2015

Design and Implementation of a Digital Information Security Service for Physical Documents

Xue Yu
University of Massachusetts Amherst

DESIGN AND IMPLEMENTATION OF A DIGITAL INFORMATION SECURITY SERVICE FOR PHYSICAL DOCUMENTS

A Thesis Presented

by

XUE YU

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

September 2015

Electrical and Computer Engineering

DESIGN AND IMPLEMENTATION OF A DIGITAL INFORMATION SECURITY SERVICE FOR PHYSICAL DOCUMENTS

A Thesis Presented

by

XUE YU

Approved as to style and content by:

Tilman Wolf , Chair

Aura Ganz, Member

David Irwin, Member

C.V.Hollot, Department Head
Electrical and Computer Engineering

ACKNOWLEDGEMENTS

I would first like to express my sincere gratitude to Professor Tilman Wolf for his guidance, patience and support as my advisor. He is a really good professor and nice guy. I am very grateful to being his student.

I would also like to thank my committee members Professor David Irwin and Professor Aura Ganz, for their invaluable time, helpful comments and suggestions about my thesis.

I need to acknowledge my labmates as well as my friends, Shuai Chen, Pengcheng Wang, Padmaja Duggisetty, Xinming Chen, Hao Cai, Thiago Teixeira and Arman Pouraghily for their support, help and kindliness. I would also like to thank Metin Ayata for his encouragement, advice and help, as well as Rui Zhang, Shuna Hu, Feng tian, Yuran Zhuo, Han Bao, Wu Wang, Yang Cui, Xiao Chen, Kan Fu, Yang Lei and Ruidong Xie, while having good time together in Amherst.

Most importantly, I would like to thank my parents and my relatives. Their unwavering love, care, and support provided me with the energy I need to accomplish this work.

ABSTRACT

DESIGN AND IMPLEMENTATION OF A DIGITAL INFORMATION SECURITY SERVICE FOR PHYSICAL DOCUMENTS

SEPTEMBER 2015

XUE YU

B.Sc., NORTHEAST PETROLEUM UNIVERSITY

M.S.E.C.E., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Tilman Wolf

This thesis presents a hybrid information security approach for data on physical documents. This system, called CryptoPaper, allows portions of a physical document be printed with a machine-readable code. This code contains original data and related information security properties. To read this code and access the encoded data, a scanning device with suitable image recognition technology is used. Using a cloud-based access control system, it can be ensured that only authorized users can interpret the machine-readable code and additional security properties can be verified. This thesis specifically explained key management mechanism, which is designed to achieve integrity, confidentiality and authenticity.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

Information security become a crucial issue especially in business or government domain [1]. And papper document are widely used in these area even personal use. Usually these documents contains sensitive information which associated with information security. For instance, personal information(SSN, date of birth, finacal issues, etc.), business secret(biding price, confidentiality agreement, etc). Current ways to protect these sensitive information on physical document is either to shred after use or to save in a safe place. But according to social media, organizations are not doing well regarding this aspect. In 2007 Dec, Company paid $50,000 penalty for tossing consumers credit report tossing consumers credit report information in unsecured dumpster [2]. In 2011 May, DYPD counterterrorism documents found in trash [3]. In 2013 Jan, Medical group fined $140K for tossing patients' health records into public dump [4]. In these cases, companies exposed privacy in both personal and governmental way which may cause potential financial problems or political issues.

As a result a realiable method need to be developed to protect sensitive information especially which need to be printed out in physical document.

1.2 Motivation

The traditional way of implementing information security with physical documents cost too much and has limitation in revoking access. For example, proper disposal of sensitive documents, tamper-evident paper or physical access control, ect, is expensive to implement for businesses. Once documents have been issued, it is difficult to revoke acess, which presents a potential for information leaks and insider attacks.

An alternative to physical documents is an all-digital document management system. In such an environment, document aceess can be fully controlled, tracked, and audited. However, such systems are complex to implement unless all entities accessing these documents commit to the same process. In scenarios, where documents need to cross organizational boundaries-while maintaining the necessary security properties-an all-digital solution becomes even more challenging. In addition, hybrid security approaches, where physical security properties(eg., locked file cabinet) can be combined with digital security properties (eg., access revocation), have not yet been deployed.

This thesis presents a hybrid information security approach for data on physical documents [5]. A system called CryptoPaper, allows portions of a physical documents be printed with a machine-readable code [6]. Only autherized people or organizations can have the access right to read the code. This code is encrypted with AES encryption algorithm and then turned into a two dimensional QR code. Once been created, comes along with a 128 bits symmetrical key which is saved in cloud database as well as the information of the code creater. At the same time cloud

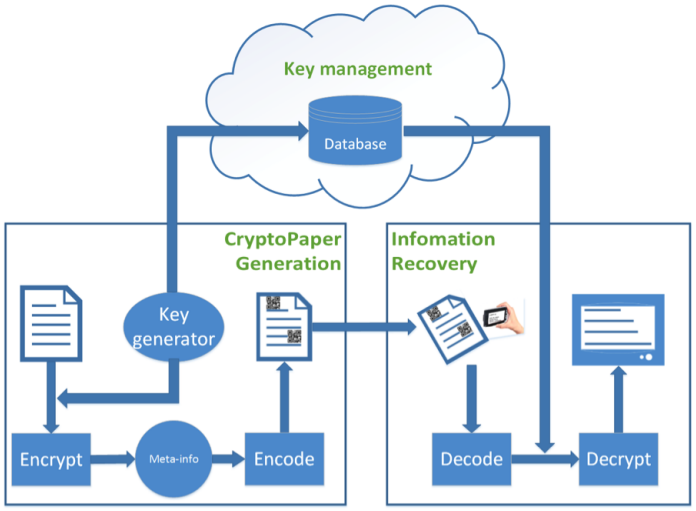

Figure 1.1. CryptoPaper System Overview.

database will record this code ID and send back the ID number to application which is kept in the meta-information of this QR code. During information recovering, after verifying user's identity CyptoPaper system will look up for the key to decrypt this code which based on the user's access right and the meta-information from the physical document.

This innovative method can keep the information safely even the physical paper is stolen or the cloud database is hacked. For the cipher text is saved in the physical document and the key is stored at seperated database. Beyond that, this thesis presents a role-based access control key management. It makes key management flexible and secure. For the access right is not assigned to a specific user but a role. If there is any position modification, the employee who is no longer wroking at the assigned postion has no longer access right.

1.3 Problem Statement

In order to make the whole system work flexibly and securely there are several problems need to be designed and implemented to meet the requirements of CryptoPaper system. This section states those promblems and the solutions will be propsed in next few chapters.

- A stable and flexibale database schema need to be designed to ensure reading and fetching data in fast speed and cost less resource.
- Confidentiality. The most significant concern is to protect the key from disclosure to unauthorized parties.
- Integrity. Protecting information from being modified by unauthorized parties. And make sure there is no errors during the process of transmission.
- Authenticity. Ensuring that authorized parties are able to access the information in fast speed when needed. And make sure the code is genuienly created by the original user.
- How to enable "CryptoPaper code generation service" provide code to multiple applications at same time.
- Eanble access rovoke. Ensuring access right can be be changed by authorized user. On the other hand, the user who is not authorized cannot make any change of the created code.

1.4 Contribution

This thesis mainly focus on how to provide a secure code generation service in CryptoPaper system which serve as a backbone of the whole project.

- It provides a role-access-control database schema which assign access right of the key to specific role instead of specific user, which enables the implementation of access revocation in case users change roles or organizations. By simply removing a role from a user's profile, their access to this role's codes can be removed.

- Achieving confidentiality by encrypting the sensitive data in the document using AES. For each QR code, CryptoPaper system use a randomly generated 128-bit key and encrypt data in 128-bit block using the Electronic Codebook (ECB) mode. The key is stored in the cloud database that manages access control. And also during transmission this 128-bit random key is protected by SSL protocol.

- To achieve integrity and authenticity, a digital signature mechanism is deployed. Based on the ciphertext which is encrypted by AES algorithm a digital signature is created and attached at the end of ciphertext and then encode together with ciphertext into QR code. When reading the code, the signature need to be verifiedto ensure data has not been modified and it is created by original creator.

- In order to implement CryptoPaper code generation service for multiple applications at same time. This service is separated from Microsoft Word application. It runs at its own so that user can encrypt data in different applications.

- User can revoke access right even if the code is created. By achieving this, cloud database record the creater's identity information and verify before making any changes of the access right. So unauthorized user would not make modification of the code.

1.5 Organization

The remainder of the thesis is organized as follows: Chapter 2 presents the background and related work on CyptoPater system. This chapter is the overview of CyptoPaper system including the whole architecture, specification of code generation part and inforamtion recovery part. Chapter 3 focus on the design of "CryptoPaper code generation service" in detail. This chapter discusses the solutions of previous concerns such as database design, how to achieve CIA properties, separating service from Micsoft Word and access revoke. Chapter 4 provides current results of how the system works. Chapter 5 presents the future work.

CHAPTER 2

RELATED WORK

2.1 Overview and System Operation

This thesis presents a hybrid information security approach for data on physical documents. This system, called CryptoPaper, allows portions of a physical document be printed with a machine-readable code [7]. This code contains original data and related information security properties. To read this code and access the encoded data, a scanning device with suitable image recognition technology is used. Using a cloud-based access control system, it can be ensured that only authorized users can interpret the machine-readable code and additional security properties can be verified [8].

The operation of CryptoPaper is illustrated in Figure 2.1. There are three major steps in the creation and use of CryptoPater documents:

1. Creation of document: When creating a CryptoPaper document, special software (in our case, a Microsoft Word plugin) is used to replace sensitive information with a 2-dimensional code (in our case, a QR code) [9]. The code contains an encrypted version of the data that needs to be protected as well as meta-information that is necessary for decryption [10]. The code is printed on

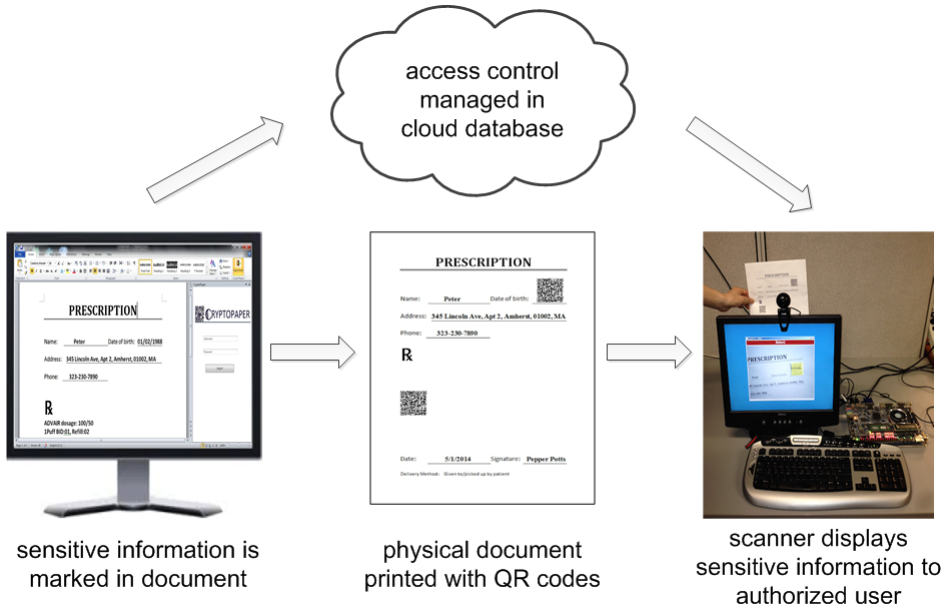

Figure 2.1. System architecure overview.

the physical document instead of the sensitive information and the digital key to access the information is stored in a cloud database.

2. Management of access control: Access control to a CryptoPaper document can now be realized through both physical constraints (e.g., by locking the document in a file cabinet) as well as virtual constraints (e.g., by limiting access to the key to decode information). The cloud database enables management of the latter type of access control. In particular, it is possible to dynamically update the list of users who have access to a document, which also enables the revocation of access (e.g., when an employee leaves an organization).

3. Reading of sensitive information with scanner: To access the encoded information on a CryptoPaper, a user has to have physical access to the document and has access rights to access the key from the cloud database in order to decrypt any encoded information. To implement this process in a convenient manner, we have developed a scanner system that automatically detects coded regions on a paper, fetches keys for which a user has access, and substitutes the codes with cleartext information on the display.

