DIGITAL RIGHTS MANAGEMENT FOR ELECTRONIC DOCUMENTS

ZHU BAO SHI

(M.Eng. Shanghai Jiaotong University, PRC)

(B.Eng. Shanghai Jiaotong University, PRC)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Wu Jiankang, for his valuable advise from the global direction to the implementation details. His knowledge, kindness, patience, open mindedness, and vision have provided me with lifetime benefits.

I am grateful to Prof. Mohan Kankanhalli for his dedicated supervision, for always encouraging me and giving me many lively discussions I had with him. Without his guidance the completion of this thesis could not have been possible.

I'd also like to extend my thanks to all my colleagues in the Institute for Infocomm Research for their generous assistance and precious suggestions on getting over difficulties I encountered during the process of my research.

This thesis draws a period for my 20-year education in schools. In addition to my teachers and classmates over the past years, I must thank my parents without whose love and nurturing I could never accomplish all these. Lastly, but most importantly, my deepest gratitude to my wife Jiayi, for her love, support and encouragement during our years in Singapore. I dedicate this thesis to her.

Table of Contents

Summary

Digital Rights Management (DRM) controls and manages rights for digital media. In the second generation of DRM, the definition of *rights* has been extended from digital rights to “all form of rights usages over both tangible and intangible assets – both in physical and digital form – including management of rights holders’ relationships.” because of pressing needs from real applications such as e-commerce and e-government.

As in the first generation definition which emphasizes on copyright, previous research efforts on DRM focus more on the copyright protection for electronic publishing. This thesis follows the second generation definition, addressing DRM issues for electronic documents in business and administrative environment. The “rights management” poses requirements of security and interoperability. The security requirement mainly concerns authentication and access control for both electronic and paper documents; while the interoperability requires a system to maintain trusted relationship among different parties by means of describing, identifying, trading, protecting, monitoring and tracking rights usages among these parties. Based on the requirements, we have proposed and developed three key novel techniques for the second generation DRM system:

(i) Authentication method for electronic documents. The method contains a digital watermark scheme and a content-based authentication technique for elec-

tronic documents. The watermark scheme utilizes the render sequences of characters. It features large information carrying capacity and robustness over document format transcoding. The authentication method is based on the NP-complete Exact Traveling Salesman Problem, which provides strong cryptographic security with short key length.

(ii) Authentication method for printed paper documents. The method utilizes the inherent non-repeatable randomness existing in the printing process. The randomness of the printing signature of a particular character or pattern results in unique features for each printed document. By registering and verifying these features, we authenticate content integrity and originality of printed documents. The authentication methods for both electronic and printed documents together solve the security requirement for the DRM system.

(iii) Model and framework for XML based access control for electronic documents and document source data. The access control model implements traditional role-based access control using XML language, with syntactic and semantic language specification and validation based on XML Schema and XML Schematron. The core permissions are described using extended ODRL standard. Adhering to a trusted access control model leads to a sound theoretical background, and adopting XML language increases the interoperability in multi-user environment. The access control model is further integrated into a complete DRM framework with security features for both electronic and paper documents.

List of Tables

List of Figures

Chapter 1

Introduction

The understanding of Digital Rights Management (DRM) has been constantly evolving since its first introduction in the 1970s. So far, the most up-to-date, comprehensive and well-accepted definition of DRM was suggested by Iannella of IPR Systems in the W3C (World Wide Web Consortium) Digital Rights Management workshop in 2001:

> “Digital Rights Management (DRM) involves the description, identification, trading, protection, monitoring and tracking of all forms of rights usages over both tangible and intangible assets – both in physical and digital form – including management of rights holders’ relationships. [Ian01]”

This definition is often referred to as the “second-generation of DRM”, whereas its ancestor, the “first-generation of DRM”, focuses on using security and encryption techniques to solve the issues of unauthorized copying and distribution of digital contents. It is now much clear that the “first-generation DRM” is more related to the “digital copyright management” than “digital rights management”. It is more based on traditional security-encryption-enforcement views. The second generation extends DRM to cover all forms of rights usages over both tangible and intangible assets – both in physical and digital form, and the management process includes the description, identification, trading, protection, monitoring and tracking. It is "digital management of rights", as opposed to "management of digital rights". In other words, DRM manages *all* rights, not only the rights applicable to permissions over digital contents.

The complete framework of DRM system contains both technical and non-technical (commercial, social and legal) aspects of rights management [oAP00, RTM01]. The commercial aspect deals with business and marketing activities, e.g., the pay-per-use versus subscription pricing model. The social aspect deals with customer education and the concept of fair use (the right to use copyrighted material without permission in certain cases). The legal aspect deals with statutory and contractual enforcement of digital rights. In this thesis, we only tackle the technical aspect of the DRM. However, the non-technical aspect remains an indispensable part to form an effective and end-to-end rights management system.

1.1 Motivation

Research activities in the digital rights management for electronic documents have been growing due to its commercial potential. It has been estimated the DRM market for electronic documents will reach $3.5b by the year 2005 [RTM01, PDF01]. However, adoption of electronic documents into any serious business and administrative transactions is very limited due to the unavailability of effective means for managing rights and usages.

Let us look at an example where a shipper consigns with a shipping company to ship some goods from port A to port B. They are required to comply with international regulations, customs, and special treatments of different shipped materials. The process is very document intensive. Various documents involved include invoices, packaging lists, certificate of origin, quality inspection certificates, letter of credits, bill of ladings, etc. Digital rights management system tries to establish a trust relationship among all the parties involved by managing these documents and controlling their usages. To achieve this, DRM system must be *interoperable* and *secure*. We now look into more detailed requirements on stages of the document management workflow.

- *Interoperability:* The interoperability requirement applies at the stages of document creation and deployment. It requires direct data exchange among different parties involved in transactions. These parties are legally independent companies, physically located at various locations, each may have their own computer systems running different software packages, with different databases, and using different data exchange format such as EDI or XML. Inability to interoperate may lead to manual processing of data. Here in the document domain, manual processing includes deploying documents by means of re-typing or DA/AD conversions such as printing, scanning, and optical character recognition (OCR). These conversions are very inefficient and error prone.

- *Security:* The security requirement can be further viewed as consisting of access control, authenticity and originality requirements.

  - *Access control:* Access control applies at the stages of document creation and deployment. It describes a set of policies for each party to access the documents. For example, a policy to allow certain internal documents be viewable by the shipping company but not the shipper.

    It also provides enforcement mechanisms to ensure all parties are complying with the policies.

  - *Authenticity:* Authenticity applies at all the stages of document management. It requires that the documents used in the transaction are genuine in terms of the contents and appearance. For example, the packaging list must be the one properly verified and signed by the authorized personnel.
  - *Originality:* Originality applies at the stage when the documents have been distributed to the end users. It requires a method to make sure that the documents are original rather than being duplicated, even though the contents are genuine. The originality requirement is particularly important for business and administrative documents, such as the bills of lading: claiming of goods with a duplicated copy is not allowed.

Techniques in the existing electronic products and services cannot meet all the requirements. The reasons include:

- Access control methods with XML based rights mark-up standards are still immature. Currently, all rights mark-up languages have been designed for media and electronic publishing industry where only access control policies for end-user are addressed. Use of these languages in business domain with respect to document creation and multi-level deployment security has not been studied and verified. Therefore, exchange of sensitive data electronically among untrusted parties is still a major concern.

- It is difficult to authenticate electronic documents while allowing data format transcoding. Traditional digital signature schemes do not work here.

For example, a shipping company located in Singapore uses A4 paper size to format all electronic documents and generates digital signatures to authenticate the documents. But a shipper located in USA requires Letter paper size. So the electronic documents sent from A to B must be reformatted. In this case the authenticity of digital signatures is voided. A more robust and content-related authentication method is hence needed.

- There is no absolute way to prevent electronic documents from being duplicated, and the duplication of electronic documents always has 100 percent perfect fidelity. As a result, justifying the originality of electronic documents is not possible. Instead, paper documents with hand-signatures are used in many circumstances. However, verifying the originality of machine generated paper documents, especially printed paper documents, remains a challenge to the research community.

In short, the requirements on managing (the description, identification, trading, protection, monitoring and tracking of) all forms of rights usages over both tangible and intangible assets – both in physical and digital form make the DRM problem much more intricate. Achievements in technologies of protecting digital contents in the past decades have little adoption by business and administrative applications so far. It may due to major concerns on the right management issues regarding interoperability and security. In this thesis we shall address these DRM issues and propose possible solutions.

1.2 Problem statement

The challenging issue that we are addressing is digital rights management for electronic documents. We concentrate our research on the management of documents for business and administrative purpose, with emphasis on interoperability, authenticity and originality. We do not address the copyright protection, which usually is not a problem in this particular domain. However, some of our research results are actually applicable to copyright protection.

We further state the issues as follows:

1. Maintain document authenticity while allowing data format transcoding.

   Data format transcoding is inevitable if the document is to be shared by heterogeneous computer systems. It is one of the major building blocks for multi-system interoperation.

2. Preserve document authenticity when an authentic electronic document is printed onto paper, uniquely identify printed original paper document, and detect its duplication.

   It is well known that paper documents are still legal instruments for most business and administrative transactions by the law. Authentication of printed paper documents is hence vital to build an end-to-end rights management system.

3. Develop an integrated DRM system framework which provides ready solutions to applications in the field of e-government and e-commerce.

   This includes system modeling, rights definition and access control mechanisms, etc.

1.3 Contribution of the thesis

Having studied the whole document flow, including its creation, processing, approval, deployment, archival and verification, and the digital rights management roles ("the description, identification, trading, protection, monitoring and tracking") in this flow, we have designed a system framework with respect to the technical aspect of digital rights management for electronic documents. Three key issues have been identified and novel methods have been developed as solutions to the three issues:

1. A document watermark and authentication method for electronic documents.

   We have developed a novel watermark scheme for electronic documents which hides information into the document during document formatting. The hidden information survives document format transcoding. Data regarding to the rights description of the document can be embedded into document using the watermark scheme. We also propose a document authentication method based on the watermark. With this method, document authenticity is maintained in an interoperable environment.

2. A document authentication method for printed paper documents.

   We have developed a novel authentication method for printed paper documents. Our method can prevent unauthorized modification or duplication of authentic printed documents. With authentication methods for both electronic documents and printed paper documents, the DRM system is complete with regard to "all forms of rights usages over both tangible and intangible assets".

3. An XML-based access control and application framework.

   We define XML based access control framework to ease document creation and exchange. The framework is based on the “role-based access control (RBAC) ” model, which provides a sound theoretical foundation. We have developed a novel implementation method to describe definitions and constraints in RBAC using pure XML technologies such as XML Schema and XML Schematron. Base the model, we integrated the proposed document authentication methods into the framework to form a complete DRM system.

These three solutions address the security and interoperability requirements in the document deployment, end-user printing and creation stages respectively, as shown in Figure 1.1: During document creation, the XML based access con-

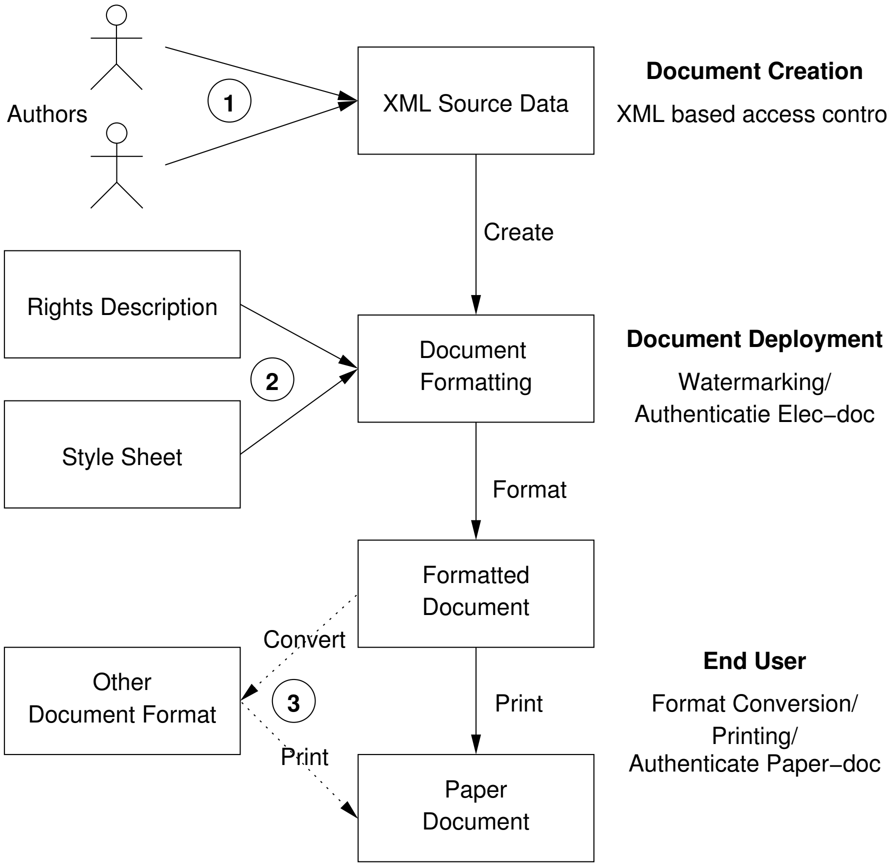


Figure 1.1: Proposed solutions in document workflow

trol framework manages author's access rights to the XML data source, which enables exchanging of idea and data within a secure and trusted environment (1). After the data source has been finalized, a document formatting system formats the data into human readable document, according to a style sheet. In this process, descriptions about the access rights to the document are embedded into the document using document watermark scheme. The watermark also serves as authenticity evidence to protect the rights descriptions and document contents. The watermarked electronic document is final version for deployment (2). When the electronic document reaches the end user, the user can either print it onto paper, or store the electronic version for archival. For the first case, our authentication method for printed paper documents can protect the paper document from unauthorized modification or duplication, thus bridges the authenticity from electronic domain to physical (paper) domain. For the second case, even though the electronic document is to be converted into other formats, the document watermark scheme guarantees that the embedded information is still preserved (3).

It can be concluded from the above workflow that the three key solutions enable rights protection along the whole life cycle of electronic documents. They manage rights over both "tangible and intangible assets – both in physical and digital form".

1.4 Overview of the thesis

We discuss related works on DRM system architectures in Chapter 2. In Chapter 3, we proposed the watermark scheme and authentication method for electronic documents, followed by an authentication method for printed paper documents in chapter 4. Chapter 5 discusses XML based access control and DRM framework.

The thesis is concluded in Chapter 6.

Chapter 2

Background

We, in this chapter, review some previous works regarding digital rights management. Our review follows three major directions: the authentication methods for electronic documents, the authentication methods for paper documents, and the frameworks and implementations of DRM systems. These works are closely related to the security and interoperability requirements of DRM system for electronic documents. They collectively form the background of our research topic.

2.1 Authentication and watermark schemes for electronic documents

Authenticity is one of the essential requirements contributing to the security of the DRM system for electronic documents. Authenticating electronic documents has been a subject of research in both cryptography and multimedia community. A general model of the authentication problem is depicted in Figure 2.1 [MV99]. Transmitter Alice transmits a message X to receiver Bob. The message is transmitted through an open channel, where Carol is capable of viewing and modifying

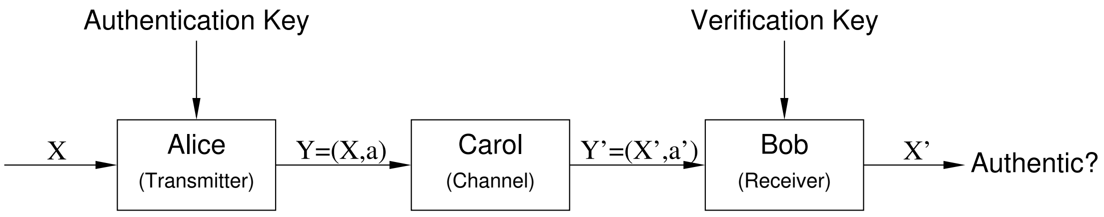


Figure 2.1: Authentication model

the message. In order for Bob to be assured that the message is indeed originated from Alice and Carol has not modified it, Alice computes an *authentication tag* (or *authenticator*) a, attaches it to the message X to form message Y. The computing of a is based on the *authentication key*, which is kept secret by Alice. When Bob receives the message, he can verify, using the *verification key*, that a is a valid authenticator for message X. Note that the verification key here can be either public, which constitutes *public verification*, or secret to receiver Bob, which constitutes *private verification*.

In the typical cryptographic perspective, Carol is considered as a malicious attacker. Her role is trying to create a fake message $Y' = (X', a')$ which she hopes that Bob would accept as authentic and originating from Alice. Digital signature schemes and message authentication code (MAC) [MvOV97] can effectively keep Carol out of the game. But problem rises when Carol is not malicious. For example, to serve the interoperability purpose as discussed in section 1.1, Carol can be sort of document format conversion software, who converts documents sent from Alice into the specific format that Bob accepts. Since Carol does not know the authentication key, she cannot just convert the document and re-create the authenticator a. Instead, she must create $Y' = (X', a)$, with $X' \neq X$ and Y' still acceptable by Bob. The problem is, how to design an authenticator a which authenticates both X and X'. We refer to this problem the *authenticator problem*.

How to associate the authenticator a with the message X to form Y is another problem that draws great interests from multimedia research community. Simply appending a to the end of X or storing it inside the file header is not a viable solution because the authenticator can always be easily removed. A more preferred solution is to embed authenticator a into message X itself, therefore extending the authentication capability to the large number of existing document formats that do not provide any explicit means of including an authenticator (for example, the industrial standard PostScript format) [MV99]. Another advantage of doing so is that it would be very convenient for the authenticator to survive document format transcoding. This partially solves the authenticator problem as well. However, how to embed information into electronic documents still remains a problem. We refer to this problem the *embedding problem*.

The authenticator problem and the embedding problem have attracted tremendous research activities in the recent decades. So far, the most widely adopted solutions are *content-based authentication* and *digital watermark*, respectively.

2.1.1 Content-based authentication

In content-based authentication, the authenticator is generated from the contents of the message, rather than the binary representation of the message. By doing so, the authenticator exhibits certain robustness that it keeps valid regardless of whatever formats or transformations the message undertakes, provided that the message content remains unchanged. This fundamentally solves the authenticator problem. Obviously, defining and extracting of contents from the message is the foremost task. As one example, in digital image domain, Bhattarcharjee [BK98] suggests the use of feature points such as edge maps in image data as the definition for image contents. Adjustments made to the image, for example, brightening, alteration of contrast, lossy compression or format transcoding will not change the edges so that the content is unchanged. However, this method is not satisfactory since it is highly probable that two distinct images have very similar edge maps (human faces, for example). Increasing the type of feature points does not solve the problem. The underlying reason is that the word “content” is itself very abstract and subject to individual’s perception. Content extraction for multimedia data is still an unsolved problem in spite of enormous advances in image understanding techniques [MV99].

Comparatively, content definition and extraction for text-based electronic documents is much easier. This is because text data have lower bandwidth and hence less abstract level (considering that the computer understands the word “apple” far better than a picture of an apple). Contents can be extracted by direct analyzing the text. For business and administrative documents, the use of structured text mark-up languages such as XML further eases content definition because it eliminates the needs for semantic natural language understanding. These favorable properties make content-based authentication for electronic documents very practical. It is natural to consider using digital signature schemes or message authentication codes onto text data as the solution to the authenticator problem. However, this solution is not applicable alone without solving the embedding problem.

2.1.2 Digital watermark

Digital watermarking has been an active research area for nearly 50 years [CM01]. It is the process of embedding some information (payload) into digital content (host) such that the payload can later be extracted or detected. Watermark schemes solve the embedding problem by treating message X as the host and authenticator a as the payload. The embedding of information is generally achieved through manipulating redundant information in the host data [BGNL96]. Redundant information presents either in the human perceptual system [CM97] or in the structure of the message [Sim98]. It is well know that multimedia data contain plenty of redundant information. For example, the least significant bit (LSB) for each pixel in digital image is considered redundant because changes made to these bits are not noticeable by human eyes. This simple property leads to a series of image watermark and authentication schemes, such as the Yeung and Mintzer's fragile watermark authentication scheme [YM97]. More advanced multimedia watermark schemes include the spread-spectrum scheme [TRvS$^+$93, vSTO94, WD96, CKLS97, WD97] for digital image, the echo-hiding scheme [GBL96] for digital audio, etc. All these schemes have been well studied in both theoretical and practical perspectives. Some excellent reviews on watermark schemes for multimedia data can be found in [PAK99, PD01, BJ97, DMH98, SHG98, HK99]. Despite these achievements, watermark schemes for text-based electronic documents have been lagging behind with respect to quantity and quality. This is due to the fact that redundant information in text data is rare and hard to explore, and any modifications to text content are easily noticeable even by casual readers [BGNL96]. In the following we focus our discussion on watermark schemes for electronic documents only.

Watermark schemes can be classified according to different criteria, which are listed in Table 2.1. For the authentication model shown in Figure 2.1, the watermark scheme is used to embed the authenticator into the document. It must be public watermark scheme, because the verifier does not know anything about the

Criterion	**Classification**
Visibility	Visible watermark Invisible watermark
Method of payload insertion	Additive watermark Quantize and replace watermark
Domain of payload insertion	Transform domain watermark Spatial domain watermark
Method of detection	Private (non-oblivious) watermark – requires original message Public (oblivious) watermark – does not require original message
Robustness	Robust watermark – survives manipulation Fragile watermark – detects manipulation

Table 2.1: Classification of watermark schemes

original document. It must be robust against format transcoding, but sensitive against unauthorized modifications. Being visible or invisible is not important for authentication purpose, but if visible, the watermark shall not interfere with the contents. Being spatial domain and quantize/replace watermark can reduce the processing complexity and the size of the document. They are preferred but not mandatory. We now review some existing document watermark schemes and examine what classes they belong to.

Existing watermark schemes for electronic documents contains two kinds of approaches: one based on the modification of the layout or appearance of the document, and the other based on the modification of the text.

Layout and appearance watermark

In layout and appearance watermark schemes, the layout of the text or the page image is altered based on the payload. These schemes are applicable both electronic documents and paper documents. In the decoding process, the paper documents must be digitized first, then the alterations are detected.

Line shift encoding. Line shift encoding algorithm was first introduced in [BLMO94], and further developed in [BLMO95b, BLMO95a, LMBO95, ML97, low98, LML98, BLM99]. In this approach, a payload is embedded into the document image by vertically displacing an entire text line. In the decoding process, the digital image of a page is obtained and the baseline or centroid of each line is calculated using horizontal profile. The distance between two adjacent lines is then measured. Since a document's initial line space is uniform, the presence or absence of a payload can be detected by analyzing the measured distances without knowing the original document image.

Theoretically, a paragraph of n lines can hold a payload of n bits. But in a real implementation, differential encoding technique [BLMO95a] is used, in which all odd text lines are kept unmoved, and even lines are either shifted up, moved down or unmoved to represent information {-1, +1, 0}. Differential encoding technique can greatly improve the accuracy of the decoding process, but at the same time it will cut the information carrying capacity by about 70%. A payload of about $0.7n$ bits can be embedded in an n-line page (e.g., 10 bits in an A4 page with double spaced 12 point font).

Experiments show that line shift encoding will survive several generations of photocopying successfully [BLMO94]. But an attacker can easily defeat it by re-spacing lines either uniformly or randomly. Since the information carrying capacity is small, embedding the authenticator using line shift encoding is very insecure. There exists non-negligible possibility that a randomly re-spaced document contains a valid authenticator (e.g., 1/1024 in the above example).

In conclusion, line shift encoding belongs to the category of invisible, public and robust watermark. It may be useful for copyright protection. But it does not meet the requirements for content-based document authentication.

Word shift encoding. Word shift encoding was introduced together with line shift encoding in [BLMO94, BLMO95b, BLMO95a, LMBO95, ML97, low98, LML98, BLM99]. This method alters a document image by horizontally shifting words within text lines to encode a payload. It features much larger information carrying capacity than line shift encoding. But since most document formatting tools use variable spaces between words to justify text, the decoding process will need the original document to determine which word has been shifted.

An attacker can eliminate the embedded information by re-spacing shifted words. In most cases this kind of attack requires much more manual interventions than attacking line shift encoded documents, because it is generally hard to do segmentation of words automatically and properly within the mixture of different fonts, symbols and equations. Word shift encoding features the same robustness as line shift encoding.

In conclusion, word shift encoding belongs to the category of invisible, private and robust watermark. It may also be useful for copyright protection. Since it is a private watermark scheme, copyright assertion must resort to trusted third-party who has access to the original document. This creates more complex issues about the proof of original, which are out of the scope of this chapter. Word shift encoding does not meet the requirements for content-based document authentication either.

Feature encoding. Feature encoding is the third method introduced in [BLMO94, BLMO95b, BLMO95a, LMBO95, ML97, LML98, BLM99]. The document image is examined for chosen text features, and those features are altered or not altered depending on the payload. Some possible choices of text features are the upward, vertical end-lines of letters – for example the tops of letters `b`, `d`, `h`, etc. These end-lines are altered by either extending or shortening their length.

An attacker will have to identify which text feature and which letters are altered in order to perform a successful attack. Obviously it has to be done manually with reference to the unaltered fonts.

Feature encoding has the same robustness as line shift encoding and word shift encoding. It is invisible, and can be considered as public watermark scheme. But the watermark detection requires a large number of altered and unaltered letters for comparison. It greatly limits the information carrying capacity.

A secure electronic publishing trial was run on October, 1995 by IEEE Communications Society (COMSOC). The issued journal *IEEE Journal on Selected Areas in Communications* contains unique digital watermark using the above mentioned methods for each recipient. The purpose of using watermark is to discourage and track illegal dissemination of the documents, but not to authenticate the documents. A report of this trial is in [Bra96].

Character spacing width sequence coding. Character spacing width sequence coding was introduced in [Cho99]. It addresses the problem that word shift encoding is not applicable to Asian languages such as Chinese, Japanese or Thai that do not have sufficiently large space as word boundary. This method alters the horizontal space between adjacent characters to encode information. The decoding process will need the original document image if the unencoded characters are not uniformly spaced, e.g., the Thai characters.

Character spacing width sequence coding has the same pros and cons of the three previous methods. Further more, for languages such as Chinese or Korean whose characters as well as spaces between characters are all fixed, a watermarked document will be quite distinguishable and suspicious.

High resolution watermarking. A document is created to have two or more components, with one of the components representing a watermark object or a background object. A high-resolution pattern is embedded in the watermark or background object, so that it is not detectable by human eye but recognizable by a special purpose device [Ada99]. The high-resolution pattern can carry information relating to the creation and controlling of the document, signatures, etc. Detection of the pattern does not require the original document so it is public watermark scheme.

The patent [Ada99] says that the high-resolution pattern is non-removable by attacks such as photocopying and scanning. But in fact a photocopier with low resolution will just blur everything on the image, thus erase the high-resolution patterns. So this method is not likely to be as robust as it suggests.

Noise placement encoding. In noise placement encoding [Max94], information is inserted in a document by adding a noise signal that is barely visible. Noise is least noticeable when it occurs at natural boundaries in an image like the edge of letters. Based on this phenomenon, two different set of fonts are designed which look alike but differ in a small number of positions. In the unencoded document the fonts are randomly selected for each character. In the encoded document, the font that has been selected for the unencoded document is switched to another or not, to transmit a bit of information.

Noise placement encoding does not survive printing or photocopying. It has large information carrying capacity since theoretically each character can hold 1 bit of information. But the detection of embedded information requires the original document for font comparison, so noise placement encoding is private watermark scheme. It is not suitable for document authentication.

Conclusion. Layout and appearance watermark schemes treat electronic documents as binary images, and try to embed data by modifying inconspicuous details in the images. They are invisible watermark schemes. Extraction of embedded data can be either public or private. Public watermark schemes usually have less information carrying capacity than private ones.

All of these watermark schemes have been proposed for copyright protection. They assume that the attacker will use image processing software packages or photocopies to remove the watermark. Such assumption is very limited. It should be noted that Optical Character Recognition (OCR) system can be used to defeat all layout and appearance watermark schemes by converting the document images back to text and re-formatting the text files. In [BLMO95a], the author argues that OCR technology does not always recognize characters correctly, and the current technology used to reconstruct a document is imperfect. There also exist special techniques which beguile OCR systems into giving incorrect outputs [CB03]. However, with human assistance, OCR attack is always possible. In fact this attack is widely used for book piracy in East Asian countries.

Layout and appearance watermark schemes are not suitable for document authentication. This is due to the fact that none public schemes have sufficient information carrying capacity to hold the content-based authenticator.

Text watermark

In text watermark, the text contents of electronic documents are altered based on the payload. The literature provides three categories of text watermark schemes: the open space watermark, the syntactic watermark and the semantic watermark.

