while unprotected .mobi files can still be read from Amazon Kindle devices as well as many reading applications. Amazon also announced the creation of their new Kindle Format 8 based on HTML 5 and CSS 3. Like the .azw, it can be protected with Amazon DRM. This format should continue to be backward compatible with Mobipocket. The Amazon e-books file format can be read from Amazon Kindle devices or with Amazon OS-specific Reading apps[4] [30].

Apple also has its own e-book file format (iBooks) based on EPUB 3 but with their own mimetype and proprietary CSS extensions [31]. The e-books in this format can be protected by Apple DRM and if obtained through their store, it will anyway be watermarked. This format is designed to be only readable with the iBooks application[5] and only on Apple mobile devices. For an author/publisher, the only way to sell e-books in this format is through the Apple store [32].

3.2 E-book Applications

For e-book, at the application level, there are three categories of software. First to create an e-book, even if some writers will still hand or typewriting their text, it is common that they use a computer with a basic text editor[6] or a word processor[7]. The editor then formats the text with a typesetting[8] or a publishing[9] or a WYSIWYG[10] editing[11] software to generate the e-book file and its metadata.

[4] `http://www.amazon.com/gp/feature.html?ie=UTF8&docId=1000493771`

[5] `https://itunes.apple.com/us/app/ibooks/id364709193?mt=8`

[6] such as notepad

[7] such as Microsoft Word (`http://office.microsoft.com/en-us/word/`) or OpenOffice.org Writer (`http://www.openoffice.org/product/writer.html`)

[8] such as LaTeX(`http://www.latex-project.org/`)

[9] such as Adobe InDesign (`http://www.adobe.com/products/indesign.html`)

[10] WYSIWYG stands for What You See Is What You Get

[11] such as Sigil (`http://code.google.com/p/sigil/`)

Nowadays, some vendors propose the same e-book in different file formats to offer the reader with a greater choice of devices, operating systems and reading applications where the e-book can be read. While some of the edition software can generate the e-book in different files formats, it also can be done with a conversion software[12]. Sometimes, such conversion is done by the reader himself if he cannot read his e-book on a specific device or with his preferred reading application.

Finally, to read the e-book file, the end user will need a software that will display the e-book on the device screen. Marshall [20, chapters 3, 4 and 5] state that these reading applications should at least provide the same functionalities as what can be achieved with a physical book, such as:

- Navigating, moving to the next/previous page, going to a specific page/chapter, etc.
- Keeping the user oriented, where he is in the book and how much left (page number out of the total).
- Annotating, that can be simple as a bookmark, line mark that the reader can use to quickly navigate to a specific part of the e-book (for example to remember where he stopped reading to restart reading from there); highlighting; a comment about a sentence, paragraph, chapter, the all e-book; a summary or a description of the e-book or part of it; etc. [33]
- Printing.
- Clipping, like copy-pasting an extract for quotation, etc.
- Social interaction, like sharing annotations, etc.

She describes these functionalities as the “live-or-die aspects of the electronic book”; not having them “ensures that readers will reject eBooks [in the largest sense (the electronic reading as a all)] out of hands”.

Some others interactions that can be done with a physical book and that should be available with an e-book do not necessary depends on the reading application. In these activities there are the acts of buying or borrowing, some parts of the social reading such as sharing quotes and comments and the actions of offering, lending, sharing, swapping, crossing or reselling an e-book.

In addition to the printing equivalent, the electronic book application offers extra features. Marshall [20, chapter 7] gives examples like:

[12] such as Calibre (`http://calibre-ebook.com/`) or Pandoc (`http://johnmacfarlane.net/pandoc/`)

- Searching, for example entering a keyword that will indicates the location of all the occurrences in the text and providing links to the found results
- Linking that can be used to facilitate the navigation within the book, like clicking on one item in the table contents to move to a specific chapter or from a searched result, etc. And can also be used to refer to external resources, for example to a web page that contains extra information, etc.
- The possibility to automatically translate the text into another language.

She also points the advantage that with a devices which can store many files, the reading application could offer searching facilities at the collection level and the use of metadata to better organize and find content within the library. The metadata can be from the book itself (like title, author, etc.) or user based (both implicit (what was read and when) and intentional (annotations)).

With new technologies, the e-book can be enriched with multimedia elements like sounds or video and thus gives more interaction to the reader. With internet connected devices (if the user store its library on-line and if the reading application is synchronized) the reader can start reading in the train on his mobile phone and continue reading on his computer once at home. Some reading application offers also text to speech feature via a speech synthesizer that reads the text aloud.

With e-book, the user behaviour can more easily be studied with analytical tools. The user can then know when, where and how fast he reads. Such type of information can be of high value for researchers and also for publishers who can study what the user likes most and can use this to propose similar contents to the reader.

Part of the e-book interactions described in this chapter are presented as use cases in annex 2

3.3 E-book Devices

At the hardware level, the minimum required to display an e-book is a device with a screen and input capability. A reader can use a computer, laptop, smart-phone, tablet or dedicated e-reader device. The dedicated e-book reading device offers numerous advantages comparing to the others, such as cheap prices and a good reading ergonomic: adapted

controls, light weight, electronic ink (E-Ink) screen, etc. [34].

E-Ink offers comfort for reading in full sunshine —which can be hard with the other type of devices due to the brightness and contrast of their screen—as well as in the shadow. This screen technology consumes very few battery giving a very long autonomy in weeks compare to days or hours of other mobile devices. Once displayed on E-Ink screen, the text will stay forever while the others types of screen will usually shut down to save energy which can be very annoying for reading. The new E-Ink device, such as the Kindle Paperwhite[13] are back lighted, allowing to read in the night. Older ones require an external source of light. Even if E-Ink prototypes with color have been demonstrated[14], they offer a small palette compare to the full range of colors of the other type of devices. And currently, on the market, most of the e-book reader are still black and white with shade of grey.

Another problem with e-book devices is that they usually only have an indirect internet access trough WLAN[15] or no access at all forcing the user to physically connect the device to a computer for loading the e-book files. This problem also applied to some tablets and laptops, for example if a Wi-Fi[16] access point is not present. However, the connection through the mobile network antenna would require a UMTS/LTE[17] subscription which has a cost.

The advantage of being specific for e-book reading can also be seen as an inconvenient. For example, if the user would need dedicated hardware to perform every other activities, such as watching films, playing games, etc., he would need to travel with a lot of different devices. Another difficulty shared with smart-phone is the small disk capacity of the early versions of these devices that is nowadays increasing and limiting this problem. Except for the mobile phones that are smaller and lighter, the e-reading devices are the most convenient for transport compared with tablet and laptop.

[13] http://www.amazon.com/gp/product/B008GEKXUO/ref=fs_cl

[14] http://www.geek.com/articles/gadgets/e-ink-promises-thinner-lighter-higher-resolution-and-color-e-paper-displays-2011095/

[15] Wireless Local Area Network

[16] Wireless connection

[17] UMTS (Universal Mobile Telecommunications System) and LTE (Long Term Evolution) are the new generation of GSM (Global System for Mobile Communications) cellular networks system and offer access to Internet through it.

4 Existing and Upcoming Technologies

4.1 Existing E-book DRM

Nowadays, there are four major e-book DRM systems in use: Adobe, Amazon, Apple and Marlin Trust Management Organization (MTMO)[1]. Nook DRM system is based on Adobe solution.

Fifth major system, the Microsoft e-book DRM technology cannot be considered in use since it was part of the Microsoft Reader that has been discontinued in August 2012 (see section 6.1 on page 32). Microsoft will probably use their PlayReady[2] technology to protect e-book. Their white paper specifies that “Microsoft PlayReady supports essentially any type of content, including games, images, and ringtones, in addition to music and video.” [35, p. 4]; but don't explicitly have the e-book in the list.

4.1.1 Adobe DRM in Detail

Adobe e-book DRM, namely Adobe Digital Experience Protection Technology (ADEPT), is part of their platform which is centred on the Adobe Content Server (ACS)[3]. ACS serves as both hosting and managing PDF and EPUB e-books distribution and encryption. Finally, to access the protected content, the user will use the Adobe Digital Editions[4] software for Windows or Mac, or an application written with the Adobe Reader Mobile Software Development Kit (SDK)[5] for e-reader device, tablet or smart-phone [36]. The figure 6 on the next page shows the flow of authoring and delivering an e-book with Adobe solution.

In the normal flow (see figure 7 on page 17), the user visits the publisher or retailer web store to buy an e-book. After the purchase confirmation, he will receive an Adobe Content Server Message (.acsm) file and by opening it with the Adobe Digital Editions, the software

[1] http://www.marlin-community.com/technology/how_marlin_works
[2] http://www.microsoft.com/playready/
[3] http://www.adobe.com/products/content-server.html
[4] http://www.adobe.com/products/digital-editions.html
[5] http://www.adobe.com/devnet/readermobile.html

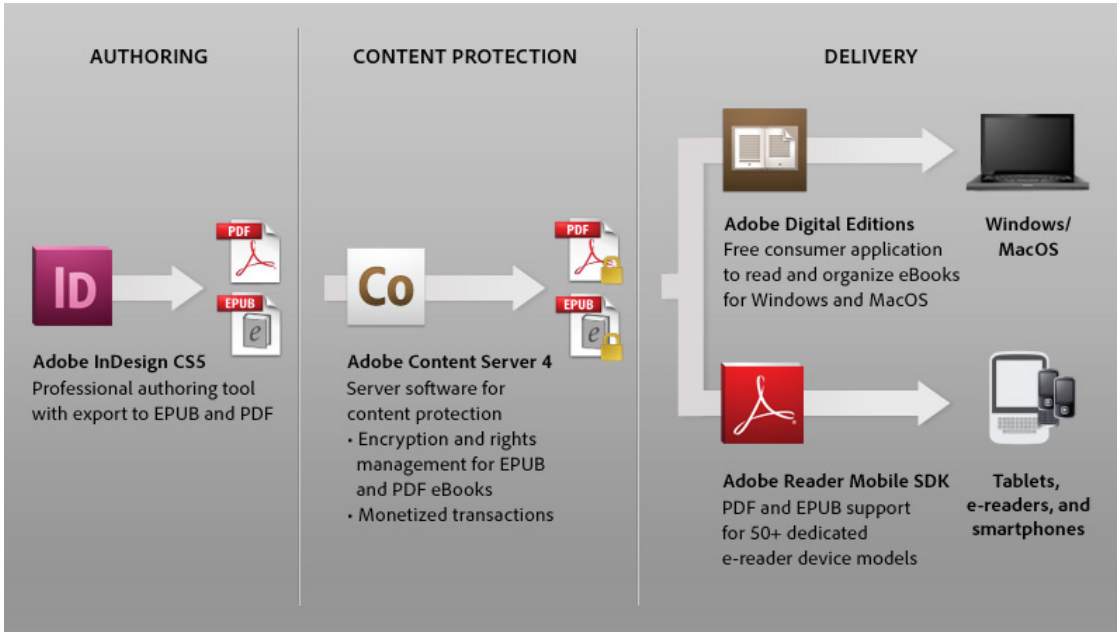

Figure 6: Adobe Digital Publishing Solution for eBooks (Copied from Adobe (2012) [37]).

will communicate online with the Content Server that will encrypt the e-book and provides it DRM protected with the publisher's authorizations for the user based on his adobe ID. Before the user can read the e-book, he must also have registered his device. That way, the platform makes sure that the file owned by the user is not distributed further.

Adobe DRM system allows the user to register up to 6 devices, so the user can access his library, annotations, bookmarks, etc. on all of them, e.g. start to read on his e-reading device, continue on his smart-phone and finish on his laptop. This can also cover the family fair use where people share a home computer.

4.1.2 Advantages and Limitations of Adobe DRM

One natural thing that user can do with physical books is to share it with friends. Unfortunately, DRM do not allow it since the e-book is protected for the user and its devices only. In order to give sharing facility Adobe offers another way to provide the e-book to the user with a password-only protection [36, Ability to share section]. The password will be encrypted with the file and it will be impossible to modify it. The publisher can choose to let the user define it and allow an open sharing or use a social DRM like approach where the publisher create the password for the user (e.g. using user credit card number, email address, etc.) to refrain the user to share it worldwide. This password-only protec-