This process of protecting sensitive information in CryptoPaper exhibits the following convenient properties:

- Only authorized users can access encoded information since both the physical document and access permissions to the digital key are necessary.
- The sensitive information is only stored in the code on the physical document (and not in the cloud database, which only holds the key to decrypt the code). Thus, a user does not need to entrust sensitive information to the database provider (and hacking attempts on the database will not leak information).
- The scanner can use established techniques to identify users (e.g., password login, fingerprint reader). This information is then used by the cloud database to ensure that decryption keys for codes are only provided to those users who have access rights to that code.
- The full range of information security properties (confidentiality, integrity, authenticity) can be implemented in the codes (assuming an appropriate public key management system).

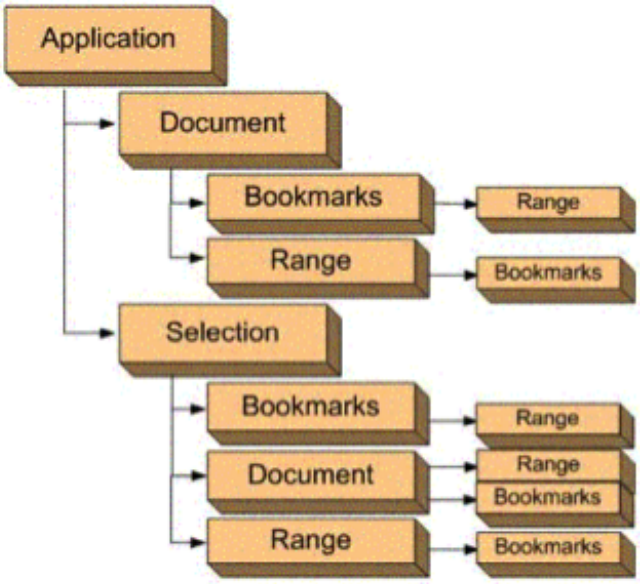

Figure 2.2. Word Object Model.

2.2 Generation of CryptoPaper

To generate the QR codes using CryptoPaper in a convenient fashion, we developed a Microsoft Word plug-in that manages the process described above. The plug-in enables users to mark sensitive text that needs to be replaced by a QR code. User needs to create an account in CryptoPaper system and log in so that to save user's code id and access right in CryptoPaper database.

2.2.1 Building pulg-in with Microsoft Visual Studio

To build this plug in, Microsoft Visual Studio provides deverlopers a rich and powerful programming environment to interact with object model by using .NET framework. Figure 2.2 shows the several object models.

- The Application Object represents the Word application, and is the parent of all of the other objects. Its members usually apply to Word as a whole. You can use its properties and methods to control the Word environment.

- The Document Object is central to programming Word. When you open an existing document or create a new document, you create a new Document object, which is added to the Word Documents collection. The document that has the focus is called the active document and is represented by the Application object's ActiveDocument property.

- The Selection Object represents the area that is currently selected. When you perform an operation in the Word user interface, such as bolding text, you select the text and then apply the formatting. The Selection object is always present in a document; if nothing is selected, the Selection object represents the insertion point. The Selection object can also be multiple noncontiguous blocks of text.

- The Range Object represents a contiguous area in a document, and is defined by a starting character position and an ending character position. You are not limited to a single Range object; you can define multiple Range objects in the same document.

- The Bookmark object is similar to the Range object in that it represents a contiguous area in a document, with both a starting position and an ending position. You use bookmarks to mark a location in a document, or as a container for text in a document. A Bookmark object can consist of the insertion

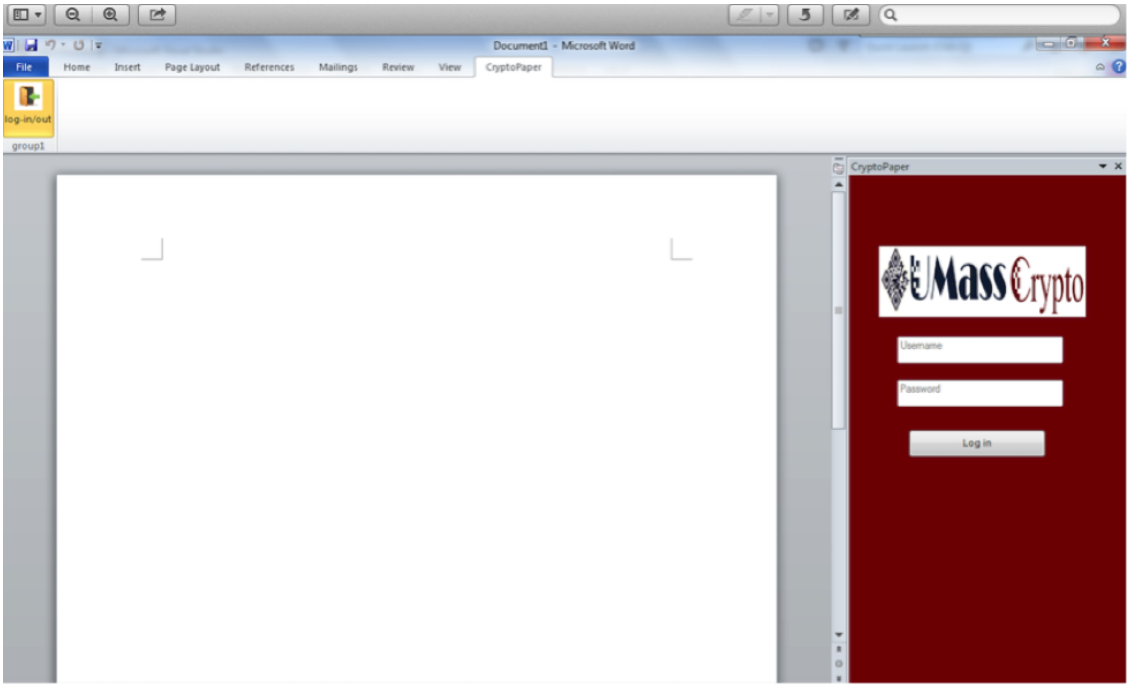

Figure 2.3. CrptoPaper Login System Plug in Editing Software Word.

point alone or be as large as the entire document. You can also define multiple bookmarks in a document.

Our CryptoPaper system uses application-level module to control the whole Word environment. We use this class to perform tasks such as accessing the object model of the Microsoft Office host application, customizing the user interface (UI) of the application, and connect information in our add-in to Amazon RDS cloud system. And also, the selection-level module realizes the function of encrypting selected plain-text and changing it into QR-code. Figure 2.3 shows the user log in interface when CryptoPater user need to generate the code.

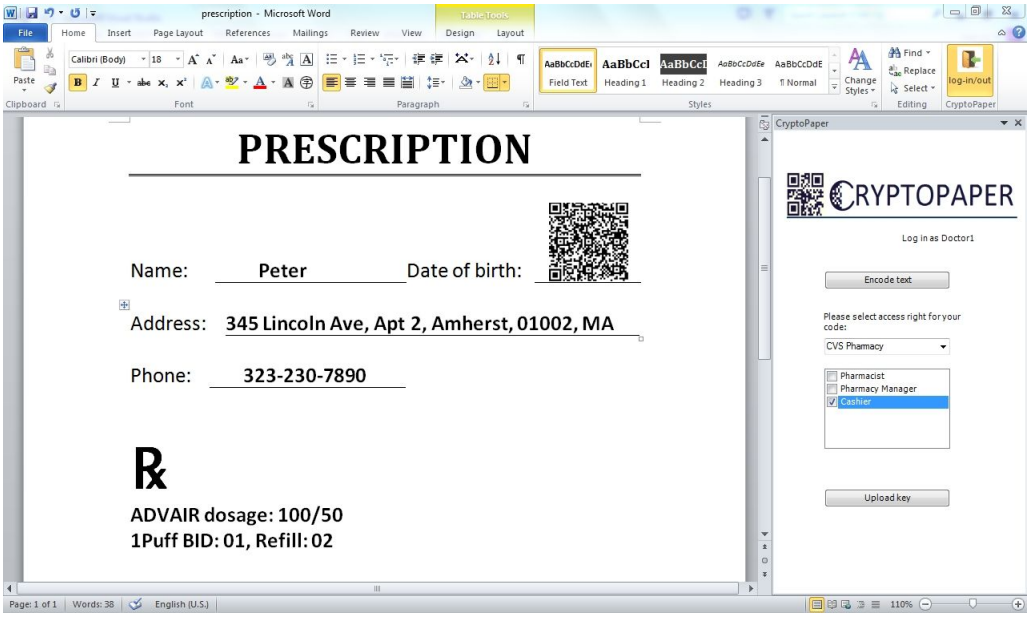

Figure 2.4. Word plug-in for generating CryptoPaper codes in document.

When generating the substitution code, the user can select the security properties that should be enabled in the code. The user can also select who can access the code (i.e., the roles within the organization—see below). The encryption key for the generated code is then automatically stored in the access control database. Figure 2.4 illustrates a user want to encrypt the Date of Birth information when editing a prescription.

2.2.2 AES Encryption

Before turning sesitive information into a two dementional code, our system choose AES encryption to encrypt the sensitive information first. AES is a cryptographic algorithm that often used to protect electronic data. Specifically, AES is

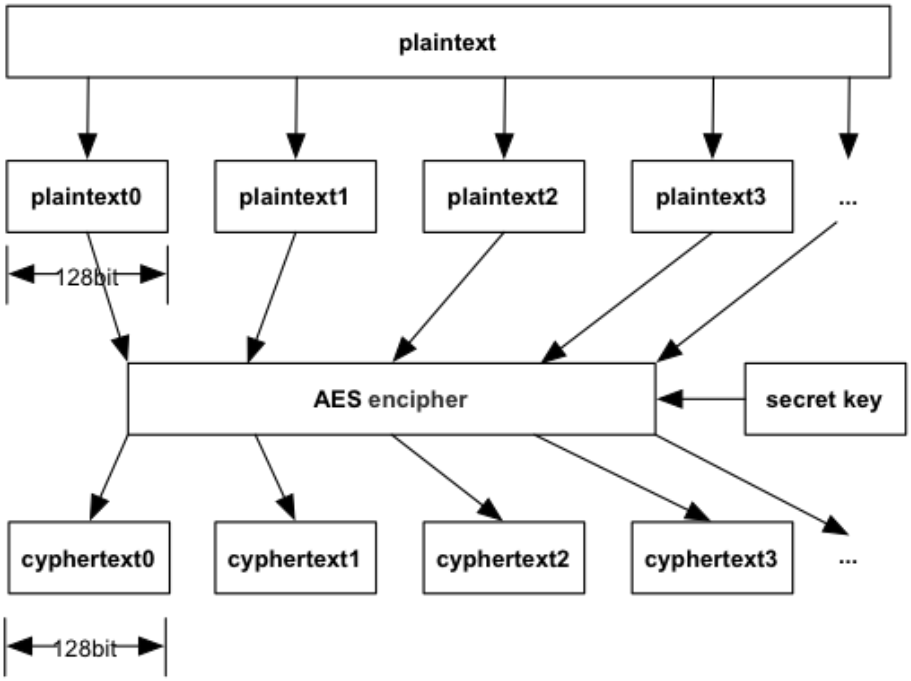

Figure 2.5. AES Encryption in ECB Mode.

an iterative, symmetric-key block cipher that can use keys of 128, 192, and 256 bits, and encrypts and decrypts data in blocks of 128 bits (16 bytes). Unlike public-key ciphers, which use a pair of keys, symmetric-key ciphers use the same key to encrypt and decrypt data. Encrypted data returned by block ciphers have the same number of bits that the input data had [11].

We use the Electronic Codebook (ECB) mode encryption to encrypt each block individually. Any blocks of plaintext that are identical and in the same message, or that are in a different message encrypted with the same key, will be transformed into identical cipher text blocks. Even though this mode opens the door for multiple security exploits, we still choose this since this is a faster encryption mode over the

other modes. In order to further secure it we use additional layers of security like encoding the cipher text into QR codes and applying a digital signature which makes it difficult for the attacker to obtain the data or modify it. The mode will enhance user experience by decreasing overall latency time in querying the database and the reaction speed of the scanner [12].

2.2.3 QR Code

The 2-dimensional code used to store information in CryptoPaper documents has several important design requirements. The code needs to have sufficient density to store sensitive information (and meta-information to decode it), needs to be machine-readable, and needs to be generated in an easy manner.

We use Quick Response (QR) codes for our system. QR codes are 2-dimensional, square bar codes that were invented by the Japanese corporation Denso Wave in 1994. QR codes can represent digital information, have options for error correction, and can be read effectively through image processing techniques. Since QR codes are widely used today and are based on mature technology, we use them to represent information on CryptoPaper documents.