Open space watermark. Open space watermark scheme [BGNL96] is based on the fact that changing the number of trailing spaces has little chance of changing the meaning of a phase or a sentence, and a casual reader is unlikely to take notice of modifications to the white spaces. This method embeds some spaces after each terminating character (e.g., a period), at the end of each line, or in the margin. Those appended spaces together can represent some payload information. Since spaces are invisible, open space watermark is invisible watermark scheme. The extraction of payload is done by counting extra spaces, so it is public watermark scheme. Open space watermark is useful as long as text remains in ASCII format, even copy-and-paste operation can not remove the payload.

Syntactic watermark. There are many circumstances where punctuation is ambiguous or when mis-punctuation has low impact on the meaning of the text contents. For example, “bread, butter, and milk” and “bread, butter and milk” are both considered correct usage of commas in list syntax [BGNL96, KH02]. This creates some flexibility in expressing the same idea. Each version of the sentence can be used to express distinct information about the payload. Syntactic watermark is private watermark scheme, because the extraction of payload needs the original text for comparison. It is invisible watermark, but inconsistent use of punctuation is noticeable, and there are cases where changing the punctuation will impact the clarity, or even the meaning of the text. This method should be used with caution [BGNL96].

Semantic watermark. Semantic watermark is similar to the syntactic watermark. It substitutes words in the text using their synonyms selectively. For example, *big* can be substituted with *large*, or *A.M.* can be substituted with *a.m.* [BGNL96, Nie99, KH02]. Assigning each synonym substitution with a value, then the set of all substitutions can be used to identify the payload. For the same reason as syntactic watermark, semantic watermark is invisible and private watermark scheme. However, the nuance of meanings of synonyms can cause problem under different context. This method should also be used with caution too.

Conclusion. Text watermark are invisible watermark schemes. Like layout and appearance watermark schemes, they are also proposed for copyright protection originally. Since the alterations are made to the text contents, text watermark can survive OCR attacks. However, changing punctuations or substituting words requires manual processing, which render text watermark schemes very ineffective.

Text watermark schemes are not suitable for document authentication either. The insufficient information carrying capacity constitutes one reason. The other reason is that changing punctuations or substituting words are not always applicable in electronic documents. For critical documents such as those for business and administrative purpose, every word may have significant importance. Improper substitution can lead to disastrous consequences.

2.1.3 Discussion

A complete solution to the authentication problem contains both the solution to the authenticator problem and the solution to the embedding problem. These two problems have been well studied for the multimedia data, but not for electronic documents. Although the authenticator problem is easily solvable in this case using cryptographic means, the embedding problem presents some unique challenges. This is mainly due to the lacking of redundant information in electronic documents which greatly limits the information carrying capacity of the embedding process. It is difficult for any existing schemes to hold a simple authenticator of several hundred bits into a page of text data, not to mention other auxiliary data such as rights descriptions. Thus, there is imperative need to develop new watermark schemes in order to solve the embedding problem.

2.2 Authentication methods for printed documents

In business and administrative environment, *authenticity* and *originality* are the two basic requirements for any paper document to be considered valid. In the traditional paper-based world, when a document is generated, it is usually signed / issued / approved by one or more authorized persons, with their signatures or seals to show the authenticity. The document with original signatures is considered to be original, authentic or legitimate. In the printed world, there are also requirements for such signatures to show the authenticity and originality of a document. Existing techniques towards meeting these requirements can be categorized into four classes: the *use of special materials*, *fingerprints*, *digital encoding*, and *visual* *cryptography / optical watermark.*

2.2.1 Use of special materials

These solutions are based on either physical means or chemical means, such as special high-resolution (>4000dpi) printers not available in the open market, special papers/inks that are very sensitive to re-produce [Bor93, KY00, Gre87, GJ00, Gre00, Zei00], and hologram labels [CJ89]. By controlling the availability of these materials, no forgery or duplication of the document is possible. However, due to the high cost of both the equipment and the efforts for controlling their use, these solutions are only used in applications which have strict security requirements, such as currency notes, checks, etc.

2.2.2 Fingerprints

The idea of fingerprinting is to make each copy of a document unique so that illegal copies are identifiable, or the person who made illegal copies is traceable. This idea was first introduced by Wagner in [Wag83], and then developed for various applications. In [NWK93], nonuniformities in disk medium are utilized as fingerprint to discourage illegal copying of files. In [Bra02], the width of each strip cut produced by a shredder is identified as the fingerprint, which in turn is used to trace the particular shredder that has been used. As for paper documents, Métois et al. [MYSS02] have proposed an identification system based on the naturally occurring inhomogeneities of the surface of paper. A special purpose imaging device is developed to capture the texture and fiber pattern of the paper. The pattern is then registered as a unique fingerprint for later retrieval and comparison. Physical fingerprints usually offer strong protection against duplication attempts.

However, the medium is not content-related. So the integrity of the contents is not protected. Furthermore, the identification of typically invisible fingerprint often requires special devices. This inevitably increases the cost of the system. As a result, these methods are only used in applications which emphasize more on medium security than content integrity, such as checks, tickets, etc.

2.2.3 Digital encoding

Originating from cryptographic theory, these approaches intend to transfer digital signature onto paper documents. Such approaches include bar codes [PSW90, PSW92] and information hiding (notably digital watermarking) [CKLS97, RG98] techniques. These methods add some machine readable information onto the document to serve as a digital signature. Only authorized persons have access to the secret information required to generate the digital signature, so the authenticity of the document is protected. However, since the information is machine readable, it can also be copied or scanned using photocopiers or scanners. The originality of the document is not protected effectively. Digital encoding methods have been widely used in applications which require machine based authentication, such as bills, ID cards, and so on.

2.2.4 Visual cryptography / optical watermark

Visual cryptography utilizes secret sharing to split a graphical pattern into different pieces in a manner that the pattern becomes visible if and only if the shares are stacked together [NS94, Sha96]. By doing this, a paper document with one share printed can be validated visually using the remaining shares. Optical watermarks is an improvement over visual cryptography in terms of the ability to hide multiple layers of graphical information and enhanced visual quality with easy alignment [HW00]. Both visual cryptography and optical watermark have been designed for manual authentication of documents. They are most suitable in applications where the convenience of verification is important such as brand protection, ticketing, etc. However, both of these techniques cannot disprove the authenticity of a photocopy or scanned-copy of an original document.

2.2.5 Discussion

We summarize the pros and cons of the existing authentication techniques for paper documents using Table 2.2. It is obvious that none of these techniques can satisfy all the security requirements for electronic documents. The inherent shortcomings of existing authentication techniques have limited their applications to niche areas. Developing a new technique suitable for business and administrative document processing is therefore imperative.

Techniques	**Authenticity**	**Originality**	**Content-related**	**Cost**
Special materials	Yes	Yes	No	Very high
Fingerprinting	Yes	Yes	No	High
Digital encoding	Yes	No	Yes	Low
Visual cryptography & optical watermark	Yes	No	Yes	Low

Table 2.2: Existing techniques for authenticating printed documents

2.3 Frameworks and implementations of DRM systems

A complete framework of DRM system for electronic documents contains not only techniques for authenticating electronic and paper documents, but also techniques for controlling the exchanging and sharing of documents among different users. Interoperability and access control are the two major requirements in the implementation of the framework. Here we review some representative works done by other researchers and companies regarding these topics.

2.3.1 Access control models and implementations

Computer systems provide access control to data and resources for reasons of integrity and confidentiality. The fundamental model of access control is suggested by Lampson in [Lam74], where the very nature of *access* suggests that there is an active *subject* assessing a passive *object* with some specific *access operations*, while a *reference monitor* grants or denies access, as shown in Figure 2.2. In the document management system, access control enables controlled sharing and exchanging of documents among users. Here the subjects are users and objects are documents. On the most elementary level, the access operations for documents may contain two types: *observation* and *alteration*. We review two most important access control models and their implementations:

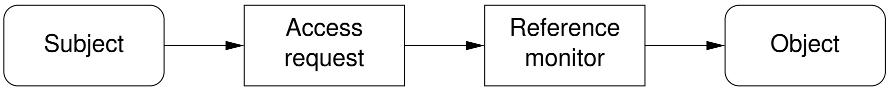


Figure 2.2: The fundamental model of access control

Access control matrix (ACM)

In the following, we refer to

- a set S of subjects,
- a set O of objects,
- a set A of access operations.

The access control matrix model defines access rights in the form of a matrix

$$M = (M_{so})_{s \in S, o \in O} \quad \text{with} \quad M_{so} \subset A \quad \text{[Lam74]}$$

The entry M_{so} specifies the set of access operations subject s may perform on object o. The access control matrix could hardly be implemented directly because otherwise the system must store a huge matrix that is very difficult for maintenance. Instead, the system stores the access rights either with the subjects or with the objects. In the first case, the access rights assigned to a subject constitute the subject's *capability*, and the corresponding access model is called *capabilities model*. In the second case, an access control list (ACL) stores the access rights to an object within the object itself, the corresponding model is called *access control list* model.

Access control matrix has the following shortcomings:

- It is difficult to get an overview of who has the access rights to a given object (for capabilities model), or what objects can a given subject access (for ACL model). Such query generally requires enumerating all objects or subjects to give an answer.
- For capabilities model, it is difficult to revoke a capability.
- For ACL model, it is difficult to revoke a subject's access rights.

Access control matrix model allows the creator of an object to assign access rights to other subjects. This is often referred to as the *Discretionary Access*

Control (DAC). DAC is the basis of intellectual property protection, wherein the author of a work automatically becomes the copyright owner of that work, and is able to decide who others can access that work. DAC does not fit the security needs for business and administrative transactions. It permits dishonest users to disclose confidential information to others. What we need is a system-wide policy decrees who is allowed to have access. Such access control policy is called *Mandatory Access Control* (MAC).

Role-based access control (RBAC)

Because of the obvious shortcomings of ACM, people developed intermediate access control models which use the concept of *groups* to group a set of subjects together, and then assign access policies to the groups. Such models include *Groups and negative permissions*, *Protection rings*, *Privileges* and *Role-based access control* (RBAC) [Gol00]. We here focus on RBAC since it "addresses many of the security needs of both the commercial and government sectors [NIS99]".

In many organizations, the access control decisions are often determined based on the employee functions, such as the specification of duties, responsibilities and qualifications. RBAC mimics these decisions. A role can be thought of as a set of procedures that a user or a set of users can perform in an organization. Here the term procedure refers to the binding of specific access operations and objects. In the context of RBAC literature, such procedures are called *permissions*. For example, the role of tellers in a bank is to execute a savings deposit transaction, requiring observation and alteration operations to the specific fields within a savings database. Here the permission is observation and alteration to the specific fields in the database. Similarly, the role of a customer in a bank contains the permission to observe the savings record owned by the customer himself. User assigned to a role derives the permissions of the role. A user can be assigned to multiple roles, thus obtains the union of permissions contained by all roles. A role can also have multiple users. Revocation of certain permissions from a user is done by excluding the user from the corresponding roles. Through this way, RBAC simplifies the management of access rights by introducing an intermediate layer between S and $A \times O$. The classification of roles and the assignment of users to roles is done by the system administrator in compliance with organization-specific protection guidelines [FK92]. It implements system-wide policy, so RBAC is an instance of MAC.

There are many RBAC models proposed in the literature, among which the NIST Standard RBAC Model [SFK00] has been proposed as a standard base for other RBAC models. Main components of NIST RBAC model include users, roles, role-hierarchy, permissions, user-role assignment and permission-role assignment. Figure 2.3 shows the details of the model:

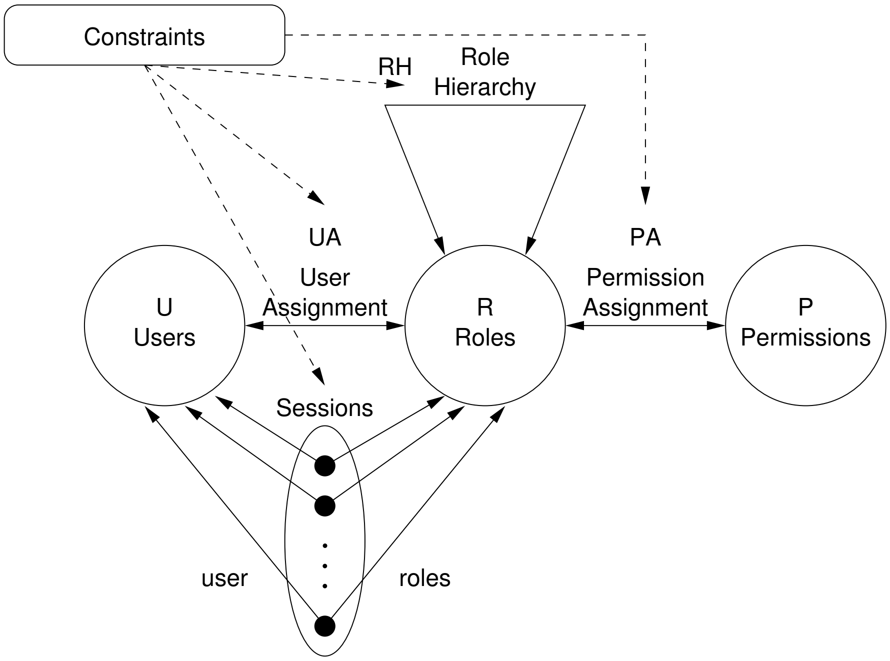


Figure 2.3: NIST RBAC model

In this model, there are sets: U (Users), R (Roles), S (Sessions), P (Permissions)

$UA \subseteq U \times R$: user-role assignment.

$PA \subseteq P \times R$: permission-role assignment.

$RH \subseteq R \times R$: a partial order of role hierarchy.

$user : S \rightarrow U$: a function mapping a session to a user.

$roles : S \rightarrow 2^R$: a function mapping a session to a set of activated roles.

$permissions : R \rightarrow 2^P$: a function mapping a role to a set of assigned permissions: $permissions(r) = \{p : P \mid (r, p) \in PA\}$

$permissions^* : R \rightarrow 2^P$: a function mapping a role to a set of activated permissions considering role hierarchy: $permissions^*(r) = \{p : P \mid \exists\, r' \leq r \cdot (r', p) \in PA\}$

There also exist some constraints in the constructions of UA, RH, PA and S which are application specific. The basic constraints are the *cardinality constraint* (controls the number of roles or users), the *principle of least privilege constraint* (requires that a user be given no more privilege than necessary to perform a job) and *separation of duties constraint* (no single individual be allowed to have all permissions).

For a user u to successfully gain permission p, the following conditions must be met:

1. $\exists\, s : S \;\cdot\; user(s) = u$, the user has started a session.
2. $ar = roles(s) \neq \emptyset$, the session contains a non-empty set of activated roles ar.
3. $p \in permissions(ar)$ or $p \in permissions^*(ar)$, the desired permission p has been assigned to the activated roles or the activated role hierarchy.

RBAC is the best candidate for the DRM system for electronic documents in business and administrative environment. This is because theoretically RBAC provides a sound and verified background for solid access control, and practically RBAC requires only minimum adjustments to the existing document management workflow during implementation. However, there have been no existing standards for the description and verification of RBAC policies. It is generally considered that the use of XML as a platform-independent data exchange format to implement RBAC models is the best solution to achieve standardization. For example, Ramaswamy in [Cha00] proposed the use of XML and associated APIs to construct a RBAC framework. He developed an XML Document Type Definition (DTD) for representing the RBAC model, which is capable of describing the user, role and role hierarchy components. But Ramaswamy's method stops at "describing" level. It does not support the enforcement of RBAC constraints. Another recent effort in implementing RBAC model is done by the Organization for the Advancement of Structured Information Standards (OASIS)[1], who has set up an *eXtensible Access Control Markup Language (XACML)* technical committee for developing XML-based frameworks and specification for access control. The next version of XACML will include a set of RBAC profile which has not yet been finalized. The current working draft [And03] of XACML RBAC profile has defined a set of elements and attributes for the description of RBAC roles, permissions, SSD and DSD components. But it falls into the same pit of being able to "describe" only. A complete implementation of RBAC should support not only the descriptions of all RBAC components, but also validations to the descriptions.

[1] `http://www.oasis-open.org`

2.3.2 Rights expression languages

Rights expression language (REL) is used to describe the specific rights owned by certain subjects over certain objects. It is closely related to the permission component in RBAC model, except that REL also describes the prerequisites for the subject to get the permission, such as making payments. A formal model of rights expression languages is introduced by Gunter in [GWW01]. In this model, the term *rights* is captured as a set of *licenses*. Each license is composed of several *realities*, which are described as "rendering work w on device d at the time t with payment x". Gunter's model is among the few theoretical works regarding formal modeling of rights expression languages. The model is heavily bound to the B2C electronic media distribution. It emphasizes on revenue model instead of information confidentiality and integrity, hence not suitable for business or administrative purpose.

More general purpose rights expression languages have been proposed by companies who are developing proprietary systems. Most current languages use standard content formats such as XML to provide an extensible core set of semantics and vocabulary. Among these languages, eXtensible rights Markup Language (XrML) [Con02] and Open Digital Rights Language (ODRL) [Ian00] have been the most advanced which meet the MPEG-21 requirements [ISO01].

XrML

XrML [Con02] can be dated back to 1996, when Mark Stefik of Xerox's Palo Alto Research Center published his idea about a "trusted system" in [Ste96]. The paper proposed that a trusted system would require machine-readable languages for defining the access procedures. Stefik published his first rights management language Digital Property Rights Language (DPRL) using Lisp language, and later changed to DPRL 2.0 using XML due to the requests from electronic publishing community. After that, Xerox and Microsoft launched a company called Content-Guard [2], who made further revision to DPRL 2.0 and renamed it to XrML.

The core element in XrML is *grant*, which contains *principal*, *right*, *resource* and *condition* elements connected using the clause "*grant principal* the *right* to access to the *resource* under *conditions*". Version 2.0 of XrML has defined 24 different rights that can be conferred on content, and the number is still growing.

Although the language has been in existence for a few years, XrML is slow to be adopted by DRM technology vendor. The main complaint is that XrML appears to be too complex, making it difficult to implement, especially in those Internet devices with low computational power and small memory footprints. Another complaint is that XrML standard is Microsoft-backed rather than an open standard. In fact, Microsoft is the only vendor to date to ship products based on XrML.

ODRL

Iannella of IPR System [3] thinks that XrML and other languages have "predominately taken a closed approach to solving problems" [Ian00], and have "focused on the content protection issues more than the rights management issues". She developed an entirely new XML-based rights management language named "Open Digital Rights Language (ODRL)" in the spirit of open source software and published it for use without restriction.

The ODRL language focuses on a simple, extensible rights-management model

[2] http://www.contentguard.com

[3] http://www.iprsystems.com

that encompasses a small set of core entities named *assets*, *rights* and *parties*. The foundation model in ODRL can be expressed using *offers* and *agreements*. *Offers* are proposals from *rights holders* (one entity of *parties*) for specific *rights* over their *assets*. *Agreements* are when *parties* enter into contracts or deals with specific *offers*. The model can also express revoking of any *offers* or *agreements* [Ian02].

The competition among XrML and ODRL has not ended till today. XrML was accepted by the ISO/IEC MPEG standards body for the developing MPEG-21 media distribution standards, and ODRL was chosen by the Open Mobile Alliance [4] (was the WAP Forum) as its rights language for all mobile content.

Both XrML and ODRL are designed to maximize the flexibility in rights description. Such flexibility is required for the diversified licensing models in digital media marketing activities, for example, the subscription model and the pay-per-view model. For business and administrative document management workflow, the rights associated with each operator seldom change as long as the actual workflow has been implemented and put into daily use. It is somewhat overkill to incorporate the entire XrML/ODRL specification into such cases.

Despite their flexibility, there are also some limitations preventing XrML/ODRL to be used for business and administrative documents:

1. Both XrML/ODRL identify the object being managed (*resource* in XrML or *asset* in ODRL) using Uniform Resource Identifier (URI)[5] or Digital Object Identifier (DOI)[6]. Typical URI implementation, for example, the URL, can only reference objects at the file level (e.g., `http://foo.com/bar.html`), or pre-defined tags inside a file (e.g., `http://foo.com/bar.html#tag`). This

[4] http://www.openmobilealliance.org

[5] http://www.ietf.org/rfc/rfc2396.txt

[6] http://www.doi.org

identification method is not precise enough for business and administrative documents. Consider an electronic contract document under negotiation, it may have some parts marked “read-only” and other parts “open-to-change”. Proper digital rights should be enforced down to arbitrary subparts inside a document, or up to a collection of documents. XrML/ODRL yet does not provide sufficient capabilities in this aspect.

2. The permission in XrML/ODRL is defined in a dictatorial way, that is to say, the permission is defined statically. Such definition is applicable to digital media distribution, because the targeted media file never changes after being delivered. However, as long as business and administrative documents are concerned, it is ofter the case that the user’s permission must depend on the contents of the document. For example, the user shall not modify the document if the document’s status is “final”. XrML/ODRL does not support conditional permission definitions.

Nevertheless, sticking to an open standard facilitates information exchange and rights enforcement among different parties. It is for this very reason that adopting XrML/ODRL into access control framework is desired. But some extensions to XrML/ODRL must be made to overcome their limitations regarding business and administrative documents.

2.3.3 Framework of DRM system

The problem we’re addressing involves digital rights management for electronic documents in business and administrative environment, with the ultimate goal of building a complete framework of DRM system. We here review some existing DRM systems built by companies and organizations who are in the DRM business.

Adobe Portable Document Format (PDF) workflow

Adobe PDF [7] is the de facto standard for distributing documents in digital form. It is created to enable people view and print a file exactly as the author designed it, without need to have the same authoring application, operation system or fonts installed. PDF supports document authentication using digital signature schemes. Start from version 1.1, PDF also provides some preliminary access control mechanisms in the form of several permissions. A recent version of PDF specification defines 7 permissions as: [Ado01]

1. Modifying the document's contents.
2. Copying or otherwise extract text and graphics from the document.
3. Adding or modifying text annotation and interactive form fields.
4. Printing the document.
5. Form fill-in and sign document.
6. Document assembly, including insertion, rotation, and deletion of pages and creation of bookmarks and thumbnails.
7. Allow only printing that are limited to a low level representation of the appearance, possibly of degraded quality.

Permission 1, 2, 3, 5, 6 apply to the access control during document deployment, within which permission 1, 3, 5, 6 provide limited functionalities to document creation in terms of form filling and content modification. Permission 4 and 7 are related to printing process at end user's site. A PDF document with any of the permission assigned will be encrypted, internally using the RC4 symmetric key encryption algorithm with MD5 hash of the user supplied password as the key.

We conclude the pros and cons of PDF format as follows:

[7] http://www.adobe.com/products/acrobat/

- Since PDF format is designed for document viewing and printing. Being platform neutral is the advantage of PDF format. But for the same reason, developing platform independent PDF authoring tool is difficult. Existing tools which create PDF file only convert other document format into PDF. In this sense, PDF does not support direct data exchange and thus is not suitable for document creation.

- PDF supports a set of access permissions which are sufficient for generic office use. But the permissions and access control functions in PDF are not flexible enough for business documents. Controlling the access rights to arbitrary subparts inside a document is required. But PDF sets permissions to the document as a whole.

- PDF also provides mechanisms to ensure the authenticity of the document using encryption and digital signatures. The authentication information is stored inside PDF file as a special record. As we have discussed before, the authenticity information does not withstand format transcoding or conversion.

- PDF supports printing control in three levels: printable, not printable and printable in low quality. Such permissions are very useful in electronic publishing applications. For business document, the protection of originality is sometimes more important. PDF format and Acrobat software do not offer numbered printing control, nor can they prove the originality of printed paper documents.

In summary, PDF offers an ideal platform for document description and presentation. Its preliminary support of permissions also meets the requirements for electronic publishing. But as far as business and administrative documents are concerned, the lack of effective means for data exchange, flexible access control and paper based originality assertion limits PDF's application into the end user's site only (and for the sake of originality requirement, permission for printing documents must be disabled).

Electronic Business XML (ebXML)

Electronic Business XML (ebXML[8]) is one of the most prevalent business protocols and framework in business-to-business (B2B) integration. It is officially established by OASIS and United Nations Center for Trade Facilitation and Electronic Business (UN/CEFACT). The goal of ebXML is to solve the interoperability problem by enabling information sharing globally among companies of all sizes.

The ebXML specification consists of five components, namely, the Core Component, the Business Process Specification Schema (BPSS), the Collaboration Partner Profile (ebCPP), the Registry Service (ebRS), and the Messaging Service (ebMS) [ebX03]. The security related services are implemented in BPSS and ebMS, wherein the parameters to determine the security of the delivery channel are defined as:

- authenticated
- confidentiality (encryption)
- authorized
- nonrepudiationOfOrigin (digitally signed message)
- nonrepudiationOfReceipt (digitally signed acknowledgment message)
- secureTransport (SSL, etc)

ebXML does not suggest which specific protocol should be used for these param-

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

eters. It is left to be determined by trading partners. Nevertheless, ebXML specification strongly recommends open Internet standards. Table 2.3 show ebXML recommended standards.

Authentication	W3C/IETF XML Signature (XMLDSIG) TLS, IPSEC
Confidentiality	W3C/IETF XML Encryption (XMLENC) S/MIME
Authorization	SAML security credentials TLS, IPSEC
Non-repudiation of message origin and content	A duty on each party to save copies of all business documents and document envelopes comprising the transaction.
Non-repudiation of receipt	The responder is required to send a signed copy of the receipt, which the requester then saves.

Table 2.3: ebXML recommended security protocol

We conclude the pros and cons of ebXML as follows:

- ebXML represents the trend of global e-commerce integration. It provides a framework that enables communication between systems in a way that is independent of individual system technologies, architectures and application domains. Adoption of ebXML shall fundamentally solve the interoperability problem. However, "global adoption" seems to be too optimistic at least for now.

- The security of ebXML emphasizes more on communication security than rights management. It specifies a set of protocols to establish a secure channel between two parties, but does not tell how to manage the data transmitted inside the channel. For example, ebXML does not tell the recipient his rights to use the data received.

- The existence of paper documents has been excluded from ebXML specification, neither does other paginated electronic documents such as PDF. Though finally it shall be possible to conduct business transaction purely electronically, for the current stage the support of paper documents cannot be ignored.

In summary, ebXML offers an ideal platform for multi-party interoperation. But the lack of rights management support has limited its application to document creation and deployment only. Concrete measures should be taken to make ebXML interface with "legacy" document based back-end systems.

2.3.4 Discussion

In the previous discussion, we have identified that the interoperability and security of DRM system can be achieved using interchangeable XML document format and role-based access control. Key technologies involved in the DRM system include access control, rights expression language, and the integrated document management framework that manages rights from XML document till paper document. Although these technologies have more or less been studied and developed, they are facing unique challenges when put into business and administrative documents domain. These challenges contain both research and implementation issues, which can be mainly captured as end-to-end document authentication and integrated access control. Without solving these two chanllenges the DRM system for electronic documents is far from complete.

2.4 Our work

The second generation DRM requires a complete and end-to-end framework for managing electronic documents. Previous works pose some difficulties in such a framework, especially in the aspect of end-to-end document authentication and integrated access control. Our work in fact addresses these problems directly. We propose authentication methods for electronic documents and printed documents with special consideration of document format transcoding and originality assertion. These two methods are linked together to provide end-to-end document authentication. We also propose a role-based access control framework using XML technologies. The access control framework can be integrated with the authentication methods to give total protection over electronic documents, from their creation, deployment, till at the end-users' site.

Chapter 3

Render Sequence Encoding

In this chapter, we present a watermark scheme and authentication method for electronic documents. The prominent feature which distinguishes our authentication method from traditional digital signature based methods is that we use content-based watermark to embed the authentication information into the document such that the information becomes an inseparable part of the document. Our method also withstands document format transcoding. It is intended to solve the authenticity and interoperability problems we raised in Chapter 1.

3.1 Introduction

In business and administrative document exchange, maintaining the authenticity of the documents is vital to the establishment of trusted relationship among all parties. Unlike traditional cryptographic point of view that authenticity applies to the binary representation of the message, DRM system demands authenticity assertion to the content of the message. This is because different parties may have different computer systems running different document processing software packages. What should be authenticated is the content of the message, not the specific document format or appearance the document takes.

Another challenge arises in document exchange is that the access rights regarding to a specific document must be maintained across different document formats. These document formats may support rights description natively (e.g., PDF), or do not support at all (e.g. PostScript). For the later case, there must exist some methods to attach rights description with the document during document delivery so that the receiver can have access to these rights.