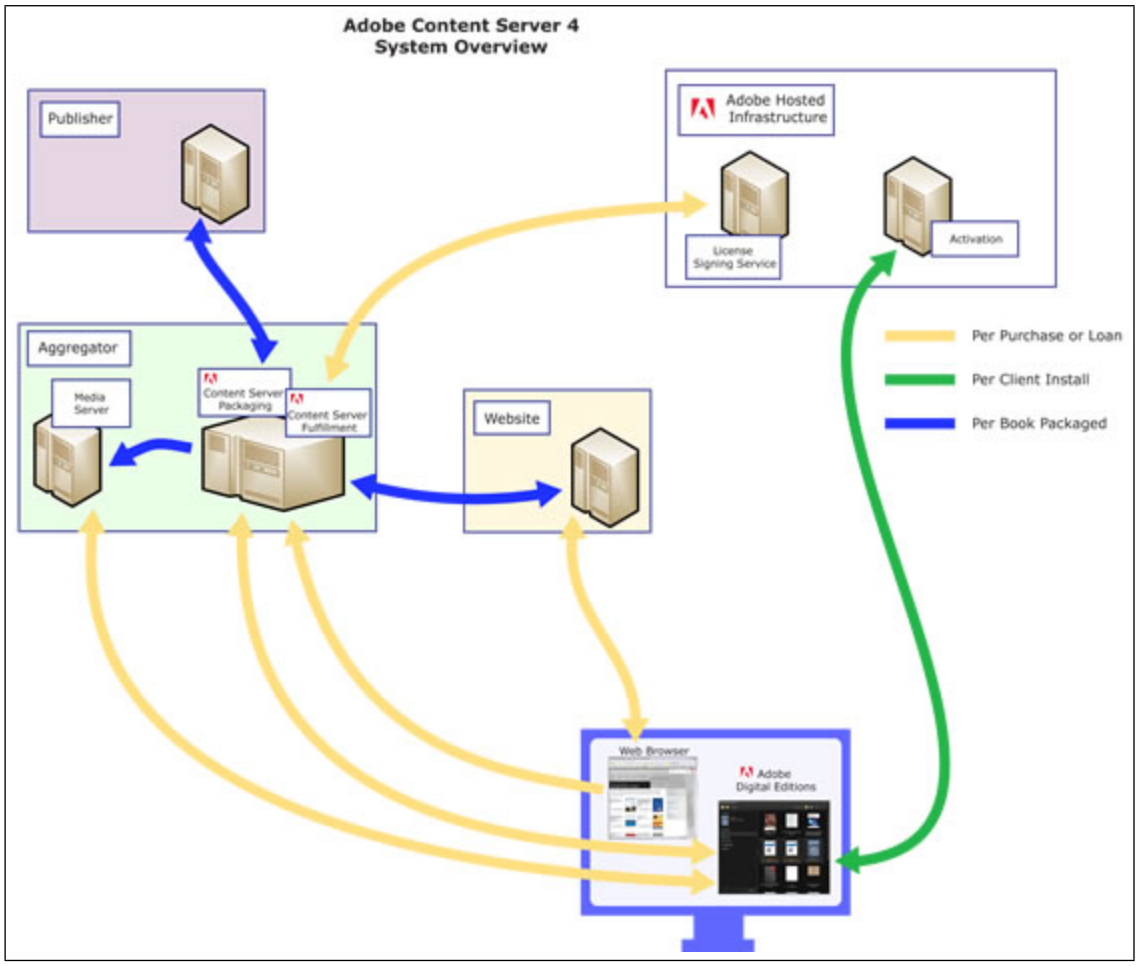

Figure 7: Adobe Content Server Architecture (Copied from Adobe (2012) [38]).

tion option is without Adobe DRM; so the file can either be DRM protected or password protected but not both.

An advantage of Adobe e-book DRM when compare to Amazon and Apple is that the user is not “imprisoned in a walled-garden system” [36, Open environment section] and can purchase DRM protected e-books and/or borrow from different vendors/libraries and can aggregate, organize and read them within the same application. Apple iBookstore allows only iOS devices to purchase files from their store and also prevent them to get e-books from a retailer who use a different DRM in their iBooks application. And to finish the Apple close loop, e-book created from their iBooks Author tool can only be sold through iBookstore [39]. Amazon Kindle device also lock the user with a single vendor. But Adobe argument is that many vendors use their DRM making the multiple vendors aggregation possible; but the Adobe Digital Editions will not allow opening an e-book encrypted with another DRM.

For publishers and authors, another advantage of Adobe DRM approach is the use of the open and well known formats PDF and EPUB, in opposition to the proprietary format from Apple iBooks and Amazon Kindle, thus simplifying the production, publication and distribution process [36, How the Adobe eBook Platform helps publishers section]. For the reader however, the advantages of these open formats get reduced. As an example, the DRM force the user to use an Adobe authorized software, preventing him to use his favourite EPUB/PDF reading application and can thus create some confusion.

Another claim of the Adobe whitepaper [36, Device interoperability section] is the possibility to read DRM protected e-books on multiple platforms. In fact, Adobe provides only the computer support with their free[6] Adobe Digital Editions application for Windows and MacOS Operating systems, excluding GNU/Linux [40]. For other devices, a third party has to develop a dedicated application using the Adobe Reader Mobile SDK, for which they have to pay a licence. Even that the price is fixed on a case-by-case basis, this can be unfair for a small publisher and is seen as another way from Adobe to monetise their DRM.

On the Adobe price politic, in 2008, the license for the Adobe Content Server was $5000[7] plus an extra $1500 per year for the support, maintenance, upgrade and the access to the Digital Signing Service [41]. And in 2012, from the Adobe Technology Partner in Europe [42], the license is $8000 plus $2000 per year support. They also charge $2495 for the installation and configuration. On top of that price, signing one e-book permanently will cost $0.22 ($0.08 for a 0-60 day expiring (e.g. for a library lending)). If such price is affordable for a big publisher, it can be too much for a small one.

As example, based on the yearly sales in Finland (see figure 8), it can be reasonable to say that a publisher sell ten thousand e-books per year. If he want to cover the license and installation (e.g. over a three year period) price, the yearly fee plus the 22 cents per e-book, he would have to charge an extra $0.77 per e-book he sell (and that do not include other charges for maintaining a server (like hardware/software installation and maintenance, electricity, domain name and internet address, etc.)).

It is not possible for a user to offer or resell his e-books. Due to the fact that the DRM is

[6] Free here has the meaning of gratis/no price.
[7] United States dollar (USD)

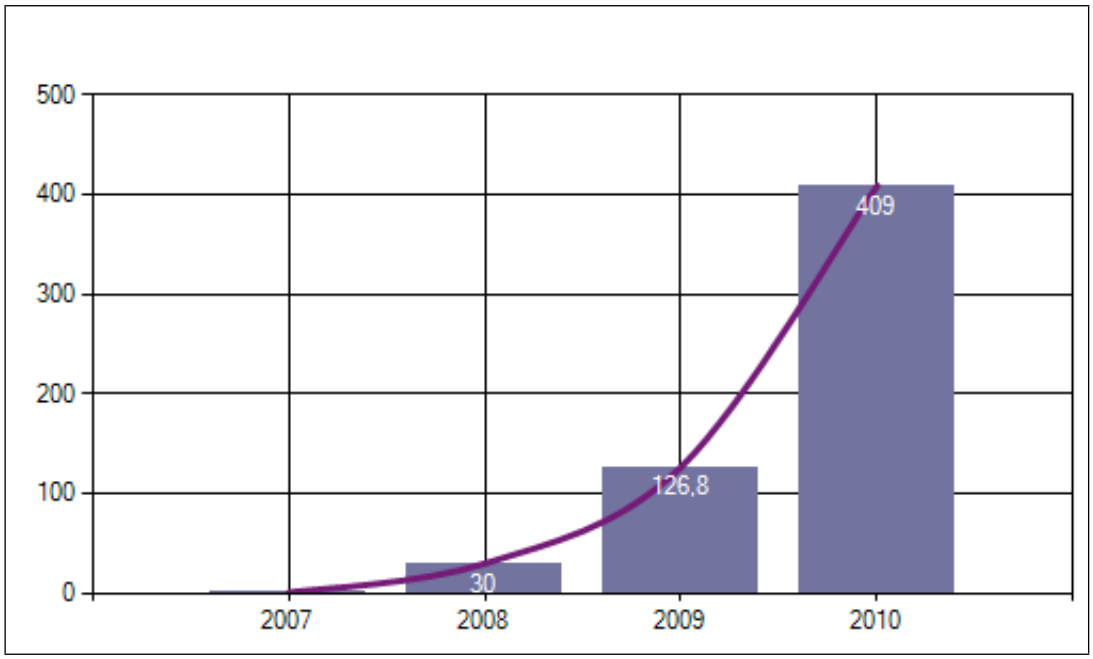

Figure 8: Downloaded e-books total. Yearly sales (1000 €) 2007-2010 in Finland. (Copied from Finnish Book Publishers Association (2012) [43]).

associated to the user and the device, it will be hard to try to read an e-book on a public computer (e.g. in an internet coffee). Also, if the user reach the limit of the 6 registered devices, buy a new computer and want an old one to be unregistered to replace it with his new one, he will face a long procedure to do so.

About two year after the release of the Adobe Digital Editions the explication on how to circumvent the Adobe ADEPT DRM was published on the I♡CABBAGES blog [44]. The author specifies that the Adobe file encryption was strong but the weakness was in how the Adobe Digital Editions hide the key. Nowadays, there are companies like epubor[8] that do business by selling DRM removal tools.

4.2 Lightweight DRM

The DRM systems suffer from many drawbacks such as interoperability problem where the reader is limited to a single platform, the requirement to be connected to internet in order to be authenticated, the risk to loose his content if the vendor goes bankrupt, etc. This chapter explore the possible upcoming standard for a Lightweight DRM proposed by the IDPF and its attempt in limiting the DRM problems.

[8] http://www.epubor.com/adobe-digital-editions-ade-drm-removal.html

The Encrypted Media Extensions possible standard from the W3C was also scouted but quickly dropped because it will not be suitable for e-books. It is specifically designed for media elements[9], namely video and audio files embedded or streamed into web pages.

4.2.1 Lightweight Content Protection (LCP)

In May 2012, IDPF made a proposal for standardizing a Lightweight Content Protection (LCP) that will be "occupying a middle ground between strong DRM and DRM-free" [27] to protect EPUB e-books. The idea is to have a protection that will be strong enough to qualify as an 'effective technical protection measure' to benefit the law protection to guarantee the publisher/vendor with a full protection mechanism and at the same time to reduce the 'hard DRM' drawbacks [7].

The LCP will work as follows: when a reader acquires an e-book, the content (texts, images, etc.) will be protected with a password encryption and the hash of the password will be associated with the file and/or the reading application. To read the e-book, the user will be prompted to enter it the first time before accessing the content. Since the password cannot be changed and will be set at purchase time, it can be defined by the vendor/publisher (to be e.g. the reader full name, email address, credit card number, etc.) [7].

With the password protection, there is no need to communicate with an authentication server, so LCP will work offline and it also means that the reader will really own the e-book (as long as he remembers the password) even if the provider ceases its activities. He will also have the freedom to have copies of his e-books on any devices he owns without limitation. The reader will be able to share his e-book; but will have to communicate the password too, so limiting him to people he trusts (friends/family) and refrain him to over-share it. Finally LCP will better respect user privacy by not spying the reader usage [7, Why Consider LCP for EPUB? and What Is LCP? sections].

With the content encryption, the publisher/vendor can also define limitation on usage like printing, copy of content, editing, etc. There will be also a possibility to set an expiration date which can be interesting for library lending. Basically, the provider will be able to set

[9] http://www.w3.org/TR/encrypted-media/#extensions

the same type of protection that he can do with other DRM [7, EPUB LCP Requirements section].

By lightweight, it means for the user device, less power and memory will be required, a fast decryption process and no communication with a server [7, What Is LCP? section]. For the provider, it will only need a simple signing/encryption mechanism, for example a web service without needing a complex server architecture [11, What Is LCP? and Requirements sections]. However, the cost in term of price as well as the licensing will be addressed later [11, Requirements section], which makes it unclear how much it will really be. And also, there are known patents[10] that exist, so a risk to have to pay royalties to third parties as an extra cost for the LCP implementation.

4.2.2 Advantages and Limitations of Lightweight DRM

A first argument for the IDPF is to make LCP a standard. This way, it will allow more devices and reading applications to be able to open and display the protected content, ensuring better interoperability. The reader would enjoy acquiring and reading e-books from different vendors/publishers using his favourite application/device, thus reducing the market fragmentation or lock-in [7, Why Consider LCP for EPUB? section]. However, the same document also states that “the resulting EPUB LCP […] would likely be published under licensing regimes. […] Use of the technology would be expected to be charged on a cost recovery basis.” [7, What Is the Recommended Process For Defining LCP for EPUB? section]. So the risk is that some application/device providers will refuse to implement the LCP and/or some e-book vendors would continue to use other DRM systems and ignore LCP.

IDPF envision some possible weaknesses. First, the LCP, like any DRM, will likely be cracked. But, they rely on the anti-circumvention law (see section 2.3) to have some level of crack protection [7, What Is LCP? section]; such as the illegality of a DRM removal tool. While modern DRM have some possibilities to recover and resist to crack, the LCP will not benefit from such features and because it is designed to not spy on the user, it means that it is not possible to monitor user activities such as suspicious ones [11, What

[10] e.g. intertrust holds over 200 patents in the digital media protection field (from: `http://www.intertrust.com/technologies/patents`)

Is LCP? section]. The others weaknesses concern the difficulties to have some business model such as “Domain authentication”, “License chaining”, “Master-slave schemes” and “Forward-and-delete” models [11, What Is LCP? section].

Another concern is how the IDPF will deal with free[11] reading software licensed under the GNU General Public License Version 3 (GPLv3) such as Calibre[12]. The GPLv3 section 3 of the terms and conditions state that “[…]you waive any legal power to forbid circumvention of technological measures[…]” [45], in other words, anyone with programming skills would legally have the right to modify the Calibre software to e.g. add a feature to ‘save the EPUB e-book unencrypted’ and the right to distribute such modified version of the application. So will IDPF forbid free application to implement the LCP and in doing so, loosing the interoperability? Or will they allow such software to implement LCP, knowing that the protection can legally be nullified?

Finally, there is no clear date when the LCP will be released not even decision to stop or continue. In the use cases and requirements, it only states that “IDPF solicit contributions of existing technology that could become the basis of a market-relevant solution for LCP within the next 12 calendar months or less” [7, Why Consider LCP for EPUB? section], meaning several month more? And also that IDPF envisions the possibility that there can be no LCP at all: “[…]it does not represent any commitment by the IDPF to establish a solution. […] it may become clear that no feasible standardized solution would be sufficiently useful or accepted, or that no solution is forthcoming that will sufficiently address critical requirements” [27].

[11] free as in freedom.