There are a range of configuration options for QR codes. The information encoded by a QR code can be made up of four standard types ("modes") of data: numeric, alphanumeric, byte/binary, Kanji. Error correction codes (ECC) can be used in four "levels," namely, 7 percent or less (level L), 15 percent (level M) or less, 25 percent (level Q) or less and 30 percent (level H) or less. The size of the code ("version") varies from 1 to 40. Version 1 is a 21 X 21 code and can store 128 bits. When the version increases, the size of the code increases by 4 in each dimension, along with

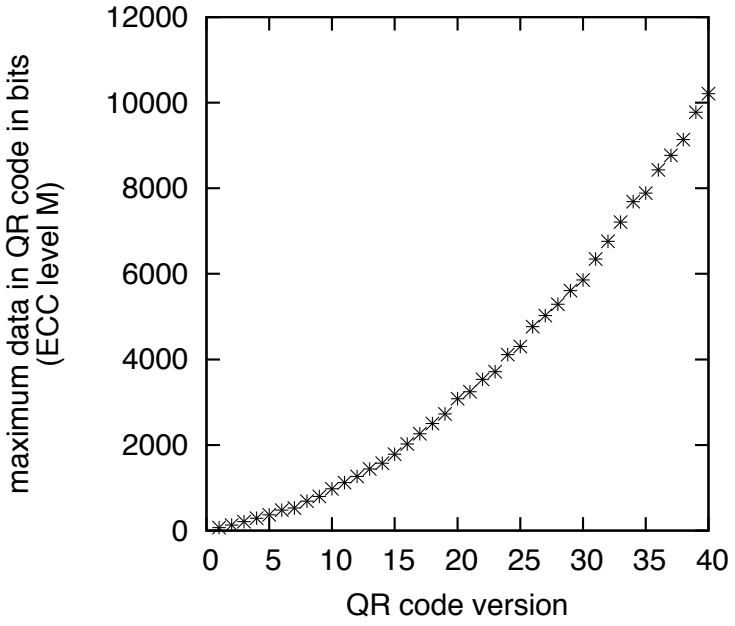

Figure 2.6. Data bits in QR codes of different versions.

the amount of data that can be stored. For example, version 40 is a 177 X 177 code that can store 10,208 bits of data (with level L error correction). Example QR codes are shown in Figure 2.6.

A key question for our system is if a QR code can store information at a sufficient level of density to replace a region of sensitive text with its encrypted digital representation and the necessary meta-information to recover the text. For a typical text document (Calibri font, 11 pt font, 1.5-line spacing), we measured an information density of around 200 bits per inch^2. For comparison, we printer QR codes with a density of approximately 32 code dots per inch. In this case, a version-1 QR code (72 bits, 0.4225 inch^2) has a data density of 170 bits per inch. For version-2 QR code (128 bits,0.5625 inch^2) has a data density of 228 bits per inch^2. For higher ver-

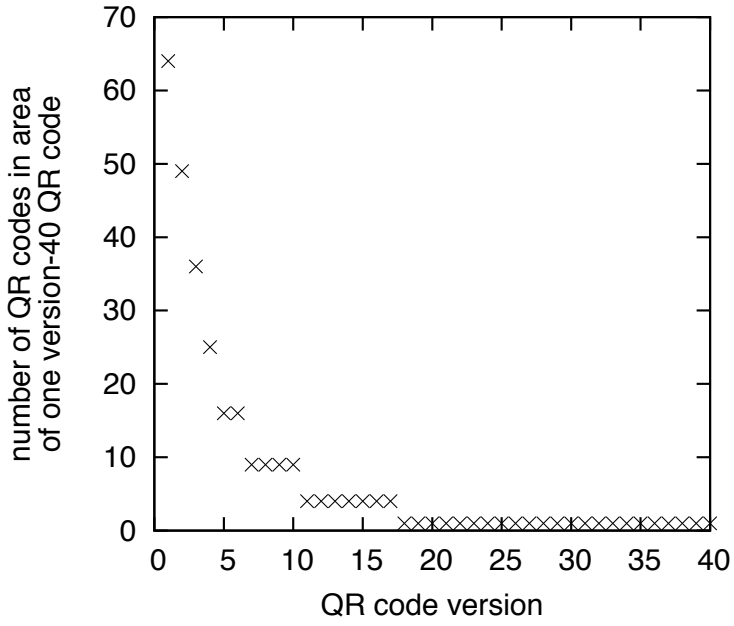

Figure 2.7. Number of QR codes that fit in space of version-40 QR code.

sions, the data density continues to increase since the overhead for finder patterns can be amortized over a larger code. Thus QR codes of version 2 and larger have information densities that exceed that of typical printed text on paper.

While the data density can increased with higher version of the QR code, there are limits to the ability to recover information with the scanner system. In order to explore this tradeoff, we consider a fixed size area and fill it with multiple QR codes of one type. Figure 2.7 shows the number of QR codes that fit into the area of one QR code of version 40. Figure 2.8 shows then the number of bits that are stored in that area. In this figure, we distinguish between the theoretical amount of data that can be placed and the practical amount of data that can be recovered successfully with our scanner. As the figure shows, very large codes of version 30 and higher

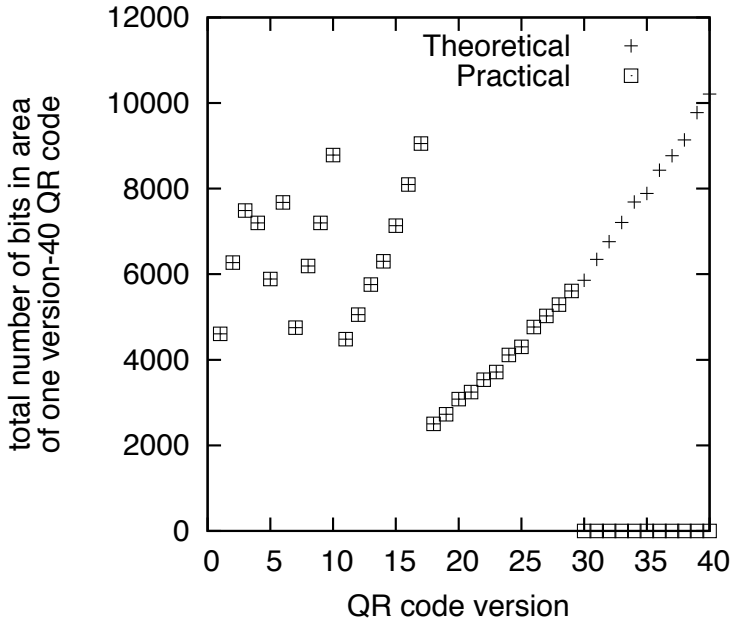

Figure 2.8. Data bits in area of version-40 QR code (theoretical and practical when read with scanner).

cannot be read and thus the practical amount of information stored in them is zero. The largest amount of recoverable data in the area can be achieved when choosing QR codes of version 3, 6, 10, or 17. We choose QR code version 17 in our prototype.

The information that is stored in the QR code in a CryptoPaper document contains not only the encrypted version of the cleartext information, but also "meta-information" that is used by the scanner to interpret and identify which code it is reading. This information is necessary to query the cloud database for the decryption key.

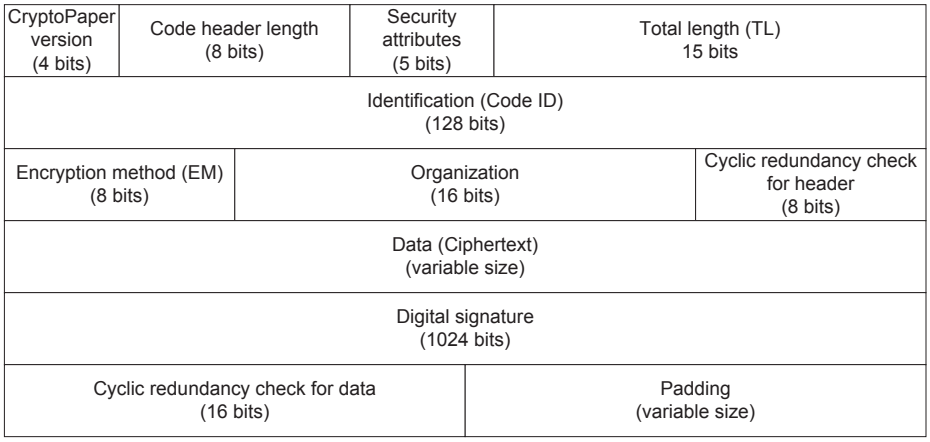

Figure 2.9. Meta-information in QR code.

Our QR codes have the structure shown in the Figure 2.9, which consists of both meta-information and data. Several fields are necessary to define the structure of the code (header length, total length, padding).

- CryptoPaper version: Version of CryptoPaper system to ensure forward compatibility.
- Security attributes: Flags indicating which security properties (described below) are implemented in a specific code,
- Code ID: Randomly generated identifier of QR code, which is used as search index in access control database to obtain decryption key.
- Encryption method: Field to specify which encryption algorithm and settings was used.

- Ciphertext: Encrypted version of sensitive data. The maximum length of the ciphertext depends on the code version used in the system.
- Digital signature: Depending on the security properties that need to be achieved with a given QR code, a digital signature may be included.
- Cyclic redundancy code for data: CRC checksum to verify integrity of code.

2.3 Information Recovery

The scanning device makes an optical recording of a CryptoPaper document, identifies any QR codes, accesses key information to decrypt code data, and displays the cleartext information as a substitute for the QR codes. Using this process, an authorized user can see the document with its original cleartext information on the scanner.

2.3.1 QR Code Detection and Processing

The first step in the process is detecting QR codes in the recorded image. The design of the QR code enables easy identification of such codes in images based on the finder and alignment patterns (see Figure 2.10 as described in [13]).

In our implementation, we capture video from a camera as a sequence of images that are processed sequentially. We use ZXing ("zebra crossing") to detect QR codes and decode them [14]. ZXing is an open-source, multi-format 1D/2D barcode image processing library implemented in Java with ports to other languages. The software can detect whether there are any QR codes in a video frame. For each of these QR

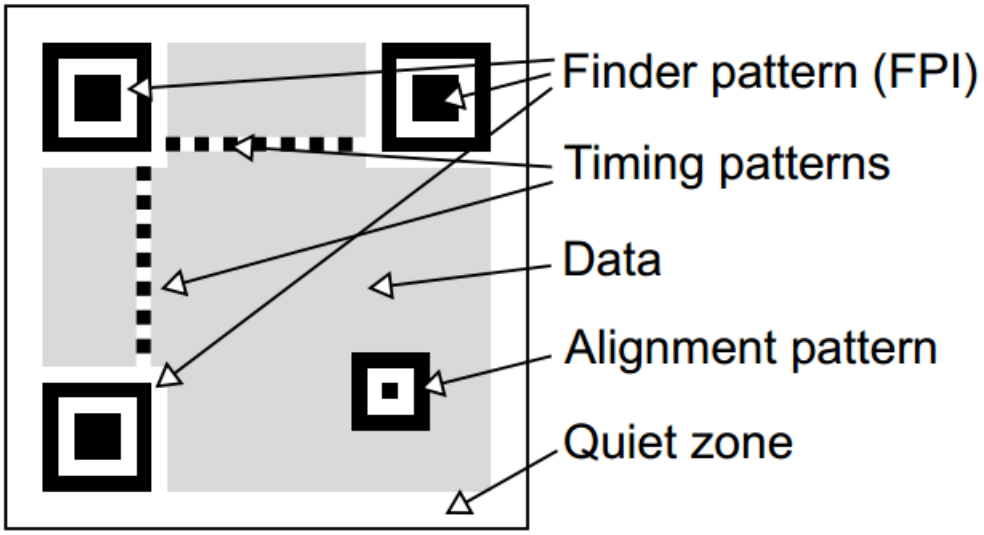

Figure 2.10. QR code finder pattern.

codes, we use the package `com.google.zxing.qrcode.decoder` in ZXing to decode them and obtain the data and meta-information stored in them.