As discussed in Section 2.1, authentication of electronic documents requires the combination of two techniques: the content-based authenticator and the document watermark scheme. We have established that the content-based authenticator can be generated using cryptographic hash or digital signature onto the critical contents of the document. A typical hash value or digital signature is a binary string of several tens to hundreds bits long. The watermark scheme used to hide this value must provide enough information carrying capacity. We also consider using the same watermark scheme to embed rights description into the document because it would be convenient for both document handling as well as interfacing with legacy systems. Obviously the addition of rights description data rises further demands on the information carrying capacity, typically with several hundreds more bits. In what follows, we propose a watermark scheme which satisfies this requirement, and present an effective document authentication method.

The application of the watermark scheme and authentication method in document delivery is shown in Figure 3.1. Given a document for protection and the corresponding rights description (co-created by the authors and system policy), we use document watermark scheme to embed the rights description into the document. We also create an authenticator regarding the document contents and rights

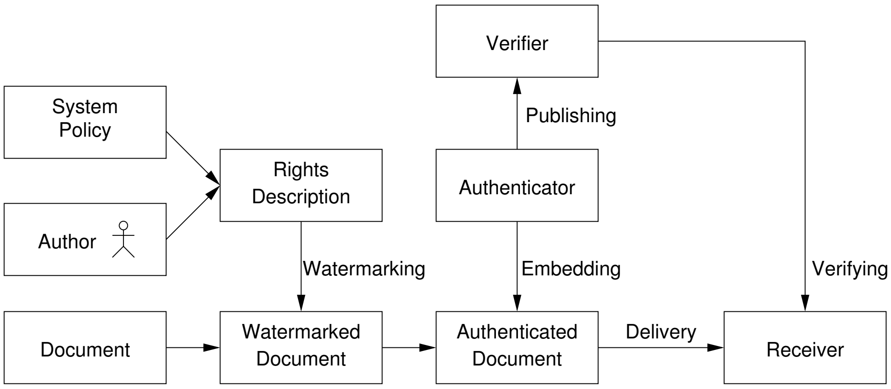


Figure 3.1: Application of watermark scheme in document management

description, and embed the authenticator again into watermarked document for delivery. On the receiving side, the receiver verifies if the delivered document is authentic using the verifier published by the sender, and extract the rights description from the document for execution. The receiver can also send the document to next recipient if it is required and allowed by the rights description, using the same procedures in Figure 3.1. With this structure, rights applicable to the document is transferred from the author to the end user, and the security of the document is protected in the workflow.

3.2 Render Sequence Encoding (RSE)

3.2.1 Motivation

The major obstacle for existing document watermark schemes to be used for authentication and rights description purpose is their low information carrying capacity. Before going on, we first consider the process that an electronic document is read and understood by human cognitive system. It is a three-stage process: the first stage is the rendering stage, when electronic document is converted into readable images by the computer's rendering system. The readable images are either displayed on computer screen, or printed onto paper. Following the rendering stage is the vision stage. It is the process that the images are captured by human visual system. The captured images are finally analyzed by the human brain and get understood in the understanding stage. Each stage in the process contains certain level of variance-tolerance. For example, small variances in rendered images will not affect the output of human visual system, and small variances in wording of sentences will not affect understanding. From information theory point of view, the variance-tolerance ability predicates the existence of redundant information in the document. By replacing the redundant information with payload data or their derivations, watermark scheme achieves information embedding, and the amount of redundant information directly determines the information carrying capacity of that watermark scheme.

Now let's look at the existing document watermark schemes we discussed earlier in Section 2.1.2. Obviously, layout and appearance watermark schemes are at the vision stage. They create small variations in document images in such a way that human visual system cannot discover them. Text watermark schemes are at the understanding stage. They create different sentences, which yield the same meaning after analyzed by the speech center in human brain. It is interesting to note that the farther the watermark scheme is away from the understanding stage, the more information carrying capacity it owns. This is because of the accumulation of variance-tolerances throughout different stages, and hence provides more redundancies, as shown in Figure 3.2.

Apparently, something missing from the existing document watermark schemes is a scheme positioned at the rendering stage. Figure 3.2 tells us that such a scheme should have larger information carrying capacity than any of the existing schemes.

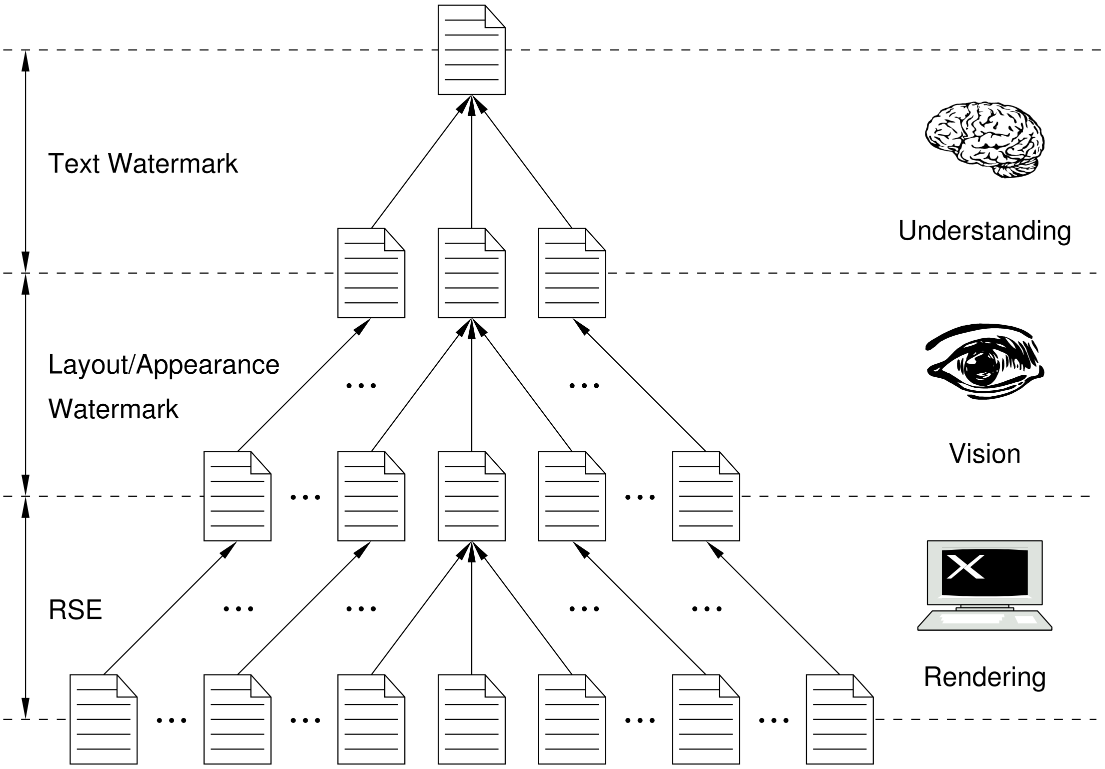


Figure 3.2: Cognition process and watermark schemes

In the next section, we propose *Render Sequence Encoding* to fill this vacancy.

3.2.2 Basis of RSE

Unlike other document watermark schemes which embed payload data into pure text documents or image-based documents, RSE embeds data into formatted documents. Formatted document refers to the document format which contains both text data and layout information. It is also called vector-based document format somewhere to be distinguished from image-based document formats. Strictly speaking, use of the word "vector" here is not precise, because in formatted documents texts are described using font definitions and glyph indexes, while in vector documents they are described using outlines. Formatted document combines the advantages of both text documents and image-based documents: the small file size and platform independent page layout. It is widely used in document exchange. Some frequent formatted document formats include PostScript (PS), Portable Document Format (PDF), Printer Control Language (PCL), Device Independent Document (DVI), etc.

In formatted documents, text data are described using 3-tuples of (*character, font, position*). For example, Figure 3.3 is a simple document which uses PostScript language to describe the sentence "`This is a sentence.`" The first three lines select 12 point Times-Roman font, line 5 moves the cursor to the target position, and line 6 draws the character string. Note that there is only one positioning command "`120 700 moveto`", which defines the position of the first character "`T`". All following characters are advanced horizontally according to the character width stored in the font definition. The description of the 3-tuple are very clear and succinct. However, such simple formatting methods are rarely used practically. Word processing software packages usually issue several positioning commands for a single text line, in order to satisfy the requirements on text justification and font kerning (adjustment of space between pairs of letters to make them more visually appealing). For example, the above sentence is formatted as Figure 3.4 by the LaTeX document preparation system (we have expanded all PostScript macros to make the code more readable). In this real sample, the sentence has been split into five code segments (line 5–6, 7–8, 9–10, 11–12, 13–14), each consists of a positioning command and a drawing command. Examining the

```
1  /Times-Roman findfont          % Font
2  12 scalefont                   % Font
3  setfont                        % Font
4  newpath
5  120 700 moveto                 % Positioning
6  (This is a sentence.) show     % Characters
7  showpage
```

Figure 3.3: A simple PostScript document

```
/Times-Roman findfont       % Font
12 scalefont                % Font
setfont                     % Font
newpath
120.5 700.0 moveto          % Positioning
(This) show                 % Characters
147.1 700.0 moveto          % Positioning
(is) show                   % Characters
159.0 700.0 moveto          % Positioning
(a) show                    % Characters
168.7 700.0 moveto          % Positioning
(sen) show                  % Characters
184.6 700.0 moveto          % Positioning
(tence.) show               % Characters
showpage
```

Figure 3.4: A PostScript document with explicit positioning commands

five positioning commands, we can find that the position parameters they take are sorted in normal reading direction, that is, from left to right. We now randomly permute the five segments so that the position parameters are no longer sorted, as shown in Figure 3.5. Obviously, Figure 3.5 has exactly the same appearance

```
/Times-Roman findfont       % Font
12 scalefont                % Font
setfont                     % Font
newpath
168.7 700.0 moveto          % Positioning
(sen) show                  % Characters
184.6 700.0 moveto          % Positioning
(tence.) show               % Characters
147.1 700.0 moveto          % Positioning
(is) show                   % Characters
120.5 700.0 moveto          % Positioning
(This) show                 % Characters
159.0 700.0 moveto          % Positioning
(a) show                    % Characters
showpage
```

Figure 3.5: A randomly permuted Postscript document

as Figure 3.4 after being rendered, but in binary level they are different ones. In fact we can create up to 5! = 120 visually same documents by using different permutations in the permuting of the five segments. From variance-tolerance perspective, the variant permutations among all documents are tolerable by the computer's rendering system. This predicates the existence of redundancies in document description. The particular form that redundant information takes here is the sequence of positioning and corresponding drawing commands.

When a formatted document is rendered either on the computer screen or on the printer, the rendering system will sequentially parse and execute all commands in the document, rather than sort all commands first and execute them as a whole. This is because otherwise the rendering system must spool a lot of commands, parse them once to determine an optimal sort order, then parse them for the second time to generate the output. Such a process will be very memory and time consuming. So typical rendering systems can preserve the permuted sequence of positioning and drawing commands. For example, when a document as Figure 3.5 is being rendered, the word "`sen`" shows up first, followed by "`tence.`", "`is`", "`This`" and "`a`". The render sequence no longer follows the normal reading direction. Thus we name our watermark scheme the *Render Sequence Encoding*. Generating a unique permutation of render sequence based on the payload data form the basis of RSE scheme.

To create more redundancies as well as to facilitate encoding and decoding efficiency, the permutation of positioning commands can be applied to characters instead of words, e.g., permuting the positions of all character "`e`"s in a whole page. We call the character being permuted the *permutation target*.

3.2.3 Implementation of RSE

In this section we present the RSE watermark scheme by discussing its encoding and decoding algorithms, information carrying capacity, and robustness against format transcoding.

Encoding algorithm

RSE embeds information into electronic document by permuting the render sequence according to a certain permutation. The encoding algorithm generates such a permutation based on the payload data. Two problems to be tackled are: 1. how to identify a permutation. 2. how to map payload data to the identification of a permutation.

An obvious method (called *normal notation* hereinafter) to identify a permutation is to list the occurrence of each element directly, for example, the notation $\{3, 4, 1, 5, 2\}$ means element 3 is permuted to the first place while 2 to the last. However, there is significant dependence in the choice of each element. Once an element had been used, it cannot appear in the following sequence again. This issue creates some difficulties in enumerating all permutations if the number of elements is large. A workaround is to find a way to list all permutations sequentially, so that we can identify each permutation using its index. Algorithm for generating all permutations sequentially based on normal notation was discovered by Johnson [Joh63] and Trotter [Tro62] independently, and was described by Gardner in [Gar74]. The algorithm is quite simple, but in order to determine the i-th permutation, all $i - 1$ permutations must be generated first, which will be time consuming for large permutation length, hence unacceptable either.

Here we use another method called *inversion notation* to identify a permutation, and introduce a fast algorithm to map arbitrary payload data to an inversion notation. Describing a permutation by means of its inversion notation was discovered by Hall [MH63]:

Definition 3.2.1

Let $\{i_1, i_2, \ldots, i_n\}$ be a normal notation of a permutation in the set $\{1, 2, \ldots, n\}$. The pair (i_j, i_k) is called an *inversion* if $j < k$ and $i_j > i_k$.

For example, the permutation $\{3, 4, 1, 5, 2\}$ has five inversions: $(3, 1)$, $(3, 2)$, $(4, 1)$, $(4, 2)$ and $(5, 2)$.

Definition 3.2.2

For a permutation $\{i_1, i_2, \ldots, i_n\}$, let a_j denote the number of inversions whose second component is j. In other words, a_j equals to the number of integers which precede j in the permutation but are large than j; it measures how much j is out of order.

The sequence of numbers $\{a_1, a_2, \ldots, a_n\}$ is called the *inversion sequence* of the permutation $\{i_1, i_2, \ldots, i_n\}$.

For example, the inversion sequence for permutation $\{3, 4, 1, 5, 2\}$ is $\{2, 3, 0, 0, 0\}$.

Theorem 3.2.1

Inversion sequence $\{a_1, a_2, \ldots, a_n\}$ of the permutation $\{i_1, i_2, \ldots, i_n\}$ satisfies this condition:

$$0 \leq a_1 \leq n-1, \quad 0 \leq a_2 \leq n-2, \quad \ldots, \quad 0 \leq a_{n-1} \leq 1, \quad a_n = 0.$$

This is so because for each $k = 1, 2, \ldots, n$ there are $n - k$ integers in the set $\{1, 2, \ldots, n\}$ which are larger than k. Brualdi in [Bru99] shows the mapping between normal notation and inversion notation is onto, and gives conversion algorithms between these two notations. The advantage of using inversion notation is that we can choose each element in an inversion sequence independently, as long as Theorem 3.2.1 is satisfied.

Procedures for mapping arbitrary payload data into inversion sequence are:

Algorithm 3.2.1 (Encoding algorithm)

For any integer X,

1. Choose an appropriate number n such that $n! > X$.

2. Calculate:

$$\begin{aligned}
a_1 &= X \div (n-1)! & X_1 &= X - a_1 \times (n-1)! \\
a_2 &= X_1 \div (n-2)! & X_2 &= X_1 - a_2 \times (n-2)! \\
&\vdots & &\vdots \\
a_k &= X_{k-1} \div (n-k)! & X_k &= X_{k-1} - a_k \times (n-k)! \\
&\vdots & &\vdots \\
a_{n-1} &= X_{n-2} & X_{n-1} &= X_{n-2} - a_{n-1} = 0 \\
a_n &= X_{n-1} = 0
\end{aligned} \tag{3.1}$$

   where $a \div b = \lfloor a/b \rfloor$, the integer part of (a/b).

3. The sequence $\{a_1, a_2, \ldots, a_n\}$ is the inversion sequence identified by data X.

Proof. of encoding algorithm

It holds without saying that $a_1 \ldots a_n \geq 0$.

From step (1) and (2) of Algorithm 3.2.1, we have:

$$\begin{aligned}
a_1 = X \div (n-1)!, \quad \text{and} \quad X < n!, \quad \text{so} \\
a_1 < n! \div (n-1)! = n \quad \Rightarrow \quad a_1 \leq (n-1)
\end{aligned} \tag{3.2}$$

Now consider $X_1 = X - a_1 \times (n-1)!$, it means X_1 is the remainder of X divided by $(n-1)!$, therefore

$$X_1 < (n-1)! \tag{3.3}$$

From 3.2 and 3.3, we get $a_2 = X_1 \div (n-2)! \leq (n-2)$. It follows that $a_3 \leq (n-3)$

till $a_n \leq (n-n) = 0$. So the outcome of Algorithm 3.2.1 is a valid inversion sequence. □

We also claim that for each X, there exists only one corresponding inversion sequence (Encoding uniqueness). To prove this, we need the following proposition:

Proposition 3.2.1

Given any integer $k > 0$,

$$k! = (k-1)! \times (k-1) + (k-2)! \times (k-2) + \cdots + 2! \times 2 + 1! \times 1 + 0! \times 0 + 1$$

Proof.

$$\begin{aligned}
& k! - (k-1)! \times (k-1) - (k-2)! \times (k-2) - \cdots - 1! \times 1 - 0! \times 0 \\
&= (k-1)! \times (k-k+1) - (k-2)! \times (k-2) - \cdots - 1! \times 1 - 0! \times 0 \\
&= (k-1)! - (k-2)! \times (k-2) - \cdots - 1! \times 1 - 0! \times 0 \\
&= (k-2)! - (k-3)! \times (k-3) - \cdots - 1! \times 1 - 0! \times 0 \\
&\ \ \vdots \\
&= 1! \times 1 - 0! \times 0 \\
&= 1
\end{aligned}$$

□

Proof. of encoding uniqueness

Suppose, on the contrary, there exist two distinct inversion sequences $\{a_1, a_2, \ldots, a_n\}$ and $\{b_1, b_2, \ldots, b_n\}$ such that:

$$X = a_1 \times (n-1)! + a_2 \times (n-2)! + \cdots + a_{n-1} + a_n$$

$$\text{and} \qquad X = b_1 \times (n-1)! + b_2 \times (n-2)! + \cdots + b_{n-1} + b_n$$

we have:

$$\begin{aligned}
& a_1 \times (n-1)! + a_2 \times (n-2)! + \cdots + a_{n-1} + a_n = \\
& b_1 \times (n-1)! + b_2 \times (n-2)! + \cdots + b_{n-1} + b_n
\end{aligned} \tag{3.4}$$

Assume $a_1 \neq b_1$ and not losing generality, assume $a_1 > b_1$, then

$$(a_1 - b_1) \times (n-1)! = (b_2 - a_2) \times (n-2)! + \ldots + (b_{n-1} - a_{n-1}) + (b_n - a_n),$$
$$(n-1)! = \frac{b_2 - a_2}{a_1 - b_1} \times (n-2)! + \ldots + \frac{b_{n-1} - a_{n-1}}{a_1 - b_1} + \frac{b_n - a_n}{a_1 - b_1} \tag{3.5}$$

Since $0 \leq a_2 \leq n-2$ and $0 \leq b_2 \leq n-2$,

$$b_2 - a_2 \leq n - 2$$

Notice $a_1 - b_1 \geq 1$, so

$$\frac{b_2 - a_2}{a_1 - b_1} \leq n - 2$$

It also holds for

$$\frac{b_3 - a_3}{a_1 - b_1} \leq n - 3$$
$$\vdots$$
$$\frac{b_k - a_k}{a_1 - b_1} \leq n - k$$
$$\vdots$$
$$\frac{b_{n-1} - a_{n-1}}{a_1 - b_1} \leq 1$$
$$\frac{b_n - a_n}{a_1 - b_1} = 0$$

Obviously Equation 3.5 contradicts with Proposition 3.2.1 because each corresponding addend in Equation 3.5 is less than or equal to the one in Proposition 3.2.1, but the last item +1 is missing. So there must exist $a_1 = b_1$.

Then we cancel $a_1 \times (n-1)!$ and $b_1 \times (n-1)!$ from both sides of Equation 3.4 and repeat the above steps. The result will be $a_2 = b_2 \cdots$ till $a_n = b_n$.

□

By now we have solved the two problems raised at the beginning of this section: 1. a permutation can be uniquely identified using its inversion sequence. 2. the encoding algorithm can uniquely map payload data to an inversion sequence.

We give an example of Render Sequence Encoding to end the introduction of encoding algorithm. In this example, we embed string "GNU" into the "Preamble" section of the "GNU General Public License" [1]. The ASCII coding for the string "GNU" is 0x474E55, or decimal 4673109. We choose $n = 11$ so that $n! = 39916800 > 4673109$. By applying Algorithm 3.2.1, we get the inversion sequence of the desired permutation as $\{1, 2, 7, 7, 1, 2, 2, 3, 1, 1, 0\}$. The permutation can be illustrated as Figure 3.6 (using the algorithm introduced in [Bru99]).

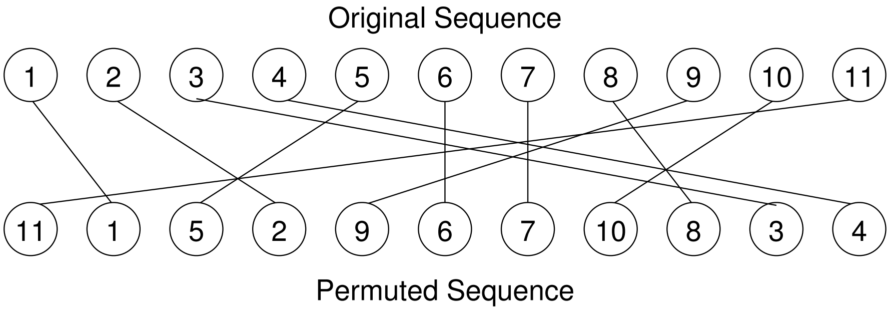


Figure 3.6: Sample permutation {1,2,7,7,1,2,2,3,1,1,0}

We choose character "e" as the permutation target. There are altogether 47 "e"s is the document, our permutation only needs 11 for one round of encoding, so we can do 4 rounds in this paragraph.

The encoding process is done by permuting the position of each character "e" according to the permutation shown in Figure 3.6, that is, the position of the 1^{st} character "e" in the encoded document is actually the position of the 2^{nd} "e" in the original document, that of the 2^{nd}'s is the 4^{th} in the original document, and so on. For illustrative purpose, we label the order of the occurrence of each "e" in the encoded document, as shown in Figure 3.7. The number "1" below the 1^{st} "e" in the word "`license`" means this "e" is the 1^{st} "e" to appear, followed by

[1] http://www.gnu.org/copyleft/gpl.html

the last “e” in the word “`software`”, so on so forth.

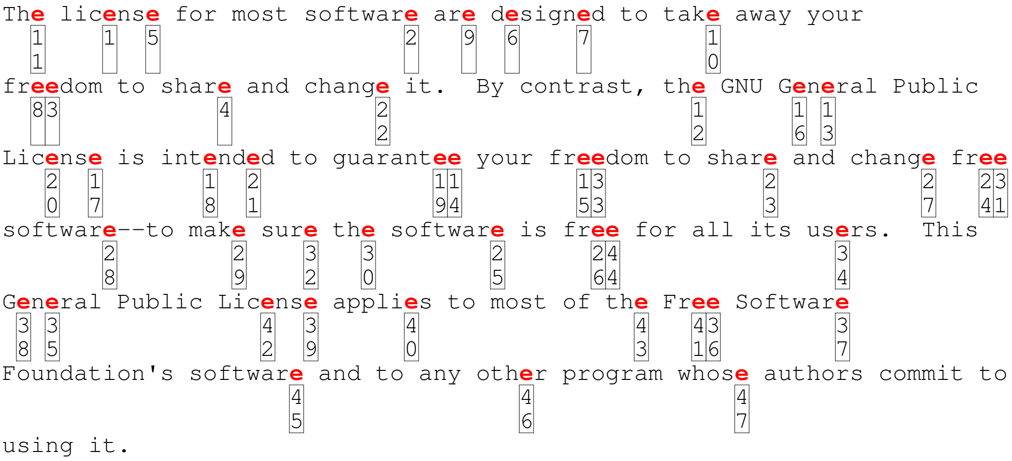

```
The license for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.
```

Figure 3.7: Sample encoded document

Decoding algorithm

To decoding a RSE encoded document, we must first extract the permutation from the document, then decode the permutation to recover the payload X. The procedures are:

Algorithm 3.2.2 (Decoding algorithm)

Given any RSE encoded document, do the following steps:

1. If the permutation target and permutation length n is known, go to step 6, otherwise go to step 2.
2. Find the permutation target by examine the positioning commands for each character.
3. Record all the positions for the permutation target and discover the render sequence. It is done by comparing the physical storage order of these positions with the logical positions they represent. The sequence is denoted $\{S_1, S_2, \ldots, S_m\}$.
4. Generate a new sequence S' such that $S'_k = S_{k+1} - S_k$.

5. Find the largest period of the sequence S', it is the permutation length n.
6. Discover the actual render sequence $\{a_1, a_2, \ldots, a_n\}$ with the permutation target and the permutation length n.
7. Calculate the encoded payload X:

$$X = \sum_{i=1}^{n} a_i \times (n-i)! \tag{3.6}$$

Equation 3.6 holds because it is just the reverse form of Equation 3.1.

It must specially noted that in order to carry out steps 2–5 to determine the permutation target and length, several prerequisites must be satisfied, including: 1. The decoder must know how to distinguish permutation targets, though it may not know which exact permutation target is used. For example, it is possible to permute character "e" to encode one message, and permute character "a" to encode another. Then the decoder must know that the permutation target contains one single character. 2. The permutation used must not contain repeatable "sub-permutations". For example, the permutation {3,1,2,6,4,5} (normal notation) is not allowed, because it will confuse the detector in finding the largest period in step 5. These two prerequisites especially the second one forces some limitations on the RSE scheme. However, for authentication purpose the permutation target and the permutation length can be made public, transferred through auxiliary channels, or pre-agreed between the encoder and the decoder. Publicizing this information does not prevent the RSE scheme to be a public watermark scheme, because they contain no information about the original document. Nevertheless, steps 2–5 can still be used as back-up mechanisms in case the extraction of permutation target and length is needed. We continue the RSE encoding example to see how the encoded information is decoded.

We first determine the permutation target is character "e" because its positions are in abnormal order. By comparing the storage order of the positioning commands with the positions they represent, we get:

$$\{S_1, S_2, \ldots, S_{47}\} = \{11, 1, 5, 2, 9, 6, 7, 10, 8, 3, 4, \ldots, 36, 37, 45, 46, 47\}$$

and by step 4, we get:

$$\begin{aligned} \{S'_1, S'_2, \ldots, S'_{46}\} = \\ & \{10, -4, 3, -7, 3, -1, -3, 2, 5, -1, -18, \\ & \ 10, -4, 3, -7, 3, -1, -3, 2, 5, -1, -18, \\ & \ 10, -4, 3, -7, 3, -1, -3, 2, 5, -1, -18, \\ & \ 10, -4, 3, -7, 3, -1, -3, 2, 5, -1, -8, \\ & \ -1, -1\} \end{aligned}$$

The largest period of the sequence S' is 11, so the encoding is on an 11-permutation (Note that the last two lines of S' are exceptions, because encoding had been truncated). We then obtain the real permutation by examine the first 11 "e"s, which is:

$$\{11, 1, 5, 2, 9, 6, 7, 10, 8, 3, 4\},$$

and the corresponding inversion sequence is:

$$\{a_1, a_2, \ldots, a_n\} = \{1, 2, 7, 7, 1, 2, 2, 3, 1, 1, 0\}.$$

By step 7, we get:

$$X = \sum_{i=1}^{n} a_i \times (n - i)! = 4673109.$$

Thus, the encoded string is decoded as "GNU".

Information carrying capacity

RSE is based on the permutation of render sequence. The length of the permutation determines how many different permutations can be generated, and hence determines the information carrying capacity. For a page of document containing n permutation targets, the maximum length of permutation is n, and the maximum number of permutations can be generate is $n!$ (n's factorial). This is a number quite enormous even if n is only moderately large. For example, 15! is more than $1,000,000,000,000$.