[12] Calibre is a e-book reading application that also offers other tools such as library management, synchronization with multiple devices, e-book file conversion, etc. (from: `http://calibre-ebook.com/about`)

5 Technical Implementation of the DRM Encryption Pillar Using W3C Standard

This work research if a DRM system could be made to protect an e-book while at the same time reduce the restriction to the fair use. Currently, one drawback of the DRM system is the lack of interoperability, like the impossibility to read an e-book on different platform. This chapter will test if the possibly upcoming W3C Web Cryptography API[1] standard could be used for the protection and encryption part of such DRM system.

The assumption is that most of the modern devices do or can have a JavaScript capable web browser and can be connected to internet directly or through another device. If asserted, then making a web based DRM system where the user could use his favourite browser on any device to read protected content could offer a better interoperability than most of the existing DRM systems.

The idea to make part of a DRM system with the Web Cryptography API originates from the W3C Use Cases [46] such as the “Banking Transactions” where a user uses a public and private cryptographic key pair to exchange messages that he will be the only one able to read and the “Authenticated Video Services” use case where the user get encrypted content from a distant service over the internet that only his device can display.

With the Web Cryptography API being in the draft state and with most of the exiting browsers having only a partial implementation of it[2], the PolyCrypt[3] implementation is used as a fall-back. That library tests and exposes the needed cryptography libraries if not yet implemented natively by the browser.

The starting point for the code is the web-crypto-samples application[4], Copyright (c) 2013 Nick Van den Bleeken under MIT License. The file `base64.js` is an exact copy from this application, used as a library. Part of his `utils.js` and `encrypt.js` scripts are derived

[1] `http://www.w3.org/TR/WebCryptoAPI`
[2] `http://www.w3.org/2012/webcrypto/wiki/Main_Page#First_implementations`
[3] `http://polycrypt.net/`
[4] `https://github.com/nvdbleek/web-crypto-samples`

into the `keydecrypt.js` code as seen in annex 3.

5.1 Coding

As a first step, this application, generate the public and private key pair on the client side if they do not exist. This operation correspond to the KEYGEN-ASYM requirement of the W3C Web Cryptography API Use Cases [46, chap. 2]. The listing 1 shows the JavaScript code used to generate the 2048 bits keys (lines 1 to 7) with the Public-Key Cryptography Standards (PKCS) RSA algorithm and the callback method if the keys get generated successfully (lines 9 to 17). After creation, the keys are stored on the user computer (lines 13 and 14) for later cryptographic operations.

```
1  var genOp = webCrypto.generateKey({
2    name : "RSAES-PKCS1-v1_5",
3    params: {
4      modulusLength : 2048,
5      publicExponent : new Uint8Array([ 0x01, 0x00, 0x01 ])
6    }
7  }, true, [ "encrypt", "decrypt" ]);
8
9  //the onerror callback function is not shown here
10
11 genOp.oncomplete = function(e) {
12   publicKey = e.target.result.publicKey;
13   privateKey = e.target.result.privateKey;
14   if(publicKey && privateKey){
15     localStorage["fi.metropolia.key.public"] = JSON.
         stringify(publicKey);
16     localStorage["fi.metropolia.key.private"] = JSON.
         stringify(privateKey);
17     //... here comes exporting keys code ...
18   }
19 }
```

Listing 1: Generating public and private cryptography key pair

If the user keeps the private key secret and shares the public key, he makes sure that he will be the only one to be able to decrypt and read the content encrypted with his public key, since only his private key can do this decryption. Since the raw public key is not suitable for the sharing process, it will first be converted into the JSON Web Key (JWK) format[5] as shown in the listing 2. This operation correspond to the EXPORT requirement of the W3C Web Cryptography API Use Cases [46, chap. 2].

[5] https://tools.ietf.org/html/draft-ietf-jose-json-web-key-25

```
1  //Export the public key
2  var exportOp = webCrypto.exportKey("jwk", publicKey);
3
4  //the onerror callback function is not shown here
5
6  exportOp.oncomplete = function(evt) {
7    //stringify the JWK public key
8    var n = JSON.stringify(evt.target.result);
9    //key ready to be shared
10   document.getElementById('public_key').innerHTML = n;
11 }
```

Listing 2: Exporting public key

After the exporting preparation operation, the user sends his now ready public key to the server which will use that key to encrypt the content specifically for him. But before encryption, the server will first import the key as presented in the PHP code of listing 3. This operation correspond to the IMPORT requirement of the W3C Web Cryptography API Use Cases [46, chap. 2]. With no PHP function found to import the JWK format strait, the key is first converted into the X.509 Privacy-enhanced Electronic Mail (PEM)[6] certificate format.

```
1  //receive the JWK key from the client
2  $pub_key = $_POST["public_key"];
3  $key_elem = json_decode($pub_key, true);
4  //convert the public key to PEM format
5  // dirty hack since it works only with a 2048 bits key
6  // (hard coded headers)
7  //from http://fi1.php.net/manual/en/function.openssl-pkey-
       get-public.php#104439
8  $prepa = wordwrap('MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A'.'
       MIIBCgKCAQEA'.str_replace(array("_", "-", "="), array("/"
       , "+", ""), $key_elem["n"]).'ID'.$key_elem["e"], 64, "\r\
       n", true);
9  $key = <<<EOF
10 -----BEGIN PUBLIC KEY-----
11 $prepa
12 -----END PUBLIC KEY-----
13 EOF;
14 //import the key
15 $res = openssl_pkey_get_public($key);
```

Listing 3: Importing public key

With the public key successfully imported, the server is ready for the encryption process. The PHP script will first get the content of the uncompressed EPUB file stored in folder

[6] https://tools.ietf.org/html/rfc2510

not accessible from internet as seen in the annex 3 encrypt.php source code file. With a maximum of 1968 bits long string that can be encrypted with the 2048 bits long key, the content of the file to be encrypted is chunked into pieces of 123 characters ($123 \cdot 16 = 1968$) and each of the substring get encrypted individually as shown in the listing 4. This operation correspond to the ENCRYPT-ASSYM requirement of the W3C Web Cryptography API Use Cases [46, chap. 2].

```
1  //the file with the encrypted content
2  $write_file = "EPUBfolder/OPS/".$list_of_content[
       $current_file];
3  $resfile = fopen($write_file, "w");
4  //reading and encrypting
5  foreach(str_split(file_get_contents("../../secret/EPUBfolder
       /OPS/".$list_of_content[$current_file]),123) as $val){
6    openssl_public_encrypt($val, $encrypted, $res); //$res is
         the imported key
7    fwrite($resfile, base64_encode($encrypted));
8  }
9  fclose($resfile);
10 echo "Encrypted file: <a href=\"$write_file\" target=\"
       _blank\">$write_file</a>";
```

Listing 4: Encrypting the content of a file with the public key

Now that the content is specifically encrypted for him, the file can be sent to the user. To decrypt the content of the file, the JavaScript will use the use the private key. This operation correspond to the DECRYPT-ASSYM requirement of the W3C Web Cryptography API Use Cases [46, chap. 2]. The listing 5 present the code that decrypt the content of the file bloc by bloc asynchronously.

```
1  var alg = { name : "RSAES-PKCS1-v1_5" };
2  var privateKey = JSON.parse(localStorage["fi.metropolia.key.
       private"]);
3  //decrypt the data bloc by bloc asynchronously
4  for(i = 0; i < blocData.length; i++){
5    (function(closureI){
6      decryptOp[closureI] = webCrypto.decrypt(alg, privateKey,
           new Uint8Array(Base64Binary.decodeArrayBuffer(
         blocData[closureI])));
7
8      //the onerror callback function is not shown here
9
10     decryptOp[closureI].oncomplete = function(evt) {
11       resultArray[closureI] = ab2str(evt.target.result);
12     };//decrypt
13   })(i) //closure
```

```
14 }
```

Listing 5: Decrypting content with the private key

Finally, with the asynchronous decryption process, all the decrypted parts are reordered and written in a local JavaScript blob object to be rendered to the user as shown in the listing 6. In order to get the image and styling associated with the EPUB XHTML pages, the URL are overwritten to let the browser getting the CSS and images files on internet.

```
1  function decryptedToBlob(){
2    //hack to transform relative path to absolute hard-coded
       URL.
3    var tmp = [];
4    var j, hack;
5    for(var i = 0; i < resultArray.length; i++){
6      tmp = resultArray[i].split("=\"");
7      if(tmp.length > 1){
8        hack = "";
9        for(j = 0; j < tmp.length - 1; j++){
10         hack += tmp[j] + "=\"";
11         if((tmp[j].endsWith("href") || tmp[j].endsWith("src"
               )) && !tmp[j+1].startsWith("http://") && !tmp[j
               +1].startsWith("#"))
12           hack += "http://hardcoded.fi/EPUBfolder/OPS/";
13       }
14       hack += tmp[j];
15       resultArray[i] = hack;
16     }
17   }
18   //array to blob to be used as the iframe source.
19   var blob = new Blob(resultArray, {type : 'application/
       xhtml+xml'});
20   document.getElementById("decrypted_blob").src = URL.
       createObjectURL(blob);
21 }
```

Listing 6: Decrypting content with the private key

Part of the source code is listed into the annex 3. The `genkey_and_export.html` and `keydecrypt.js` representing the client side of the application responsible for the local operation of generating the keys, exporting the public key, decrypting the content with the private key and rendering the result to the user. The `encrypt.php` file represent the server side of the application responsible for importing the user public key and encrypting the content. The source code files that are not listed in annex 3 are the non-modified files used as library, namely the PolyCrpt[7], Copyright (C) Raytheon BBN Technologies Corp,

[7] `https://github.com/polycrypt/polycrypt/tree/master/src`

2013, published under BSD 3-clause license[8] and the `base64.js`[9] file , Copyright (c) 2013 Nick Van den Bleeken under MIT License[10].

A demonstration of this application is running at `http://users.metropolia.fi/~patricka/wcrypt-book/genkey_and_export.html` and the source code is available at the same place or in github. This software is licensed under the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This application encrypt the public domain EPUB book Moby-Dick; or, The Whale by Herman Melville for the sole purpose of demonstration; the e-book originate from Project Gutenberg[11] where it is freely available.

5.2 Known Problems and Possible Improvements

With this application, the protection is only guaranteed during the transport between the server and the client machine. Once on the client machine, it is very hard to hide the private key and obfuscate the JavaScript decryption code. If the user retrieve the private key, he could write an alternative decryption application that could for example save unencrypted copy of the files that he could later share. With the JavaScript being an interpreted language, the reverse engineering process to modify the decryption code would be easier than with compiled language.

Currently, once decrypted, the content can easily be copied. But even if the selecting, copy-pasting and the print screen would be disabled, the user could still take physical picture of the content to save an unencrypted version of the text.

Another concern with JavaScript as stated by Matasano Security [47], Matthew Green [48] or the PolyCrypt[12] team itself is that this programming language and the current implementation of the Web Cryptography API can not be use for serious cryptographic operation. The language will suffer from many flaws such as “side-channel attacks”, problem with the garbage collector where “secrets are lurking in memory potentially long after they're

[8] `https://github.com/polycrypt/polycrypt/blob/master/license.txt`
[9] `https://github.com/nvdbleek/web-crypto-samples/blob/master/js/lib/base64.js`
[10] `https://github.com/nvdbleek/web-crypto-samples/blob/master/LICENSE`
[11] `http://www.gutenberg.org/ebooks/2701`
[12] `http://polycrypt.net/`

needed” and many more. So at best, secure JavaScript cryptography could be possible only after it would be natively implemented and solidified by browser and seriously reviewed.

This application use the asymmetric decryption operation intensively and this operation is rather slow with the current implementation of the Web Cryptography API. A possible improvement would be to only encrypt part of the file, for example the first and last three paragraphs of each chapter, that would be enough to make the content unusable. Another option would be to use the asymmetric keys only to exchange symmetric session keys, as proposed by the use case “Authenticated Video Services” [46], with the symmetric decryption being much faster.

Some possible improvements for this application would be to use the off-line storage provided by the HTML5 language, so the full encrypted content could be stored on the user computer to be read without internet connection and also making possible to do the decryption of the next chapter in the background so the user will not have to wait. The content, once decrypted could be better presented to the user, for example by using the `epub.js`[13] script. One know bug of this application is with the encryption and decryption of some special typographical characters such as the curly quotation marks or the em dash.

[13] `https://github.com/futurepress/epub.js`

6 DRM-Free

6.1 DRM Opponents and Limitations

The opponent to the DRM, such as the Free Software Foundation (FSF) through their 'defective by design' [49] campaign define DRM as "Digital Restriction Management"because DRM restrict the fair use like limitation of the private copies and backup, hard or impossible to share, swap, offer or resell purchased files. It also "imprisons"the user in non-free[1] reading software where the user is forced to agree with policy that may downgrade his rights with new software upgrade otherwise he looses his files. They raise privacy concerns since such software can monitor user hard disk and spy computer usage. They also denounce a back-door that allow vendor to remotely delete the e-books form the devices of the users; with the example of Amazon removing the George Orwell 1984 e-book from hundreds of users[2] or more recently clearing the full collection from one reader[3].

Even if the EU DIRECTIVE 2001/29/EC [14, article 6(4)] request that the DRM respect some of the exceptions (from [14, article 5]); it makes for example optional the private copy and as such, in the Finnish COPYRIGHT ACT [15, section 50c(1)] this exception is not present; so in Finland, a DRM can legally forbid the private copy that confirm the FSF concerns. There are many others fair uses that may be restricted by DRM such as reproduction by the press, communication to the public [14, article 5(3)(c)], use for purpose of caricature, parody or pastiche [14, article 5(3)(k)], use during religious celebration [14, article 5(3)(g)].

Dusollier [50, chapter 7] complains that "the DRM is more a protection for the services of the provider than a protection of the copyright"[4]. The fair use exceptions, that DRM must follow, concern almost only public institutions (libraries, schools, museums, hospitals or prisons) and thus can be unfair for a private user. As example, an individual user want to

[1] Free here has the meaning freedom/liberty in opposition to proprietary (so nothing related to the price/-cost).