Once a QR code has been decoded, the scanner displays the cleartext information in the location of the QR codes in video stream. We use the `WebcamPanel.Painter` class in our webcam library to paint the QR codes in the video with semitransparent red rectangle areas. Then, we display the decoded and decrypted information in these areas with a different color. First, we need to find the positions of several QR codes in the scanned content. This can be done through the package `com.google.zxing.qrcode.detector` in ZXing. The software recognizes the position of QR code by obtaining the coordinates of three finder patterns. Then, it returns the coordinates of three corners of QR codes. After calculating all these parameters, we use the `fillRect` method in the Java Graphics library to fill the

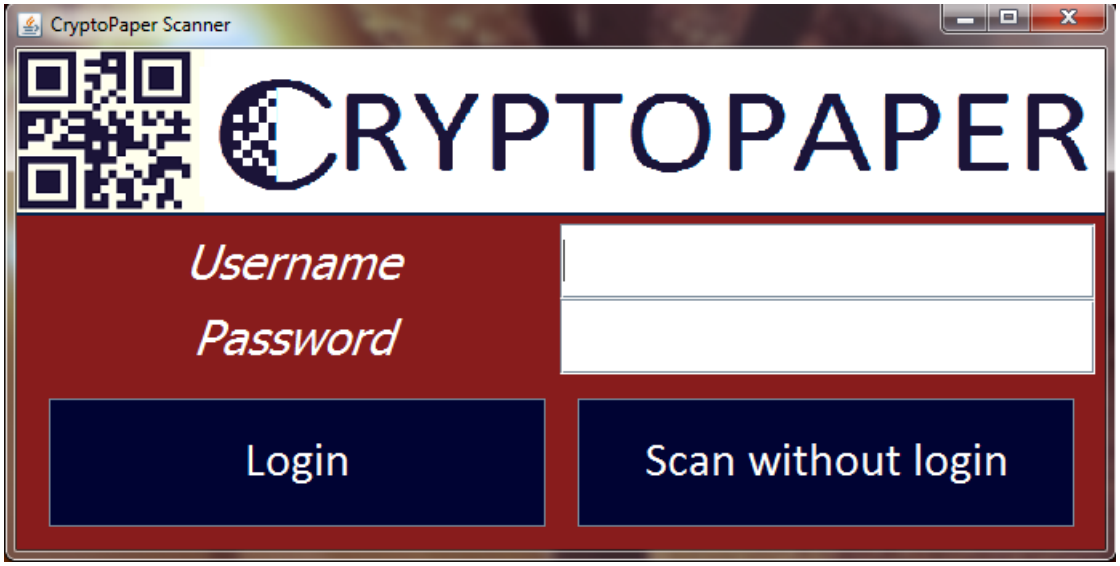

Figure 2.11. Scanner login interface.

specified rectangle. Then, we implement `WebcamPanel.Painter` to paint the information inside the QR codes on the rectangles. Thus, we can clearly visualize the QR codes being replaced by the actual information in the scanned content.

2.3.2 Connection with Access Control Database

To obtain the cryptographic key to decode a QR code, we use the Java Database Connectivity (JDBC). JDBC provides methods for querying and updating data in a database and allows the scanner to connect to our Amazon RDS using JDBC in Java.

The query to the database, a user first has to log into the scanner system (see Figure 2.11). The login system connects securely over the Internet using SSL to the CryptoPaper database. Login is denied to a user who is not registered in the system.

The system hashes the passwords of users (and other sensitive information) before storing them in the database.

Once the QR code has been read and the cryptographic key has been obtained by an authorized user, the scanner can decode the cryptographic information stored in the code. Depending on the security properties implemented in the code, the scanner can not only display the cleartext information, but can also verify integrity and authenticity of the information. The security properties of the decoded information can be displayed in the QR code area in addition to the cleartext information.

CHAPTER 3

SYSTEM SECURITY DESIGN

3.1 Key Management

We use AES algorithm to encrypt and decrypt data, it is a symmetric key. So it is the same key involves in both generation and detection part. That's why we need to put this key in a safe and convenient place so that when an authorized reader need to lookup for the key, it is easy to fetch.

We choose cloud database which stores the cryptographic keys associated with every QR code used in CryptoPaper. Both the code identifier and the encryption key used for sensitive data in the code are generated when the QR code is created. The generation software stores both in the access control database, along with information about who can access the key. Figure 3.1 show the overview of whole process.

When a scanner attempts to read a code, it requests the decryption key from the database. To enable a lookup, the code identifier, which is part of the QR code meta-information, is provided to the database as a search index. The information left in the QR code is shown in Figure3.2. "Code id" is an index for AES key to be created and saved in database. Digital signature is used for identity integrity and authentication which I will introduce in Security Property part. Cipher text is a variable size of encrypted data.

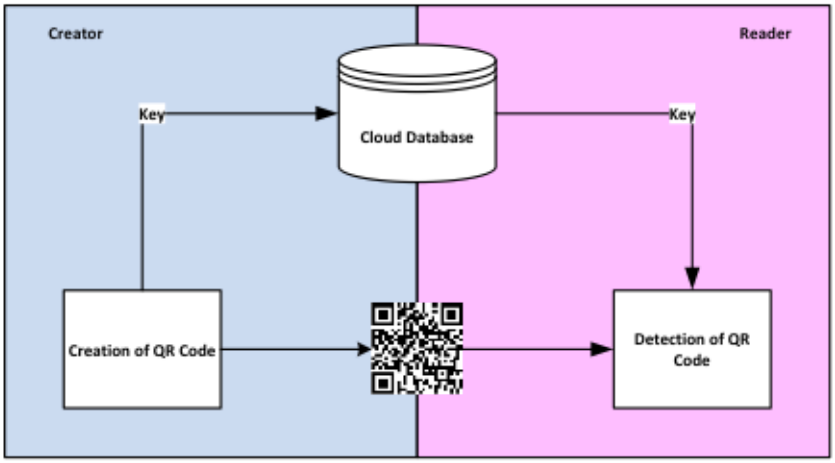

Figure 3.1. Key Management Overview.

It is important to note that the access control database only manages code identifiers and cryptographic keys. The database does *not* store the data encrypted in a code. That data remains solely on the printed CryptoPaper document.

3.2 Security properties

For information security, CIA triad is the heart of it which involes Confidentiality, Integrity and Authenticity. This project is designed to acheive these properies and implemented. [15]

3.2.1 Confidentiality

Confidentiality refers to preventing the disclosure of information to unauthorized individuales or systems. Measures undertaken to ensure confidentiality are designed

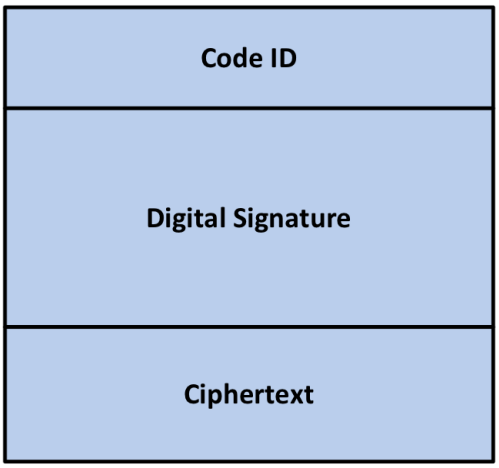

Figure 3.2. Information stored in single QR code.

to prevent sensitive information from reaching the wrong people, while making sure that the right people can in fact get it: Access must be restricted to those authorized to view the data in question. We achieve confidentiality by encrypting the sensitive data in the document using AES algorithm. For each QR code, we use a randomly generated 128-bit key and encrypt data in 128-bit block using the Electronic Codebook(ECB) mode. As well as the password login required and SSL(Secure Sockets Layer) mechanism is deployed during key transmission [16].

SSL is cryptographic protocols designed to provide communication security over the Internet [17]. It uses asymmetric cryptography to authenticate the counterparty with whom they are communicating, and to exchange a symmetric key.

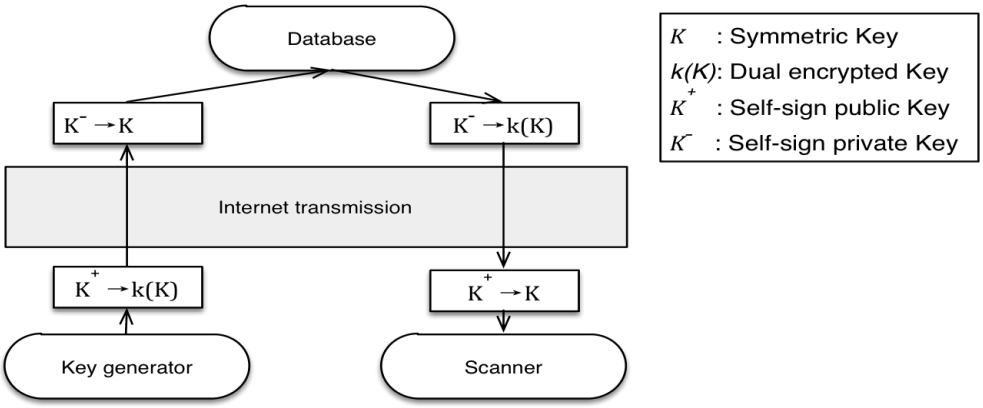

Figure 3.3. SSL Mechanism.

Public-key refers to a cryptographic mechanism. It has been named public-key to differentiate it from the traditional and more intuitive cryptographic mechanism known as: symmetric-key, shared secret, secret-key and also called private-key.

Symmetric-key cryptography is a mechanism by which the same key is used for both encrypting and decrypting; it is more intuitive because of its similarity with what you expect to use for locking and unlocking a door: the same key. This characteristic requires sophisticated mechanisms to securely distribute the secret-key to both parties.

In our case, we upload our symmetric key to cloud database after cloud database destribuing a public key. After encrypting with the public key our original symmetric key turn into a dual encrypted key. Only determined cloud database can decrypt the transmitted data because only the real organization own the private key. That ensure only authorized organization can get the information we transmitted.

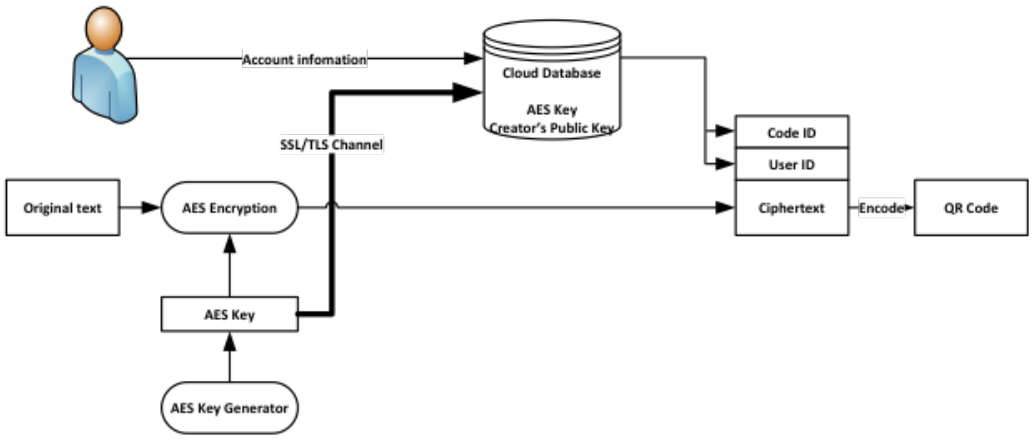

Figure 3.4. Key management with confidentiality property.

When decrypting it, user need to first verify their identity. Then CryptoPaper database check whether this user has access right to read this code. If he/she is permitted, this user get key from cloud database with SSL protocol protection. The process is showed in Figure3.5. So even if the cloud database is attacked, the attacker still cannot get the sensitive information, since the document(ciphertext) is hold by creator.

3.2.2 Intergrity and Authenticity

Intergrity involves maintaining the consistency, accuracy, and trustworthiness of data over its entire life cycle. Data must not be changed in transit, and steps must be taken to ensure that data cannot be altered by unauthorized people.

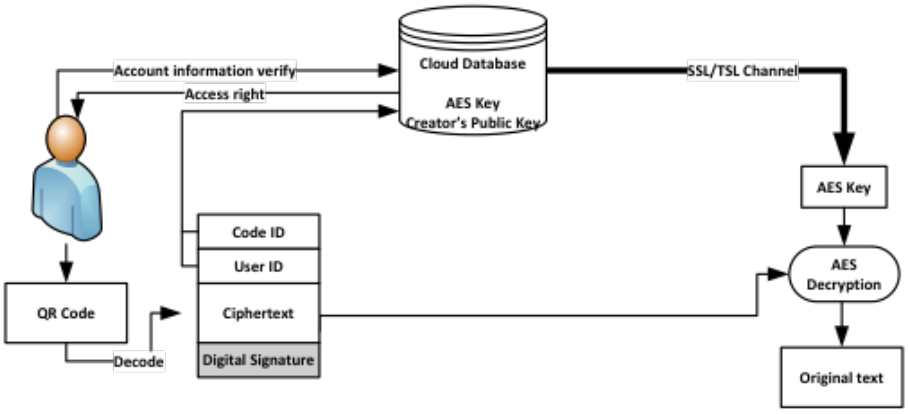

Figure 3.5. Decrypting Code by Acheiving confidentiality property.

It is necessary to ensure that the data is genuine. It is also important for authenticity to vaildate that both parties involved are who they claim to be. In the other hand, if someone or some organization who has created the code previously but denies to admit he/she has created this particular code. A method need to be worked out to avoid this kind of situation.