The number of bits that can be encoded using n-permutation is:

$$N_{bits} = \lfloor \log_2 n! \rfloor$$

from Stirling's formula [Fel67]:

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

where $\pi = 3.141\ldots$ and $e = 2.718\ldots$ is the base of natural logarithm, we get:

$$N_{bits} \sim \left\lfloor 1.3257 - 1.4427n + \frac{1+2n}{2}\log_2 n \right\rfloor \tag{3.7}$$

The relationship between the number of permutation targets and the maximum encoded bits is shown in Figure 3.8. The fist page of this chapter contains 106

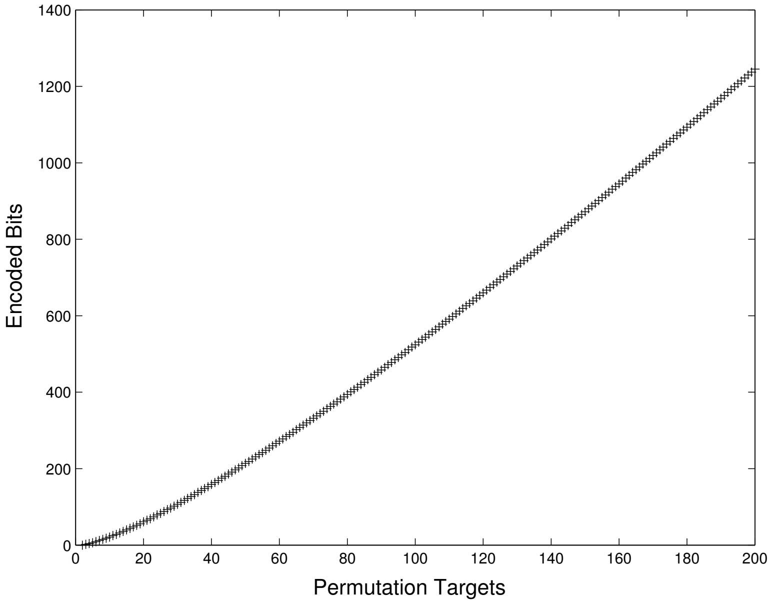


Figure 3.8: Permutation Targets vs. Encoded Bits

“e”s, which means a payload of 560 bits long can be embedded. It is 57 times large than using Brassil’s line shift encoding scheme.

It is expected that, by adding positioning commands into the original document, the file size will be increased. We use experiments to show the effect of size increase versus the number of permutation targets. This experiment takes a page of pure text data as input, outputs two PostScript documents, one unencoded, and the other encoded with selectable permutation targets. We choose up to 10 most frequently used characters in the original text as the permutation targets (in the experiment, {e, t, o, r, i, a, s, n, h, u}). Table 3.1 shows the relationship among the choice of permutation targets, the number of permutation targets, the encoded bits and the file size in “ps.bz2 (PostScript with bzip2 compression)” format. We find the addition of permutation targets dramatically increases the information carrying capacity of RSE, but the enlargement of file size is comparatively smaller.

3.2.4 Robustness

An important requirement on the watermark scheme for authentication purpose is the robustness against document format transcoding. This property helps to ensure document security in an interoperable environment. We use experiments to show RSE scheme satisfies this requirement.

In the experiment setup, we create a virtual printer driver which stores the position of printed characters into local file instead of printing onto real printer. The printer driver is programmed as a Ghostscript [2] device, and servers as the back end of the CUPS [3] printing system. Through this architecture, any document

[2] http://www.ghostscript.com

[3] http://www.cups.org

Choice of Permutation targets	Number of Permutation targets	Number of encoded bits	File size (ps.bz2)
{ø}	0	0	1701
{e}	284	1910	3121
{e,t}	511	3866	4026
{e,t,o}	710	5706	4756
{e,t,o,r}	883	7373	5144
{e,...,i}	1037	8899	5601
{e,...,a}	1187	10416	5947
{e,...,s}	1338	11972	5844
{e,...,n}	1466	13310	6082
{e,...,h}	1555	14250	6198
{e,...,u}	1628	15026	6271

Table 3.1: File size & Encoded bits vs. Permuted characters

formats other than Postscript are converted into Postscript, and the permuted render sequence is captured at the virtual printer driver.

The encoded source document is in Postscript format. We convert it into PDF format (using ps2pdf13 and Adobe Distiller respectively), PCL format (using ljet4 with Ghostscript), PCLXL format (through a HP Windows printer driver) and EPS format (using epswrite with Ghostscript). We then send the converted files info CUPS system and examine the layout of permutation targets. We have performed the same experiments for 50 different encoded source file with distinct permutation targets. All of them preserve the render sequence successfully.

In another experiment, we create an encoded Postscript document, edit it using Adobe Illustrator by changing several characters, then save back to a PDF file.

When the edited PDF file gets printed, we discover that the encoded information at the places where changes are made has been destroyed, while the encoded information at the unchanged places is preserved perfectly. This property shows the fragility of RSE against modifications. It enables a tamper detection application without resort to the original version. We'll discuss it the next section.

3.2.5 Discussion

We have proposed a watermark scheme named RSE for electronic documents. The scheme features large information carrying capacity and robustness against document format transcoding. These two favorable properties enable RSE to be used in the document management system as discussed in Section 3.1.

RSE watermark scheme is invisible watermark scheme. The word "invisible" here means "strictly invisible". It is different from the traditional "invisible" watermark schemes wherein the "invisible" actually means "unnoticeable". Because of so, it is possible to incorporate other document watermark schemes into RSE to achieve more information carrying capacity. For example, all layout and appearance watermark schemes can be implemented using RSE by slightly modifying the positions or adding some extra drawing commands, and all text watermark schemes can be done by modifying the characters.

RSE watermark scheme is public watermark scheme, which means the extraction of embedded payload data does not need the original unwatermarked document. However, for efficiency and accuracy consideration, it is better for the encoder and the decoder to make pre-agreement on the permutation targets and length. Whether this information can be made public or not depends on whether the verification is to be done publicly or privately.

3.3 Document authentication

We have proposed Render Sequence Encoding as a solution to the embedding problem. RSE provides enough information carrying capacity to embed rights description for the document as well as other auxiliary data. In this section, we solve the authenticator problem, and propose an efficient document authentication method which integrates with the RSE scheme to protect embedded rights description and document contents.

How to design content-based authenticator for electronic documents is a well studied topic in cryptographic literature through the use of digital signature schemes [MvOV97]. A simple content-based authentication method for electronic document can be designed as:

1. Generate authentic document:
   (a) Extract all text contents or certain critical text contents from the document.
   (b) Generate digital signature for the contents.
   (c) Embed the digital signature into the document using RSE scheme.
2. Verify document:
   (a) Extract the embedded digital signature from the document.
   (b) Extract the same contents from the document as in the generating procedure document.
   (c) Verify the extracted contents with the digital signature.

Here digital signature has been used as the authenticator in Figure 2.1. This method seems applicable but practically it gets some problems. The most significant one is still RSE's information carrying capacity. As one example, the widely adopted "Digital Signature Standard (DSS) [4]" which uses the DSA signature scheme with 1024-bit secret key will produce two 160-bit strings as the signature. This 320-bit signature contains only numerical values. It must be attached with the signer's identification for the verifier to access the signer's public information. A typical implementation of DSS such as the GnuPG [5] program produces standard DSS signature as long as 520 bits. Encoding a 520-bit digital signature into a page of ordinary document is not difficult for RSE watermark scheme (see Figure 3.1). But at the presence of rights descriptions and other auxiliary data, it would be too much for RSE watermark especially when the document is short. A authentication method with short authenticator length is hence needed.

The reason why most existing digital signature schemes produce long signatures is that they are based on the security of cryptosystems built on top of the hardness of number theory problems. Examples of these problems include the factorization of integers and the discrete logarithm. These problems use operations over integers from hundreds to thousands of bits to prevent exhaustive search attacks, so the output is among that range also. Other drawbacks of using these problems include the facts that the hardness of these problems is not proved (so efficient algorithms and computers may threaten them), and the arithmetic operations are very expensive (modular multiplications, modular exponentiations, etc). Since 1989, there have been several attempts to build cryptosystems based on the NP-complete problems which use operations over small numbers or even bits. The results include the Permuted Kernels Problem (PKP) [Sha90, PC94], the Syndrome Decoding (SD) [Ste94], the Perceptron Problem (PP) [Poi95], the Constrained Linear Equations (CLE) [Ste95], and the Exact Traveling Salesman

[4] FIPS 186-2, http://csrc.nist.gov/publications/fips/index.html

[5] http://www.gnupg.org

Problem (XTSP) [Luc94, Luc95]. Most of these schemes have been proposed on the zero-knowledge interactive proof background without touching the authentication requirements. Here we propose an authentication method based on XTSP. We first explore the hardness of XTSP, and then present the authentication method.

3.3.1 Mathematical background

TSP and XTSP

The Traveling Salesman Problem (TSP) is one of the most widely studied combinatorial optimization problems. The definition of TSP is:

Definition 3.3.1

Let G define a graph (V, E) where V is a set of vertices, and E is a set of edges between members of V. For each edge $e \in E$, $c(e)$ gives the cost for that edge and $C = (c_{i,j})$ is the cost matrix associated with E. The TSP problem is to find a tour T which visits each vertex once and only once (formally a Hamiltonian cycle) in G, with the lowest total cost.

The most common practical interpretation of the TSP is that of a salesman seeking the shortest tour through n cities. TSP contains some special cases: if $c_{i,j} = c_{j,i}$ for all $i, j \in V$, the problem is called *symmetrical* TSP, otherwise *asymmetrical* TSP; if $c_{i,j} + c_{j,k} \geq c_{i,k}$ for all $i, j \in V$, C is said to satisfy the *triangle inequality* and the problem is called *Euclidean* TSP. The proof of NP-completeness of TSP can be found in [CLRS01, LLKS85]. Compared with other NP-complete problems, TSP is among the oldest ones and has been studied long before the theory of NP-completeness was developed. Although not provable, the hardness of TSP is backed by decades of research.

The Exact TSP is a variation of TSP:

Definition 3.3.2

Let G define a graph (V, E) where V is a set of vertices, and E is a set of edges between members of V. The XTSP problem is to find a tour T which visits each vertex once and only once in G, with the total cost equals to a given cost L.

The major difference between TSP and XTSP is that XTSP asks for a tour with exact total cost instead of the lowest cost. The NP-Completeness of XTSP has been proved by Stefan Lucks in [Luc94]. In the following, we review some existing TSP algorithms to see whether they also apply to XTSP.

Algorithms for TSP

TSP is perhaps the most well known combinatorial optimization problem. Its simple definition along with its notorious difficulty has stimulated (and still stimulates) many efforts to find an efficient algorithm [Pun02]. Due to the NP-complete nature of TSP, only approximate algorithms can be expected. We here review two major approaches: the heuristic algorithms and the exact algorithms.

Heuristic algorithms. Heuristic algorithms for TSP do not aim to find the lowest cost tour but a tour that is reasonably low cost. The algorithms have been following two streams: one stream emphasizes on guaranteed *worst-case* performance, and the other emphasizes on good *empirical* performance.

The heuristic with guaranteed worst-case performance works on symmetrical TSP with the cost matrix C satisfies triangle inequality. It usually starts with the minimum spanning tree S of graph G [AHU74], then create a tour T which shares as many edges as possible with S. The best result so far is given in [Chr76], where the total tour cost is guaranteed to be less than 1.5 times of the cost of the minimum spanning tree. The algorithm has polynomial time complexity. But it should be mentioned that no heuristic with a guaranteed worst-case performance is known for the asymmetrical TSP, or symmetrical TSP which is not Euclidean.

The heuristics with good empirical performance can be classify into *tour construction heuristic* which gradually builds a solution by adding a new vertex at each step, and *tour improvement heuristic* which improve a feasible solution by various exchanges. There also has been *composite heuristic* which combines the tour construction heuristic and tour improvement heuristic together. For a good review about these algorithms, please refer to [Lap92]. The heuristic with good empirical performance works for both symmetrical and asymmetrical TSP. However, the algorithm has exponential complexity. It can be proved that if the cost matrix C does not satisfy the triangle inequality, good approximate tours cannot be found in polynomial time unless $P = NP$ [CLRS01].

The core of heuristic algorithms is the iterative loop of optimizations. For each iteration of optimization, the algorithm determines a better edge to be inserted into the tour, or a worse edge to be removed from the tour, until the final tour is reasonably good. Different algorithms have different criteria for making the choices, but from statistical point of view, they are all trying to maximize the probability that an inserted edge is in the final optimal tour and a removed edge is not in the final optimal tour. The cost and the connectivity of edges give great hints on the estimation of the probability, for example, edges with lower cost and edges close to the convex hull of all vertices gain more points [Rei94].

Now consider the XTSP case. We argue that heuristic algorithms cannot solve XTSP. This is because:

1. Heuristic algorithms only find approximate solutions, not exact solutions.

2. The cost of each edge gives no hint on whether the total cost will equal

to the given cost L or not. Without the hint, optimization is pointless. However, exception should be made if the given cost value L is very large or small, or the cost of $c_{i,j}$ has a significant uneven distribution. Under such circumstance, it is possible to determine several vertices in the solution, thus reduces the size of the problem. So in order to improve the hardness of XTSP, $c_{i,j}$ should have uniform distribution, and L should be appropriate ranged (suppose $c_{i,j}$ conforms to a uniform distribution over $[a \ldots b]$, then L should be close to $\frac{n \times (a+b)}{2}$, where n is the number of vertices).

Exact algorithms. Exact algorithms formulate TSP as an integer linear program (ILP), and then solve TSP based on the linear relaxations of integer programs. The most commonly used one is the branch-and-bound (BB) algorithm, we here briefly describe the BB algorithm proposed by Carpaneto and Toth [CT80], which seems to be the basis of other BB algorithms.

The Carpaneto and Toth's BB algorithm (refereed to as CTBB hereinafter) uses the assignment problem (AP) as lower-bound criterion. Given n cities and a cost matrix $(c_{i,j})$, the assignment problem is to assign each city i a city j $(j \neq i)$, such that the total cost is minimum, and each city has been assigned and assigned to other city exactly once. The AP can be regarded as a relaxed TSP which allows sub-tours, see Figure 3.9 for illustration. If the AP solution happens to be a Hamiltonian tour, then it is a solution to TSP as well. If the AP solution contains sub-tours, CTBB algorithm tries to eliminate the sub-tours through the following steps:

1. *(Initialization)*

   Set the solution space $S = X_0$, the best tour $T = \phi$, its cost $Z = \inf$, where X_0 is the original TSP problem.

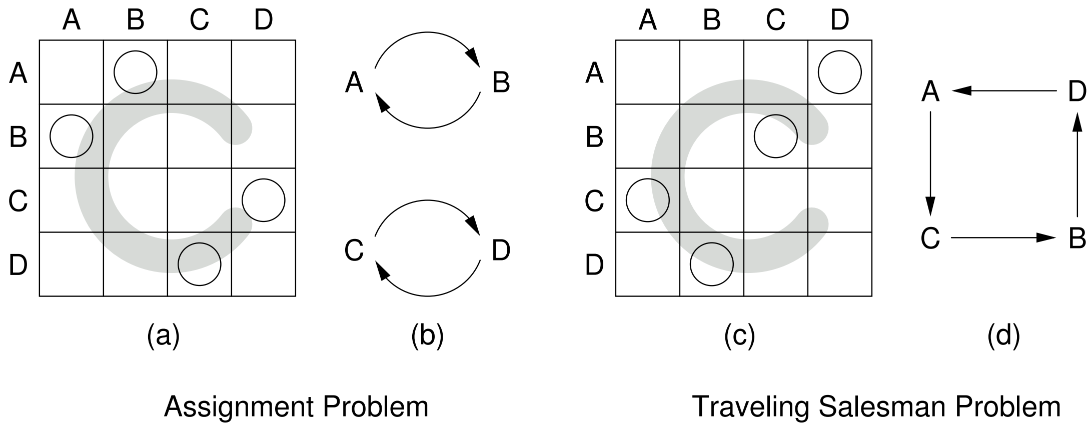


Figure 3.9: Assignment Problem vs. Traveling Salesman Problem: (a) An instance of AP, the leftmost circle means assigning A to B, and so on. (b) Corresponding tour of (a). (c) An instance of TSP, the leftmost circle means traveling from A to B, and so on. (d) Corresponding tour of (c).

2. *(Subproblem selection)*

   If $S = \phi$, stop. T is an optimal tour costs Z.

   Otherwise, select an $X \in S$, and remove it from S.

3. *(Lower-bounding)*

   Let A be the AP solution of X whose cost is Z^*.

   If $Z^* > Z$, this is a worse tour, discard it and go to step 2.

   If $Z^* < Z$ and A is a Hamiltonian tour, this is a better tour. Let $T = A$, $Z = Z^*$, go to step 2.

   Otherwise, go to step 4.

4. *(Branching)*

   Expand X by breaking the sub-tours, generate subproblems $X_1, X_2, \ldots, X_n$, with their AP costs $Z_1^*, Z_2^*, \ldots, Z_n^*$.

   Let $S = S \bigcup \{X_i | Z_i^* < Z, i = 1 \ldots n\}$, go to step 2.

The detailed procedures for breaking sub-tours and generating subproblems can be found in [CT80]. It has been determined that AP tour can be computed in

$O(n^3)$ [MT87]. The CTBB algorithm does not have polynomial complexity, but Carpaneto and Toth had solved 240-vertex TSPs in less than one minute on a CDC 6600. With today's CPU power, TSPs with hundreds of thousands vertices are considered solvable.

Now let's examine if CTBB algorithm works for XTSP as well. Among the sub-tour elimination steps, the most important one is step 3. During the step, tour with higher cost is discarded and tour with lower cost is accepted as new possible solution. However, for the XTSP problem, one cannot either simply discard a tour because its cost does not equal to the given cost L, or accept a tour because its cost is close to the given cost L. This is because the total cost being close to L or not gives no information about whether a tour is similar to the solution or not. So branching cannot be done. This result can be generalized to other branch-and-bound algorithms. Lucks in [Luc94] gives similar results using experiments.

In conclusion, both heuristic and exact algorithms are not adaptable to the XTSP. Although not provable, current results show the only way to solve XTSP is exhaustive search. For an n-city XTSP, the solution space contains $(n-1)!$ different tours. With $n = 41$, the security of XTSP is comparable to 160-bit digital signature schemes (since $log_2(40!) \approx 160$).

Modular XTSP

Definition 3.3.3

The modular XTSP is to find a Hamiltonian tour T in graph G such that

$$Cost_C(T) = L \pmod{2^l}$$

wherein l is a parameter which determines the security of modular XTSP.

For a graph of n vertices, there are altogether $(n-1)!$ different Hamiltonian tours, so T can be described using $\lceil log_2(n-1)!\rceil$ bits. It is nature to consider both the length of $c_{i,j}$ and l equal to $\lceil log_2(n-1)!\rceil$ so that the modular XTSP can yield $(n-1)!$ uniformly distributed Ls. In fact $l = \lceil log_2(n-1)!\rceil$ defines the most secure case of modular XTSP. This result has been rigorously proved in [Luc94].

3.3.2 RSE authentication method

The purpose of this section is to find a short authenticator for embedding into the document using RSE scheme. A preferable way of doing this is to use the permutation directly as the authenticator, thus maximizes the utilization of RSE. For the total $n!$ permutations over n permutation targets, we treat a very small subset of the permutations as valid, then we can authenticate the document by examine the presence of the permutation. Of course the verification of permutations must be content-related, so as to prevent substitution attacks.

Given an n-permutation, we can create a directed Hamiltonian cycle by concatenating the first and last elements. And given a directed Hamiltonian cycle of length n, we can create n corresponding permutations, by regarding each node in the cycle as the starting point, as shown in Figure 3.10. This relationship between permutation and Hamiltonian cycle enables us to authenticate the permutation by authenticating its corresponding Hamiltonian cycle. Since the permutation is encoded into document using RSE watermark scheme, we name our authentication

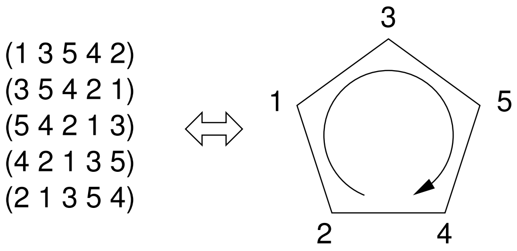


Figure 3.10: Permutations and corresponding Hamiltonian cycle

method the RSE authentication method.

For efficiency consideration, RSE authentication method authenticates a batch of documents together. These documents can have the same contents but created for different recipients or just have different contents. In business and administrative environment, the need for differentiating recipients, or preparing a series of documents for a single transaction is very frequent, so our method is adaptable. Suppose there are N documents to be authenticated, we execute the following procedures (as shown in Figure: 3.11):

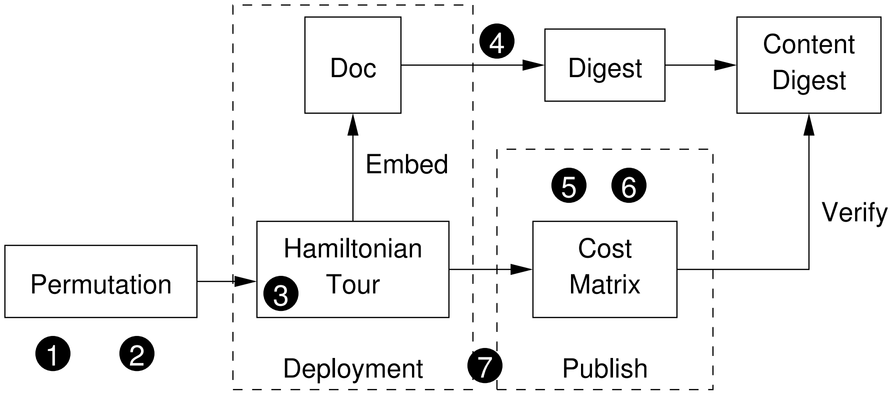


Figure 3.11: RSE authentication flowchart

1. Choose a number n such that $n \geq 41$ and $n \times (n - 1) > N$.
2. For each document, assign a distinct n-permutation $P_i, i = 1 \ldots N$. Not losing generality, we require $P = (1, \ldots)$ (normal notation, so there are $(n - 1)!$ different Ps). Generating of permutations can be done by using a Pseudo Random Number Generator (PRNG) to generate some random numbers, convert the random number to $(n - 1)$-permutation using RSE encoding algorithm, then insert 1 as the first element and adjust the following elements accordingly.
3. Generate Hamiltonian tour T_i from P_i, $i = 1 \ldots N$.
4. For each document, extract all text content or certain critical text content,

then create a $l = \lceil log_2(n-1)! \rceil$ bits message digest of the contents using a collision-resistant one-way hash function $H(\cdots)$. The digests are denoted using $L_i, i = 1 \ldots N$.

5. Create an all-zero $n \times n$ cost matrix C, then solve the equation:

$$Cost_C(T_i) = L_i \pmod{2^l}, \qquad i = 1 \ldots N \tag{3.8}$$

   by adjusting $c_{i,j}$ in C. Since there are $n \times (n-1)$ unknowns in matrix C ($c_{i,i} = 0$ for $i = 1 \ldots N$) and Equation 3.8 only contains $N < n \times (n-1)$ restrains, C is always solvable using linear algebra method.

6. Assign all unused $c_{i,j}$ random values. Then calculate

$$c_{i,j} = c_{i,j} \pmod{2^l}, \qquad i,j = 1 \ldots N \tag{3.9}$$

   With this step, $c_{i,j}$ has been limited to l bits.

7. Finally, publish the cost matrix C as the verification key, and embed P_i into corresponding document using RSE watermark scheme.

For verification of the document, the verifier first calculates the message digest L'_i from the document using the same one-way function $H(\cdots)$. The verifier then extracts the permutation P_i from the document, converts it to Hamiltonian tour T_i, and verifies if

$$Cost_C(T_i) = L'_i \pmod{2^l}, \qquad l = \lceil log_2(n-1)! \rceil$$

The selection of one-way hash function $H(\cdots)$ needs special consideration. It must be able to output $\lceil log_2(n-1)! \rceil$ bits message digest. There has no such variable length one-way functions been proposed except for the HAVAL [ZPS92] (outputs 128, 160, 192, 224, 256 bits) and SHA-V (outputs 128, 160, 192, 224, 256, 288, 320 bits) algorithms. While truncating hash values to a lower number of bits is possible, concatenating shorter values to form a longer value reduces security.

We recommend selecting proper n values so that $\lceil log_2(n-1)! \rceil$ are comparable to hash function outputs. Recommended values are $n = 41, (log_2(40!) \approx 160)$, $n = 47, (log_2(46!) \approx 192)$, $n = 52, (log_2(51!) \approx 224)$, $n = 58, (log_2(57!) \approx 256)$, $n = 63, (log_2(62!) \approx 288)$, $n = 68, (log_2(67!) \approx 320)$. For number of documents more than $68 \times (68-1) = 4556$, it is possible to partition documents into several groups, then generate cost matrix C for each group.

Once a cost matrix C has been fixed, adding more documents for authentication requires re-calculating the whole C. This is because otherwise an attacker can compare the cost matrix before and after adding new documents to determine the newly used edges. For verification, it means the verifier must always keep his copy of matrix C up-to-date. This resembles the verification of digital signature where the verifier must retrieve the signer's public information from a trusted server. Nevertheless, if the cost matrix C has been fully utilized (authenticate $n \times (n-1)$ documents), the verifier needs only retrieve an average l-bit value for one document. The communication bandwidth is much lower than that of digital signature schemes. Considering other advantages such as the size of the authenticator (permutation vs. encrypted message digest) and the computation complexity (modular addtion vs. modular exponential or logarithm), RSE authentication scheme is much superior than digital signature as long as electronic documents are concerned.

3.3.3 Security analysis

The security of the authentication scheme is easily verified:

- For the total $(n-1)!$ possible Hamiltonian tours, we treats N of them to be valid. So the possibility that a random permutation be erroneously considered valid is:

$$\frac{N}{(n-1)!} < \frac{n \times (n-1)}{(n-1)!}$$

For $n = 41$, this figure is about 2×10^{-45}, which means such a coincidence is really rare. We do not specifically require distinct Hamiltonian tours produce different costs. It is very unlikely because the space for cost values is at least as large as the space for Hamiltonian tours (2^l vs. $(n-1)!$). There is no way to prevent collisions except enumerating all tours, which is an astronomical figure.

- In order for an deliberate attacker to forge a document which can pass the verification process, he must be able to do one of the following things:

  1. He creates a new document and generates a hash value L'. In order to embed a correct n-permutation P' into the document, he must solve the modular XTSP to find a Hamiltonian tour T' that satisfies

     $$Cost_C(T') = L' \pmod{2^l}$$

     The mathematical background shows it is intractable. Figure 3.12 shows such kind of attack (blocks marked with red diagonal lines indicate vulnerability.)

  2. He selects a Hamiltonian tour T' and calculates

     $$L' = Cost_C(T') \pmod{2^l}$$

     Now he must reverse the one-way hash function $H(\cdots)$ in order to create a meaningful document and relevant information that hashes to L', as illustrated in Figure 3.13. It is also intractable since the one-way hash function is collision resistant.

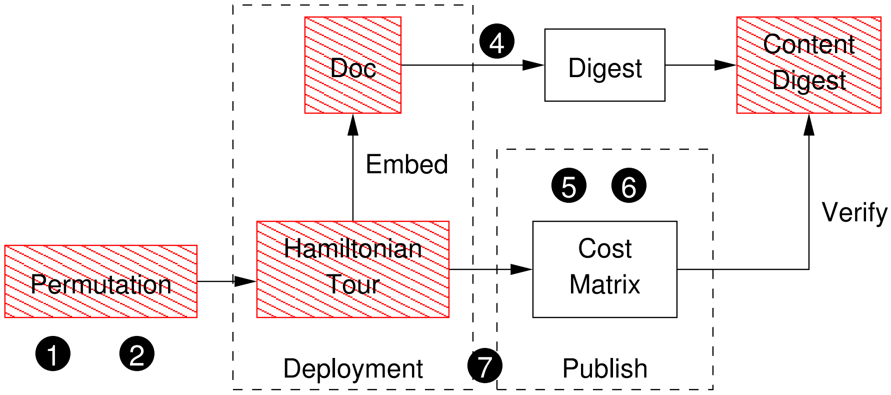


Figure 3.12: Attacking RSE authentication scheme (Method 1)

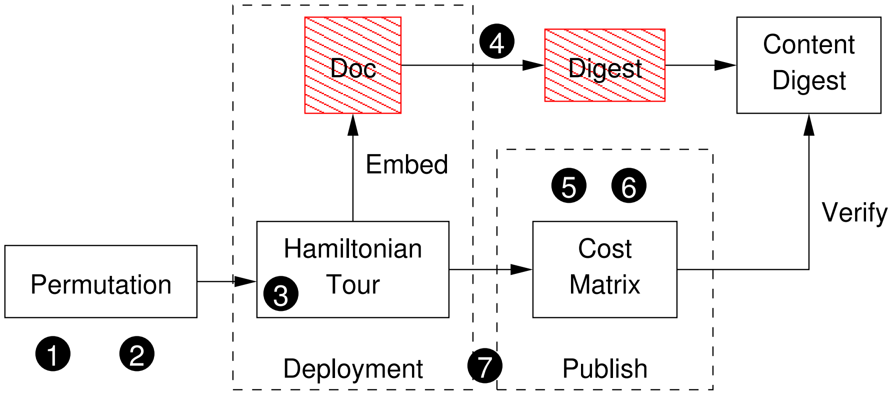


Figure 3.13: Attacking RSE authentication scheme (Method 2)

- Back to Figure 2.1. In RSE authentication method, the verification key is the cost matrix C, the authenticator is the permutation P or Hamiltonian tour T, and the embedding of $P/(T)$ into electronic document is through RSE watermark scheme. It can be figured out that our method does not have a specific authentication key, which means the sender Alice has nothing secret. This issue makes impersonating attack possible. However, Alice has the special power to publish C. If Alice puts C onto a trusted server, then attacker had to compromise the servers to conduct a successful attack. The attacking is shown in Figure 3.14.