[2] `http://www.defectivebydesign.org/blog/1248`

[3] `http://www.defectivebydesign.org/node/2250`

[4] My translation

copy part of the text of a DRM protected e-book for a quotation with purpose of review [14, article 5(3)(d)] and if the DRM restrict the copy of the text, he would have to circumvent it to reach his goal. So he will not infringe the copyright law; but he will be guilty for circumventing a technological measure. While if he was removing the DRM to distribute unauthorized copies, he would already violate the copyright; so the DRM is not playing its role, except that now the user will be punished twice for infringing the copyright and the technical measure.

Because DRM can limit the number of devices the user can use to read his e-book and by knowing that the user will probably buy a new device every year or two, it means that he will have problems when he reaches that limit. He will not understand that he was paying for a service instead of owning his files. This might also stop the social sharing (e.g. to lend an e-book to his friends or family members) restricting one aspect of the reading experience.

The Electronic Frontier Foundation (EFF) [51] shares the same kind of concerns that the FSF. EFF goes further against big companies by stating that the DRM is an anti-competitive practice. For example, if the vendor proposes the e-book in a single protected proprietary file format, forcing the reader to access it only with a specific hardware and/or software, that will restrict the user to access DRM protected content from a concurrent with the same device/application; it will lock the user to a single vendor. That type of strategy is seen from Amazon and Apple. On the other side, if the publisher imposes protection, a small reseller will have problem to pay for a DRM system; closing the market for the big companies. For Doctorow [52, chapter 28], the protection does not benefit the writer nor the publisher nor the end user but put the power in the hands of the DRM provider.

EFF also says that “putting DRM on e-books is short-sighted, futile, and doomed” [53], because the best e-book DRM will suffer from the analog hole. In other words, the protection ends once displayed on screen. So anyone with a digital camera and an Optical Character Recognition (OCR) software can get an almost perfect DRM-free copy in few minutes. For example, in figure 9, a book ripper designed to digitalize physical book[5] is used to free up a DRM protected e-book from a reader device.

[5]Where the book is placed in the middle and when turning every pages, take a picture of the odd page with the left camera (the right one for the even page). When the end of the book is reach, the pictures are treated with an OCR software that will produce the e-book.

Figure 9: e-book ripper (Copied from bkrpr.org (2009) [54]).

Another fear for the user is if the vendor goes bankrupt or if the DRM provider stops to maintain the system. This happen with Microsoft announcing that they will not support any more the .lit file format: “Microsoft is discontinuing Microsoft Reader effective August 30, 2012, which includes download access of the Microsoft Reader application from the Microsoft Reader website” [55]; so the users may loose the access to their files when their devices will die.

Since the DRM can also be location aware[6], it can be problematic for the user who travel or move abroad. This can also impede a user to buy a book from an other country if for example, the book is not available in his area (Apple iBooks store is not accessible in every countries). This looks again like a way to control the business.

Illegal offers exist for e-books; but it is minority. Based on a French survey, it count only for 5% through file sharing websites, 4% with peer to peer (P2P) and 1% in streaming[7]; compare to legal offer with 41% through the big operator (Amazon, Apple Store, Google books, etc.) and 28% on specialized web-stores (Fnac, VirginMedia, Cultura, etc.), etc. [56, p. 9]. The same survey also reports that 17% of the readers had at least once

[6] the patents exist (e.g. `http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=/netahtml/PTO/srchnum.html&r=1&f=G&l=50&s1=20060059096.PGNR.`)

[7] except for streaming, the study do not specify if the download was legal or not (e.g. an e-book under a creative common license (or in the public domain) can be legally downloaded via P2P).

acquired an e-book illegally. The reasons in doing so was for 69% that the legal offer was too expensive, for 40% that the legal offer did not exist and for 14% that there were having problems with DRM [56, p. 13]. So, to reduce the download of e-books, the publishers/vendor could concentrate on attractive and/or transparent[8] price, have richer collections available and abandon DRM.

Going DRM-free can have a positive impact on the readers and be used as a marketing argument. In facts, Bragelonne, a French publisher, followed that path in the end of 2010. Six month later, they reported that this strategy made them among the leaders in e-books selling in their genre. The journalist use that success-story as an extra example to show that the “DRM is a barrier to business” [57]. More generally, being DRM-free does not prevent to do business.

Watermark and Social DRM also raise some privacy concerns, like the risk of having private data (such as the reader full name, email, the credit card number, etc.) visible to everyone. They can also lead to an unfair punishment if for example, someone loose his reading device and his files goes illegally shared, he risks to be punished for crime he did not committed. And this can also affect people who have a poor knowledge of technology, who can by mistake make their files available to the public.

With DRM, Watermark and Social DRM, the reader must provide private information in order to purchase an e-book. He don't has the possibility to buy it anonymously as he could do with physical book by paying cache in a bookshop. This could lead to self-censorship where the user would refrain himself from buying a controversial e-book with the risk of restricting his freedom of opinion.

6.2 DRM Alternative

The file sharing on internet (authorized or not) is a reality and the DRM does not stop nor prevent it. So, an alternative to the technical measure is to have no DRM. Among the advocates of such approach, let mention Stallman [58] with his essay Freedom—or Copyright? and Aigrain [59] with his book Sharing. Both also promote the legalization

[8] e.g. in Finland the Value Added Tax (VAT) is 23% for e-book while it has a reduced rate of 9% for paper book (`https://www.vm.fi/vm/en/10_taxation/04_value_added_tax/index.jsp`).

of non-market file sharing. The non-market file sharing is done between users without commercial counterpart and without using a commercial third party service.

Stallman [58] and Aigrain [59, section 3.2] say that non-market sharing is useful for culture by providing a better access to it, making it more divers and also by making available out of print and orphan works. It can also help in promoting unknown authors, when publishers put focus on possible best-selling tittle (from which they hope to get more revenues), the readers will more likely share works that they enjoyed or think that are of interest.

They also say that the authors can still get fair revenues by selling DRM-free e-books giving example of Stephen King [58] and even e-books release under the Creative Commons licenses in synergy with paper books such as Cory Doctorow or John Sundman [59, section 7.4]. In scientific/technical publishing, this is already a reality with publishers like Springer who provide e-books with no DRM to guarantee “Perpetual access & ownership” [60] or O'Reilly who adds services to attract customers such as “lifetime access”, provide the files in multiple formats so the user can read them in any device and “free updates to reflect published changes and corrections” [61].

In fiction and others genre, some vendors, for example, Weightless Books[9] sell all their e-books DRM-free. In publishing the Tor/Forge announced that they will release their full catalogue without DRM by July 2012 [62] and that discussion went with their parent company Macmillan [63]. One of the older DRM-free publisher is Baen Books[10] who was in 2007 stating:

> We don't treat our customers like criminals, and they don't act like them. We've found that if you treat your readers with respect, they become repeat customers, as the success of our decade-old Webscription program can attest. [64]

An alternative way of selling e-book was experimented by Humble Bundle[11]. They offered a bundle of thirteen DRM-free e-books during two weeks with the ‘Pay What You Want’ pricing system and a possibility to set a percent that will go to charity. The result was over 84 thousand purchases of the bundle generating 1,2 million $ [65, Humble eBook Bundle]. This was seen successful enough to give birth to the story bundle[12] based on

[9] http://weightlessbooks.com/about/
[10] http://www.baenebooks.com/t-DRM.aspx
[11] http://www.humblebundle.com
[12] http://storybundle.com/

the same principle.

Stallman [58] and Aigrain [59, section 5.2, chapter 6 and 7] also propose a new source of income for the authors to remove their fear to publish their works DRM-free in digital form. The idea is to collect a flat-rate tax based on internet subscribers that would be re-distributed to authors and how that would be distributed. Aigrain goes further by proposing that the collected money should also fund future cultural project and a part to serve for archiving.

Aigrain [59, chapter 6] also provides more idea about the rights to share; the sharing of cultural artefacts would be possible only after they have first been made available to the public in digital form. For example, scanning a paper book and sharing it will still be copyright infringement; except if the author has explicitly permitted it. “This saves an essential element of media chronology: the possibility [for the author] to schedule the public performance, analogic distribution and digital distribution at different times”. In addition to the sharing, he proposes the remix right to create modified works. He also insists on the facts that the users must respect the attribution of the authors for their works and can’t remove nor modify the meta-data identifying a file. With Stallman [58], they insist on the non-market sharing part. For example, websites that want to sell copies, give direct access to a library of files through subscription system or by using ads will still need to negotiate commercial licenses with the authors.

At the political level, the ‘free culture’ is a motto of the pirate parties[13]. The idea is also supported by the European Green Party [66]. The subject is discussed at various stages in some governments such as France, Belgium, Germany, Brazil and Switzerland where a postulate “toward a fair copyright compatible with internet user freedom”[14] [67] has been proposed and has been accepted by the government who has formed a working group[15] that will provide the results at the end of 2013.

Stallman [58] concludes by advising that before this “information utopia” battle is won, the users should not buy DRM products –except if there is a way to break it– to avoid the establishment of a “pay-per-view world” imposed by publishers.

[13] http://piraattipuolue.fi/english

[14] my translation

[15] http://www.ejpd.admin.ch/content/ejpd/fr/home/dokumentation/mi/2012/2012-08-09.html (in French (also available in German and Italian))

7 Results

The main products of this thesis are the participation for the Next Media project in different forms. One is the deliverable (number 1.3.1.4) with the title “Protection mechanisms for electronic books”. It is composed of almost the same text and content as the chapters 1, 2, 4, 6 and 9 as of this work. It was returned to the consortium in February 2013.

Another participation to the Next Media project consisted of a presentation to the Finnish National library and the company Ellibs representatives in April 2012. The presentation and discussion concentrated on the topics of how could Finnish libraries lend e-books as they do with paper books and also about the archiving e-books. An other presentation for the Finnish Book Publishers Association development group in October 2012 concentrated in presenting the different alternatives for protecting e-books.

For the eReading work-package of the Next Media project, there are two presentations in seminars, one in June 2012 and the next one in December 2012. Another seminar participation is the Next Media Result Seminar in February 2013, with a short, fifteen minutes presentation. The slide show of this presentation is available in the Next Media website[1] and the video recording available on the Vimeo platform[2]. A last presentation is a 45 minutes conference for the Finnish Book Publishers Association and two authors unions members in February 2012. The slide show of all the presentation are published in github[3].

The last task for the Next Media project is the participation to the reflection group about e-book protection initiated by the Finnish Book Publishers Association. The outcome is the DRM recommendation send to the Finnish publishers in June 2013. The main result being majors Finnish publishers who abandon DRM in favour of the watermark and Social DRM lighter protection system [4].

[1] http://www.nextmedia.fi/779350088/en/read/Next_Media_Result_Seminar_6.2.2013
[2] http://vimeo.com/59498270
[3] ...
[4] https://www.sttinfo.fi/release?releaseId=8700912

The application produced for this thesis demonstrates that the upcoming W3C Web Cryptography API can be used to encrypt data. It also shows some of the limitation of DRM systems such as the difficulty to hide private key. It also outline the risks in using the current implementation of the Web Cryptography API for serious cryptographic operations.

8 Discussion

During this research work, I studied the existing and upcoming protection techniques, namely: Digital Right Management (DRM), Watermark, Social DRM and Lightweight Content Protection (LCP) and tried to implement a protection system with the Web Cryptography API. What I found is that it does not exist possible improvement of the DRM techniques because of the paradigm problem of trying to impose the physical world to the electronic world which are fundamentally different.

However, a technical solution that could guarantee a fair revenue to the author of a e-book while not restricting the user freedom to share copy of it, could be to implement a script that would present to the reader a donate button. With such button, if the reader likes the e-book, he could very easily offers to the author a small sum of money as a reward for his work. A similar system already exist on the internet with for example Flattr[1].

At the non technical level, there is also important lewis that can be used to fight copyright infringements. At the market level, it is important that the publishers offer a rich collection of titles in electronic form, makes it easily available, propose attractive prices and ideally do not impose DRM restriction. With that approach, the user would be less tempted to search for the illegal alternatives. Another approach that would be interesting to study could be to test Crowdfunding where the author writes and publishes his new book only if he gets its funding beforehand.

Another important lewis is to continuously educate people about the copyright. With Crowdfunding or a donate option, this work could be made easy as “did you offer 1€to your favourite author this month?” [58]. Finally, the other proposition is to collect flat-rate tax that would be distributed to the authors. Such proposition requires a work at the political and legal level in order to adapt the copyright laws to the reality of the internet.

While this research work was limited to the protection of electronic book, many of these proposition could be extended to the arts and sciences in general. This problem and

[1] https://flattr.com/

the proposed solution already apply to the music and cinema and with the advance of technology such as 3D printer, this could also reach other domains such as sculpture and carving. There is research on holographic screen, so maybe one day, we will be able to have live arts such as dancing and theatre in our living room as well.

9 Conclusions

The research conducted shows that among the technical options to protect e-book from illegal uses, none is perfect. Trying to impose the rules of the physical world to the electronic one is not suitable.

For the e-book, the ‘hard DRM’is not an appropriate solution as it does not prevent nor stop the illegal file sharing, it restricts the freedom of legitimate reader (while cracker get it fully), adds extra costs in term of price and infrastructure for the publisher/vendor and tends to fragment the market (or creates monopoly). However, in some cases, such as company confidential documents, where the reader and provider agree on the need for protection, the DRM may be a good solution.