Digital signature [18] is a mathematical scheme for demonstrating the integrity and authenticity of a digital message or document. Proving that this message or document is effectively coming from a given sender, much like a signature on a paper document. For instance, the sender uses its private key to encrypt the message or document, sender then sends the message or document along with its public key. Since sender's public key [19] is the only key that can decrypt this message or document, a successful decryption concstitutes a Digital Signature Verfication,

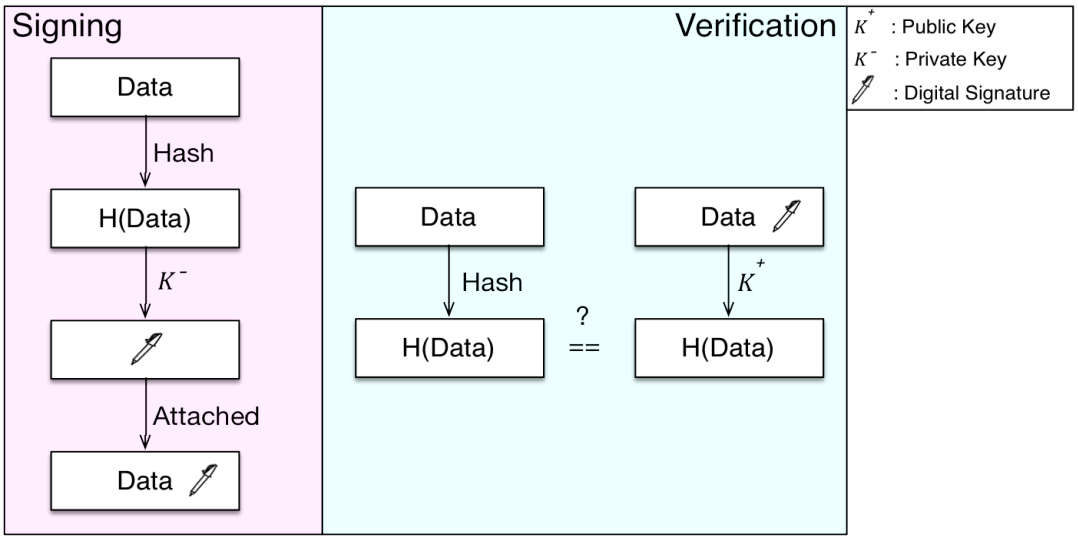

Figure 3.6. Digital Signature Mechanism.

meaning that there is no doubt that it is sender's private key that encrypted the message or document. So it ensure that this message or docuemnt is came from the sender, at the same time, sender cannot deny it is him/her who sent this message or document.

In this project, I first encrypt original text with randomly generated AES key and hash the ciphertext by hash function. Using creator's own private key to encrypt the hash value then turn into creator's digital signature which is attached to the ciphertext.

A authorized reader can first check whether this code is the original version or whether is created by creator, that is, whether it is genuine. The process goes into two parts: first fetch creator's public key from database and decrypt the digital signature which turn back into a hash value. The second part is to hash the ciphertext by using

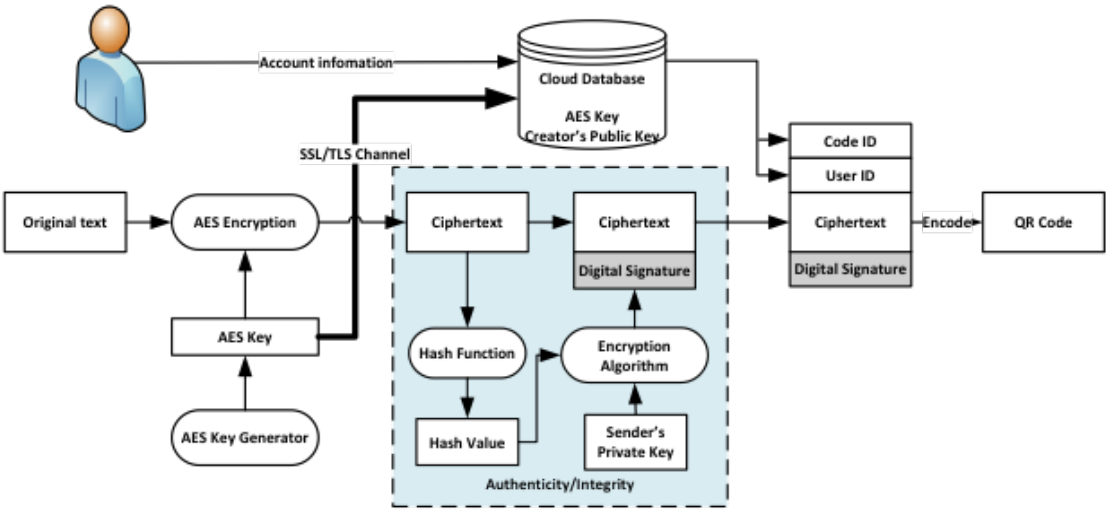

Figure 3.7. Achieving Integrity and Authenticity by Using Digital Signature.

same hash function to get a hash value. Then compare these two hash value. If they are excatly same, it shows the code is created by creator and no error or modification. Figure3.17 shows the process.

3.3 Access Control Database

The access control database, which manages the keys to decode QR codes and which ensures that only authorized users can access those keys, is another critical component in CryptoPaper. This database provides a logical connection between the QR codes generation process and the scanning system that attempts to interpret a QR code. The main functions of this database are to implement key management, role-based access control.

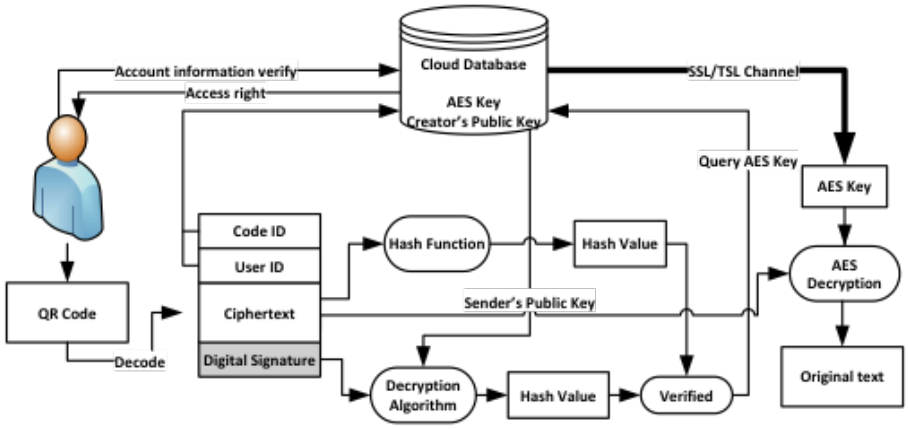

Figure 3.8. Decypting Code when Using Digital Signature.

3.3.1 Role-Based Access Control

To enable the implementation of typical security policies used in organizations, we use role-based access control in CryptoPaper. Users are associated with roles that enable access to certain types of QR codes. When generating codes, users can determine which roles in their organization (or organizations that they interact with) can access a code [20].

The use of role-based access control enables the implementation of access revocation in case users change roles or organizations. By simply removing a role from a user's profile, their access to this role's codes can be removed.

To make the database scalable, storage of code identifiers and cryptographic keys can be done independently for each organization (unless roles cross organization boundaries).

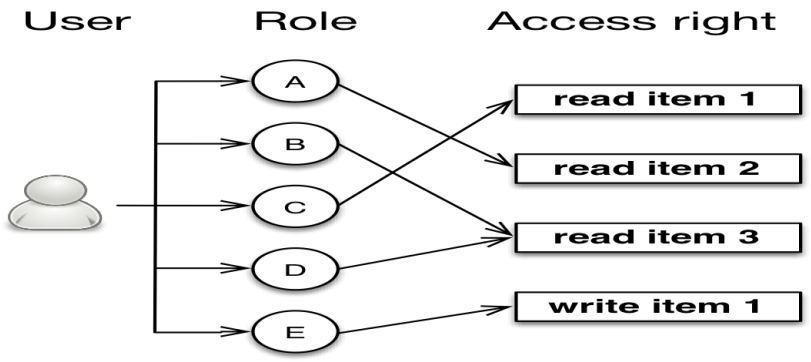

Figure 3.9. Role-based Access Control.

3.3.2 Database Structure

We have implemented a CryptoPaper database in the Amazon cload Relational Database. Figure 3.10 provides an overview of this database. The tables used in our implementation are:

- The `User_info` table contains basic user information.

Table 3.1. Users Info Table

Name	Type	Length	Description
User_id	VARCHARE	45	Auto added, auto increment
usename	VARCHAR	70	Use email as usename
first_name	VARCHAR	40	-
last_name	VARCHAR	40	-
is_active	BOOLEAN	1	Some user may deactivate the account
data_joined	DATETIME	-	The date and time user signed up
last_login	DATETIME	-	Auto added when user login

- When users log in, the user identifier and password they provide is verified against the data stored in the `User_credential` table.

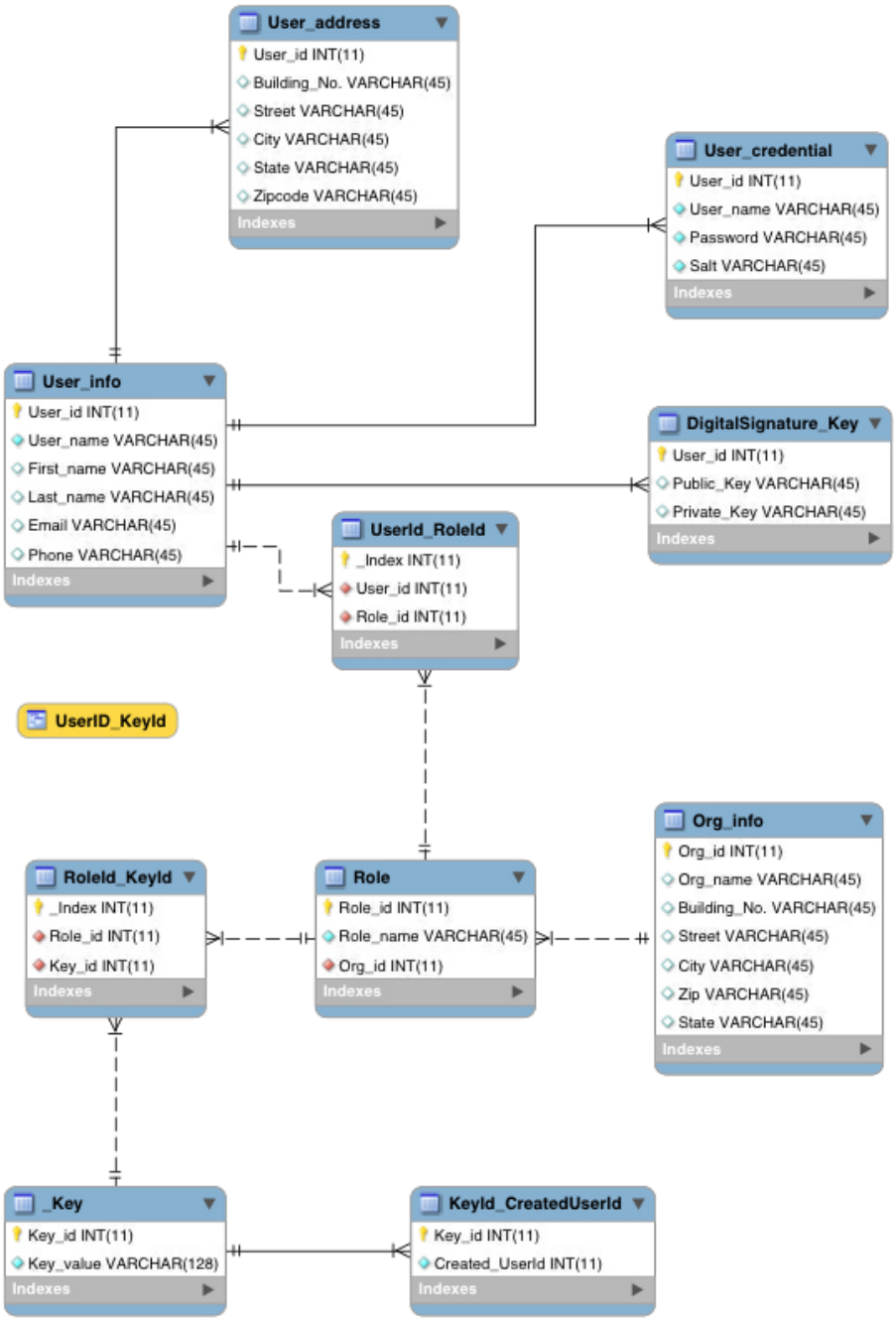

Figure 3.10. CryptoPaper access control database structure.