In conclusion, RSE authentication method relies on theoretical security and operational security. If both requirements are satisfied, the method can effectively

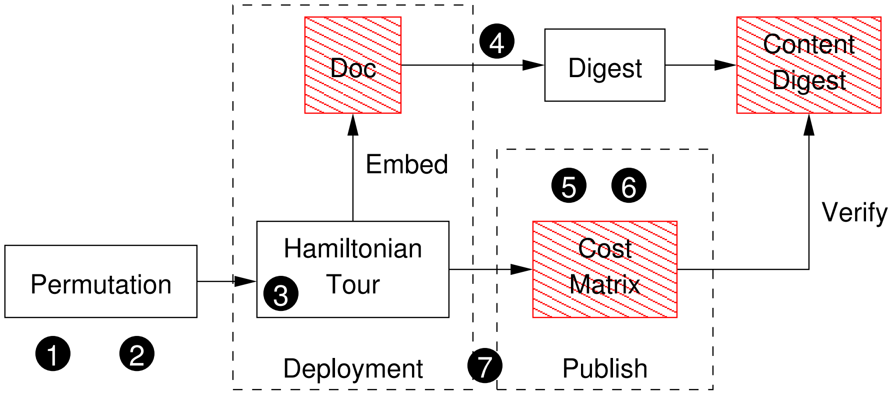


Figure 3.14: Attacking RSE authentication scheme (Method 3)

ensure that a document has not been tampered with and has indeed originated from a specific source. Our method is file format neutral, which solves the content-based authentication problem in an interoperative environment. This is the major advantage over traditional digital signature based schemes.

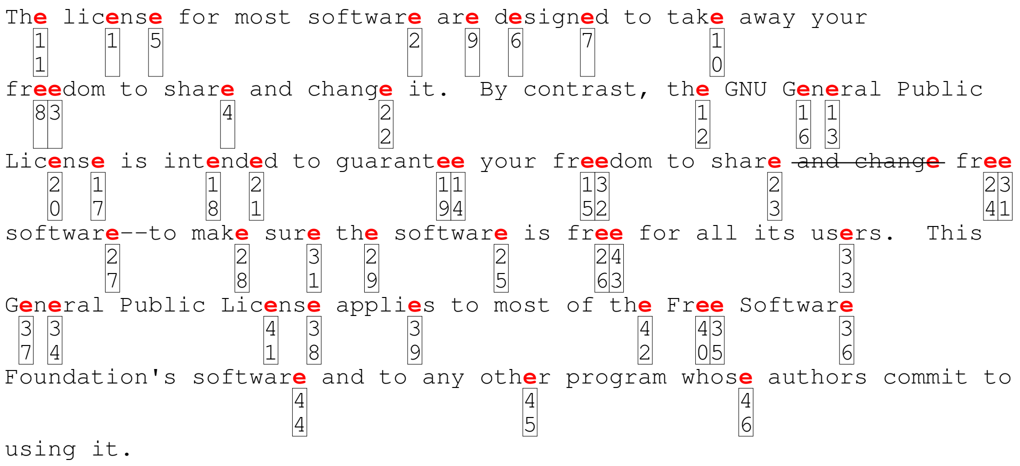
3.4 Tamper detection and copyright protection

In this section, we briefly discuss some other possible applications for RSE. We propose a tamper detection method and a copyright protection method. Both methods offer some advantages over their counter parts in electronic document domain.

3.4.1 Tamper detection with RSE

In Section 3.2.4, we use experiments to show the fragility of RSE against modifications. Modifying an RSE encoded document will only destroy the embedded information at the places where modifications are made. We encode the same permutation into the document for multiple rounds, or onto multiple permutation targets, then the modified places can be located by comparing all permutations.

Figure 3.15 shows an example of tamper detection with RSE.



```
The license for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share ~~and change~~ free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.
```

Figure 3.15: A tampered document

The sample document has been modified by deleting words “`and change`” in line 3. As a result, the 27^{th} letter “`e`” is missing and all “`e`”s after it are shifted forward by one position. Here we consider the situation that we do not know the original render sequence beforehand. If on the contrary we know it then detecting modified places is much easier. In Figure 3.15, the render sequence is:

$$\{11, 1, 5, 2, 9, 6, 7, 10, 8, 3, 4,$$
$$22, 12, 16, 13, 20, 17, 18, 21, 19, 14, 15,$$
$$32, 23, 24, 31, 27, 28, 31, 29, 25, 26,$$
$$43, 33, 37, 34, 41, 38, 39, 42, 40, 35, 36,$$
$$44, 45, 46\}.$$

We assume the modifications only appear at a small part of the document. This assumption is reasonable because most unauthorized modifications are aimed at changing a few critical words rather than the whole document. We use the RSE decoding algorithm to determine the permutation length. Here we find the permutation length is 11 from the 1^{st}, 2^{nd} and the 4^{th} lines. Since the majority of the document has not been modified, the number 11 is credible, and modification must be in line 3. Substituting line 3 with the permutation obtained from line 1, 2 and 4, the proper sequence for line 3 should be

$$\{33, 23, 27, 24, 31, 28, 29, 32, 30, 25, 26\}.$$

Thus, we detect the modification as missing a permutation target “e” between the 23^{rd} and 24^{th} letter “e”s.

The addition of permutation targets can be similarly detected. In cases when more accurate tamper detection is needed, we may increase the number of permutation targets, e.g., permuting each vowel respectively, or permuting characters together with words or even sentences.

The tamper detection method is especially handy if used together with the RSE authentication method. This is because the length of the permutations used in the RSE authentication method is very short. For most of the cases only a small part of available permutation targets are permuted. It allows us to encode the same permutation multiple rounds by taking advantage of the remaining permutation targets, so the RSE decoding algorithm can be more accurate and the tampered location is more easily identified.

3.4.2 Copyright protection with RSE

For most of the document watermark schemes we introduced in Section 2.1, their original intention is to discourage illegal dissemination of copyrighted document. To achieve this, they use watermark schemes to embed distinct hidden marks for each recipient. When an illegally disseminated document is found, its hidden mark is extracted so the original recipient is caught. The most important feature for a watermark scheme to be used for copyright protection is robustness. In the previous works, the robustness has been identified as persistence over photocopies, scanning, printings, and so on. This is a little out-of-date nowadays as electronic distribution of documents is the main form. Current on-line document distribution systems are usually equipped with access control software packages to limit legitimate users' rights. For electronic documents these rights mainly refer to the rights to view or print the documents. Making illegal copies of a document by dumping the contents from the screen is not an easy job because the software can disable such operations. However, if the access control software allows a legitimate user to print the document, the control is lost. This is because an attacker can use "virtual printing" technique, which redirects the printing job to a file. If the printer driver the attacker uses is a PostScript printer driver, then the dumped file is in PostScript language. It is very easy to convert the PostScript file into other formats such as PDF. Thus, an unprotected electronic document is obtained.

RSE watermark scheme can be used as a method to discourage "virtual printing" attack. Our experiments in Section 3.2.4 show the RSE embedded information can survive format transcoding and printing. So we can design an access control software package, which encodes the user's identity or other distinguishable information during printing of copyrighted material using RSE watermark scheme. Later format transcoding cannot remove the information so the original user who prints the document is traceable. It should be mentioned that RSE encoded information persists only if the converted document is still a formatted document. This seems to be most of the cases since formatted document has tremendous advantage over image-based document (consider indexing, searching, file size...).

However, copyright protecting through watermark schemes is still a topic worth debating. It has been identified in [PAK98] that no watermark schemes is likely to withstand all attacks. So the copyright watermark scheme is secure only if the attacker does not realize the existence of hidden information. This is “security through obscurity (STO)”, which is not a good practice in computer security domain. The protection of copyrighted content is still the task of the legal systems. Technical means can only play a supportive role in this drive.

3.5 Conclusion

In this chapter, we started by identifying the requirements for protecting electronic document in an interoperative environment, which are the authentication requirement and the rights description transferring requirement. The solution to these two requirements resorts to a watermark scheme with sufficient information carrying capacity, which we proposed as Render Sequence Encoding.

The major difference between RSE watermark scheme and existing watermark schemes is that our scheme takes advantage of the redundancies in the page description languages of electronic documents. The redundancies are captured as render sequences. By manipulating render sequences, we achieve information carrying capacity that is several orders of magnitude larger than all existing schemes. RSE watermark scheme is robust in terms of surviving file format transcoding. This feature archives interoperability by bridging rights description and authenticator from one document format to another.

Based on the RSE watermark scheme, RSE authentication method adopts modular XTSP to authenticate the document. The security of RSE authentication method is guaranteed by the intractability of XTSP. The advantage of RSE authentication method over digital signature is its small authenticator size. With this feature RSE authentication is adaptable to very short documents. Another feature of RSE authentication is its compatibility with most popular document formats. It can be integrated into existing systems with only minor changes at the creation and rendering ends of the whole workflow. RSE authentication method facilitates the “management of rights holders’ relationship” by establishing trust among parties involved in document exchange. It is thus a major building block in the whole DRM system for electronic documents.

Chapter 4

Print Signatures for Document Authentication

In this chapter, we present a method to authenticate printer paper document by utilizing the inherent non-repeatable randomness existing in the printing process. The randomness results in unique features for each printed paper document, which are captured as the *print signature*. We present theoretical and experimental details on how to register and verify this print signature. This method facilitates the realization of managing " all forms of rights both in physical and digital form".

4.1 Introduction

The definition of second-generation DRM tell us that DRM involves the management of all forms of rights usages over both tangible and intangible assets – both in physical and digital form. The inclusion of rights in physical form is the fundamental difference between the second-generation DRM and first-generation DRM. For electronic documents, managing rights in physical form means managing rights over printed paper documents. This involves authenticating printed paper documents, and controlling the access rights to the printed paper documents. The later one is usually enforced by physical means such as safe-box or lockers, which is outside the scope of this thesis. We here concentrate the authentication methods for printed paper document.

Research in the authentication of physical documents has been growing because of its commercial potential. Compared with its counterpart in digital domain, authenticating physical documents especially paper documents is much less advanced. Recent research shows paper documents still form the basis of today's business transactions and administrative processes, and "will continue to occupy an important place in office life, but will increasingly be used in conjunction with an array of electronic tools [AH01]". For that reason, authenticating printed paper documents, which is the link between electronic tools and paper documents, becomes extremely important.

In business and administrative environment, *authenticity* and *originality* are the two basic requirements for any paper document to be considered valid. As discussed in Section 2.2, existing authentication methods either cannot protect the originality and authenticity of printed document simultaneously, or cost too much to be widely used. In view of this, we present our novel print signature authentication method which has the following advantages:

- *Security*: The print signature is unique for each printed document. Any duplication attempt can be detected during the authentication phase. The content of the document is also used in the validation process. Thus, both authenticity and originality of printed paper documents are secured.
- *Convenience*: Our system can be implemented in a fully automated manner for high–speed batch processing. It can also be incorporated in handheld

devices for manual operation.

- *Low Cost*: Our solution works on any ordinary laser printers. No special material or accessory is required. The cost of automatic verification devices is quite low as well.

The chapter is organized as follows. In Section 4.2, we'll discuss the basis of our method. In Section 4.3, the detailed authentication process is analyzed. Experimental results are given in Section 4.4, and followed by the conclusion in Section 4.5.

4.2 Basis of the method

Authenticity and originality are two major requirements for printed document which need to be authenticated. It can be concluded from Section 4.1 that physical methods (special material and fingerprinting) prove more effective for establishing originality whereas cryptographic methods (digital encoding and optical watermarking) protect authenticity better. Our proposal combines the advantages of both these approaches. We first discuss new properties for protecting the originality of documents, then consider the integration issues with cryptographic techniques to lead to a complete solution.

4.2.1 Print signatures

Figure 4.1 depicts the major components of a laser printer's imaging unit, which develops a piece of printed paper over six steps [BK94]: A photosensitive surface (photoreceptor) is uniformly charged with static electricity by a corona wire (1). Then the charged photoreceptor is exposed to an optical image through laser

[1] http://www.howstuffworks.com/laser-printer.htm

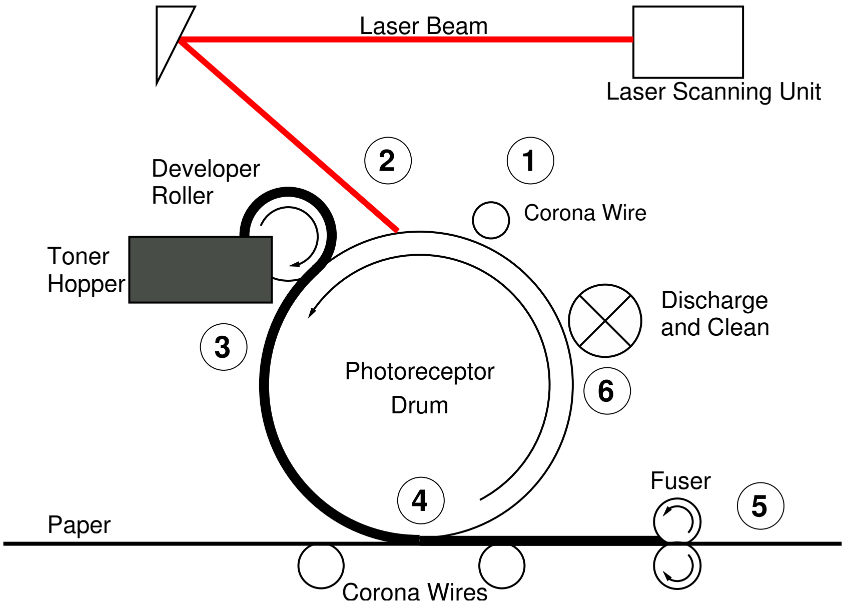


Figure 4.1: How laser printer works[1]

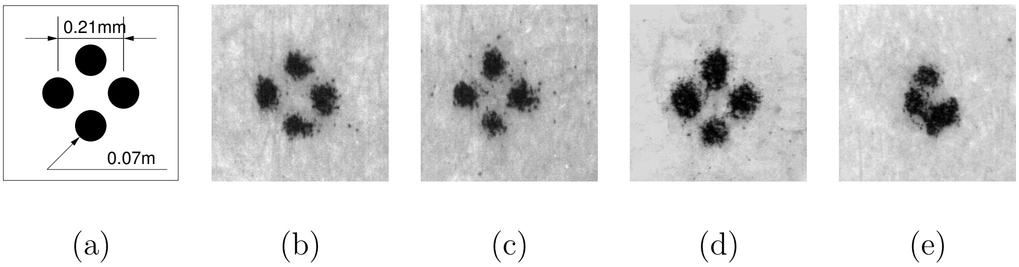
beam, which discharges it at desired positions to form a latent or invisible image (2). Development is done by spreading toner, a kind of fine powder, over the surface. The powder adheres only to the charged areas, thereby making the latent image visible (3). In the next step, electrostatic field transfers the developed image from the photosensitive surface to a sheet of paper (4). The transferred image is then fixed permanently to the paper by fusing the toner using pressure and heat (5). The last step cleans off all excess toner and electrostatic charge from the photoreceptor to make it ready for the next cycle (6).

As no process repeats exactly, we expect to observe variations in each step. Such variations include the unevenness of the photosensitive surface and paper surface, the variable granularity of the toner powder, unstable heat and pressure of the fuser, amount of excess toner remaining on the photoreceptor, and many other such factors. The net outcome of all these variabilities is that some toner powder gets randomly misplaced at undesired positions. Such misplacement is non-repeatable for each print run. This is because any repeatable defect can be detected during the quality control process and thus is fixed by improving the printer design. It is much harder to fix random phenomena hence they persist. Therefore, the pattern of misplaced toner powder on each paper is unique. We refer to this unique pattern by *print signature* as a metaphor for the manual signature on paper documents.

To study the characteristics of print signatures, we created a representative test pattern as shown in Figure 4.2(a). The pattern comprises of four rounded dots. The diameter of the dots is $0.07mm$ and the horizontal and vertical distance between two adjacent dots is $0.21mm$. These two numbers are selected by taking both the physical limitations of the printer and experimental results into consideration. The size of the dots is larger than the theoretically smallest dots the printer can print (in this case 1/600 inch for a 600 dpi printer), so that the dots are clearly visible after printing. On the other hand, the dots are enough small for the random misplacement of toner powder to be significantly noticeable around their boundaries. The distance between two adjacent dots and the configuration of dots ensure that the printed dots will not merge together, which is very useful for our later segmentation process. The number of dots balances the authentication performance and required computational resources. We will provide more details on this topic in the next section.

Figure 4.2 shows some experimental printouts and photocopies examined under a 200× microscope. Image (b) and (c) are the test pattern printed using HP[2] LaserJet 8100 (600dpi) office printer. Image (d) is the same test pattern printed on a high resolution HP LaserJet 4050 (1200 dpi) printer. Apparently, the dissimilarity among these patterns is large. Even for the same printer, we obtain a large variance. Image (e) is a photocopy of image (b) using a 600×600 dpi digital

[2] http://www.hp.com

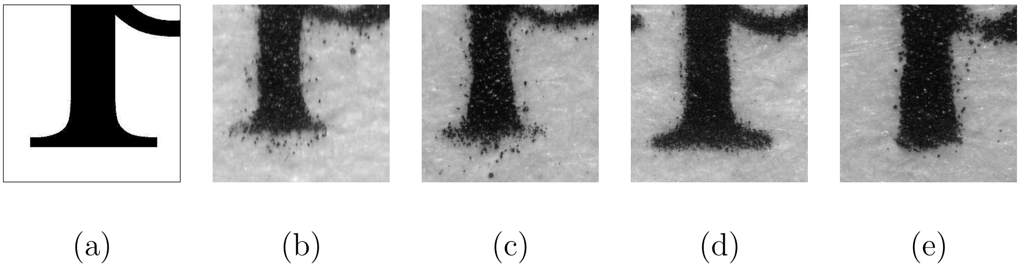


(a) (b) (c) (d) (e)

Figure 4.2: Printouts and Photocopies of the testing pattern: (a) Testing pattern. (b) Testing pattern printed using LJ8100. (c) Testing pattern printed using LJ8100 again. (d) Testing pattern printed using LJ4050. (e) Photocopy of (b).

photocopier Minolta Di152f[3]. It is quite obvious that the photocopied image is very different from the original one.

Besides the test pattern, occurrences of random toner powder misplacement can also be noticed at boundaries of printed characters, as shown in Figure 4.3, where images (a–e) are the source character, two test printouts on LaserJet 8100, one test printout on LaserJet 4050, and a photocopy of (b) on the Minolta photocopier respectively. We observe the same phenomenon noticed in the previous experiment that the print signature is random and non-repeatable for each print run.



(a) (b) (c) (d) (e)

Figure 4.3: Printouts and photocopies of character "p": (a) Source pattern "p". (b) Source pattern printed using LJ8100. (c) Source pattern printed using LJ8100 again. (d) Source pattern printed using LJ4050. (e) Photocopy of (b).

We have performed many such experiments and have consistently observed this

[3] http://www.minolta.com/flash-copier.html

occurrence for several types of laser printers. The experiments demonstrate the uniqueness and randomness of our proposed print signature. Our method utilizes some features of this phenomenon to authenticate the originality of printer paper documents.

4.2.2 Basis of the method

Without loss of generality, we describe our proposed method based on the type of print signature shown in Figure 4.2. We call this test pattern used in the experiment *secure pattern* as it enables certain security features. With some minor modification, our method can also apply to the print signature detected on printed characters as well as hand signatures.

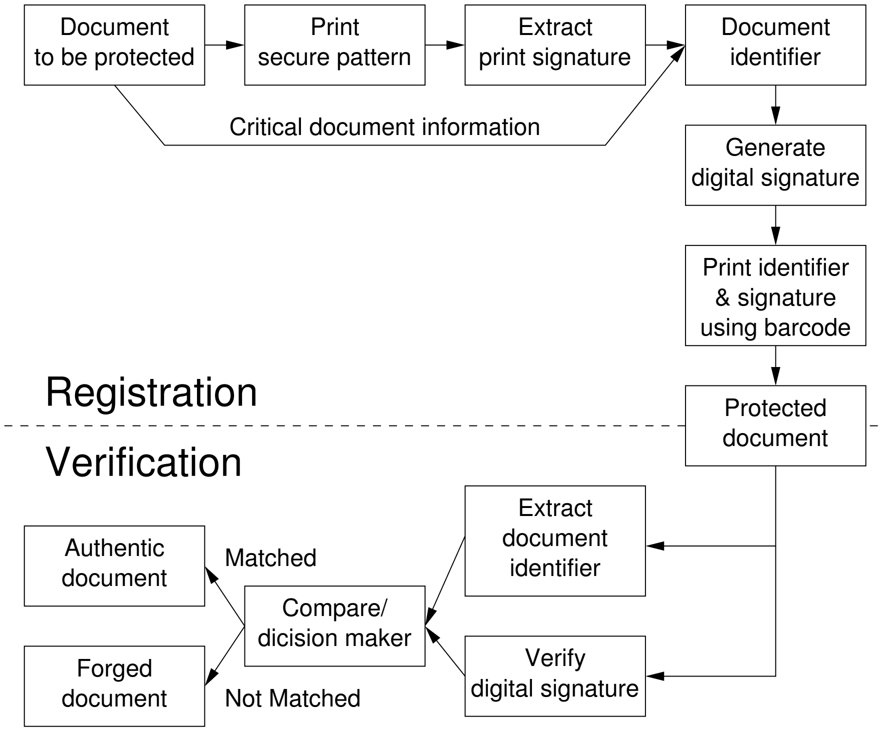
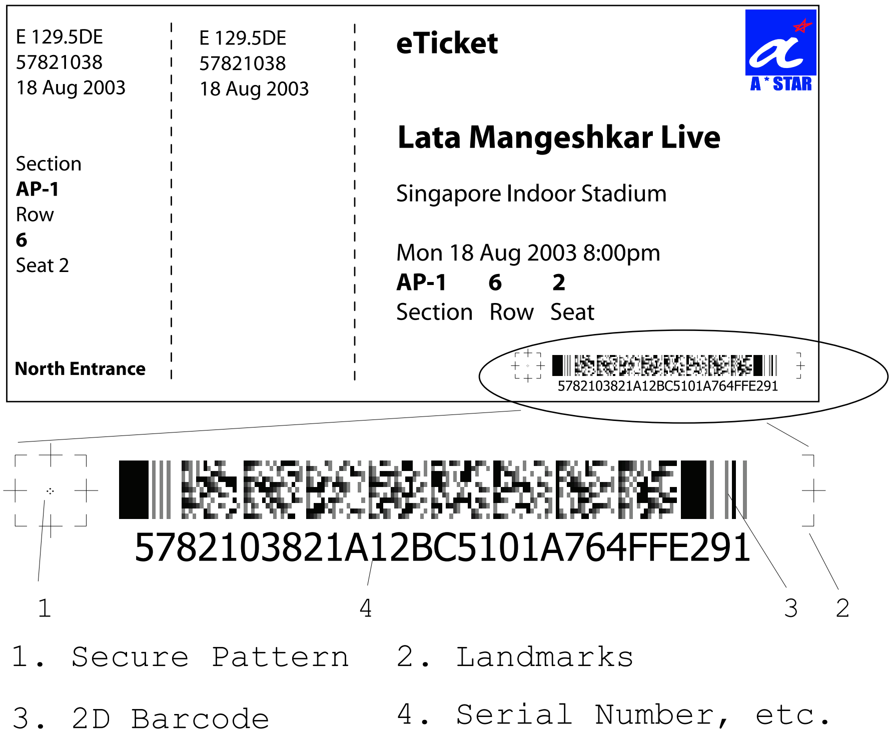


Figure 4.4: System diagram

As illustrated in Figure 4.4, our method contains two procedures: registration and verification.

- *Registration:* Given a document to be protected, we print the secure pattern onto some blank area of the paper. Several auxiliary landmarks are also printed around the pattern to facilitate alignment. The printed paper is then examined by a microscope. Features describing the print signature such as the shape of the dots are detected and extracted. The feature description, together with some critical information about the document (such as the seat number in a concert ticket), forms a unique identifier for this specific document and specific print run. A digital signature is then generated for the identifier. The digital signature and the identifier are printed onto the same document using digital encoding methods such as bar codes or OCR fonts. These printed information and the secure pattern are used for later verification. Figure 4.5 is a sample concert e-ticket protected using our method.



E 129.5DE
57821038
18 Aug 2003

Section
AP-1
Row
6
Seat 2

North Entrance

E 129.5DE
57821038
18 Aug 2003

eTicket

Lata Mangeshkar Live

Singapore Indoor Stadium

Mon 18 Aug 2003 8:00pm

AP-1	**6**	**2**
Section	Row	Seat

5782103821A12BC5101A764FFE291

Figure 4.5: Protected e-ticket

- *Verification:* In order to verify the authenticity and originality of a printed document, we first perform feature extraction as in the registration process to get the feature description of the print signature. Also, the encoded information is read from the document using either a bar code reader or an OCR scanner. The digital signature is verified first to ensure there have been no modifications to the document identifier. We then compare the extracted feature and contents on the paper with the document identifier, through a decision process. If the results match, the document is considered to be authentic and original. Otherwise it considered to be a faked one or have been tampered.

4.2.3 Feasibility analysis

We formalize the registration and verification procedures as follows:

The registration process can be described using:

$$S = (\{F(P), I\}, Sig(\{F(P), I\})) \tag{4.1}$$

where S is the printed information; $Sig(\cdot)$ is the digital signature scheme. P is the print signature; $F(\cdot)$ is the feature extraction function which is used to generate the description of print signature; and I is some critical information related to the document.

The verification process can be described as:

$$\begin{aligned} V1 &= V_{sig}(\{F(P), (I)\}, Sig(\{F(P), I\})) \\ V2 &= DM(F'(P'), F(P), I, I') \end{aligned} \tag{4.2}$$

where $V1$ is the verification of digital signature, $V2$ is the verification of critical document information and print signature. $DM(\cdot)$ is the discriminative decision function; $F'(\cdot)$, P' are the feature extraction function and the *print signature* respectively. It should be noted that P and P', $F(\cdot)$ and $F'(\cdot)$ may not necessarily be the same. This is because $F(\cdot)$ and $F'(\cdot)$ are built into two different devices, and the similarity between P and P' depends on the condition of the paper (e.g. any salt and pepper noise) and the inspection environment (e.g. illumination, focusing of the microscope, etc.).

Suppose an attacker intends to forge a document either by recreating a new document or by modifying the contents of an authentic document. In this case, his major task is to create a valid digital signature which can pass the verification procedure $V1$. This task is computationally infeasible unless the digital signature scheme used is compromised.

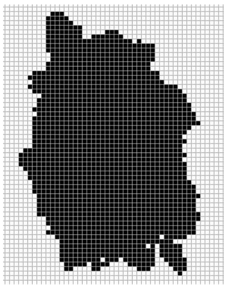


Figure 4.6: Quantized dot image

The attacker can also photocopy/scan–reprint an authentic document and claim it to be the original. The underlying task is to create a print signature P' which is the same as P, or satisfies $F'(P') = F(P)$ in order to pass $V2$. In [QG03], the authors have shown that completely recreating P through photocopying or scanning–reprinting with commercially available tools is impossible because of the nonlinear distortions and halftoning effects. Since it can be argued that P can be duplicated using professional equipment with higher resolution, let us refer to Figure 4.6. This is the leftmost dot of Figure 4.2 (b) being examined under a 200× microscope with an CCD array of 320×288 pixels. After quantization, the dot covers an area of 40 × 59 pixels. Considering the physical size of the dot is about 1/360 inch, this specific shape needs at least $59/(1/360) = 21240$ dpi resolution printer to reproduce. The number is 17 times larger than today's highest-end laser printer which has a resolution of 1200 dpi. This analysis shows that even if the attacker knows how an authentic print signature looks like, he does not have any method to create it. What he can do is exhaustively create and test various P', trying to find a collision wherein $F'(P') = F(P)$. In the following sections, we'll show that the probability of successfully creating such a P' is extremely low.

4.3 Authentication Process

The authentication function essentially compares the print signatures and assesses the degree to which a retrieved print signature matches the registered one. In what follows, we will first describe the feature extraction of print signature, then describe our matching algorithm and analyze its performance as well as security.

4.3.1 Feature Extraction for Print Signature

Feature extraction for print signature is performed during registration as well as authentication. It takes captured images of the print signature P as the input, and extracts the most descriptive features such as shapes, profiles, or spatial configuration of P as the output.