Concerning the Lightweight Content Protection (LCP), it remains interesting on following the IDPF work. Even if the LCP still has unclear costs, it could be an alternative to ‘hard DRM’for vendor or library when the publisher imposes technical protection. If it would be implemented, it could also reduce some of the drawbacks with better interoperability, no spying and a real ownership for the reader. But, as with the ‘hard DRM‘, it could be cracked or the users may fight against it.

The Social DRM and watermark combination might be the less problematic solution. It can be better accepted by the users, offers some level of protection and help deter on copyright infringement. However, the reader may reject it because of privacy concern.

For these reasons, being DRM-free would be the better option. It might show that the authors trust and respect their readers. Copyright and licenses, such as the Creative Commons, do provide protection and being DRM-free does not prevent to do business. It is also relevant to continue the work at the educational, legal and market levels.

References

1 international community UN. The Universal Declaration of Human Rights. 1948;Available from: http://www.un.org/en/documents/udhr.

2 M A C Dekker. Flexible Access Control for Dynamic Collaborative Environments. University of Twente. Enschede; 2009.

3 D Pgdba Satish. Digital Rights Management : An Introduction. Hyderabad, India: Icfai University Press; 2007.

4 Frank Hartung, Friedhelm Ramme. Digital rights management and watermarking of multimedia content for m-commerce applications. Communications Magazine, IEEE. 2000 nov;38(11):78 -- 84.

5 Boštjan Marušič, Philippe de Cuetos, Laurent Piron, Zvi Lifshitz. TIRAMISU: That's unobtrusive DRM in the home domain. INDICARE; 2005. Available from: http://indicare.org/tiki-read_article.php?articleId=125 [cited November 14, 2012].

6 Martyn Daniels. Hard and Soft DRM: Part 1; 2009. Available from: http://bookseller-association.blogspot.com/2009/12/hard-and-soft-drm-part-1.html [cited January 18, 2013].

7 Bill Rosenblatt. EPUB Lightweight Content Protection: Use Cases & Requirements. International Digital Publishing Forum; 2012. Available from: http://idpf.org/epub-content-protection [cited May 28, 2012].

8 Fabien A P Petitcolas, Ross J Anderson, Markus G Kuhn. Information hiding-a survey. Proceedings of the IEEE. 1999 jul;87(7):1062 --1078.

9 Insup Lee. EMTM 553: E-commerce Systems - Lecture 7a: Digital Watermarking. University of Pennsylvania; 2001. (slide show presentation). Available from: http://www.cis.upenn.edu/~lee/00emtm553/watermark.ppt [cited November 15, 2012].

10 Kevin Franco. Packaged Digital Rights Messaging; 2012. Available from: http://kevinfranco.blogspot.fi/2012/05/packaged-digital-rights-messaging.html [cited November 15, 2012].

11 Bill Rosenblatt. EPUB Lightweight Content Protection: Request for Proposals. International Digital Publishing Forum; 2012. Available from: http://idpf.org/epub-content-protection-rfp [cited July 30, 2012].

12 Martyn Daniels. Hard and Soft DRM: Part 2; 2009. Available from: http://bookseller-association.blogspot.com/2009/12/hard-and-soft-drm-part-2.html [cited January 18, 2013].

13 World Intellectual Property Organization (WIPO). WIPO Copyright Treaty. CLEA - Collection of Laws for Electronic Access; 1996. Available from:

http://www.wipo.int/wipolex/en/treaties/text.jsp?file_id=177635 [cited February 28, 2012].

14 European Parliament, Council. DIRECTIVE 2001/29/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 22 May 2001 on the harmonisation of certain aspects of copyright and related rights in the information society. EUR-Lex, http://eur-lex.europa.eu/; 2001. Available from: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2001:167:0010:0019:EN:PDF [cited February 22, 2012].

15 Ministry of Education and Culture. FINLAND COPYRIGHT LEGISLATION 2010, Unofficial Translations in English. Finlex.fi / Ministry of Justice and Edita Publishing; 2010. Available from: http://www.finlex.fi/en/laki/kaannokset/1961/en19610404.pdf [cited February 27, 2012].

16 Senate and House of Representatives of the United States of America. Digital Millennium Copyright Act. GPO - U.S. Government Printing Office; 1998. Available from: http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=105_cong_public_laws&docid=f:publ304.105.pdf [cited February 28, 2012].

17 Bill Rosenblatt. The Harry Potter Watermarking Experiment. Copyright and Technology; 2012. Available from: http://copyrightandtechnology.com/2012/04/08/the-harry-potter-watermarking-experiment [cited January 28, 2013].

18 Definition for e-book - Oxford Dictionaries Online (World English). Oxford University Press; 2012. Available from: http://oxforddictionaries.com/definition/e-book [cited May 16, 2012].

19 electronic book noun - definition in British English Dictionary & Thesaurus - Cambridge Dictionary Online. Cambridge University Press; 2011. Available from: http://dictionary.cambridge.org/dictionary/british/electronic-book [cited May 16, 2012].

20 Catherine C Marshall. Reading and Writing the Electronic Book. San Rafael, California, United States: Morgan & Claypool; 2010.

21 La Nouvelle Edition du 29/03 - Part. 3 Marc Levy. SOCIETE D'EDITION DE CANAL PLUS; 2012. Online video clip (in French). Available from: http://player.canalplus.fr/#/617029 [cited July 23, 2012].

22 OpinionWay. Baromètre des usages du livre numérique - Vague 1—Mars 2012. Société Française des Intérêts des Auteurs de l'écrit (Sofia), Syndicat national de l'édition (Sne) and Société des Gens de lettres (SGDL); 2012. (in French). Available from: http://www.opinion-way.com/pdf/opinionway_barometre_des_usages_du_livre_numerique_-_vague_1.pdf [cited November 20, 2012].

23 Adobe PDF history. Adobe Systems Incorporated; 2012. Available from: http://www.adobe.com/products/acrobat/adobepdf.html [cited May 29, 2012].

24 EPUB. International Digital Publishing Forum; 2012. Available from: http://idpf.org/epub [cited May 29, 2012].

25 Garth Conboy, Matt Garrish, Markus Gylling, William McCoy, MURATA Makoto, Daniel Weck. EPUB 3 Overview. International Digital Publishing Forum; 2011. Available from: http://idpf.org/epub/30/spec/epub30-overview.html [cited May 29, 2012].

26 James Pritchett, Learning Ally, Markus Gylling. EPUB Open Container Format (OCF) 3.0. International Digital Publishing Forum; 2011. Available from: http://idpf.org/epub/30/spec/epub30-ocf.html [cited May 31, 2012].

27 Request for Comments: Use Cases & Requirements for Lightweight Content Protection for EPUB. International Digital Publishing Forum; 2012. Available from: http://idpf.org/lcp_draft_reqs_announce [cited May 28, 2012].

28 About Mobipocket. Mobipocket.com;. Available from: http://www.mobipocket.com/en/Corporate/AboutMobipocket.asp [cited January 04, 2013].

29 Nate Hoffelder. RIP: Mobipocket 2000-2011. The Digital Reader; 2011. Available from: http://www.the-digital-reader.com/2011/11/02/rip-mobipocket-2000-2011/#.UOrIvndqOjZ [cited January 04, 2013].

30 Kindle Format 8. Amazon.com, Inc.;. Available from: http://www.amazon.com/gp/feature.html?ie=UTF8&docId=1000729511 [cited January 04, 2013].

31 glazou. iBooks Author, a nice tool but... Glazblog; 2012. Available from: http://www.glazman.org/weblog/dotclear/index.php?post/2012/01/20/iBooks-Author-a-nice-tool-but [cited January 04, 2013].

32 iBooks Author: Publishing and distribution FAQ. Apple Inc.; 2012. Available from: http://support.apple.com/kb/HT5071 [cited January 04, 2013].

33 Uldis Bojars. Use Case Social Annotation. W3C; 2011. Available from: http://www.w3.org/2005/Incubator/lld/wiki/Use_Case_Social_Annotation;http://www.w3.org/2005/Incubator/lld/wiki/index.php?title=Use_Case_Social_Annotation&oldid=5794 [cited January 16, 2012].

34 Sam Costello. Comparing e-Readers: iPad vs Kindle vs Nook. About.com; 2012. Available from: http://ipod.about.com/od/ipadcomparisons/a/comparing-ipad-kindle-nook.htm [cited November 14, 2012].

35 Microsoft PlayReady Content Access Technology White Paper. Microsoft Corporation; 2008. (to be downloaded in docx format). Available from: http://www.microsoft.com/playready/documents/ [cited December 18, 2012].

36 Adobe eBook Platform: Authoring and delivering eBooks across devices?. Adobe Systems Incorporated; 2010. Available from: http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/

products/content-server/pdfs/adobe-ebook-platform-whitepaper.pdf [cited January 30, 2012].

37 Adobe Digital Publishing Solution for eBooks. Adobe Systems Incorporated; 2010. Available from: http://www.adobe.com/digitalpublishing/ebook/ [cited March 14, 2012].

38 Adobe Content Server / Architecture. Adobe Systems Incorporated; 2012. Available from: http://www.adobe.com/products/content-server/architecture.html [cited March 14, 2012].

39 Ed Bott. Closing thoughts on Apple's greedy, “crazy evil” iBooks license. ZDNet; 2012. Available from: http://www.zdnet.com/blog/bott/closing-thoughts-on-apples-greedy-crazy-evil-ibooks-license/4414 [cited March 20, 2012].

40 Adobe Digital Editions Forum - When will Linux version be released?. Adobe Systems Incorporated; 2007. Available from: http://forums.adobe.com/message/1230686 [cited March 20, 2012].

41 Adobe Content Server Forum - ADEPT and Content Server 4 --- MIA?. Adobe Systems Incorporated; 2008. Available from: http://forums.adobe.com/thread/312867 [cited March 02, 2012].

42 Buy Adobe Content Server license. CtrlPublishing Group; 2012. Available from: http://www.ctrlpublishing.com/consultants/products/adobe-content-server/buy-adobe-content-server-license [cited March 02, 2012].

43 Yearly sales Digital publications (1 000 €) 2007-2010. Finnish Book Publishers Association; 2012. (graph generated on the fly in a pop-up by clicking a cell). Available from: http://tilastointi.kustantajat.fi/PublicReporting/Yearly.aspx?reportName=YearlySalesEurDigital.xml&language=ENG [cited March 15, 2012].

44 I♡CABBAGES. Circumventing Adobe ADEPT DRM for EPUB; 2009. Available from: http://i-u2665-cabbages.blogspot.com/2009/02/circumventing-adobe-adept-drm-for-epub.html [cited March 27, 2012].

45 GNU General Public License. Free Software Foundation, Inc.; 2007. Available from: http://www.gnu.org/licenses/gpl-3.0.html [cited January 02, 2013].

46 Arun Ranganathan. Web Cryptography API Use Cases. W3C; 2013. Available from: http://www.w3.org/TR/webcrypto-usecases [cited April 12, 2014].

47 Javascript Cryptography Considered Harmful. Matasano Security; 2013. Available from: http://www.matasano.com/articles/javascript-cryptography [cited April 12, 2014].

48 Matthew Green. The anatomy of a bad idea;.

49 Matt Lee. What is DRM? Digital Restrictions Management. Free Software Foundation, http://www.fsf.org/; 2006. Available from: http://www.defectivebydesign.org/what_is_drm [cited February 24, 2012].

50 Séverine Dusollier. Internet et droit d'auteur. Droit & Nouvelles Technologies; 2001. (in French). Available from: http://www.droit-technologie.org/upload/dossier/doc/50-1.pdf [cited February 22, 2012].

51 DRM. Electronic Frontier Foundation (EFF); 2005. Available from: https://www.eff.org/issues/drm [cited April 04, 2012].

52 Cory Doctorow. Context. San Francisco, California, United States: Tachyon Publications; 2011.

53 Fred von Lohmann. Why DRM on e-Books Will Fail. Electronic Frontier Foundation (EFF); 2009. Available from: https://www.eff.org/deeplinks/2009/06/why-drm-will-fail-e- [cited April 04, 2012].

54 bkrpr.org. e-book ripper. Flickr; 2009. Available from: http://www.flickr.com/photos/bkrpr/3403655734/in/photostream/ [cited April 04, 2012].

55 Closing the book. Microsoft Corporation; 2010. Available from: http://www.microsoft.com/reader/ [cited August 26, 2012].

56 OpinionWay. Baromètre des usages du livre numérique - Vague 2—Septembre 2012. Société Française des Intérêts des Auteurs de l'écrit (Sofia), Syndicat national de l'édition (Sne) and Société des Gens de lettres (SGDL); 2012. (in French). Available from: http://www.opinion-way.com/pdf/barometre_livre_numerique_-_vague_2_-_pour_presentation_orale_vf.pdf [cited November 20, 2012].

57 Julien L . Un éditeur supprime les DRM et devient un leader sur le marché des ebooks. PressTIC / Numerama; 2011. (in French). Available from: http://www.numerama.com/magazine/18904-un-editeur-supprime-les-drm-et-devient-un-leader-sur-le-marche-des-ebooks.html [cited November 27, 2012].

58 Richard M. Stallman. Freedom—or Copyright?. GNU Operating System / Free Software Foundation, http://www.fsf.org/; 2008. Available from: http://www.gnu.org/philosophy/freedom-or-copyright.html [cited April 16, 2012].

59 Philippe Aigrain. Sharing : Culture and the Economy in the Internet Age. Amsterdam, Netherlands: Amsterdam University Press; 2012.

60 eBooks. Springer; 2012. Available from: http://www.springer.com/librarians/e-content/ebooks [cited April 26, 2012].

61 Ebook Usage, Devices, and Formats. O'Reilly Media, Inc.; 2012. Available from: http://shop.oreilly.com/category/customer-service/ebooks.do [cited April 26, 2012].