Table 3.2. User_credential Table

Name	Type	Length	Description
User_id	VARCHAR	45	Auto added, auto increment
password	VARCHAR	128	Encrypted by SHA-256
Salt	VARCHAR	45	Use to hash the password

- The `Digital_Signature_key` table saves the public and private key pair, which is used for digital signatures. The public and private key pair is generated when a user registers in our system. The primary key, `User_id`, in this table is the foreign key of the `User_credential`, `User_address` and `UserId_RoleId` tables.

Table 3.3. User Address Table

Name	Type	Length	Description
User_id	VARCHAR	45	Auto added, auto increment
Building_No.	VARCHAR	45	Or Apt No.
Street	VARCHAR	45	-
City	VARCHAR	45	-
Zip	VARCHAR	20	-
State	VARCHAR	45	-

- The `Role` and `Organization` table saves every identified role in every identified organization. Since every each organization may has same position, the primary key `Org_id` in `Organization` talbe is a foreign key of `Role` table. This schema enhance the convenience and flexibility of adding or deleting role operation.

Table 3.4. Role Table

Name	Type	Length	Description
Role_id	INT	11	Primary Key
Role_name	VARCHAR	45	-
Org_id	INT	11	Foreign Key

Table 3.5. Organization Table

Name	Type	Length
Org_id	INT	11
Org_name	VARCHAR	45
Building_No.	VARCHAR	45
Street	VARCHAR	45
City	VARCHAR	45
Zip	VARCHAR	11
State	VARCHAR	11

- `UserId_RoleId` table enable the access right is assigned to the specific role instead of specific user.

- When a QR code ciphertext is created, the key is saved in the `_Key` table and the user can set the access right to the key, i.e., a key-role pair indicating which role can read this specific key. This information is saved in the `RoleId_KeyId` table.

- At the same time, the creator user identifier is also recorded in the `KeyId_CreatorUserId` table to enable users to revoke a code's access rights and retrieve their encryption history.

- A `UserId_KeyId` view is created each time a query from a scanner is received during the decryption process. This way, the scanner gets the entire authorized user identifiers associated with a specific code with a single query.

We also use Amazon Relational Database Service (Amazon RDS) [21] to build our database, which is a web service that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity

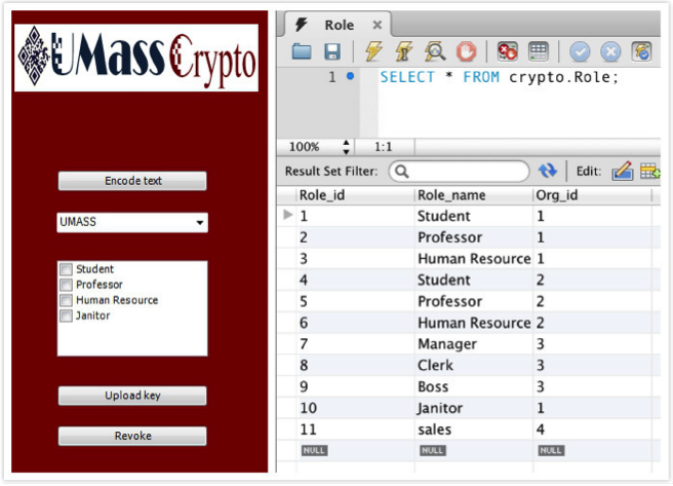

Figure 3.11. Role Table in CryptoPaper Database to Perform Add/Delete Function.

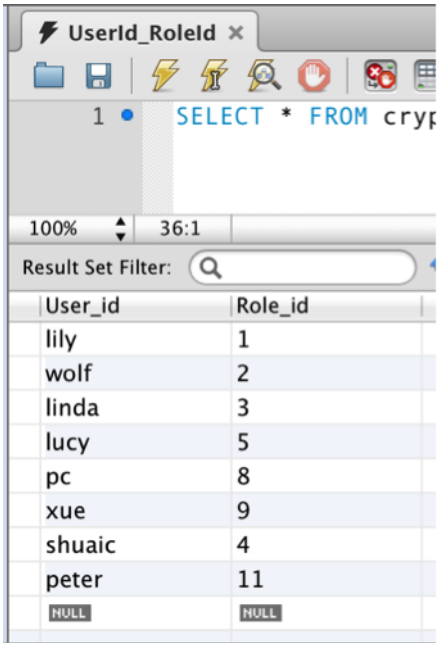

Figure 3.12. UserId_RoleId Table in CryptoSystem Database with Various Roles in an Organization.

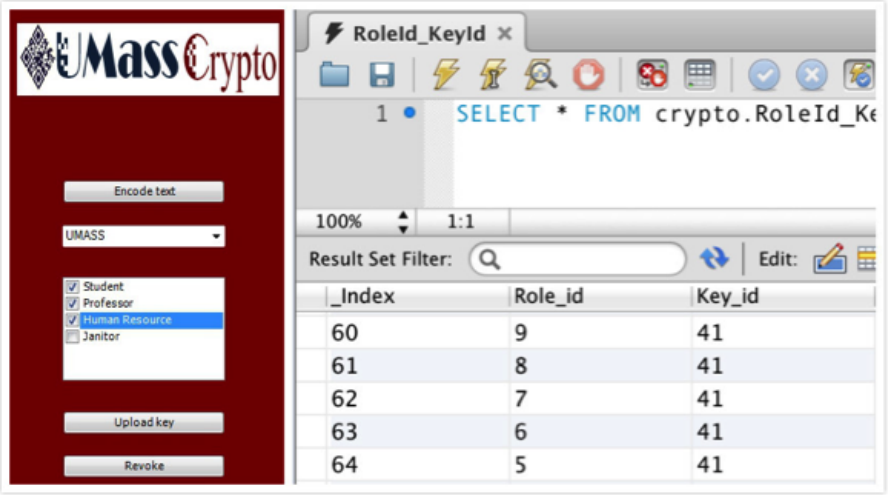

Figure 3.13. RoleId_KeyId Talbe in CryptoSystem Database Used to Provide Multiple Levels of Access.

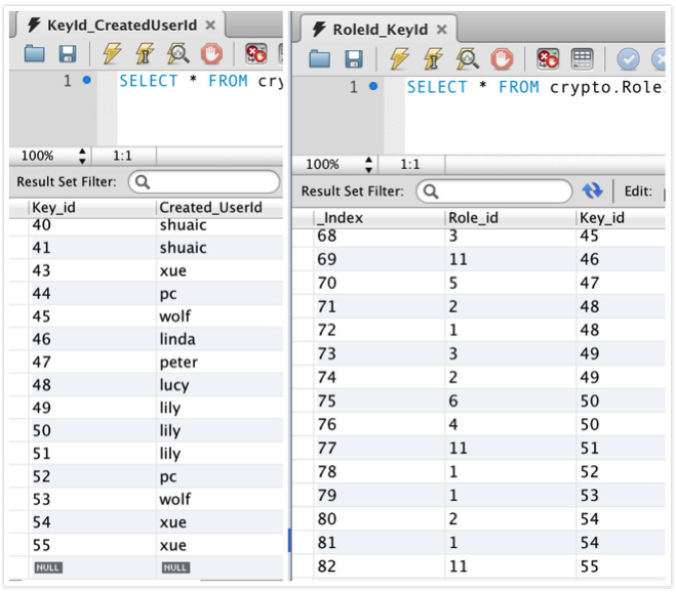

Figure 3.14. RoleId_KeyId, KeyId_CreatedUserId tables in CryptoSystem Database used to provide access rights.

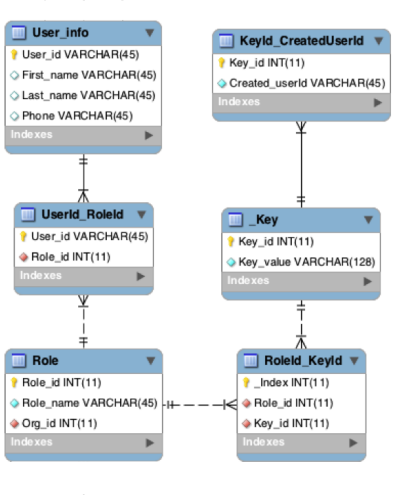

Figure 3.15. Get creator Id from Code Id

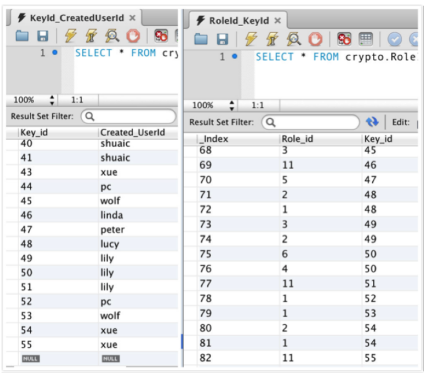

Figure 3.16. Modify Role_Key pair

while managing time-consuming database administration tasks, freeing you up to focus on your applications and business.

Our implementation enables complete role-based access control, dynamic management of users and roles, and dynamic change of access rights for codes. Scanners can access key information quickly, and invalid operation (e.g., duplicate uploading of keys) can be detected and avoided.

3.3.3 Access Right Revocation

This database scheme also should allow to revoke the access right after code is generated. By doing this, we need to first get the code Id then we find this specific key and lookup `KeyId_CreatedUserId` Table. Then compare whether the sent request user is the same with creator for a specific key. If it is same, the column of this key need to be modified.

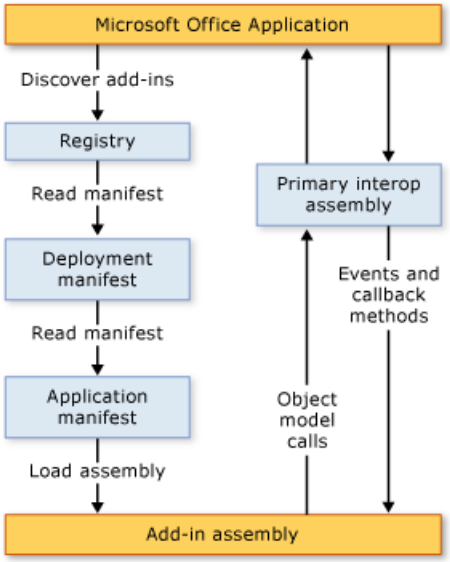

Figure 3.17. How Add-in Work with Microsoft Applications.

3.4 Separating CryptoPaper Service from Microsoft Word

CryptoPaper code generation service can run outside of Microsoft Word so that muliple applications can use at same time.

3.4.1 How add-in work with Microsoft Office Application

When a user starts a Microsoft Office application, the application uses the deployment manifest and the application manifest to locate and load the most current version of the add-in assembly. The following illustration shows the basic architecture of these add-ins.

The following steps occur when a user starts an application: The application checks the registry for entries that identify add-ins that were created by using the Office developer tools in Visual Studio. If the application finds these registry entries,

the application loads VSTOEE.dll, which loads VSTOLoader.dll. These are unmanaged DLLs that are the loader components for the Visual Studio 2010 Tools for Office Runtime. VSTOLoader.dll loads the .NET Framework and starts the managed portion of the Visual Studio Tools for Office runtime. The Visual Studio Tools for Office runtime checks for manifest updates, and downloads the most recent application and deployment manifests. The Visual Studio Tools for Office runtime performs a series of security checks. If the add-in is trusted to run, the Visual Studio Tools for Office runtime uses the deployment manifest and application manifest to check for assembly updates. If a new version of the assembly is available, the runtime downloads the new version of the assembly to the ClickOnce cache on the client computer. The Visual Studio Tools for Office runtime creates a new application domain in which to load the add-in assembly. The Visual Studio Tools for Office runtime loads the add-in assembly into the application domain. The Visual Studio Tools for Office runtime calls the RequestComAddInAutomationService method in your add-in, if you have overridden it. You can optionally override this method to expose an object in your add-in to other Microsoft Office solutions. The Visual Studio Tools for Office runtime calls the RequestService method in your add-in, if you have overridden it. You can optionally override this method to extend a Microsoft Office feature by returning an object that implements an extensibility interface. The Visual Studio Tools for Office runtime calls the ThisAddInStartup method in your add-in. This method is the default event handler for the Startup event.