In our test setup, the "IntelPlay QX3" [4] computer microscope is used to capture the image of the print signature P. We have selected this cheap (costing 50 US dollars) microscope as a low-cost scanner. As a result, the quality of captured images is not always satisfactory. As shown in Figure 4.2(b-e), only the rough

[4] http://www.intel.com/support/intelplay/qx3/

shape of the four dots is invariant under illumination and focus changes. Therefore, we binarize and segment the four dots before extracting the shape as the descriptor of the print signature.

- *Binarization:* Binarization involves the selection of an optimal threshold such that the major features are preserved during conversion from a grayscale to a binary image. Good binarization accuracy is very important for the registration process which requires a precise description of the print signature. Illumination conditions and focus have a major influence on this. In a bad-illuminated and out-of-focused image, the edges of the four dots will be blurred, thus destroying the accuracy of binarization and later shape retrieval. To overcome this in the registration procedure, we first capture a set of images under different focus and illumination conditions. Then the images are binarized using the optimal threshold value defined by Otsu's algorithm [Ots79] which attempts to maximize the inter-class variance between the class of pixels above the threshold, and the class of pixels below. The average form of these images is used as binarization result. During authentication, we use Otsu's threshold directly.
- *Segmentation:* Since we have the a priori knowledge of how the secure pattern looks like, we can easily segment the four dots from the background. In case the four dots cannot be segmented successfully, this information is fed back to the binarizer, instructing it to adjust the threshold and redo the binarization. A set of segmented images for Figure 4.2(b) are shown in Figure 4.7.
- *Feature extraction:* The security of our method relies on the performance of feature extraction and matching for the print signature. In the test setup, shapes of the four dots have been identified as the main feature of print

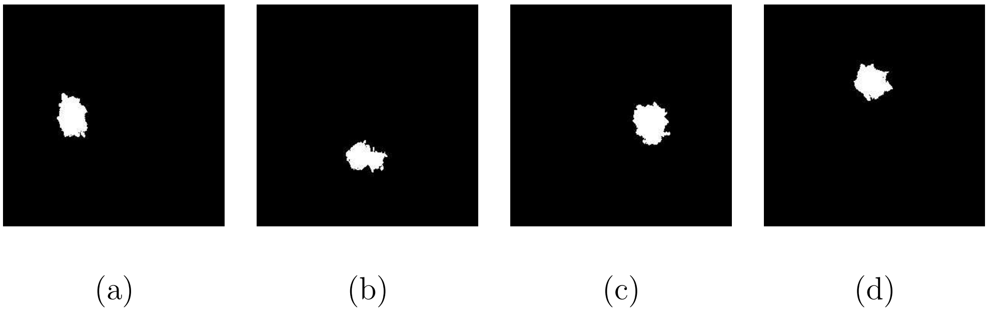


Figure 4.7: Segmented secure pattern: (a) Leftmost dot. (b) Bottommost dot. (c) Rightmost dot. (d) Topmost dot.

signature, so the problem is reduced to a shape matching problem. In the computer vision literature, shape matching methods have been studied for a long time, and a variety of solutions have been discovered. [VH99] provides a good review of these developments. However, our requirements are a bit different. Traditional shape matching algorithms are usually robust against affine transforms like translation, scaling or rotation. In our method, since landmarks are used to assist alignment, this robustness is not required. We can therefore use a simple radius profile as the descriptor for the shape of the dots.

For each dot, the radius is calculated from its centroid to the perimeter. Typical perimeter length of a dot in our experiment is between 180 to 200 pixels. Using all of them as the shape descriptor is not acceptable because:

1. The radius values are not independent of each other. Instead, the position of each pixel on the perimeter is determined by the pixels beside it. If the radius values of all pixels along the perimeter are used, there will be a lot of redundancy in the data set.
2. Finally the descriptor will be encrypted and encoded using symbolic encoding methods such as bar codes or OCR fonts. These methods have limited storage capacity.

Therefore, we partition the radius profile proportionally into several segments using polar coordinates and then calculate the average radius value r' for each segment, as shown in Figure 4.8.

Considering that the correlation between adjacent radius values of fan-shaped segments is low, we assume the obtained set of r' are independent variables. Thus, the profile can be represented as:

$$\vec{R'} = (r'_1, r'_2, \ldots, r'_{N-1}, r'_N)^T$$

Since we are only interested in the shape of the dots but not the size, we normalize the profile using its mean value:

$$\bar{r} = \frac{1}{N}\sum_{i=1}^{N} r'_i$$

$$\vec{R} = (\frac{r'_1}{\bar{r}}, \ldots, \frac{r'_N}{\bar{r}})^T = (r_1, r_2, \ldots, r_{N-1}, r_N)^T \tag{4.3}$$

Here N is the number of segments. It is a critical parameter for the overall security. We leave the discussion about N for the next section.

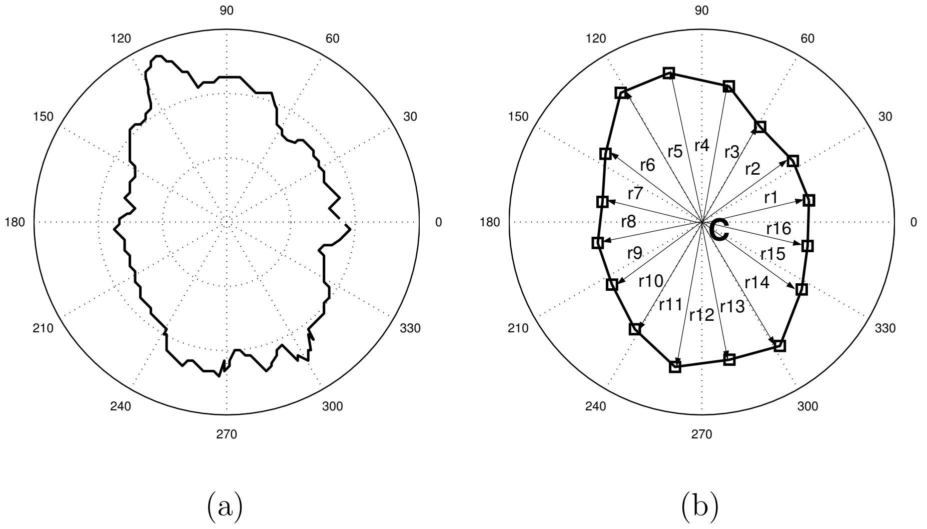
Thus, the feature description of our print signature is represented as:

$$F(P) = \{\vec{R}_1, \vec{R}_2, \ldots, \vec{R}_M\} \tag{4.4}$$

where M is the number of dots used in the secure pattern.

4.3.2 Profile Matching

Referring to the matching function 4.2, DM is a discriminative decision function that measures the similarity between extracted profiles of the print signature and the registered profiles. It must be carefully selected so that no authentic document is rejected (false-alarm rate is low) and no forged document is accepted (false-acceptance rate is also low). Given a reference profile $\vec{R}_{ref}$ for one dot, we use the



(a) (b)

Figure 4.8: Profile of print signature: (a) Profile of dot. (b) Average profile (16 segments).

following Euclidean distance classifier to differentiate profiles:

Define Euclidean distance as:

$$\begin{aligned} D(R, R_{ref}) &= \|\vec{R} - \vec{R}_{ref}\|^2 \\ &= \sum_{i=1}^{N} (r_i - r_{ref,i})^2 \end{aligned} \tag{4.5}$$

For threshold T, we consider:

$$\begin{aligned} D(R, R_{ref}) < T &\rightarrow R \text{ and } R_{ref} \text{ are the same} \\ D(R, R_{ref}) \geq T &\rightarrow R \text{ and } R_{ref} \text{ are different} \end{aligned} \tag{4.6}$$

and DM:

$$\begin{aligned} &R \text{ and } R_{ref} \text{ are the same for all } M \text{ dots} && \rightarrow \text{ACCEPT} \\ &R \text{ and } R_{ref} \text{ are not the same for any of } M \text{ dots} && \rightarrow \text{REJECT} \end{aligned} \tag{4.7}$$

Here, T, N and M are to be determined together with their performance analysis.

Matching using profiles can also be used for other types of print signatures such as the one detected on characters as shown in Figure 4.3. The profile of arbitrary shape is obtained by calculating the distance from its outermost perimeter to its centroid. But since there is no fixed location for the print signature to be detected, the location of a specific print signature must be encoded into the barcode, and a positioning mechanism is required to locate the print signature precisely.

4.3.3 Performance Analysis

In Equation 4.7, T and N are two very important parameters which determine the performance of the classifier. When T increases, the classifier becomes more robust against noise, but the false-acceptance rate increases. Conversely, when N increases, D also increases, the classifier becomes more sensitive to variance, but the false-alarm rate increases as well. We regard the radius values of the dots as random variables and then use a statistical model to estimate the optimal values for T and N.

To simplify the analysis, let us assume $M = 1$, that is, only one dot is used in the secure pattern. As shown in Equation 4.3, for a single dot A, profile $\vec{R}$ can be considered as a joint distribution of independent random variables $r_1 \ldots r_N$. The randomness of these variables comes from the environmental conditions when capturing the image, and the threshold value used to binarize the image. To study the distribution of these variables $r_1 \ldots r_N$, we captured 100 images for the same dot using different illumination and focusing conditions. For each image, we use 5 distinct threshold values for binarization. So altogether we obtained 500 binarized images. Then for each different N from 8 to 64, we partition the profile into N segments and calculate the average profile $\vec{R}_{j,N}$, $j = (1 \ldots 500)$. The distribution of each radius value r_i in $\vec{R}_{j,N}$ can be determined by computing the histogram of $r_{j,N,i}$, $j = (1 \ldots 500)$, $i = (1 \ldots N)$ for each i. In our experiment, the shapes of the histograms had a Gaussian profile, so we assume r_i obeys a Gaussian distribution. By using the “Bera-Jarque Normality Test” [Jud88], our hypothesis is verified with a P-value (significance level, the larger the better) of 40%.

Another useful result that we obtained from the experiments is that the standard deviation for each set of r_i is almost the same. This can be explained as follows: the randomness is homogeneous for all directions. We use symbol σ to denote the standard deviation hereafter.

Let

$$\bar{R} = (\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_{N-1}, \bar{r}_N)^T$$

denote the mean profile of dot A from the above experiments, and

$$R = (r_1, r_2, \ldots, r_{N-1}, r_N)^T$$

denote the profile of dot A obtained from one test, we have

$$\frac{r_i - \bar{r_i}}{\sigma} \sim N(0,1), \qquad i = (1 \ldots N)$$

which means that $\frac{r_i - \bar{r_i}}{\sigma}$ obeys the Gaussian normal distribution.

Now consider

$$D(R, R_{ref}) = \sum_{i=1}^{N} (r_i - r_{ref,i})^2$$

Note that the reference value R_{ref} is obtained from the registration process where the average value of multiple images is used. So $r_{ref,i}$ must be very close to $\bar{r_i}$. Thus,

$$D(R, R_{ref}) \doteq \sum_{i=1}^{N} (r_i - \bar{r_i})^2$$

It therefore follows that:

$$\frac{D(R, R_{ref})}{\sigma^2} = \sum_{i=1}^{N} (\frac{r_i - \bar{r_i}}{\sigma})^2$$

So

$$\frac{D(R, R_{ref})}{\sigma^2} \sim \chi^2(N)$$

is a chi-square cumulative distribution with N degrees of freedom.

We demand that the "false-alarm" rate be lower than 0.5%, or,

$$P(D(R, R_{ref}) > T) < 0.005$$

Table 4.1 shows the acceptable T and N values under this requirement.

N	σ^2	$\chi^2(N) = 0.995$	T
72	0.001589	106.7	0.16940
36	0.001241	61.58	0.07641
32	0.001246	56.33	0.07017
24	0.001063	45.56	0.04841
16	0.001015	32.27	0.03478
8	0.000821	21.96	0.01801

Table 4.1: Choice of segments and threshold under false-alarm rate $< 0.5\%$

In Section 4.2.3, we raised the question that whether it is possible to find another profile P' such that $F'(P') = F(P)$. We rephrase the problem under the current context to ask the question: given a discriminative function D and parameters N and T, how large is the probability for two distinct profiles R and R', to have $D(R, R') < T$.

To answer this question, we must know the distribution of radius r across different dots. Our experiment on 400 different dots shows that the distribution of r is Gaussian, with an average P-value of "Bera-Jarque Normality Test" of 37%. The result is easy to explain: the average radius value of the dots is our designed dot's radius in the secure pattern. By the central limit theorem, the distribution of radius value must conform to a Gaussian distribution under the sum of a large number of random influences.

Now consider two distinct profiles R and R', and

$$r_i, r_i' \sim N(\bar{r}, \sigma_r^2), \qquad i = (1 \dots N)$$

where $\bar{r}$ is the average radius value and σ_r is the standard deviation of radius r, we have

$$(r_i - r_i') \sim N(0, 2\sigma_r^2),$$
$$\frac{r_i - r_i'}{\sqrt{2\sigma_r^2}} \sim N(0,1), \qquad i = (1 \dots N)$$

Obviously, for

$$D(R, R') = \sum_{i=1}^{N} (r_i - r_i')^2$$
$$\frac{D(R, R')}{2\sigma_r^2} \sim \chi^2(N)$$
$$P(D(R, R') < T) = P(\frac{D(R, R')}{2\sigma_r^2} < \frac{T}{2\sigma_r^2})$$

Substituting T and N using the values shown in Table 4.1, we have the false-acceptance rate as shown in Table 4.2.

N	σ_r^2	$P(D(R, R') < T)$
72	0.02387	4.468×10^{-34}
36	0.02228	1.333×10^{-18}
32	0.01846	8.636×10^{-15}
24	0.02078	3.787×10^{-13}
16	0.01846	3.950×10^{-08}
8	0.01291	4.680×10^{-04}

Table 4.2: The false-acceptance rate

We can find that the probability of successfully creating a print signature whose profile P' can pass our verification process is very low even for only one dot. For a secure pattern with M dots, since we accept a document to be authentic only when the profiles of all M dots are matched, the false-alarm rate will be

$$1 - (1 - P_{f.alarm})^M,$$

and the possibility of false–acceptance wherein all M faked dots are matched will be

$$P_{f.accpt}^M$$

Use $P_{f.alarm} = 0.5\%$, $N = 32$ and $M = 4$ as an example, the false-alarm rate becomes 1.98% and the false-acceptance rate is reduced to 5.562×10^{-57}. The false-alarm rate is slightly increased but the false-acceptance rate is greatly decreased. Of course, the use of more dots requires more computation. The choice of M should be balanced between security concerns and acceptable resource costs.

It must be pointed out that our reasoning is based on the assumption that the radius r is a continuous random variable. In practice, since the value must be quantized, the false-alarm and false-acceptance rate will be amplified. However, this problem can be mitigated by the use of high resolution image sensors.

4.4 Experimental results

In this section, we will present extensive experimental results to demonstrate the feasibility of our proposed method for document authentication. Our experiments are intended to test whether an authentic document can successfully pass the authentication process, and a forged document can be successfully rejected. We use $N = 32/T = 0.07017$ as shown in Table 4.1, because these values seem to be optimal in terms of good discriminative power as well as low storage requirement. The secure patterns used in the experiments are composed of 1–4 small dots respectively. The printers used are a HP LaserJet 8100 monochrome printer and a HP LaserJet 4600 color printer. For the color printer, the secure pattern was printed using the black channel. The paper we used in the experiments includes plain paper, color paper, translucent paper and card paper. These types of paper are widely used in all kinds of business and administrative documents such as certificates, bills of lading, invoices, licenses and checks.

We printed authentic documents with combinations of secure patterns, printers and papers. Some captured pattern images are shown in Figure 4.9. The reference profile for a secure pattern was obtained by averaging the results from multiple tests as was discussed in Section 4.3.1. For each authentic document, we capture another 50 images by changing illumination, focusing, and by applying some small mis-alignments. The Euclidean distances between the profile of these images and the reference profile are marked using 'o' in Figure 4.10. We also created 50 forged copies for each authentic document by reprinting/scanning–reprinting the same document. We did not perform the photocopying test because our initial experiments had shown that the quality of photocopied documents is very bad. The print signature was destroyed to such an extent that we could not even segment the dots. The Euclidean distances between the profile of these forged images and the reference profile are marked using 'x' in Figure 4.10.

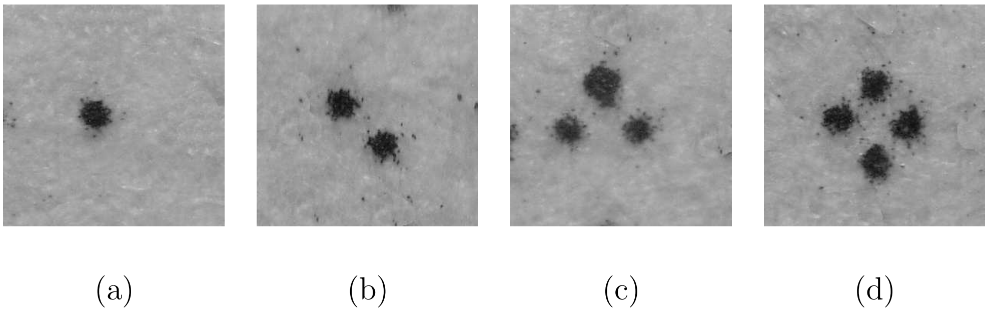
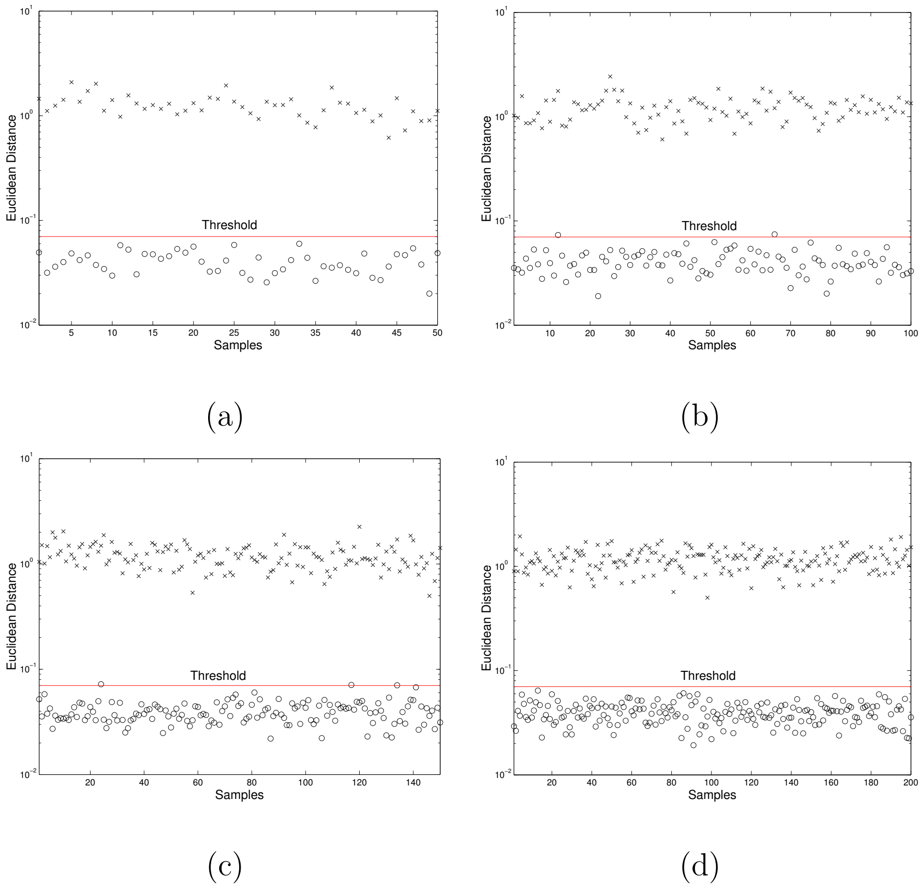


Figure 4.9: Experimental print signatures: (a) Plain paper. (b) Color paper. (c) Translucent paper. (d) Card paper.



(a)

(b)

(c)

(d)

Figure 4.10: Experimental results for print signature: (a) Plain paper. (b) Color paper. (c) Translucent paper. (d) Card paper.

As we can see from the results, no forged document has been accepted. But as the number of dots increases, there have been a few occasions that authentic documents are rejected. This result conforms to our designed false-alarm rate of 0.5% for one dot and 1.98% for four dots. Rejecting all forged documents at the expense of erroneously rejecting a few authentic documents is acceptable since in most cases forged documents can cause a lot more damages. However, if we perform another round of validation when a document is rejected, the false-alarm rate can be greatly reduced.

4.5 Conclusion

In this chapter, we have proposed a novel authentication technique for printed paper document. The print signature is based on the inherent randomness present in the physical printing process. The security of the method is guaranteed by both the digital signature and the print signature. The method has been demonstrated to be secure against forgery and duplication attacks.

As the laser printing technology improves, the printing resolution will become even higher. However, as long as the underlying mechanism is unchanged, we still expect to see the random phenomenon on each copy of printed paper. This will only entail the use of microscopes of even higher resolution.

This method can be readily extended to other document types such as offset-printed documents, ink-jet printed documents, or manually signed documents. It basically reduces to the task of finding unique randomness in each copy of the document to be used as a signature. For example, the ink trail for each manually signed document is unique. As long as the uniqueness is found, a new document authentication method based on the same principle can be developed.

Print signature authentication method can be integrated with RSE authentication method to form an end-to-end document authentication system, as shown in Figure 4.11.

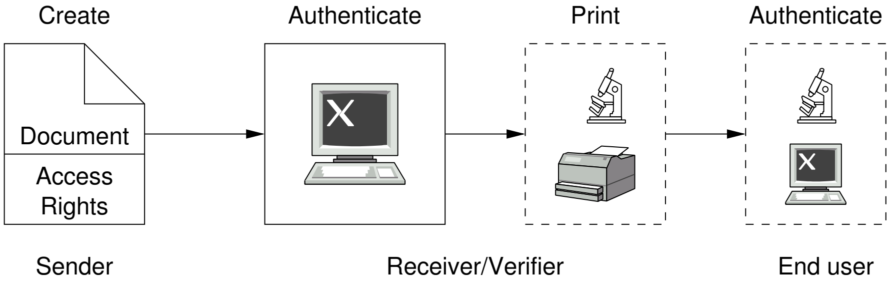


Figure 4.11: Integrating RSE and print signature

1. The sender creates a document together with the render sequence authenticator. He also specifies some access rights to the document, such as whether the document can be printed, and how many copies can be printed. The authenticator and the access rights are encoded into the document using RSE watermark scheme.
2. When the document reaches at the receiver's side, the receiver authenticates the document first using RSE authentication method. He then extracts and executes the access rights. If the document is printable, he creates a printable version of the document, selects some critical document information to be protected, encodes the information into document again using RSE watermark scheme, then sends the documents to the printing system for printing.
3. The printing system prints the document with print signature. The "critical document information" required is obtained from RSE embedded information.
4. The printed paper document is finally given to the end user for verification.

It can be seen from the system that the communications among different parties contain nothing special except normal file exchanges. This is attributed to the RSE watermark scheme and paper-based authentication method, which enable additional security without modifying the existing protocol. The document's authenticity and originality is well protected in the whole creation–deployment–usage process. It realizes the objective of "managing all forms of rights both in physical and digital form".

Chapter 5

Model and Framework for XML Based Access Control

In this chapter, we present an XML based access control framework. Access control is required for interactive document creation, where users' privileges are governed by the system policy. The access control framework is further incorporated with other document security features to form an integrated DRM system framework, which provides end-to-end rights management for both electronic and paper documents.

5.1 Introduction

In business and administrative environments, a DRM system should facilitate exchanges of ideas among authors during document drafting, leading to the final version. Versioning and interfacing with databases and other existing document frameworks for automatic document processing are preferable. Such cooperative document creation scenario poses great challenges upon the interoperability and security of the system, which can be briefly identified as two basic requirements:

1. Defining a set of standard interchangeable semantics and vocabulary for business and administrative document description. This standard forms the foundation of the intra- and inter-organizational document exchange. It includes not only document format definition, but also a set of standard methods of database interfacing (for data collection) and form generation (for creating human-readable documents).
2. Formulating a set of standard rights expression languages (RELs), and designing access control method to enforce users' rights. Access control applies to both document drafting and document deployment stages. It requires that the authors and the end users conform to a pre-defined set of rights usages so as to maintain trusted relationship among them.

The eXtensible Markup Language (XML) offers an ideal platform for document exchange because of its favorable features of opening standards, extensibility, easy database interfacing and platform neutrality. Recent evidence [Mar03] shows global adoption of XML as a standard business language has become a trend. Although rival business XML specifications exist (e.g., ebXML[1], boleroXML[2] and RosettalNet[3]), we have sufficient reason to believe that XML will be the foundation of business and administrative documents for the future. In this sense, the adoption of XML format basically fulfills the first requirement.

An important feature of XML is that it can represent information at different levels of sensitivity. Developing access control mechanisms that define which part of the document is accessible by whom thus becomes a hot topic in the research community. In this chapter, we consider a role based access control (RBAC) framework as one possible solution. The framework is implemented using "pure" XML

[1] http://www.ebxml.org

[2] http://www.bolero.net

[3] http://www.rosettalnet.org

technologies include ODRL, XML Schema and XML Schematron. It conforms to the standard RBAC model, so its security is verified and trustworthy.

We have introduced the basis of access control model and rights expression languages in Section 2.3. In the next, we first present our XML based RBAC framework in Section 5.2, then present an integrated DRM framework for electronic documents in Section 5.3. The chapter is concluded in Section 5.4.

5.2 XML based RBAC framework

It has been identified that role-based access control "addresses many of the security needs of both the commercial and government sectors [NIS99]". The implementation of RBAC framework requires two set of specifications: one defines the RBAC components and their relationship, for example, the *user*, *role*, *permission*, and *user-role assignment* components; the other defines the constraints in the construction of the framework, notably the *cardinality* and *separation of duties* constraints. For illustrative purpose, we present our XML based RBAC framework within the context of a real-world document workflow in the shipping industry. As we've discussed in Section 1.1, the process is very document intensive. We first briefly describe the existing process, and then consider how the process can be formulated and implemented using XML and RBAC model.

5.2.1 Document workflow in shipping application

Figure 5.1 is a simplified workflow of bills of lading (Bs/L) document workflow used in the shipping industry. For a shipper to consign with a shipping company to ship some cargo, the following steps are carried out:

1. The shipper submits a new B/L request to the shipping line, specifying

information such as port of loading, port of discharging, cargo details, etc.

2. On receiving the B/L request, a port clerk from the shipping company processes the information, updates specific fields for the shipper to confirm. The shipper reviews the updated information, either confirms or makes further amendments for the port clerk to confirm. The process is carried out iteratively until both sides have confirmed.
3. The confirmed B/L request is sent to the port manager for approval.
4. On approval, the port manager prints the original Bs/L (typically contains 3 original negotiable Bs/L and 2 copies of non-negotiable Bs/L) and signs each negotiable B/L. The document set is then sent to the discharging port by courier service for the shipper to claim his cargo.

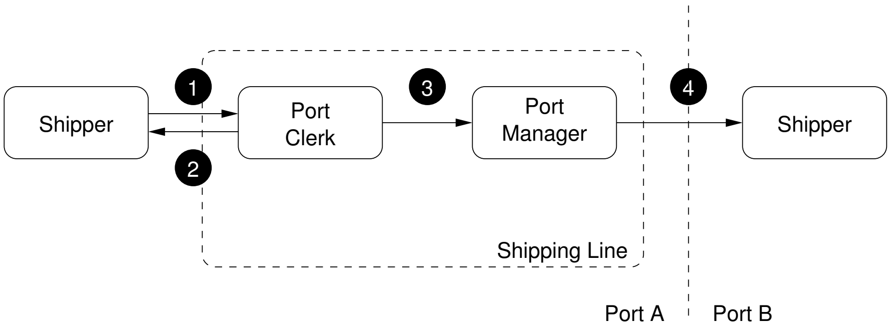


Figure 5.1: Document workflow in shipping industry

The 4th step in the B/L workflow is very costly, and usually results in additional delay in the entire transaction. The current workaround is that some shipping companies give their valued customers pre-printed blank B/L so that their customers can print the details and sign on behalf of the shipping companies. However the shipping companies are unable to control or track what is actually printed on these forms and who has access to the forms. Alteration, forgeries and fraud using such documents are common. The shipping industry is in urgent need for an Internet-based Bs/L document workflow system which provides convenient document delivery as well as strict access control to the all parties participated in the workflow.