62 Tor/Forge E-book Titles to Go DRM-Free. Macmillan; 2012. Available from: http://www.tor.com/blogs/2012/04/torforge-e-book-titles-to-go-drm-free [cited April 26, 2012].

63 Charlie Stross. More on DRM and ebooks; 2012. Available from: http://www.antipope.org/charlie/blog-static/2012/04/more-on-drm-and-ebooks.html [cited April 26, 2012].

64 Subterranean Press to publish ebooks through Baen's Webscription service. Baen Publishing Enterprises; 2007. Available from: http://www.baen.com/Press/2007-09-24.htm [cited May 10, 2012].

65 Prior Bundle Statistics. Humble Bundle, Inc.; 2012. Available from: http://support.humblebundle.com/customer/portal/articles/281031-prior-bundle-statistics [cited November 21, 2012].

66 Digital Rights are Civil Rights. European Green Party; 2008. Available from: http://europeangreens.eu/euroarchive/fileadmin/logos/pdf/policy_documents/resolutions/Montreuil/3._Digital_rigths_are_civil_rights.pdf [cited April 17, 2012].

67 Luc Recordon. 12.3326 – Postulat: Vers un droit d'auteur équitable et compatible avec la liberté des internautes. Curiavista, Services du Parlement; 2012. (in French (also available in German and Italian)). Available from: http://www.parlament.ch/f/suche/pages/geschaefte.aspx?gesch_id=20123326 [cited April 16, 2012].

68 Gautam Pant, Padmini Srinivasan, Filippo Menczer. Crawling the Web. In: M Levene, A Poulovassilis, editors. In Web Dynamics: Adapting to Change in Content, Size, Topology and Use. Springer-Verlag; 2004. p. 153--178.

69 Marc Najork. Web Crawler Architecture. In: Ling Liu, M Tamer Özsu, editors. Encyclopedia of Database Systems. Springer US; 2009. p. 3462--3465.

70 Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze. Introduction to Information Retrieval. Cambridge University Press; 2008.

71 research verb - definition in British English Dictionary & Thesaurus - Cambridge Dictionary Online. Cambridge University Press; 2011. Available from: http://dictionary.cambridge.org/dictionary/british/research_2?q=researcher [cited January 10, 2012].

72 library - definition of library by the Free Online Dictionary, Thesaurus and Encyclopedia. Farlex, Inc; 2012. Available from: http://www.thefreedictionary.com/library [cited January 10, 2012].

1 Web Crawler

Pant et al. [68] define **Web Crawlers** as programs that exploit the graph structure of the Web to move from page to page. Crawlers are also known as spiders, robots or simply bots, and in their infancy were also named wanderers, fish or worms. In 1993, the first web crawler, the World Wide Web Wanderer[1], was used to compile statistics about the growth of the web [69]. They quickly became a central component of search engines to collect web pages that are to be indexed (for example the googlebot[2]). Crawlers are used for many other applications such as archiving the web (e.g. Heritrix[3] crawler) or specialized data mining, for example for price comparison services (e.g. the ShopWiki[4] crawler) or searching for copyright violation (e.g. the Digimarc Guardian[5] or the Belgian Librius SINBAD[6] crawlers) and more (e.g. spambot (used to collect email addresses and send unsolicited messages), etc.).

The operations of web crawlers, as described by Manning et al. [70, pp. 405-420], are as follow:

1. The crawler starts with one or more web page addresses constituting the *seed set*, picks one URL[7] from the set and download that web page.
2. The fetched page is then parsed with two objectives:
   a) Extract the text, images, etc. for the data mining process, or in the case of a search engine to fed the indexer.
   b) Extract the links to other pages, files, etc. that are added to a *URL frontier*, corresponding to the resources that have yet to be fetched by the crawler. Initially, the URL frontier contains the seed set.
3. Once the resource has been fetched and parsed, its URL is either removed from the URL frontier or time stamped for a later visit.

Manning et al. [70, pp. 405-420] also list the features web crawler must or should pro-

[1] `http://www.mit.edu/people/mkgray/net/background.html`
[2] `http://support.google.com/webmasters/bin/answer.py?hl=en&answer=182072`
[3] `https://webarchive.jira.com/wiki/display/Heritrix/Heritrix`
[4] `http://www.shopwiki.com/w/Help:Bot`
[5] `http://www.attributor.com/solutions/solutions.php?X=1.1`
[6] `http://www.librius.com/over-librius/wat-doet-librius/Librius-SINBAD/`
[7] Uniform Resource Locator

vide. Ideally, a crawler must be **robust** to avoid spider trap, which are server generating web pages making the crawler downloading million of pages from the same domain and be **polite**. The politeness is both implicit, for example the crawler will wait some time before downloading a page from the same domain to avoid server overloading and bandwidth consumption and explicit by following the Robots Exclusion Protocol define in the robot.txt[8] which set what resources a crawler can or not visit and fetch. Malware and spam robots are know to not follow these rules and some commercial crawler such as the Attributor (now Digimarc Guardian) do not respect the exclusion neither, even if understandably necessary, webmasters may block it more aggressively[9]. The features crawler should have concern efficiency, quality, freshness, scalability, extensibility, etc.

[8] http://www.robotstxt.org/

[9] http://incredibill.blogspot.fi/2007/11/attributor-post-mortem-copyright.html

2 Use Cases

In this chapter will be presented an exhaustive list of actions that can be perform with e-books in form of use cases. First comes the definition of the user roles. These actors are the users who interact with the e-books:

Author A person or a group of person who create (write the text, draw illustration, etc.) a book.

Individual Refers to a single person who read an e-book.

Teacher Is a person whose work is to bring knowledge to others (usually students).

Researcher A person or a group of person running studies on a specific subject in order to reach a new understanding or discovering new information [71].

Publisher A company or a person who publish books and e-books and promote them.

Seller A person or a company who sell or rent e-books.

Library Is an institution who make available to the public by lending cultural, educational and research material, such as books, magazines, CDs, films, etc. A library is also responsible for maintaining such material collection and may also serve for archiving it for future generations [72].

2.1 Detailed Use Cases: Read, Buy, Borrow, Lend, Rent, Offer, Resell, Annotate an E-book

Use case: Read an e-book

Actors: Individual, Teacher, Researcher.

Prerequisite: The user got an e-book either by buying it (cf. 2.1 on the next page) or borrowing it (cf. 2.1 on page 3). He has an appropriate device (computer, tablet, etc.) and there is an application installed on it able to open and display the e-book.

Story: The user has open the reading application on his device. From the application, he clicks the open menu, browses where the e-book file is located and selects it. The

application opens the e-book at its first page. At this stage, the user can start reading. Usually the reading application will also provide to the reader some navigation facilities like moving forward/backward to the next/previews page, going to a specific page, chapter, bookmark (see 2.1 on page 6), etc. Such application can also provide zooming (make the text smaller/bigger) functionalities.

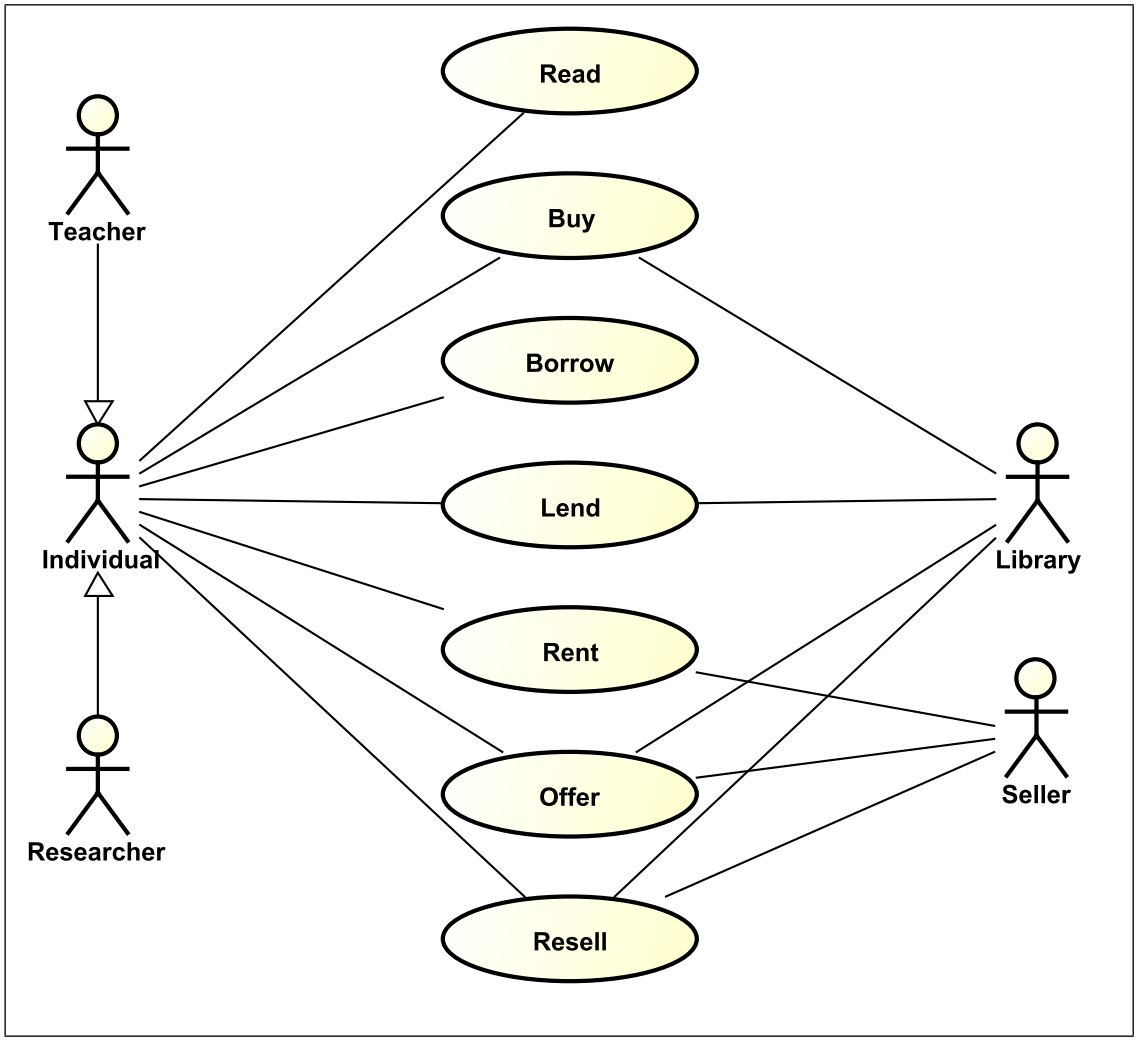

Figure 10: Use case diagram for section 2.1: Read, Buy, Borrow, Lend, Rent, Offer, Resell an e-book.

Use case: Buy an e-book

Actors: Individual, Teacher, Researcher, Library.

Prerequisite: The user has an appropriate device with an internet connection (in here the assumption is that the buying transaction will be done on the internet through an on-line book store) and a web browser. He also disposes of a payment method (through credit card, paypal, etc.).

Story: The user uses his web browser to navigate to the on-line book store. There, he will browse the list of available e-books or use a search facility. Once the user finds the e-book he wants, he orders it (usually by adding it into a shopping cart). By confirming the order, he will be requested for payment (e.g. with credit card). Once the payment is confirmed he will get his e-book through download. The user now owns the e-book and has it as a file on his device.
An alternative would be that the e-book goes on the cloud (cf. 2.2), in that case the e-book file is not downloaded on the user device but is stored on-line.

Use case: Borrow an e-book

Actors: Individual, Teacher, Researcher.

Prerequisite: A lender makes an e-book available (cf. 2.1 on the following page) to the user on-line through a website, by sending it by email, by sharing it with a social network website or by sharing it with a physical storage medium such as a memory card, USB flash drive, etc. The user has an appropriate device (computer, tablet, etc.) and there is an application installed on it able to open and display the e-book.

Story: The user navigates where the e-book is (e.g. the library website, the sharing place of another user, etc.). He specifies the borrowing period (usually a few hours to a couple of weeks) if not predefine by the lender. The user then retrieves (e.g. by downloading) the e-book file. He now has the read access for the borrowing period. At the same time, the lender looses his read access for that same period.
There should be also option for the user so he can return the e-book before the end of the borrowing period and/or a symmetrical option where the lender can request the e-book back.

~~Alternative story for library: A library owns 5 copies of an e-book and all five copies are borrowed. A user does not want to wait that one of the e-books comes back. The library will charge a fee from the user and use that money to get a temporary extra copy from the publisher and lends that extra copy to the borrower. This is not to be confused with the renting (cf. 2.1 on the next page) use case; in here the library does not make any benefit, the fee goes integrally to the publisher for getting the temporary extra copy of the e-book.~~ 2012-01-30 Olli meeting about library DRM => concerning libraries, for the

customers borrowing have to be free of charge!

Use case: Lend an e-book

Actors: Library, Individual, Teacher, Researcher.

Prerequisite: The user owns an e-book. He has a device, where the e-book is stored, with an internet connection or a connector for a removable storage medium. The borrower will have to have the symmetrical device facilities. The lender has an application installed on his device that help him to modify the e-book access rights.

Story: The user changes the rights of his e-book to the lend state. He specifies the lending period (from few hours to couple of weeks) or alternatively provides a lending range (so the borrower will decide how long he will borrow it). The user puts his e-book on-line (through a website, a social network website) or sends it by email or puts it on a physical storage medium. Once the borrower gets the e-book file, he instantly gets the read access to it and at the same time, the lender looses his read access and his right to lend it (to avoid concurrent lending or renting (cf. 2.1) of the same e-book) for that same lending period.
There can be an option where the lender can request the e-book back before the end of the lending period and/or an option for the borrower to return it earlier.