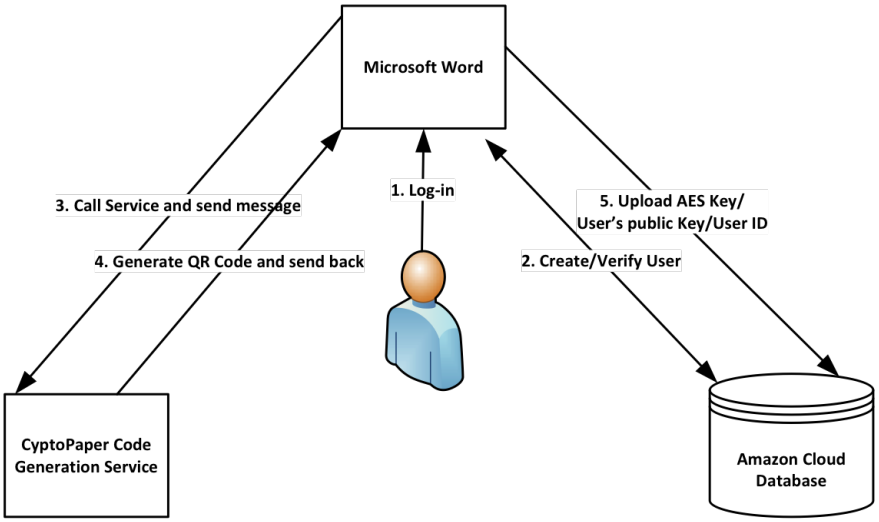

Figure 3.18. CryptoPaper Service Communication Process.

3.4.2 CryptoPaper Code Generation Service Communication Process

When user open document editing software eg. Microsoft Word the current user need to log in his/her account if he/she has already signed up. Then he can call CryptoPaper code generation service rather than generate code inside Word Application, a interface shows up. And from Word, user can choose whatever he/she wants to encrypt and send message to CryptoPaper service. Once CryptoPaper Service receive message, service encrypt data with AES algorithm and create a digital signature with creator's private key. After the generation service will send back the packet to Word. The process is shown on Figure 3.18.

To implement it we use "SendMessage" API which is provide by Windows. First pass the destination of this message. By identifing the applicaion we can choose method from FindWindow class or GetProcess class. After the message is received,

Figure 3.19. Seperated CryptoPaper Service

Figure 3.20. Message Received from Word

it can be shown on the application textbox. then create CryptoPaper code then pass back to Word. Figure 3.19 and Figure3.20 show the interface of seperated CryptoPaper service.

CHAPTER 4

EVALUATION

In this chapter, we shows the test results of key management module coordinates with code generation part and information recovery part. And several scenarios which to demonstrate how this system perform. Section 1 test whether an unathorized use can use this system. Section 2 test whether a user can select different access right by different roles. And how it works when detecting the codes. Section 3 test when an unauthorized user want to change the access right. Section 4 test how the system works when an authorized user want to change the access rightand how long time takes to relfect to information recovery part.

4.1 User Log in Protection

CryptoPaper system was tested for confidentiality by checking if a unauthorized/unregistered user can log in to our system. The result expected was to deny access even after multiple attempts and hence the system remains confidential only to registered users. Figure 4.1 demonstrates when a unanthorized/unregistered user get denied by this system during code generation since there is account information of this user in CryptoPaper database. Figure 4.2 shows the same scenario in information recovery module.

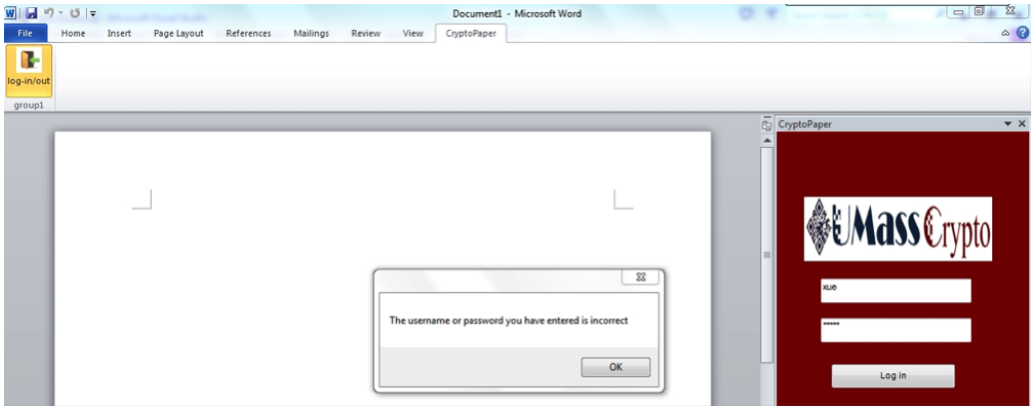

Figure 4.1. Login Access Denied for Unauthorized Users during CryptoPaper Generation.

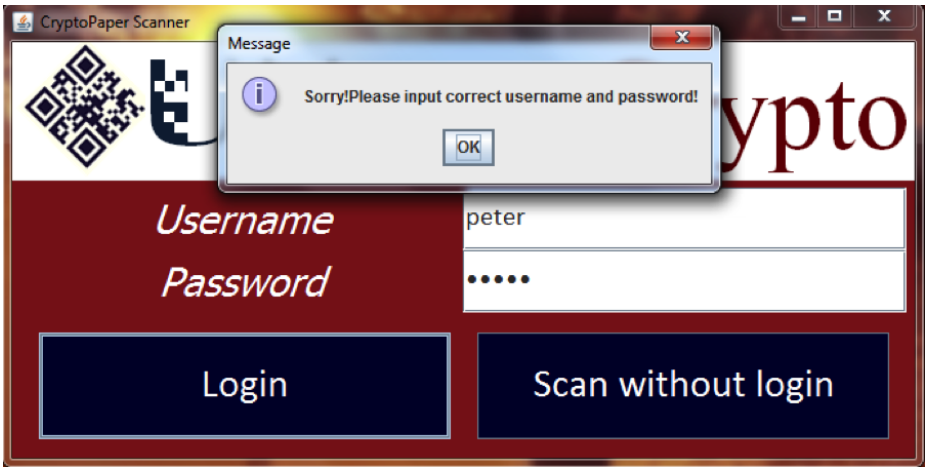

Figure 4.2. Login Access Denied for Unauthorized Users during Information Recovery Module.

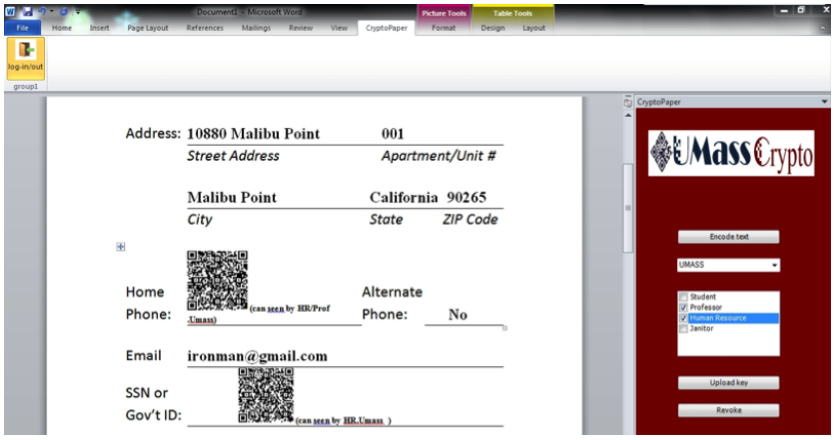

Figure 4.3. User Selection of Access Levels of the First QR Code during CryptoPaper generation.

4.2 Access Level Changes by the User

We test our system for different access levels set by a user. As shown in the Figure 4.3 and Figure 4.4 the user selects the access to be provided for the first QR code only to a HR or a professor in organization named UMASS and the second QR code only to a HR who belongs to the organization named UMASS. The result expected at the Information recovery module was to allow access only for a user who belongs to UMASS and is a professor or a HR. The result is shown in Figure 4.4.

4.3 Check If One User Can Change the Access Rights of Codes Created by Another User

To check if the system is reliable and robust we perform this test if one user can change the access rights of codes created by another user.Figure 4.6 shows the test

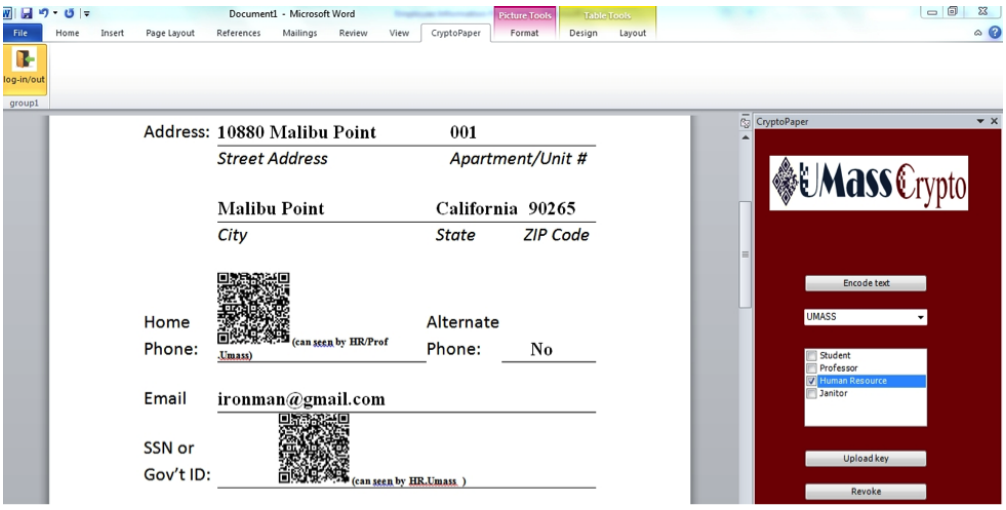

Figure 4.4. User Selection of Access Levels of the Second QR Code during CryptoPaper generation.

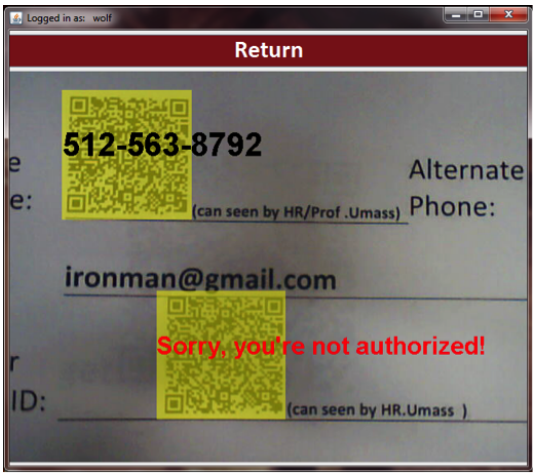

Figure 4.5. The User wolf Who is a Professor in UMASS Is not Authorized to Read the Second QR Code But Can Read the First QR code.

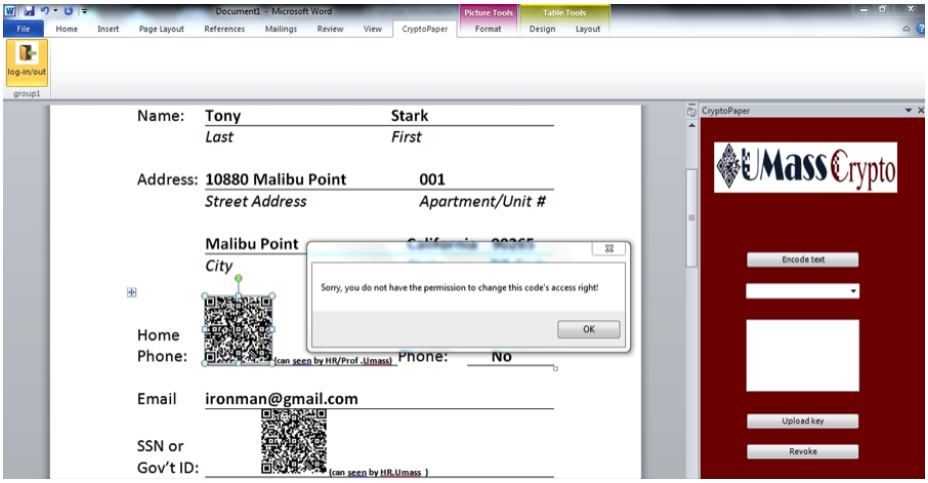

Figure 4.6. User Denied Access to Change the Access Rights of a Code Owned by Another User.

result during code generation. The expected result during detection is to deny access and the result obtained is shown in Figure 4.5 .

4.4 Check If the Updates in a Database Are Reflected in the Recovery Module

We check the performance of our system and reliability by checking if the updates in a database are properly reflected. To do this we denied the access rights to anyone belonging to organization UMASS to view the code. The user wolf, who is a professor in UMASS was logged in to read the code after this change. The expected result was to deny access to wolf to view the code. The expected result is shown in Figure 4.7 and Figure 4.8. The updates only takes around 3 seconds in inforamation recovery part.