5.2.2 RBAC for B/L workflow

The heart of the B/L document workflow is the B/L source data, which are described using XML format to facilitate data exchange. The access control is to grant or deny user's operation to these data. We attach one record of B/L (Listing A.1, `bl.xml`) source data at the appendix of this thesis. This file describes the required data for generating a valid B/L document. Note that the `<status>` element at line 21 shows the current state of the B/L. It has 5 possible values:

- `both_not_confirmed` – Both the shipper and the port clerk have not confirmed the B/L.
- `wait_shipper_confirm` – The port clerk has confirmed the B/L, waiting for the shipper to confirm.
- `wait_clerk_confirm` – The shipper has confirmed the B/L, waiting for the port clerk to confirm.
- `both_confirmed` – Both the shipper and the port clerk have confirmed the B/L, waiting for the port manager to approve.
- `approved` – The port manager has approved the B/L. It is ready for printing and deployment.

Access rights of different users are determined not only by the users' privilege, but also by the status of the B/L, which are explained below.

Role definition and privilege identification

Four categories of users (roles) can access the B/L source data:

- Shipper – Modify `<shipper_id>`, `<loading_port>`, `<discharging_port>`,

<consignment> elements, read <negotiable>, <nonnegotiable> elements, and update <status> element if the original status was 'both_not_confirmed' or 'wait_shipper_confirm'. When the B/L has been approved by the manager (<status>='approved'), shipper can print negotiable B/L if left copies <negotiable/left> is larger than zero, or print non-negotiable B/L if left copies <nonnegotiable/left> is larger than zero.

- Port clerk – Read all the elements, modify <clerk_id>, <vessel_name>, and update <status> element if the original status was 'both_not_confirmed' or 'wait_clerk_confirm'.
- Port Manager – Inherit all privileges from the port clerk, additionally can modify <manager_id> element and update <status> element if the original status was 'both_confirmed'.
- Application – This is a role hidden in the workflow. It performs all the background jobs to maintain the integrity of the B/L, which include:
  - Update <bl_id> and <internal_ref> elements when a new B/L request has been submitted.
  - Update the <negotiable> element with (<printed>=0, <left>=3) and <nonnegotiable> element with (<printed>=0, <left>=2) when the B/L has been approved by the port manager.
  - Update <negotiable> and <nonnegotiable> elements when the shipper prints the B/L.

Role hierarchy

The role hierarchy is shown in Figure 5.2. Roles higher in the hierarchical chain inherit the privileges from roles lower in the chain.

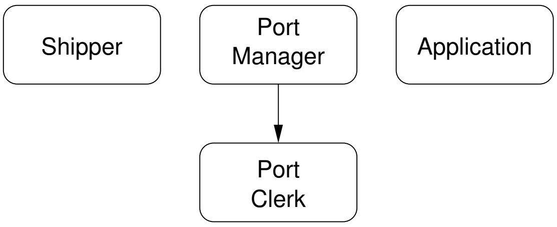


Figure 5.2: Role hierarchy for the B/L workflow

Constraints

A set of constraints help to maintain the integrity of the B/L document and the workflow, which include:

1. The maximum number of users that can be assigned to "Application" role is one.
2. The following pair of roles cannot be assigned to the same user (static separation of duties):
    - "Shipper" and "Application"
    - "Shipper" and "Port manager"
    - "Shipper" and "Port clerk"
    - "Port clerk" and "Port manager"
3. The following pair of roles cannot be activated in a same session (dynamic separation of duties)
    - "Application" and "Port clerk"
    - "Application" and "Port manager"

The definitions of roles, privileges, role hierarchy and constraints directly map to the corresponding components in NIST RBAC model as shown in Figure 2.3.

5.2.3 B/L RBAC framework

In our B/L RBAC framework, the access control policies are described and stored in XML format. The adoption of XML into RBAC framework requires that the XML description can: 1. Define all the components in RBAC model accurately. 2. Exercise all constraints in RBAC model strictly. We put these definitions and constraints into a single framework using "pure" XML technologies, as shown in Figure 5.3.

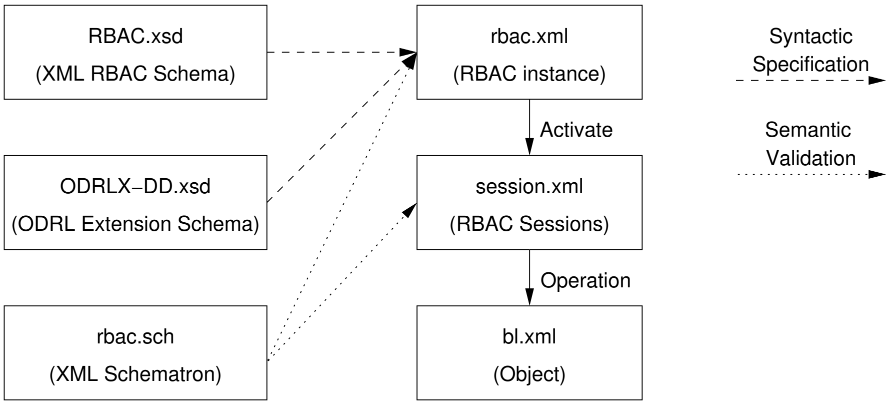


Figure 5.3: XML based RBAC framework

In this framework, the access control policy is described in `rbac.xml`. XML Schema `RBAC.xsd` and `ODRLX-DD.xsd` provides syntactic specification for the definition of RBAC components. XML Schematron `rbac.sch` provides semantic validation to `rbac.xml` and the run-time session database `session.xml`. Through `session.xml`, the system is able to grant or deny access operations to the object, `bl.xml`. In a typical implementation, `RBAC.xsd`, `ODRLX-DD.xsd` and `rbac.sch` are designed by the system integrator or developer to formulate the RBAC control model. The RBAC policy description `rbac.xml` is created by the system administrator which defines company specific permissions, roles, users and user-role assignment. It is to be regarded as one instance of the RBAC implementation. Run-time session database `session.xml` is created by the application, which dynamically records user logins and session activations. We have included a set of example files in the appendix of this thesis. In what follows, we present some details about `RBAC.xsd`, `ODRLX-DD.xsd` and `rbac.sch` to examine how the RBAC model is implemented using these files.

Components definition using XML Schema

The definitions of RBAC components in `rbac.xml` must conform to application specific data types, content relationships and structures. XML Schema [W3C01a, W3C01b, W3C01c] is the technology to establish such specification. The advantage of XML Schema over its alternative the Document Type Definition (DTD) is that XML Schema supports complex constraints for XML components such as elements, attributes and data types, thus strengthens the veracity of the specification. We use XML Schema `RBAC.xsd` to define the "User", "Permission", "Role", "User-Role Assignment" components in RBAC model, as shown in Figure 5.4 (generated using XMLSpy [4] software). At the top level of RBAC Schema are

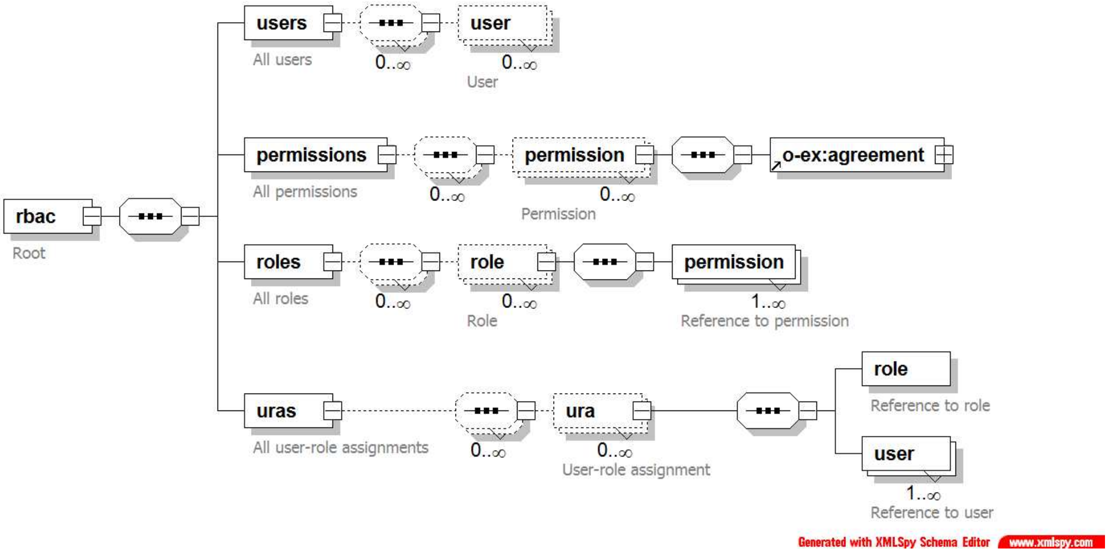


Figure 5.4: RBAC Schema (RBAC.xsd)

[4] http://www.xmlspy.com

the definitions of `<users>`, `<permissions>`, `<roles>` and user-role assignments (`<uras>`) components. Each `<role>` component contains one or more references to the `<permission>` components to implement the Permission Assignment (PA) task in Figure 2.3. User-role assignments `<uras>` element contains a sequence of `<ura>` elements, each of which combines a role with the users assigned to that role through XML reference too. The advantage of using element references instead of defining elements locally is that we can separate permission and user definitions under other namespaces or databases so as to encourage data re-use. The complete listing of `RBAC.xsd` is attached in Listing A.2, inside which two elements deserve more explanations:

Schema for the `<role>` element. The RBAC model for B/L workflow in Section 5.2.2 defined four types of roles: "application", "clerk", "manager" and "shipper". The RBAC Schema hence must provide means for limiting the role definition inside these types. We include an attribute `id` in the `<role>` element to serve this purpose. The value of the `id` attribute has been restricted as enumeration of four options: "application", "clerk", "manager" and "shipper" (lines 63–73). We also use `xs:ID` data type to define `id`, so no duplicated `id` value are allowed. Through this definition, any illegal implementation of `<role>` elements can be invalidated using the Schema.

Schema for the `<permission>` element. The definition of the `<permission>` element is the most complicated. Simple permissions like "observation" or "alteration" are too general to give precise descriptions. Here we base our permission definition on ODRL specification which is a lot more advanced. The term "permission" in RBAC model allows a set of "operations" onto a set of "objects".

In ODRL syntax, the corresponding terms of “permission”, “operation” and “object” are `<rights>`, `<asset>` and `<permission>`, respectively. ODRL assembles these terms inside an `<agreement>` element, meaning that the `<rights>` contain `<permission>` over `<asset>`. ODRL also allows `<constraint>` element as a child of `<permission>`, which adds restrictions to the permissions. By this structure, ODRL is capable of defining very complex rights, especially in the context of digital media distribution. However, we must make two extensions for ODRL to be used to express permissions on XML data source.

The first extension is to add XPath [W3C99] expression into ODRL’s `<asset>` element. It is to solve the `UID` inability problem we identified in Section 2.3.2. The XPath expression is added as a substitution to the ODRL context model. It introduces a new `<selector>` element with an `xpath` attribute which points to the specific XML nodes as the ODRL asset. For example, we can identify the `<shipper_id>` element in `bl.xml` using:

```
<asset>
 <context>
   <selector xpath="bill_of_lading/shipper_id"/>
 </context>
</asset>
```

The second extension is to add an XPath based condition evaluator into ODRL’s `<constraint>` element. With this extension we can support conditional permission definitions which are required in business and administrative environments, for example, the shipper can modify the `<shipper_id` only of the `<status>` was `'both_not_confirmed'` or `'wait_shipper_confirm'`. The extension is added as a substitution to the ODRL constraint element model. It introduces a new `<assert>` element with a `test` attribute which defines a XPath evaluator. An `<assert>` element inside an ODRL `<constraint>` element means that the constraint is effective only if the assertion is successful. For example, we can define shipper's privilege of modifying `<shipper_id>` element as:

```
<agreement>
 <asset>
  <context>
   <selector xpath="bill_of_lading/shipper_id"/>
  </context>
 </asset>
 <permission>
  <modify>
   <constraint>
    <assert test=
     "(/bill_of_lading/status='both_not_confirmed')||
      (/bill_of_lading/status='wait_shipper_confirm')"/>
   </constraint>
  </modify>
 </permission>
</agreement>
```

The above definition uses ODRL `<agreement>` element to combine extended `<asset>` and `<permission>`, meaning that the system allows for `'modify'` permission onto `<asset>` only if the evaluation of `<assert>` constraint is true.

The two ODRL extensions are defined using a separated Schema file `ODRL-DD.xsd`, as listed in Listing A.3. We've also included a commented RBAC policy definition (`rbac.xml`) in Listing A.4 as an illustration of the definitions of other RBAC components as well as the assembly of all these components.

Constraints specification using XML Schematron

By introducing XML Schema, we have successfully realized the role definition, privilege identification, role hierarchy and user-role assignment tasks. Here, Schema serves as the design tool, establishing a framework on which implementations can be built. XML Schema not only specifies the syntactic structures for RBAC policy (`rbac.xml`) creation, but also provides limited semantic constraints in term of preventing duplicated elements using `xs:ID` or string enumeration (e.g., `role/@id`) data type. However, such constraint alone is not sufficient to constitute a solid framework for access control. Quite a few domain-specific policy constraints (see Section 5.2.2) have not been addressed. The reason why XML Schema cannot support these constraints is that the constraints restrict the content of XML document, while XML Schema only restricts the structure of the document.

Currently there have been two approaches proposed as complements to the DTD or XML Schema for XML content validation – the *XML Constraint Specification Language (XCSL)* [Ram01] and the *XML Schematron* [Dod01]. We use XML Schematron in our framework because Schematron is pure XML based technology and seems to gain more support.

Schematron validates XML content with a simple action: *find* a context node in XML document based on XPath criteria, then *check* to see if some other XPath expressions are true for the node. In the Schematron constraint definition, the `<rule>` element sets the context; within the `<rule>` element are one or more `<assert>` elements, each of which evaluates an XPath expressions and emits predefined strings or invoke optional `<diagnostic>` procedures if the assertion fails. A skeleton of XML Schematron is:

```
<rule>
 <assert test="..." diagnostics="...">Error text</assert>
 <diagnostics>
  <diagnostic id="...">Error text</diagnostic>
 </diagnostics>
</rule>
```

We now illustrate how different types of constraints can be expressed using XML Schematron.

The first type of constraint is data type validation in XML element referencing/dereferencing. Strictly speaking data type validation is not a semantic constraint, but XML Schema does not provide any means to support such validation. For example, the `<role>` element in `rbac.xml` contains references to the `<permission>` elements. But the references are defined using `xs:IDREF` type in XML Schema which cannot differentiate what type of elements it is really referencing. As such, it is always possible to reference a `<user>` element in place of `<permission>` meanwhile the violation cannot be detected by XML Schema. By using XML Schematron, we can read the reference identifier from `<role/permission/@ref>` attribute, and check if it really refers to a `<permission>` element, as shown in lines 5–13 in `rbac.sch` (Listing A.5).

The second type of constraint is the cardinality constraint defined in Section 5.2.2. For example, the B/L RBAC model requires that only one user can be assigned to the "application" role. If we use XML Schema to support this constraint, we would have to define a special user-role assignment (`<ura>`) element which allows only one `<user>` element as its child. It is very cumbersome approach when more and more cardinality constraints are required. With XML Schematron, we can count the number of users under each role using XPath's `count()` function, and determine if the number violates the constraint. For example, lines 14–24 in `rbac.sch` count the number of users under the `<ura>` element whose `<role>` element references to "application" role, and emit error messages when the number does not equal to one.

The third type of constraint defines the "separation of duties (SOD)" constraint in the NIST RBAC model. It specifies a set of conflicting roles that a single user cannot be assigned to these roles simultaneously. For example in our B/L workflow, no users can be assigned to both "shipper" role and "application" role. XML Schema cannot exercise this constraint. With XML Schematron, we can specify this constraint using XPath query and evaluation. We first obtain two sets of users who are assigned to the "shipper" role and the "application" role, then use XPath = operator to check if there is an intersection. The specific Schematron code is listed in lines 25–34 in Listing A.5. This Schematron constraint can also be used to support "dynamic separation of duties (DOD)'. It basically reduces to the task of finding the intersection between two set of users who have activated conflicting roles.

It is worth to note that the above three types of constraints respectively realized the restrictions on XML attribute value, number of elements, and relationship among elements. Together with the XML Schema which is capable of restricting the type of elements, all the primitive XML building blocks have been included. So it is sufficient to conclude that other constraints can also be implemented using XML Schema and Schematron technologies.

Conclusion

Through the use of XML Schema and Schematron, we have defined a framework to implement role-based access control using XML technologies. We have also brought in extended ODRL for rights expression, which greatly eases the need for complex usage controls. Unlike XACML [And03] which stops at the "description" level, our method combines both definition and validation into a single framework. The validation does not require any proprietary utilities but standard XML Schema and Schematron validator. Since the construction of our XML framework is based on the standard RBAC model, and all the components and constraints in RBAC model have been successfully realized, our framework can be considered as a secure and effective implementation of RBAC framework.

5.3 Towards an integrated DRM framework

The RBAC model and framework we discussed in the previous section is aimed at strengthening the security of XML data source and XML document in the document's creation and deployment stages. Using XML as the only file format to conduct business or administrative transactions is hardly to be true at least in the near future. People still need formatted electronic documents and paper documents everywhere for viewing and exchanging. By adopting the "Render Sequence Encoding" and "Print Signature" method we proposed in the previous chapters, we can build an end-to-end integrated DRM framework for electronic documents workflow.

Still using the shipping company sample, an end-to-end DRM framework is depicted in Figure 5.5.

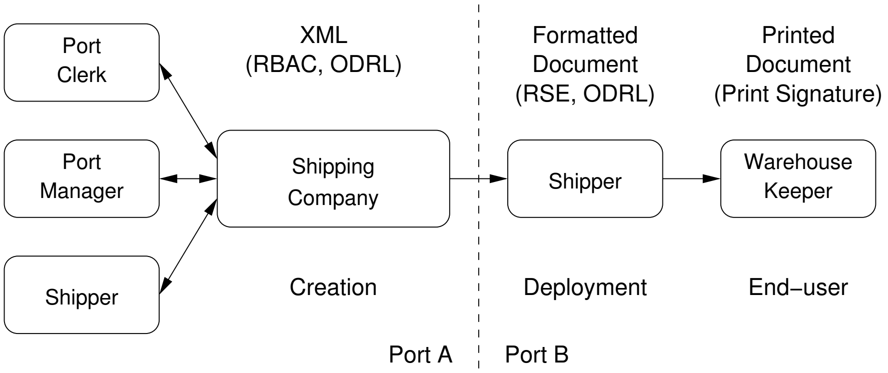


Figure 5.5: An integrated DRM framework for shipping companies

The figure illustrates the creation, deployment and end-user stages of B/L workflow. At the creation stage, the DRM system mainly interfaces with authors like port clerks, port managers and shippers for the creation of data needed by a final B/L. Access control among these users is the major task in this stage. Our XML based RBAC framework is designed to address this problem by managing their rights and usages on a role-based manner. The security measures in this stage are taken from two perspectives: XML Schema and Schematron protect the system architecture by eliminating all illegal policy specifications and user activities, and the ODRL protects the integrity of B/L data by limiting the usage rights.

When the original B/L is to be delivered to the discharging port, it must be formatted into human readable electronic documents. Existing XML tools which do document formatting include XSL Formatting Objects (XSL-FO) [5] style sheets and formatting engines. The authenticity and integrity of formatted documents are the major targets of protection in this stage. Our “Render Sequence Encoding (RSE)” method addresses this requirement by hiding content-related information into documents during formatting. Access control over formatted documents is enforced by ODRL rights descriptions. For example, the shipper is allowed to print negotiable B/L three times. The license for printing is described as:

```
<?xml version="1.0" encoding="UTF-8"?>
<o-ex:rights
  xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
  xmlns:o-ex="http://odrl.net/1.1/ODRL-EX">
  <o-ex:agreement>
    <o-ex:asset>
      <o-ex:context>
        <o-dd:uid>
          urn:shippingline.com/bl_shipper.pdf
        </o-dd:uid>
      </o-ex:context>
    </o-ex:asset>
    <o-ex:permission>
      <o-dd:display/>
      <o-dd:print>
        <o-ex:constraint>
          <o-dd:count>3</o-dd:count>
        </o-ex:constraint>
      </o-dd:print>
    </o-ex:permission>
    <o-ex:party>
      <o-ex:context>
```

[5] http://www.w3.org/TR/xsl/

```
        <o-dd:uid>
          x500:c=SG;o=Shipper;cn=ShipperUser
        </o-dd:uid>
      </o-ex:context>
    </o-ex:party>
  </o-ex:agreement>
</o-ex:rights>
```

It allows a shipper holding a valid X.500 identity to display the B/L unlimited times, but print only 3 times. Using ODRL facilitates interoperation between shipping company and shippers, because ODRL is an open standard which is easy to follow and implement. Transferring of ODRL license is archived using RSE, by embedding ODRL license into the formatted document itself.

The shipper at discharging port authenticates the document and print negotiable B/L according to embedded ODRL license. The printed B/L will be finally presented to a warehouse keeper for claiming of the cargo. The authenticity and originality of the B/L become extremely important in this stage. The “Print Signature” method is used to provide this protection. Instead of giving their customers pre-printed blank Bs/L, the shipping company gives/loans their shippers specialized printers and computer systems capable of printing Bs/L and making print signatures. The printing process is done interactively with shipping companies’ database server. Holding the Bs/L with valid print signature, shippers are ready to claim cargo shipped to them. At the verification side, the warehouse keeper has been equipped with devices capable of authenticating Bs/L by verifying the print signature. The verification process is done off-line with no need to access any database. It is also possible for several shipping companies to share one single authentication device, because digital signatures can sufficiently differentiate them apart. Through this manner, the authenticity and originality property has been successfully migrated from the electronic world to the paper world. Thus, the whole integrated framework is complete and end-to-end.

It must be noted that although our discussion has been carried out in the context of Bs/L workflow for shipping industry, the DRM framework is easily extended to other electronic document workflow which demands security and interoperability. The implementation basically reduces to the task of identifying roles, permissions and constraints in the existing systems, describing these components using XML technologies, and linking the document creation system to the RSE-enabled document formatting system and print signature-enabled document printing system.

5.4 Conclusion

In this chapter, we have proposed an XML based access control framework for XML documents. The framework is built on top of the traditional role-based access control model. The advantage of adopting XML technology is that XML is platform, operating system and application neutral. It serves as a glue layer among interacting systems to achieve interoperability. Our approach uses XML Schema and Schematron as the modeling language to provide both syntactic and semantic constraints, so our framework is a verifiable and complete implementation of RBAC.

Our XML RBAC framework incorporates ODRL as rights expression. Using a mature and open rights expression language instead of developing a proprietary one can greatly improve the flexibility and reliability of the framework. We also made extensions to the ODRL specification which use standard XPath query and expression evaluation to support elemental level and conditional rights description for XML documents. Although the extensions are proposed under RBAC context, they are actually independent technologies and can be used in other XML security applications.

Based on the XML RBAC framework, we proposed an integrated DRM framework for electronic documents. This large framework incorporates the RSE and Print Signature technologies proposed in previous chapters for persistent protection of electronic documents. It ensures the authenticity, integrity and originality for both electronic documents and paper documents. The framework can find applications in many business and administrative document workflow systems.

Chapter 6

Conclusion and Future Work

This thesis presents multiple aspects of digital rights management for electronic documents. We have shown that digital rights management contains different requirements for different applications. In the business and administrative environment, a DRM system should protect authenticity, integrity and originality of the documents, facilitate inter-operations, and provide strict access control.

DRM is "digital" management of rights. The rights apply to the whole life cycle of targeted assets, in both tangible and intangible form. This DRM definition motivated us to conduct our research along the creation – deployment – end user stages of document workflow, and extend the management issues from electronic documents to printed documents.

For the document creation stage, managing authors' relationship is the foremost task. The solution is access control, which is governed by global policies to limit the permission of each author, and enable cooperation among them. We have proposed a role-based access control framework using XML technologies. The framework is specified using XML Schema and XML Schematron. These two specifications provide syntactic and semantic validation to the policy description, such that misuse or violations to the policy can be detected both statically and at run time. We advocate using XML as the exchanging format among authors, and our access control model is able to specify permissions to any subparts in a single XML file. The permissions are expressed using extended ODRL, and integrated into RBAC framework.

For the document deployment stage, maintaining the interoperability and authenticity of the electronic document is most important. We've identified the challenging problem of authenticating electronic documents while allowing document format transcoding. The specific solution we propose is digital watermark. We have designed a specialized watermark algorithm named "Render Sequence Encoding", which is most suitable for formatted electronic documents. The RSE watermark exhibits large information carrying capacity and robustness against document format transcoding. We use RSE to bridging rights description across different versions and formats of electronic documents. RSE authentication algorithm also protects the authenticity of the documents. Defeating the authenticity requires solving an NP complete Exact Traveling Salesman Problem (XTSP). The security of our watermark scheme is guaranteed by the intractability of XTSP which has been studied for several decades.

When the document reaches at the end users' site and gets printed, authenticity information from electronic world must be transferred to the paper world. We proposed "Print Signature" method to authenticate printed document. The method utilizes the inherent non-repeatable randomness existing in the printing process, which results in unique features for each printed document. We have designed a set of procedures to register and verify the unique features. The "Print Signature" method fills the authenticity gap between electronic documents and paper documents. It realizes “the management of rights in both tangible and intangible form for electronic document”.

The three solutions we proposed are linked together to form a complete end-to-end document DRM framework. The rights are managed across spatial, temporal and physical domains. This is what the second generation DRM demands for, and what our major contribution is.

Our future work involves more investigation into rights expression languages in administrative environment. This environment contains many hierarchical structures. It requires that rights expression languages can support this structure, resolve conflict between lower and higher layers, and derive rights along layers in a proper way. Confidentiality is also extremely important in administrative environment, but not extensively studied in this thesis. We will apply more formal security models in the future studies to address these problems.

RSE scheme also deserve more investigation. In Section 3.4 we have proposed a tamper detection method based on the recovery of render sequence. This method only survives the situations in which modifications are small. If the document has been severely modified, we’re not able to recover the original sequence so we cannot locate the tampered areas. A possible solution to this problem is adopting error correction codes such as the Reed-Solomon codes or Davey-MacKay codes into the generation of permutation, so that insertion / deleting / substitution to the permuted targets can be detected and corrected. Another direction of enhancement to RSE scheme involves improving the authentication algorithm for permutation. Our current method is based on the XTSP problem. The disadvantage of using XTSP is that XTSP authenticates Hamiltonian tours instead of permutations. It reduces the solutions spaces from $n!$ to $(n-1)!$. If it is possible to base the authentication algorithm onto other hard problems which takes permutations as the solution, then the security of the system is improved and we can use shorter keys. A possible candidate for such hard problem is Shamir's permuted kernels problem (PKP). The solution to PKP is a permutation. But to verify that a permutation is a valid solution, the verifier needs a n-dimensional vector. This is in contrast to the case of XTSP where only one number is needed to authenticate a Hamiltonian tour. More work is required here to find a better solution.

Using Print Signature to authenticate printed documents has been extensively studied in this thesis. Taking advantage of randomness in the printing process is the basic idea of Print Signature. It can be expected that the other side of this problem is to discover fixed features in the randomness. These features may be unique for each printer. If such features are identified, we can use them to trace the specific printer that has been used to print the document. This leads to a new direction of media forensics and traitor-tracing. It provides another level of protection which is very useful in governmental or military document management systems.

The Internet is rapidly growing to be the major information medium. It enables information delivery at light speed with low cost. The management of the content being transferred by Internet involves not only copyrighted materials, but also materials whose significance extends beyond intellectual property categories. Business and governmental documents are the most representative instances. DRM systems in this field are not for restricting the usage, but for enabling the information sharing among users, in a controlled manner.

Bibliography

[Ada99] Phillip M. Adams. Media-independent document security method and apparatus. US Patent Number 5,974,548, 1999.

[Ado01] Adobe System Incorporated. Pdf reference, third edition, version 1.4. Available at http://partners.adobe.com/asn/acrobat/docs/File_Format_Specifications/PDFReference.pdf, November 2001.

[AH01] J. S. Abigail and R. H. R. Harper. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, 2001.

[AHU74] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[And03] Anne Anderson. XACML RBAC profile (working draft). Available at http://www.oasis-open.org/committees/download.php/2405/wd-xacml-rbac-profile-01.doc, 2003.

[BGNL96] W. Bender, D. Gruhl, N.Morimoto, and A. Lu. Techniques for data hiding. *IBM System Journal*, 35(3&4):313–336, 1996.

[BJ97] Wolfgang Raymond B. and Delp Edward J. Overview of image security techniques with applications in multimedia systems. In *SPIE In-*

ternational Conference on Multimedia Networks: Security, Displays, Terminals and Gateways, volume 3228, pages 297–308, November 1997.

[BK94] S. J. Bigelow and E. Kuaimoku. *Easy Laser Printer Maintenance and Repair*. Windcrest/McGraw–Hill, Blue Ridge Summit, PA, 1994.