Use case: Rent an e-book

Actors: Seller, Individual, Teacher, Researcher.

Prerequisite: The owner of the e-book has a device, where the e-book is stored, with an internet connection or a connector for a removable storage medium. The renter will have to have the symmetrical device facilities. The owner has an application installed on his device that help him to modify the e-book access rights.

Story: The owner changes the rights of his e-book to the rent state. He specifies the renting period (typically from few hours to couple of weeks). Once the renter pays the fee, he gets (by downloading or through removable storage medium) the e-book on his device. He has now the read access to the e-book for the renting period. At the same

time, the owner looses his right to read and rent the e-book (to avoid concurrent renting or lending (cf. 2.1 on the previous page)) for that same period. **TO CHECK: DOES THE OWNER HAS TO PAY A SPECIAL TAX TO PUBLISHER/AUTHOR OF THE E-BOOK?**

Alternative story: The seller (owner) can rent as many copies of the e-book he wants. In that case he will pay to the publisher (or the author) an extra fee per rented copies. The idea here is to propose a business approach similar to VOD (Video On Demand).

Use case: Offer an e-book

Actors: Individual, Teacher, Researcher, Seller, Library.

Prerequisite: The owner of the e-book has a device, where the e-book is stored, with an internet connection or a connector for a removable storage medium. The receiver will have to have the symmetrical device facilities. The owner has an application installed on his device that help him to modify the e-book access rights.

Story: The user modifies the ownership of the e-book to be the receiver. He then offers the e-book to that person. He looses all the rights on the e-book. The receiver now owns the e-book and has the full rights (read, lend, etc.) on it.
For a library, if they want to reduce storage (like save server disk space), they can offer e-books that were only rarely borrowed.
For a seller, he can offer e-book for good customer or make special offers like buy ten e-books and pay for nine.

Use case: Resell an e-book

Actors: Individual, Teacher, Researcher, Seller, Library.

Prerequisite: The owner of the e-book has a device, where the e-book is stored, with an internet connection or a connector for a removable storage medium. The buyer will have to have the symmetrical device facilities. The owner has an application installed on his device that help him to modify the e-book access rights.

Story: Once the user gets the money payment from the buyer, he modifies the ownership

of the e-book. He then transmit (through internet, email or with a removable storage medium) the e-book to the buyer. The buyer now owns the e-book and has the full rights on it. At the same time, the ancient owner looses them.
Library sometimes resell books that were rarely read (or broken); so something similar may append with e-books. That allow them to free some space (in this case, disk space). For a seller, we can image a system similar to a second hand shop where they buy e-books from individuals and resell them.

Use case: Annotate an e-book

As define in the W3C Incubator Use Case Social Annotation [33], is consider as an annotation: a bookmark, line mark that the reader can use to quickly navigate to a specific part of the e-book (as example remember where he stop reading to restart reading from there); a comment about a sentence, paragraph, chapter, the all e-book; a summary or a description of the e-book or part of it; etc.

Actors: Researcher, Teacher, Individual.

Prerequisite: The owner of the e-book has a device, where the e-book is stored, with a reading application that help him to add bookmark. He may also have another application that allow him to write text (a notepad), record voice, etc.

Story: The user reads or browses the e-book to a specific place (page, line, chapter, etc.) and mark it. By saving this mark, he already has a bookmark. He then write his comment in a text editor or record his voice comment. He optionally add a title or a description of his comment. By aggregating the comment with the bookmark, he has created an annotation.

2.2 List of Other Use Cases

The use cases were the very start point of this work. The goal of it was to get an as complete as possible overview of all the possible actions that can be done with an e-book in its larger definition (the file, application and/or device). While relevant, they did not feet well in the thesis itself and went into this annex. When that decision was took, the formal writing of the use cases was stopped. The table 1 presents the others cases that are not

described in the section 2.1. The description of the use cases is using keywords only and could be harder to read.

Table 1: Short version of use cases

Use Case	Description
Share annotations of an e-book	Prerequisite: The user has created an annotation (cf. 2.1 on the preceding page). -internet connection - personal website, social network website, etc. Stories: shares his annotations - accesses and uses others annotations
Copy an e-book	backup, archive, transfer on multiple devices (e.g. using a USB stick)
Copy (part of) the content of an e-book	for a quotation, for a review, etc.
Offer a copy of an e-book	Like offer but the original owner keep a copy of the book. Note: Require author consent [14, Article 4.1] or not e.g. if under any of the creative commons licenses (even the NC).
Sell a copy of an e-book	Same as offer; but with money transaction. It almost become a parallel distribution. Note: Require author consent [14, Article 4.1] or not e.g. if under some of the creative commons licenses (all that are not NC).
Print an e-book	Does that become a physical book?
Offer printed copy of (part of) an e-book	E.g. for teacher, as teaching material
Sell printed copy of an e-book	To check if illegal?
Convert an e-book to another format	E.g. some device/application require a specific format (can not read the others)
Read on multiple devices	E.g. user start to read in the train on mobile phone and continue home on the computer. Requires to have copies on multiple devices. Synchronization would be nice to have (continue to read on the next device where stopped on the previous).
Read on multiple devices concurrently	E.g. Wife on the mobile phone and husband on the computer. Requires to have copies.
publisher / vendor cases	Sell Return to shop for resale (after reading, keep a copy) DRM-free after entering the public domain Access after publisher/seller bankrupt Track/trace preview and other promotion mechanism
Archive	As a backup, keep a copy on another machine/disk in case of crash or as a pure archiving for the future generation. Similar to the copy case.

Use Case	Description
Store a copy of an e-book on the cloud	keep a copy on a distant server that can easily be accessed by different devices over the internet. Similar to copy and archive cases.
anonymously buy/own/borrow an e-book	E.g. if you are interested in religion and science, you do not want to be blocked at the border because you have read the Koran and chemistry books.
Use as teaching/research material	More because it is one of the exception [14, Article 5.3(a)]. Requires the copy content, print cases and more.
Share library	Offer a collection of e-books. Or share the list of e-books that other users could borrow.
E-book swapping	similar to the book swapping, once read, offer the e-book to a friend in exchange of one of his e-book.
BookCrossing	Require a physical book. BookCrossing a reading device with e-books on it?

3 Source Code

The `genkey_and_export.html` HTML file in listing 7 is the starting point of the application where the user is presented the action of generating the public and private key pair, export the public key by calling the `keydecrypt.js` and shares it with the distant server by sending it to the `encrypt.php`.

```
1  <!doctype html>
2  <html>
3  <head>
4  <title>Generate Key Pair and Export</title>
5  <meta charset="utf-8">
6  <script src="polycrypt/common/util.js"></script>
7  <script src="polycrypt/front/polycrypt.js"></script>
8  </head>
9  <body>
10 <h1>Test Web Cryptographic API - JavaScript Generate Key
       Pair &amp;amp; Export</h1>
11 <p>This application test the possible future <a href="http
       ://www.w3.org/TR/WebCryptoAPI/" target="_blank">W3C Web
       Cryptography API</a> standard with the following scenario
       :</p>
12 <ol>
13 <li>The user will run the JavaScript to generate the public/
       private key pair (if they don't already exist).</li>
14 <li>The user share his/her public key.</li>
15 <li>The server, will use that public key to encrypt data
       specifically for the user.</li>
16 <li>The user will run the JavaScript to decrypt the message
       with his/her private key.</li>
17 </ol>
18 <p>From <a href="https://dvcs.w3.org/hg/webcrypto-usecases/
       raw-file/tip/Overview.html" target="_blank">W3C uses
       cases</a>, this application demonstrate: KEYGEN-ASYM (
       with 2048 RSAES-PKCS1-v1_5 key pair), EXPORT (in JSON Web
        Key (JWK) format), DECRYPT(-ASYM). the IMPORT and
       encryption part (ENCRYPT-ASYM) are done in <a href="http
       ://www.php.net/manual/en/book.openssl.php" target="_blank
       ">php</a>.<br>
19 If the browser don't support the Web Cryptography API, it
       will use the <a href="http://polycrypt.net/" target="
       _blank">PolyCrypt</a> library as the fallback.<br>
20 The sample code (under MIT License) Copyright (c) 2013 <a
       href="https://github.com/nvdbleek/web-crypto-samples"
```

```
      target="_blank">Nick Van den Bleeken</a> was used as the
      starting point. The <a href="https://github.com/nvdbleek/
      web-crypto-samples/blob/master/js/lib/base64.js" target="
      _blank">base64.js</a> file is from him and parts of his <
      a href="https://github.com/nvdbleek/web-crypto-samples/
      blob/master/js/utils.js" target="_blank">utils.js</a> and
       <a href="https://github.com/nvdbleek/web-crypto-samples/
      blob/master/js/encrypt.js" target="_blank">encrypt.js</a>
       scripts are derived into my keydecrypt.js script.</p>
<p id="logMsg"></p>

<h2>1. Generate public/private keys</h2>
<form>
  <button id="genKey" type="button">Generate Keys</button>&
     nbsp;<span id="time_warn" style="color:orange;"></span>
</form>

<h2>2. Send the public key for receiving the encrypted
   message</h2>
<form action="encrypt.php" method="post">
  <textarea id="public_key" name="public_key" readonly rows=
     "14" cols="80">Generate keys first.</textarea><br>
  <input type="submit" name="send" value="Send Public Key"
     id="send_key" disabled>
</form>
<input type="hidden" id="decrypt">
<script src="base64.js"></script>
<script src="keydecrypt.js"></script>
</body>
</html>
```