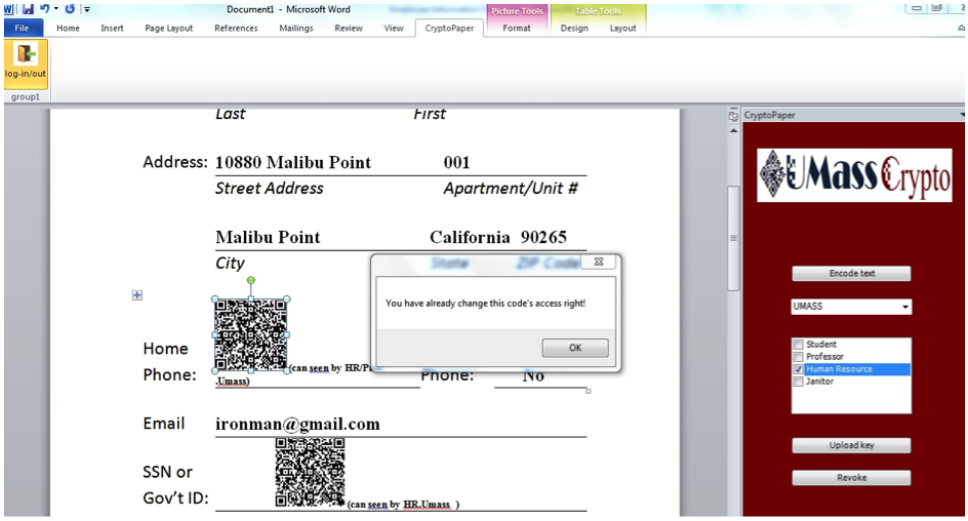

Figure 4.7. Access Rights Changed by the Same User during Code Generation.

Figure 4.8. The User wolf Can No Longer View the Code.

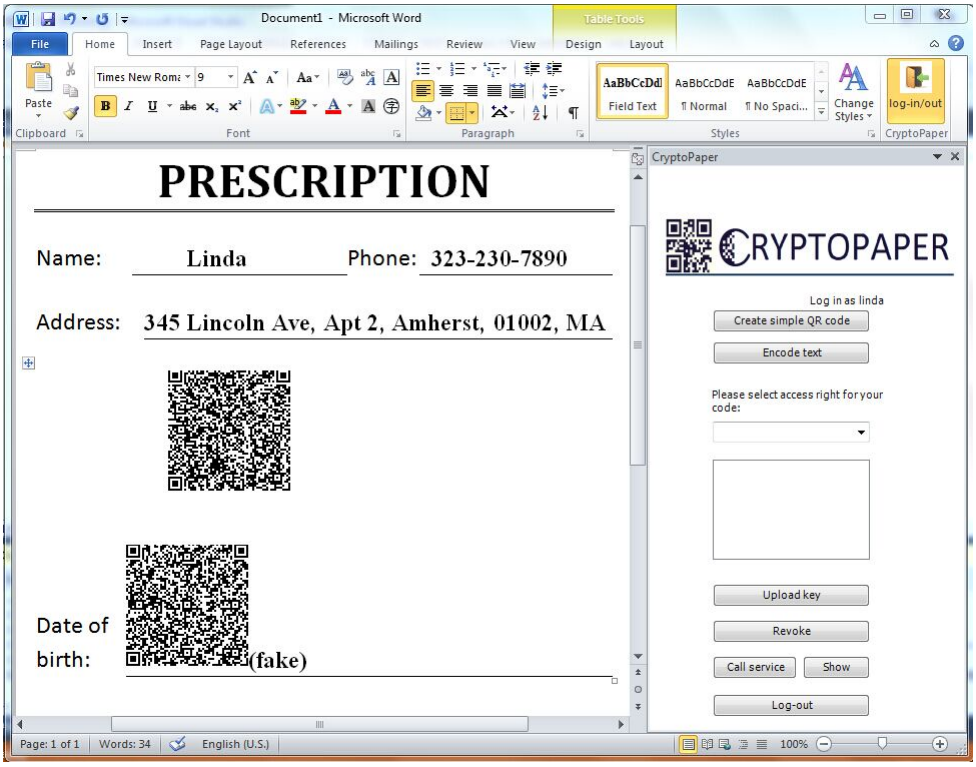

Figure 4.9. Check when digital signature is changed.

4.5 Check If Digital Signature is been Maliciously Modified

After generating a normal code, we copy the character inside the QR code but change the digital signature see Figure4.9. The result show in Figure4.9. If someone change any part of the data, the digital signature will be changed. So a valid digital signature will guarantee the genuine of the data.

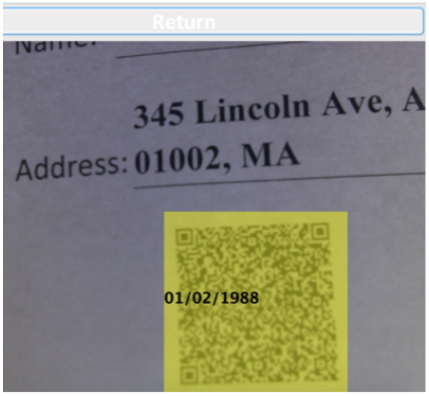

Figure 4.10. Show Original Text if it is valid signature

Figure 4.11. without valid Digital Signature

4.6 Check Whether does the Code Which Generated by Seperated CryptoPaper Service

A seperated CryptoPaper service should work as well as the the function inside Microsoft Word.Figure4.12show the operation when call independent service. After generating the code show in the document. Test it whether could be read with authorized account and unauthorized account. The result is shown in Figure4.13 and Figure4.14.

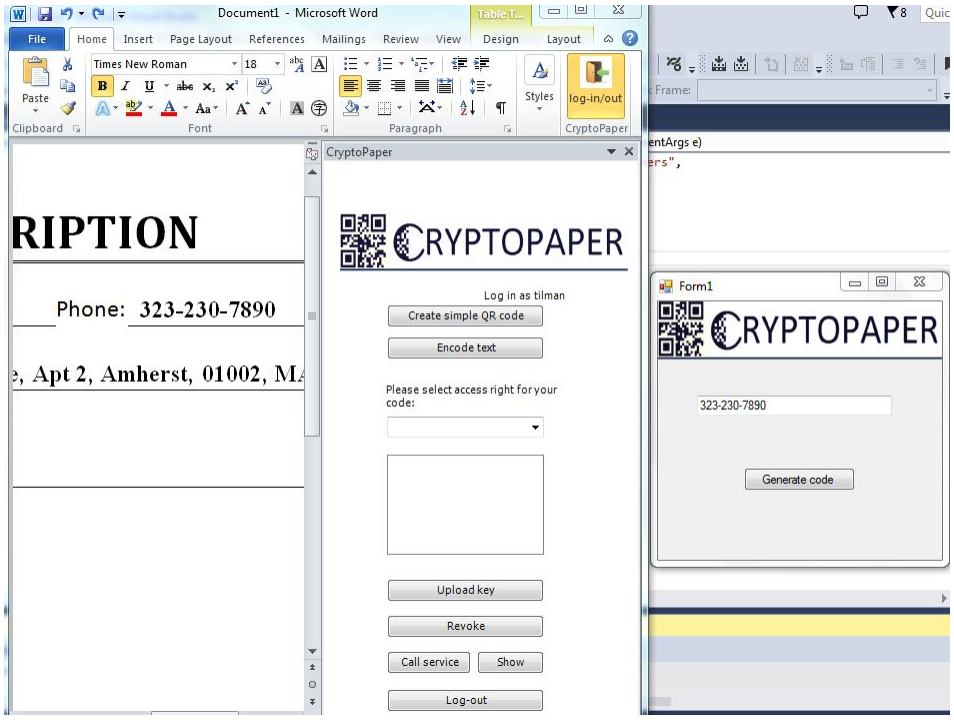

Figure 4.12. Generate Code from Independent CryptoPaper Service

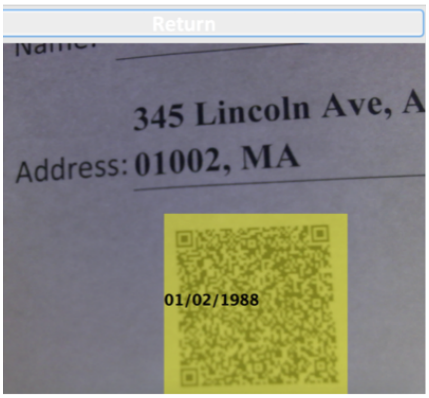

Figure 4.13. Show Original Code generated by Independent Service with authorized reader account

Figure 4.14. Code generated by Independent Service with autho-vice with unauthorized reader account

CHAPTER 5

CONCLUSION

Throughout this thesis, we have designed a digital information security service for physical documents, which is successfully implemented. In Chapter one, the backgroud and the motivation of this project were discussed. In addition, the problems what we had to address and the challenges what we achived are briefly explained. The second chapter of the thesis talk about the relevant work, which is about generation and information recovery part.

The third chapter of this thesis explains my main work, system security design. The key management tells how and where the symmetric key be generated and where to pass and save. And also how to manage the asymmetric key for digital signature. Second part in this chapter is about how to achive and ensure security properties which are confidentiality, intergrity and authenticity. Third part is related how to design a role-based access control database schema to not depend on specific person but role to access the key. The forth part is about how to separate the service from Microsoft Word and can provide service to multiple applications parallel.

The evaluation work are implemented in forth chapter. This service passed all the tests and proved its functionality.

BIBLIOGRAPHY

[1] M. Codish, K. Marriott, and C. Taboch, *FY2008 Investigation Report on Information Security Incidents (Ver.1.3)*. NPO Japan Network Security Association, 2008.

[2] `http://www.ftc.gov/news-events/press-releases/2007/12/company-will-pay-50000-penalty-tossing-consumers-credit-report`, 2007.

[3] `http://www.huffingtonpost.com/2011/05/16/nypd-counterterrorism_n_862493.html`, May 2011.

[4] `https://nakedsecurity.sophos.com/2013/01/15/medical-patients-health-records-dump/`, 2013.

[5] T. Anan, K. Kuraki, and J. Takahashi, "Paper encryption technology," *FUJITSU Sci. Tech. J.*, vol. 46, pp. 87–94, 2010.

[6] X. Lu and Z. Lu, "A publishing framework for digitally augmented paper documents: Towards cross-media information integration.," in *PCM* (Y. Zhuang, S. Yang, Y. Rui, and Q. He, eds.), vol. 4261 of *Lecture Notes in Computer Science*, pp. 494–501, Springer, 2006.

[7] M. J. Gormish, "Interaction between paper and electronic documents," in *Proceedings of the 2005 ACM Symposium on Document Engineering, Bristol, UK, November 2-4, 2005*, p. 133, 2005.

[8] C. M. Li, P. Hu, and W. C. Lau, "Demo: Authpaper - protecting paper-based documents/credentials using authenticated 2d barcodes.," in *MobiSys*, p. 348, 2014.

[9] *Embedded data glyph technology for hardcopy digital documents*, vol. 2171, 1994.

[10] M. H. Eldefrawy, K. Alghathbar, and M. K. Khan, "Hardcopy document authentication based on public key encryption and 2d barcodes.," in *ISBAST*, pp. 77–81, IEEE Computer Society, 2012.

[11] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. Berlin, Heidelberg, New York: Springer Verlag, 2002.

[12] N. I. of Standards and Technology, "Advanced encryption standard," *NIST FIPS PUB 197*, 2001.

[13] I. Szentandrási, A. Herout, and M. Dubská, "Fast detection and recognition of QR codes in high-resolution images," in *Proceedings of 28th Spring conference on Computer Graphics.ACM*, pp. 1–8, Comenius University in Bratislava, 2012.

[14] S. Owen, "Zxing." `https://github.com/zxing/zxing`, 2013.

[15] `http://en.wikipedia.org/wiki/Information_security#Key_concepts`, 2013.

[16] C. Feher, Y. Elovici, R. Moskovitch, L. Rokach, and A. Schclar, "User identity verification via mouse dynamics," *Inf. Sci.*, vol. 201, pp. 19–36, Oct. 2012.

[17] S. A. Thomas, *SSL and TLS Essentials: Securing the Web with CD-ROM*. New York, NY, USA: John Wiley & Sons, Inc., 2000.

[18] C. NIST, "The digital signature standard," *Commun. ACM*, vol. 35, pp. 36–40, July 1992.

[19] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, pp. 120–126, Feb. 1978.

[20] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, pp. 38–47, Feb. 1996.

[21] R. Lawrence and K. Barker, "Integrating relational database schemas using a standardized dictionary," in *Proceedings of the 2001 ACM Symposium on Applied Computing*, SAC '01, (New York, NY, USA), pp. 225–230, ACM, 2001.