[BK98] S. Bhattacharjee and M. Kutter. Compression tolerant image authentication. In *Proceedings of IEEE International Conference in Image Processing*, volume 1, pages 435–439, 1998.

[BLM99] Jack T. Brassil, Steven H. Low, and Nicholas F. Maxemchuk. Copyright protection for electronic distribution of text documents. *Proceedings of the IEEE (USA)*, 87(7):1181–1196, 1999.

[BLMO94] Jack T. Brassil, Steven H. Low, Nicholas F. Maxemchuk, and Lawrence O'Gorman. Marking text features of document images to deter illicit dissemination. In *Pattern Recognition, Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 2, pages 315–319, October 1994.

[BLMO95a] Jack T. Brassil, Steven H. Low, Nicholas F. Maxemchuk, and Lawrence O'Gorman. Electronic marking and identification techniques to discourage document copying. *IEEE Journal on Selected Areas in Communications*, 13(8):1495–1504, 1995.

[BLMO95b] Jack T. Brassil, Steven H. Low, Nicholas F. Maxemchuk, and Lawrence O'Gorman. Hiding information in documents images. In *Conference on Information Sciences and Systems (CISS-95)*, 1995.

[Bor93] Borowski Jr. *et al.* Surface treated security paper and method and device for producing surface treated security paper. US Patent Number 5,193,854, 1993.

[Bra96] Jack T. Brassil. SEPTEMBER – secure electronic publishing trial (poster). In *Proceedings of the 1st ACM International Conference on Digital Libraries, March 20-23, 1996, Bethesda, Maryland, USA*, page 177. ACM, 1996.

[Bra02] Jack T. Brassil. Tracing the source of a shredded document. In *5th International Workshop on Information Hiding*, October 2002.

[Bru99] Richard A. Brualdi. *Introductory Combinatorics*. Prentice Hall, 1999.

[CB03] Monica Chew and Henry S. Baird. Baffletext: a human interactive proof. In *Proceedings of the SPIE : Document Recognition and Retrieval X*, volume 5010, pages 305–316, 2003.

[Cha00] Ramaswamy Chandramouli. Application of XML tools for enterprise-wide RBAC implementation tasks. In *Proceedings of the ACM Workshop on Role-Based Access Control 2000*, pages 11–18, 2000.

[Cho99] Nopporn Chotikakamthorn. Document image data hiding technique using character spacing width sequence coding. In *International Conference on Image Processing (ICIP-99)*, volume 2, pages 250–254, 1999.

[Chr76] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, 1976.

[CJ89] Constant and N. James. Holographic identification system using incoherent light. US Patent Number 4,820,006, 1989.

[CKLS97] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2001.

[CM97] Ingemar J. Cox and Matt L. Miller. A review of watermarking and the importance of perceptual modeling. In *Proceedings of SPIE Human Vision and Electronic Imaging II*, volume 3016, February 1997.

[CM01] Ingemar J. Cox and Matt L. Miller. Electronic watermarking: The first 50 years. In *Proceedings of the IEEE International Workshop on MultiMedia Signal Processing*, 2001.

[Con02] ContentGuard. XrML: Extensible rights markup language. Available at http://www.xrml.org, 2002.

[CT80] G. Carpaneto and P. Toth. Some new branching and bounding criteria for the asymmetric traveling salesman problem. *Management Science*, 26:736–743, 1980.

[DMH98] Swanson M. D., Kobayashi M., and Tewfik A. H. Multimedia data-embedding and watermarking technologies. *Proceedings of the IEEE*, 86(6):1064–1087, 1998.

[Dod01] Leigh Dodds. Schematron: Validating XML using XSLT. In *Proceedings of XSLT UK Conference*, 2001.

[ebX03] ebXML.org. Electronic business XML. Available at http://www.ebxml.org, 2003.

[Fel67] W. Feller. A direct proof of Stirling's formula. *American Mathematical Monthly*, 74:1223–1225, 1967.

[FK92] David Ferraiolo and Ruhn Kuhn. Role-based access controls. In *Proceedings of the 15th NIST-NCSC National Computer Security Conference*, pages 554–563, 1992.

[Gar74] M. Gardner. Mathematical games. *Scientific American*, pages 122–125, November 1974.

[GBL96] Daniel Gruhl, Walter Bender, and Anthony Lu. Echo hiding. In *Proceedings of the first international workshop on information hiding*, volume 1174, pages 295–315, 1996.

[GJ00] E. B. Greene and D. Jonathan. Security document. US Patent Number 6,089,610, 2000.

[Gol00] Dieter Gollmann. *Computer Security*. John Wiley & Sons, 2000.

[Gre87] E. B. Greene. Negotiable instrument. US Patent Number 4,634,148, 1987.

[Gre00] E. B. Greene. Coatings and ink designs for negotiable instruments. US Patent Number 6,155,604, 2000.

[GWW01] Carl A. Gunter, Stephen Weeks, and Andrew K. Wright. Models and languages for digital rights. In *Proceedings of the 34th Annual Hawaii International Conference on system Science (HICSS-34)*, volume 9, page 9076, 2001.

[HK99] Frank Hartung and Martin Kutter. Multimedia watermarking techniques. *Proceedings of the IEEE*, 87(7):1079–1107, 1999.

[HW00] S. Huang and J. K. Wu. Optical watermark. WIPO-PCT Patent Number WO 0,223,481, 2000.

[Ian00] Renato Iannella. Open digital rights management. Position paper for the W3C DRM Workshop, 2000, Available at `http://www.iprsystems.com`, 2000.

[Ian01] Renato Iannella. Open digital rights management. Position paper for the W3C DRM Workshop, 2001. Available at `http://www.w3.org/2000/12/drm-ws/pp/Overview.html`, 2001.

[Ian02] Renato Iannella. Open digital rights language (ODRL). Available at `http://odrl.net/1.1/ODRL-11.pdf`, 2002.

[ISO01] ISO. Mpeg-21 requirements for a rights data dictionary and a rights expression language. Available at `http://mpeg.telecomitalialab.org`, 2001.

[Joh63] S.M. Johnson. Generation of permutations by adjacent transpositions. *Mathematic of Computation*, 17:282–285, 1963.

[Jud88] G. G. Judge. *Introduction to the Theory and Practice of Econometrics, 2nd Edition*. John Wiley & Sons, New York, 1988.

[KH02] Mohan S. Kankanhalli and K. F. Hau. Watermarking of electronic text documents. *Electronic Commerce Research*, 2:169–187, 2002.

[KY00] Kimura and Yoshihiro. Woven security label. US Patent Number 6,068,895, 2000.

[Lam74] Butler W. Lampson. Protection. *ACM Operating Systems Review*, 8:18–24, 1974.

[Lap92] Gilbert Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59:231–247, 1992.

[LLKS85] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. *The Traveling Salesman Problem – A Guided Tour of Combinatorial Optimization*. John Wiley & Sons, 1985.

[LMBO95] Steven H. Low, Nicholas F. Maxemchuk, Jack T. Brassil, and Lawrence O'Gorman. Document marking and identification using both line and word shifting. In *INFOCOM (2)*, pages 853–860, 1995.

[LML98] Steven H. Low, Nicholas F. Maxemchuk, and Aleta M. Lapone. Document identification for copyright protection using centroid detection. *IEEE Transactions on Communication*, 46(3):372–383, 1998.

[low98] Performance comparison of two text marking methods. *IEEE Journal on Selected Areas in Communications*, 16(4):561–572, 1998.

[Luc94] Stefan Lucks. How to exploit the intractability of exact TSP for cryptography. In *Fast Software Encryption*, pages 298–304, 1994.

[Luc95] Stefan Lucks. How traveling salespersons prove their identity. In *IMA: IMA Conference on Cryptography and Coding, LNCS lately (earlier: Cryptography and Coding II, Edited by Chris Mitchell, Clarendon Press, 1992)*, 1995.

[Mar03] Monica Martin. ebXML adoption update December 2003. Available at http://www.ebxml.org/documents/ebxml_adopt_update_122203.pdf, 2003.

[Max94] Nicholas. F. Maxemchuk. Electronic document distribution. *AT&T Technical Journal*, pages 73–80, 1994.

[MH63] Jr. M. Hall. Proceedings symposium in pure mathematics. *American Mathematical Society, Providence*, 6:203, 1963.

[ML97] Nicholas F. Maxemchuk and Steven H. Low. Marking text documents. In *International Conference on Image Processing (ICIP-97)*, pages 13–16, 1997.

[MT87] S. Martello and P. Toth. Linear assignment problems. *Annals of Discret Mathematics*, 31:259–282, 1087.

[MV99] Nasir Memon and Poorvi L. Vora. Authentication techniques for multimedia content. In *Proceedings of SPIE*, volume 3528, pages 412–422, 1999.

[MvOV97] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[MYSS02] E. Métois, P. Yarin, N. Salzman, and Joshua R. Smith. Fiberfingerprint identification. In *Third Workshop on Automatic Identification*, March 2002.

[Nie99] Jakob Nielsen. Fingerprinting plain text information. US Patent Number 5,953,415, 1999.

[NIS99] NIST. Common criteria for it security evaluation v2.1. Available at http://csrc.nist.gov/cc/index.html, 1999.

[NS94] M. Naor and A. Shamir. Visual cryptography. In *Proceedings of EUROCRYPT 94*. Springer, 1994. Lecture Notes in Computer Science No. 950.

[NWK93] A. D. Narasimhalu, W. Wang, and M. S. Kankanhalli. Method for utilizing medium nonuniformities to minimize unauthorized duplication of digital information. US Patent Number 5,412,718, 1993.

[oAP00] Association of American Publishers. Digital rights management for ebooks: Publisher requirements, November 2000.

[Ots79] N. Otsu. A threshold selection method from gray-level histogram. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-9(1):62–66, 1979.

[PAK98] Fabien A.P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Attacks on copyright marking systems. In *Second Workshop on Information Hiding*, pages 218–238, 1998. Lecture Notes in Computer Science, Vol. 1525.

[PAK99] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Information hiding — a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999.

[PC94] Jacques Patarin and Pascal Chauvaud. Improved algorithms for the permuted kernel problem. In Douglas R. Stinson, editor, *Advances in Cryptology – proceedings of CRYPTO '93*, pages 391–402. Springer, 1994. Lecture Notes in Computer Science No. 773.

[PD01] Christine I. Podilchuk and Edward J. Delp. Digital watermarking: Algorithms and applications. *IEEE Signal Processing Magazine*, 18(4):33–44, 2001.

[PDF01] PDFZone.com. Digital rights management: A primer. Available at http://www.pdfzone.com/news/100864.html, 2001.

[Poi95] David Pointcheval. A new identification scheme based on the perceptrons problem. In *Proceedings of Eurocrypt '95*, pages 319–328, 1995. Lecture Notes in Computer Science No. 921.

[PSW90] T. Pavlidis, J. Swartz, and Y. P. Wang. Fundamentals of bar code information theory. *Computer*, 23(4):74–85, April 1990.

[PSW92] T. Pavlidis, J. Swartz, and Y. P. Wang. Information encoding with two–dimensional bar codes. *Computer*, 24(6):18–28, June 1992.

[Pun02] Abraham P. Punnen. *The Traveling Salesman Problem and Its Variations. Volume 12 Combinatorial Optimization*, volume 12 of *Combinatorial Optimization*. Kluwer academic publishers, 2002.

[QG03] N. Degara Quintela and F. Pérez González. Visible encryption: Using paper as a secure channel. In Ping Wah Wong and Edward J. Delp, editors, *Security and Watermarking of Multimedia Contents V, Proceedings of SPIE*, volume 5020, 2003.

[Ram01] José Carlos Ramalho. Constraining content: specification and processing. In *Proceedings of XML Europe'01*, 2001.

[Rei94] Gerhard Reinelt. *The Traveling Salesman: Computational Solutions for TSP Applications*, volume 840 of *Lecture Notes in Computer Science.* Springer-Verlag Heidelberg, 1994.

[RG98] Rhoads and B. Geoffrey. Identification/authentication system using robust, distributed coding. US Patent Number 5,745,604, 1998.

[RTM01] Bill Rosenblatt, Bill Trippe, and Stephen Mooney. *Digital Rights Management: Business and Technology.* Henry Minds/John Wiley & Sons, New York, 2001.

[SFK00] Ravi Sandhu, David Ferraiolo, and Richard Kuhn. The NIST model for role-based access control: Towards a unified standard. In *Proceedings of 5th ACM Workshop on Role-Based Access Control*, pages 47–64, 2000.

[Sha90] A. Shamir. An efficient identification scheme based on permuted kernels. In *Advances in Cryptology – proceedings of CRYPTO '89*, pages 606–609, 1990. Lecture Notes in Computer Science No. 435.

[Sha96] A. Shamir. Method and apparatus for protecting visual information with printed cryptographic watermarks. US Patent number 5,488,664, 1996.

[SHG98] J. Su, F. Hartung, and B. Girod. Digital watermarking of text, image, and video documents. *Computer & Graphics*, 22(6):687–695, 1998.

[Sim98] Gustavus J. Simmons. A survey of information authentication. *Proceedings of IEEE*, 76(5):603–620, May 1998.

[Ste94] Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *Advances in Cryptology – proceedings of CRYPTO '93*, pages 13–21. Springer, 1994. Lecture Notes in Computer Science No. 773.

[Ste95] Jacques Stern. Desiging identification schemes with keys of short size. In *Advances in Cryptology – proceedings of CRYPTO '94*, pages 164–173. Springer, 1995. Lecture Notes in Computer Science No. 839.

[Ste96] Mark Stefik. *Internet Dreams: Archetypes, Myths, and Metaphors*, chapter Letting Loose the Light: Igniting Commerce in Electronic Publication, pages 219–253. MIT Press, Cambridge, Massachusetts, 1996.

[Tro62] H.F. Trotter. Algorithm 115. *Communications of the ACM*, 5:434–435, 1962.

[TRvS+93] A. Z. Tirkel, G. A. Rankin, R.M. van Schyndel, W. J. Ho, N. R. A. Mee, and C. F. Osborne. Electronic watermark. In *Digital Image*

Computing, Technology and Applications – DICTA '93, pages 666–673, 1993.

[VH99] R. C. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. Technical Report UU-CS-1999-27, Utrecht University, the Netherlands, 1999.

[vSTO94] R. M. van Schyndel, A. Z. Tirkel, and C. F. Osborne. A digital watermark. In *International Conference on Image Processing*, volume 2, pages 86–90, 1994.

[W3C99] W3C. XML Path language. Available at http://www.w3.org/TR/xpath, 1999.

[W3C01a] W3C. XML Schema part 0: Primer, W3C recommendation. Available at http://www.w3.org/TR/xmlschema-0, 2001.

[W3C01b] W3C. XML Schema part 1: Structures, W3C recommendation. Available at http://www.w3.org/TR/xmlschema-1, 2001.

[W3C01c] W3C. XML Schema part 2: Datatypes, W3C recommendation. Available at http://www.w3.org/TR/xmlschema-2, 2001.

[Wag83] N. R. Wagner. Fingerprinting. In *Proceedings of the 1983 IEEE Symposium on Security and Privacy*, pages 18–22, 1983.

[WD96] Raymond B. Wolfgang and Edward J. Delp. A watermark for digital images. In *International Conference on Image Processing*, pages 219–222, September 1996.

[WD97] Raymond B. Wolfgang and Edward J. Delp. A watermarking technique for digital imagery: further studies. In *International Conference on Imaging, Systems, and Technology*, pages 279–287, 1997.

[YM97] M. Yeung and F. Mintzer. An invisible watermarking technique for image verification. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 680–683, October 1997.

[Zei00] Zeira *et al.* Verification methods employing thermally-imageable substrates. US Patent Number 6,107,244, 2000.

[ZPS92] Zheng, Pieprzyk, and Seberry. HAVAL – A one-way hashing algorithm with variable length of output. In *AUSCRYPT: Advances in Cryptology–AUSCRYPT '90, International Conference on Cryptology*. LNCS, Spring-Verlag, 1992.

Appendix

Listing A.1: B/L XML source data (bl.xml)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<bill_of_lading>
 <bl_id>bl001</bl_id>
 <shipper_id>shipper001</shipper_id>
 <clerk_id>clerk001</clerk_id>
 <manager_id>manager001</manager_id>
 <internal_ref>000001</internal_ref>
 <vessel_name>star virgo</vessel_name>
 <loading_port>shanghai</loading_port>
 <discharging_port>singapore</discharging_port>
 <consignment>
  <product>
   <name>firecrack</name>
   <weight>1000</weight>
  </product>
  <product>
   <name>chewing gum</name>
   <weight>200</weight>
  </product>
 </consignment>
 <status>both_not_confirmed</status>
 <negotiable>
  <printed>0</printed>
  <left>0</left>
 </negotiable>
 <nonnegotiable>
  <printed>0</printed>
  <left>0</left>
 </nonnegotiable>
</bill_of_lading>
```

Listing A.2: RBAC XML Schema (RBAC.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://example.net/RBAC"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
 elementFormDefault="qualified" attributeFormDefault="qualified">
 <xsd:import namespace="http://odrl.net/1.1/ODRL-EX"
  schemaLocation="ODRL-EX-11.xsd"/>
 <xsd:element name="rbac">
  <!-- RBAC root element -->
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="users">
     <!-- users element: defines all users. -->
     <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
       <xsd:element name="user" minOccurs="0"
        maxOccurs="unbounded">
        <!-- user element: defines each user. -->
        <xsd:complexType>
         <xsd:attribute name="id" type="xsd:ID" use="required"/>
         <xsd:attribute name="name" type="xsd:string"/>
        </xsd:complexType>
       </xsd:element>
      </xsd:sequence>
     </xsd:complexType>
    </xsd:element>
    <xsd:element name="permissions">
     <!-- permissions element: defines all users. -->
     <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
       <xsd:element name="permission" minOccurs="0"
        maxOccurs="unbounded">
        <!-- permission element: defines each permission. -->
        <xsd:complexType>
         <xsd:sequence>
          <xsd:element ref="o-ex:agreement"/>
           <!-- Each permission contains an ODRL agreement. -->
         </xsd:sequence>
         <xsd:attribute name="id" type="xsd:ID" use="required"/>
         <xsd:attribute name="name" type="xsd:string"/>
        </xsd:complexType>
       </xsd:element>
      </xsd:sequence>
     </xsd:complexType>
    </xsd:element>
    <xsd:element name="roles">
     <!-- roles element: defines all roles. -->
     <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
       <xsd:element name="role" minOccurs="0"
        maxOccurs="unbounded">
        <!-- role element: defines each role. -->
        <xsd:complexType>
         <xsd:sequence>
          <xsd:element name="permission" maxOccurs="unbounded">
           <!-- ref. to permissions assigned to the role. -->
           <xsd:complexType>
```

```
            <xsd:attribute name="ref" type="xsd:IDREF"
             use="required"/>
           </xsd:complexType>
          </xsd:element>
         </xsd:sequence>
         <xsd:attribute name="id" use="required">
          <!-- valid choices of role id attribute-->
          <xsd:simpleType>
           <xsd:restriction base="xsd:ID">
            <xsd:enumeration value="application"/>
            <xsd:enumeration value="clerk"/>
            <xsd:enumeration value="manager"/>
            <xsd:enumeration value="shipper"/>
           </xsd:restriction>
          </xsd:simpleType>
         </xsd:attribute>
        </xsd:complexType>
       </xsd:element>
      </xsd:sequence>
     </xsd:complexType>
    </xsd:element>
    <xsd:element name="uras">
     <!-- uras element: defines all user-role assignments. -->
     <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
       <xsd:element name="ura" minOccurs="0"
        maxOccurs="unbounded">
        <!-- ura element: defines each user-role assignment. -->
        <xsd:complexType>
         <xsd:sequence>
          <xsd:element name="role">
           <!-- ref. to the role element. -->
           <xsd:complexType>
            <xsd:attribute name="ref" type="xsd:IDREF"
             use="required"/>
           </xsd:complexType>
          </xsd:element>
          <xsd:element name="user" maxOccurs="unbounded">
           <!-- ref. to user elements assigned to the role -->
           <xsd:complexType>
            <xsd:attribute name="ref" type="xsd:IDREF"
             use="required"/>
           </xsd:complexType>
          </xsd:element>
         </xsd:sequence>
         <xsd:attribute name="id" type="xsd:ID" use="required"/>
        </xsd:complexType>
       </xsd:element>
      </xsd:sequence>
     </xsd:complexType>
    </xsd:element>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

Listing A.3: Extended ODRL XML Schema (ODRLX-DD.xsd)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://example.net/ODRLX-DD"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
 elementFormDefault="qualified" attributeFormDefault="qualified">
 <xsd:import namespace="http://odrl.net/1.1/ODRL-EX"
  schemaLocation="ODRL-EX-11.xsd"/>
  <xsd:element name="selector"
   substitutionGroup="o-ex:contextElement">
   <!-- Extension of ODRL context model. -->
   <xsd:complexType>
    <xsd:attribute name="xpath" use="required">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:minLength value="1"/>
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:attribute>
   </xsd:complexType>
  </xsd:element>
  <xsd:element name="assert"
   substitutionGroup="o-ex:constraintElement">
   <!-- Extension of ODRL constraint model. -->
   <xsd:complexType>
    <xsd:attribute name="test" use="required">
     <xsd:simpleType>
      <xsd:restriction base="xsd:string">
       <xsd:minLength value="1"/>
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:attribute>
   </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Listing A.4: RBAC policy definition (rbac.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<rbac:rbac xmlns:rbac="http://example.net/RBAC"
 xmlns:ox-dd="http://example.net/ODRLX-DD"
 xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
 xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://example.net/RBAC RBAC.xsd
 http://example.net/ODRLX-DD ODRLX-DD.xsd
 http://odrl.net/1.1/ODRL-EX ODRL-EX-11.xsd
 http://odrl.net/1.1/ODRL-DD ODRL-DD-11.xsd">
 <!-- Definition of users -->
 <rbac:users>
  <rbac:user rbac:id="u001" rbac:name="Alice"/>
  <rbac:user rbac:id="u002" rbac:name="Cindy"/>
  <rbac:user rbac:id="u003" rbac:name="Chris"/>
  <rbac:user rbac:id="u004" rbac:name="Michael"/>
  <rbac:user rbac:id="u005" rbac:name="Melvin"/>
  <rbac:user rbac:id="u006" rbac:name="Steven"/>
  <rbac:user rbac:id="u007" rbac:name="Sarah"/>
 </rbac:users>
 <!-- Definition of permissions -->
 <rbac:permissions>
  <rbac:permission rbac:id="p001"
   rbac:name="shipper modify shipper_id">
   <!-- ODRL rights: modify shipper_id if status is
    both_not_confirmed or wait_shipper_confirm -->
   <o-ex:agreement>
    <o-ex:asset>
     <o-ex:context>
      <ox-dd:selector ox-dd:xpath="/bill_of_lading/shipper_id"/>
     </o-ex:context>
    </o-ex:asset>
    <o-ex:permission>
    <o-dd:display/>
    <o-dd:modify>
     <o-ex:constraint>
      <ox-dd:assert ox-dd:test=
       "(/bill_of_lading/status=’both_not_confirmed’)||
       (/bill_of_lading/status=’wait_shipper_confirm’)"/>
      </o-ex:constraint>
     </o-dd:modify>
    </o-ex:permission>
   </o-ex:agreement>
  </rbac:permission>
  <rbac:permission rbac:id="p002"
   rbac:name="clerk modify vessel_name">
   <!-- ODRL rights: modify vessel_name if status is
    both_not_confirmed or wait_clerk_confirm -->
   <o-ex:agreement>
    <o-ex:asset>
     <o-ex:context>
      <ox-dd:selector ox-dd:xpath="/bill_of_lading/vessel_name"/>
     </o-ex:context>
    </o-ex:asset>
    <o-ex:permission>
     <o-dd:display/>
     <o-dd:modify>
      <o-ex:constraint>
       <ox-dd:assert ox-dd:test=
        "(/bill_of_lading/status='both_not_confirmed')||
        (/bill_of_lading/status='wait_clerk_confirm')"/>
      </o-ex:constraint>
     </o-dd:modify>
    </o-ex:permission>
   </o-ex:agreement>
  </rbac:permission>
  <rbac:permission rbac:id="p003"
   rbac:name="manager modify manager_id">
   <!-- ODRL rights: modify manager_id if status is
    both_confirmed -->
   <o-ex:agreement>
    <o-ex:asset>
     <o-ex:context>
      <ox-dd:selector ox-dd:xpath="/bill_of_lading/manager_id"/>
     </o-ex:context>
    </o-ex:asset>
    <o-ex:permission>
     <o-dd:display/>
     <o-dd:modify>
      <o-ex:constraint>
       <ox-dd:assert ox-dd:test=
        "/bill_of_lading/status='both_confirmed'"/>
      </o-ex:constraint>
     </o-dd:modify>
    </o-ex:permission>
   </o-ex:agreement>
  </rbac:permission>
  <rbac:permission rbac:id="p004"
   rbac:name="application issue license">
   <o-ex:agreement>
    <o-ex:asset>
     <o-ex:context>
      <ox-dd:selector ox-dd:xpath="/bill_of_lading/negotiable/*"/>
     </o-ex:context>
    </o-ex:asset>
    <o-ex:asset>
     <o-ex:context>
      <ox-dd:selector ox-dd:xpath=
       "/bill_of_lading/nonnegotiable/*"/>
     </o-ex:context>
    </o-ex:asset>
    <o-ex:permission>
     <o-dd:display/>
     <o-dd:modify>
      <o-ex:constraint>
       <ox-dd:assert ox-dd:test=
        "/bill_of_lading/status='approved'"/>
      </o-ex:constraint>
     </o-dd:modify>
    </o-ex:permission>
   </o-ex:agreement>
  </rbac:permission>
 </rbac:permissions>
 <!-- Definition of roles -->
 <rbac:roles>
```

```
  <rbac:role rbac:id="application">
   <rbac:permission rbac:ref="p004"/>
  </rbac:role>
  <rbac:role rbac:id="clerk">
   <rbac:permission rbac:ref="p002"/>
  </rbac:role>
  <rbac:role rbac:id="manager">
   <!-- implementation of role hierarchy-->
   <rbac:permission rbac:ref="p003"/>
   <rbac:permission rbac:ref="p002"/>
  </rbac:role>
  <rbac:role rbac:id="shipper">
   <rbac:permission rbac:ref="p001"/>
  </rbac:role>
 </rbac:roles>
 <!-- Definition of user-role assignment -->
 <rbac:uras>
  <rbac:ura rbac:id="ura001">
   <rbac:role rbac:ref="application"/>
   <rbac:user rbac:ref="u001"/>
  </rbac:ura>
  <rbac:ura rbac:id="ura002">
   <rbac:role rbac:ref="clerk"/>
   <rbac:user rbac:ref="u002"/>
   <rbac:user rbac:ref="u003"/>
  </rbac:ura>
  <rbac:ura rbac:id="ura003">
   <rbac:role rbac:ref="manager"/>
   <rbac:user rbac:ref="u004"/>
   <rbac:user rbac:ref="u005"/>
  </rbac:ura>
  <rbac:ura rbac:id="ura004">
   <rbac:role rbac:ref="shipper"/>
   <rbac:user rbac:ref="u006"/>
   <rbac:user rbac:ref="u007"/>
  </rbac:ura>
 </rbac:uras>
</rbac:rbac>
```

Listing A.5: RBAC Schematron validator (rbac.sch)

```
<?xml version="1.0" encoding="US-ASCII"?>
<schema xmlns="http://www.ascc.net/xml/schematron">
 <ns uri="http://example.net/RBAC" prefix="r"/>
 <title>Validation of B/L RBAC Policy</title>
 <!-- Data type validation -->
 <pattern name="Attribute 'role/permission/@ref' should reference
  to a 'permission' element.">
  <rule context="/r:rbac/r:roles/r:role/r:permission">
   <assert test="/r:rbac/r:permissions/r:permission/@r:id=@r:ref">
    Error: Permission id <value-of select="@r:ref"/> not found.
   </assert>
  </rule>
 </pattern>
 <!-- Cardinality constraint -->
 <pattern name="Role 'application' should not contain more than
  one users.">
  <rule context=
   "/r:rbac/r:uras/r:ura/r:role[@r:ref='application']">
   <assert test="count(parent::*/r:user)=1">
    Error: Role 'application' contains
    <value-of select="count(parent::*/r:user)"/> users.
   </assert>
  </rule>
 </pattern>
 <!-- Separation of duties constraint -->
 <pattern name="User should not be assigned to both 'application'
  and 'shipper' roles.">
  <rule context="/r:rbac/r:uras">
   <assert test=
    "not ((r:ura/r:role[@r:ref='application']/../r:user/@r:ref)=
    (r:ura/r:role[@r:ref='shipper']/../r:user/@r:ref))">
    Error: Conflicting users in 'application' and 'shipper' roles.
   </assert>
  </rule>
 </pattern>
</schema>
```