Listing 7: Generates keys and exports - HTML file

The `keydecrypt.js` JavaScript file in listing 8 is responsible for the operation on the client side. It generates the public and private keys pair, exports the public key and decrypts the content of files with the private key.

```
/*
 * With help from the Web cryptography API samples
 * Copyright (c) 2013 Nick Van den Bleeken
 * https://github.com/nvdbleek/web-crypto-samples
 */

/*
    ************************************************************

 * Web Cryptography API EBook
 * Copyright (c) 2014 Patrick Ausderau
 *
```

```
11  * This program is free software: you can redistribute it
       and/or modify
12  * it under the terms of the GNU General Public License as
       published by
13  * the Free Software Foundation, either version 3 of the
       License, or
14  * (at your option) any later version.
15  *
16  * This program is distributed in the hope that it will be
       useful,
17  * but WITHOUT ANY WARRANTY; without even the implied
       warranty of
18  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
        the
19  * GNU General Public License for more details.
20  *
21  * You should have received a copy of the GNU General Public
        License
22  * along with this program.  If not, see <http://www.gnu.org
       /licenses/>.
23  ************************************************************
       */
24 var metropolia = metropolia || {};
25 metropolia.keydecrypt = (function() {
26   var keydecrypt = {
27     version : "1.0"
28   };
29
30   //log to html
31   var msg = document.getElementById('logMsg');
32   //key pair
33   var publicKey;
34   var privateKey;
35   //decrypt
36   var resultArray;
37   var blocCount;
38
39   //utils
40   str2ab = function (str) {
41     var buf = new ArrayBuffer(str.length);
42     var bufView = new Uint8Array(buf);
43     for (var i=0, strLen=str.length; i<strLen; i++) {
44       bufView[i] = str.charCodeAt(i);
45     }
46     return buf;
47   }
48
49   ab2str = function (buf) {
50     return String.fromCharCode.apply(null, new Uint8Array(
         buf));
```

```
51  }
52
53  //browser native or polycrypt fallback
54  var webCrypto;
55  var jwkAsObject = false; // Some implementations want the
        jwk as an object
56  if (window.crypto && window.crypto.subtle) {
57    console.log('Using standard Web Crytography API.');
58    msg.innerHTML += 'Using standard Web Crytography API.<br
          >';
59    webCrypto = window.crypto.subtle;
60  } else if (window.msCrypto && window.msCrypto.subtle) {
61    console.log('Using MS Web Crytography API.');
62    msg.innerHTML += 'Using MS Web Crytography API.<br>';
63    webCrypto = window.msCrypto.subtle;
64  } else {
65    console.log('No native Web Crytography API, falling back
          to polycrypt.');
66    msg.innerHTML += 'No native Web Crytography API, falling
          back to polycrypt.<br>';
67    webCrypto = window.polycrypt;
68    jwkAsObject = true;
69  }
70
71  keydecrypt.generateKeyPair = function() {
72    console.log("key gen?");
73    msg.innerHTML += 'Go to key pair generation<br>';
74    //generate public-private keys
75    if(!(localStorage["fi.metropolia.key.public"] &&
         localStorage["fi.metropolia.key.private"])){
76      document.getElementById("time_warn").innerHTML = "
           Please wait. The key pair generation may take from
           0.2 to more than 5 seconds.";
77      var genOp = webCrypto.generateKey({
78        name : "RSAES-PKCS1-v1_5",
79        params: {
80          modulusLength : 2048,
81          publicExponent : new Uint8Array([ 0x01, 0x00, 0x01
              ])
82        }
83      }, true, [ "encrypt", "decrypt" ]);
84      genOp.onerror = function(e) {
85        console.log('Error generating key pair');
86        console.log(e);
87        msg.innerHTML += 'Error generating key pair<br>';
88        document.getElementById("time_warn").innerHTML = "";
89      }
90      genOp.oncomplete = function(e) {
91        publicKey = e.target.result.publicKey;
92        privateKey = e.target.result.privateKey;
```

```
93          if(publicKey && privateKey){
94            localStorage["fi.metropolia.key.public"] = JSON.
                 stringify(publicKey);
95            console.log(publicKey);
96            //kids, never do that at home. the private key
                 should be stored securely.
97            localStorage["fi.metropolia.key.private"] = JSON.
                 stringify(privateKey);
98            msg.innerHTML += 'key generated: ';
99            for(i = 0; i < 12; i++){
100             msg.innerHTML += publicKey.key.substring(i*100,
                   (i*100)+99) + '<br>';
101           }
102           msg.innerHTML += '<br>';
103           //Export the public key
104           var exportOp = webCrypto.exportKey("jwk",
                 publicKey);
105           exportOp.onerror = function(evt) {
106             console.log('Error exporting public key');
107             msg.innerHTML += 'Error exporting public key<br>
                   ';
108             document.getElementById("time_warn").innerHTML =
                    "";
109           }
110           exportOp.oncomplete = function(evt) {
111             console.log('Exported public key');
112             console.log(evt.target.result);
113             msg.innerHTML += 'Exported public key<br>';
114             //var n = evt.target.result.n.replace(/=/g, '').
                   replace(/\-/g, '+').replace(/_/g, '/');
115             var n = JSON.stringify(evt.target.result);
116             console.log(n);
117             localStorage["fi.metropolia.key.public.export"]
                   = n;
118             //localStorage["fi.metropolia.key.public.export
                   "] = evt.target.result;
119             document.getElementById('public_key').innerHTML
                   = n;
120             //document.getElementById('public_key').
                   innerHTML = evt.target.result;
121             document.getElementById('send_key').disabled =
                   false;
122             document.getElementById("time_warn").innerHTML =
                    "";
123           }
124         }else{
125           console.log('Error generating key pair?!?');
126           console.log(e);
127           msg.innerHTML += 'Error generating key pair<br>';
```

```
128             document.getElementById("time_warn").innerHTML = "
                    ";
129           }
130         }
131       }else{
132         msg.innerHTML += 'key already exist<br>';
133         document.getElementById('public_key').innerHTML =
                localStorage["fi.metropolia.key.public.export"];
134         document.getElementById('send_key').disabled = false;
135         document.getElementById("time_warn").innerHTML = "";
136       }
137     };
138
139     keydecrypt.decrypt = function(data) {
140       console.log('decrypt');
141       console.log(data);
142       if (privateKey == null) {
143         console.log('not in cache, try from local storage');
144         msg.innerHTML += '<br>Private key not in cache, try
                from local storage';
145         privateKey = JSON.parse(localStorage["fi.metropolia.
                key.private"]);
146       }
147       if (privateKey == null) {
148         console.log('this time we have a problem :D');
149         msg.innerHTML += '<br>Private key lost :(';
150         return;
151       }
152
153       //TODO fix polycrypt
154       var alg;
155       if (jwkAsObject) {
156         alg = "RSAES-PKCS1-v1_5"
157       }
158       else {
159         alg = { name : "RSAES-PKCS1-v1_5" };
160       }
161       var blocSize = 344;
162       var i;
163       var blocData = [];
164       resultArray = [];
165       blocCount = 0;
166       document.getElementById("time_warn").innerHTML = "Please
              wait. The asynchronous decryption process has
             started, it will take few seconds.";
167       document.getElementById("decrypt_progress").value = 0;
168       for(i = 0; i < data.length/blocSize; i++){
169         blocData.push(data.slice(i * blocSize, (i + 1) *
                blocSize));
170       }
```

```
171     console.log(blocData);
172     var decryptOp = new Array(blocData.length);
173     document.getElementById("decrypt_progress").max =
           blocData.length;
174     //decrypt the data
175     for(i = 0; i < blocData.length; i++){
176       (function(closureI){
177         decryptOp[closureI] = webCrypto.decrypt(alg,
              privateKey, new Uint8Array(Base64Binary.
              decodeArrayBuffer(blocData[closureI])));
178         //console.log("ALIVE? " + closureI)
179
180         decryptOp[closureI].onerror = function(evt) {
181           console.log('Error decrypting data');
182           msg.innerHTML += '<br>Error decrypting data :(';
183           blocCount++;
184           document.getElementById("decrypt_progress").value
                = blocCount;
185           document.getElementById("decrypt_progress").
                innerHTML = '(' + blocCount + '/' + blocData.
                length + ')';
186           if(blocCount === blocData.length)
187             decryptedToBlob();
188         }
189
190         decryptOp[closureI].oncomplete = function(evt) {
191           decryptedData = evt.target.result;
192           blocCount++;
193           document.getElementById("decrypt_progress").value
                = blocCount;
194           document.getElementById("decrypt_progress").
                innerHTML = '(' + blocCount + '/' + blocData.
                length + ')';
195
196           if (decryptedData) {
197             //console.log('Decrypted data: ' + closureI);
198             //msg.innerHTML += ' Decrypted data: ' +
                  closureI + ' (' + blocCount + '/' + blocData.
                  length + ')';
199             resultArray[closureI] = ab2str(decryptedData);
200             document.getElementById('decrypted').innerHTML
                  += '\nDecrypted data: ' + closureI + ' (' +
                  blocCount + '/' + blocData.length + ')\n' +
                  resultArray[closureI];
201           } else {
202             console.log('Error decrypting data 2');
203             msg.innerHTML += '<br>Error decrypting data :(';
204           }
205           if(blocCount === blocData.length)
206             decryptedToBlob();
```

```
207           };//decrypt
208         })(i) //closure
209       }
210
211     };
212
213     function decryptedToBlob(){
214       console.log(resultArray);
215       document.getElementById("time_warn").innerHTML = "";
216       //hack to transform relative path to absolute hard-coded
              URL.
217       var tmp = [];
218       var j, hack;
219       for(var i = 0; i < resultArray.length; i++){
220         tmp = resultArray[i].split("=\"");
221         if(tmp.length > 1){
222           hack = "";
223           for(j = 0; j < tmp.length - 1; j++){
224             hack += tmp[j] + "=\"";
225             if((tmp[j].endsWith("href") || tmp[j].endsWith("
                    src")) && !tmp[j+1].startsWith("http://") && !
                    tmp[j+1].startsWith("#"))
226               hack += "http://users.metropolia.fi/~patricka/
                      wcrypt-book/moby-dick/OPS/";
227             console.log("tmp: " + tmp[j]);
228           }
229           hack += tmp[j];
230           console.log("hack: " + hack);
231           resultArray[i] = hack;
232         }
233       }
234       //array to blob to be used as the iframe source.
235       var blob = new Blob(resultArray, {type : 'application/
             xhtml+xml'});
236       document.getElementById("decrypted_blob").src = URL.
             createObjectURL(blob);
237     }
238
239
240     // Hook up event listeners
241     document.getElementById('genKey').onclick = function() {
          metropolia.keydecrypt.generateKeyPair()};
242     document.getElementById('decrypt').onclick = function() {
          metropolia.keydecrypt.decrypt(document.getElementById('
          encrypted').innerHTML)};
243
244     return keydecrypt;
245   })();
```

Listing 8: Generates keys, exports public key and decrypts - JavaScript file

The `encrypt.php` PHP file in listing 9 handles the server side operations. It imports the user public key and uses it to encrypt a EPUB file content that is sent back to the user for decryption and rendering.

```
1  <!--
2  /*
       ************************************************************************

3  * Web Cryptography API EBook
4  * Copyright (c) 2014 Patrick Ausderau
5  *
6  * This program is free software: you can redistribute it
       and/or modify
7  * it under the terms of the GNU General Public License as
       published by
8  * the Free Software Foundation, either version 3 of the
       License, or
9  * (at your option) any later version.
10 *
11 * This program is distributed in the hope that it will be
       useful,
12 * but WITHOUT ANY WARRANTY; without even the implied
       warranty of
13 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
        the
14 * GNU General Public License for more details.
15 *
16 * You should have received a copy of the GNU General Public
        License
17 * along with this program.  If not, see <http://www.gnu.org
       /licenses/>.
18 ************************************************************************
       */
19 -->
20 <!doctype html>
21 <html>
22 <head>
23 <title>Encrypt and Decrypt</title>
24 <meta charset="utf-8">
25 <script src="polycrypt/common/util.js"></script>
26 <script src="polycrypt/front/polycrypt.js"></script>
27 </head>
28 <body>
29 <h1>Test Web Cryptographic API - PHP Encrypt and JavaScript
       Decrypt Message</h1>
30
31 <p id="logMsg">The message is encrypted on server side with
       PHP.<br>The key has been converted to PEM format from <a
       href="http://www.php.net/manual/en/function.openssl-pkey-
```

```
      get-public.php#104439" target="_blank">ppostma1</a> hack
      .<br></p>
32
33
34  <?php
35    $pub_key = $_POST["public_key"];
36    if($pub_key){
37      ?>
38      <h2>1. Message has been encrypted with your public key</
          h2>
39      <p>If done properly, you are the only one who can
          decrypt this message.</p>
40      <pre><?php
41      $key_elem = json_decode($pub_key, true);
42      var_dump($key_elem);
43      //Ugly hack to transform the public key to PEM format
          because it works only with a 2048 bits key
44      //from http://fi1.php.net/manual/en/function.openssl-
          pkey-get-public.php#104439
45      $prepa = wordwrap('MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A'.'
          MIIBCgKCAQEA'.str_replace(array("_", "-", "="), array
          ("/", "+", ""), $key_elem["n"]).'ID'.$key_elem["e"],
          64, "\r\n", true);
46      $key = <<<EOF
47  -----BEGIN PUBLIC KEY-----
48  $prepa
49  -----END PUBLIC KEY-----
50  EOF;
51      $res = openssl_pkey_get_public($key);
52      print_r(openssl_pkey_get_details($res));
53      echo "\n";
54      //content to be encrypted is in a file outside the www
          folder.
55      //the epub file has been "unzipped" and the next/
          previous content is known from its package.opf
          configuration file
56      $list_of_content = array();
57      $current_file = $_POST["current_file"];
58      if(!$current_file)
59        $current_file = 0;
60      if($_POST["next"])
61        $current_file++;
62      elseif($_POST["prev"])
63        $current_file--;
64      $prev_file = true;
65      $next_file = true;
66      $tmp;
67      foreach(file("../../secret/moby-dick/OPS/package.opf")
          as $line){
68
```

```
69        if(strpos($line, '<item id="') !== false && strpos(
              $line, '.xhtml') !== false ){
70          $tmp = substr($line, strpos($line, 'href="') +
                strlen('href="'), strpos($line, '.xhtml') +
                strlen('.xhtml') - (strpos($line, 'href="') +
                strlen('href="')));
71          //normal files (preface, chapters,...)
72          if(strpos($line, '<item id="x') !== false)
73            array_push($list_of_content, $tmp);
74          //"special" files (cover, title,...)
75          elseif(strpos($line, '<item id="cover"') !== false)
76            $list_of_content[0] = $tmp;
77          elseif(strpos($line, '<item id="titlepage"') !==
                false)
78            $list_of_content[1] = $tmp;
79          elseif(strpos($line, '<item id="copyright"') !==
                false)
80            $list_of_content[2] = $tmp;
81        }
82      }
83      if($current_file <= 0)
84        $prev_file = false;
85      elseif($current_file >= count($list_of_content) - 1)
86        $next_file = false;
87
88      $write_file = "moby-dick/OPS/".$list_of_content[
            $current_file];
89      $resfile = fopen($write_file, "w");
90      //smart quotes don't get encrypted nicely :(
91      //$search = array(chr(145), chr(146), chr(147), chr(148)
            , chr(151));
92      //$replace = array("'", "'", '"', '"', '-');
93      //123 * 16 = 1968bits = max to encode with a 2048 key
94      foreach(str_split(file_get_contents("../../secret/moby-
            dick/OPS/".$list_of_content[$current_file]),123) as
            $val){
95        //the quote and other special char don't get encrypted
              very friendly
96        //$val = str_replace($search, $replace, $val);
97        openssl_public_encrypt($val, $encrypted, $res);
98        fwrite($resfile, base64_encode($encrypted));
99      }
100     fclose($resfile);
101     echo "</pre>";
102     echo "Encrypted file: <a href=\"$write_file\" target=\"
            _blank\">$write_file</a>";
103     ?>
104     <form>
105        file content:<br>
```

```
106         <textarea id="encrypted" name="encrypted" readonly
               rows="14" cols="80"><?php
107           echo file_get_contents($write_file);
108         ?></textarea><br>
109         <button id="decrypt" type="button">Decrypt</button>&
               nbsp;<span id="time_warn" style="color:orange;"></
               span>
110       </form>
111
112       <h2>2. Your file get decrypted with your private key</h2
             >
113         <progress id="decrypt_progress" value="0" max="5"></
               progress><br>
114         <textarea id="decrypted" cols="125" rows="20" style="
               border: 1px solid green"></textarea>
115         <form action="encrypt.php" method="post">
116           <textarea id="public_key" name="public_key" readonly
                 rows="0" cols="0" style="visibility: hidden;
                display: none;"><?php echo $pub_key; ?></textarea
                >
117           <!--input type="hidden" id="public_key" name="
                public_key" value=""-->
118           <input type="hidden" id="current_file" name="
                current_file" value="<?php echo $current_file; ?>
                ">
119           <input type="submit" value="< Previous" name="prev"
                id="prev" <?php echo $prev_file?"":"disabled";
                ?>>       <input type="
                submit" value="Next >" name="next" id="next" <?
                php echo $next_file?"":"disabled"; ?>>
120         </form>
121         <input type="hidden" id="genKey">
122       <h2>3. The result reordered and human readable</h2>
123         <iframe id="decrypted_blob" style="width: 60em; height
               : 1390px; border: 2px solid green;"></iframe>
124       <script src="base64.js"></script>
125       <script src="keydecrypt.js"></script>
126       <?php
127     }else{
128       ?>
129       <p>Seems that you didn't sent your key? <a href="
             genkey_and_export.html">Try again</a></p>
130       <?php
131     }
132
133   ?>
134   </body>
135   </html>
```

Listing 9: Imports public key, encrypts content - PHP file