July 2015

Design and Implementation of Digital Information Security for Physical Documents

Pengcheng Wang
University of Massachusetts Amherst

DESIGN AND IMPLEMENTATION OF DIGITAL INFORMATION SECURITY FOR PHYSICAL DOCUMENTS

A Thesis Presented

by

PENGCHENG WANG

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

May 2015

Electrical and Computer Engineering

DESIGN AND IMPLEMENTATION OF DIGITAL INFORMATION SECURITY FOR PHYSICAL DOCUMENTS

A Thesis Presented

by

PENGCHENG WANG

Approved as to style and content by:

Tilman Wolf, Chair

Aura Ganz, Member

Michael Zink, Member

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

ACKNOWLEDGMENTS

First of all, thank god for giving me this chance to do this project! And I would like to express my sincere appreciation to my advisor Prof. Tilman Wolf for his invaluable guidance during the project and his instructions on my work. Also I want to thank my committee members Prof. Aura Ganz and Prof. Michael Zink for their kind help in my thesis proposal and defense.

Then I would like to thank my team members Shuai Chen, Yu Xue, Shuo Guo and Padmaja Duggisetty for making contributions to reaching the final design of the project.

I will also give my thanks to PhD student Xinming Chen for his advice and help when we design our project.

In addition, I will thank my girlfriend Jie Zhao for her encourage when I encounter problems during the project.

Last but not least, I would thank my parents for their selfless love!

ABSTRACT

DESIGN AND IMPLEMENTATION OF DIGITAL INFORMATION SECURITY FOR PHYSICAL DOCUMENTS

MAY 2015

PENGCHENG WANG

B.S, NANJING UNIVERSITY

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

Directed by: Professor Tilman Wolf

The objective of this thesis is to improve the security for physical paper documents. Providing information security has been difficult in environments that rely on physical paper documents to implement business processes. Our work presents the design of a digital information security system for paper documents, called "CryptoPaper", that uses 2-dimensional codes to represent data and its security properties on paper. A special scanner system is designed for "CryptoPaper" which uses image recognition techniques and cloud-based access control to display plaintext of encrypted and encoded data to authorized users.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background

Nowadays physical documents are widely used for business, government, and personal use. According to TAPPI [1], Americans use more than 90 million short tons of paper and paperboard every year. That's an average of 700 pounds of paper products per person each year. These documents often contain lots of sensitive information to which access needs to be controlled (e.g., personal information such as date of birth, home address, phone number, etc.). Or the documents may need to maintain integrity (e.g., contracts, identification documents).

Many individuals and companies consider it important to protect their information for a variety of reasons, including financial, competitive, and privacy-related purposes. People who wish to obtain this information may be computer crackers, corporate spies, or other malicious individuals. This information may be directly beneficial to them, such as industrial secrets or credit card numbers. It may also be indirectly beneficial to them. For example, computer passwords do not have inherent value. However, they provide computer system access that may be used to get other information or to disable a person/company electronically [2].

If not dealt with properly, these paper documents will leak sensitive information which may lead to large economic loss. Examples of security incidents related to releasing sensitive information are discussed in [3]. The inability to discard worthless documents even though they appear to have no value is known as compulsive hoarding syndrome. In December 2007, the Federal Trade Commission announced a

$50,000 settlement with American Mortgage Company of Northbrook, Illinois, over charges the company violated the FTC's Disposal, Safeguards, and Privacy rules by failing to properly dispose of documents containing consumers' credit and personally identifiable information [4]. A $50,000 settlement might seem low when measured against the potential for financial harm to individuals as a result of the company's negligence, but in addition to the negative PR for American Mortgage, the settlement includes an obligation to obtain an audit, every two years for the next 10 years, from a qualified, independent, third-party professional to ensure that its security program meets the standards of the order. Any similar failures by this company during the next decade will be met with more severe punishment. That, indeed, is a very costly lesson.

Thus physical information security has become a serious problem especially when you consider the increasing financial cost of a security breach, not to mention the significant reputation damage that follows. Statistics show it can take as long as a year to recover from a security breach! Although we live in a World of Email and E-Books and we can store all of our documents in the cloud, paper is still necessary and we can't imagine a world without physical paper. Paper is much more usable than a computer monitor and easier to hold and look at. Even if we store document in the cloud, we're still worried about the risk of being hacked by others on the Internet. The 2014 celebrity photo hack [5] of Apple's cloud services suite iCloud is a perfect example for this and has raised universal worries about cloud security.

1.2 Motivation

To protect the privacy of sensitive information on these paper documents, humans have developed lots of different ways. Generally speaking, they can be divided into three methods. One is that documents may use special paper that is tamper-evident [6]. Another method is locking documents in file cabinets [7] to prevent unauthorized access. The last one is using paper shredding service [8] to destroy the documents which contain privacy. The first one is used rarely in people's everyday life. A lot of people will have paper documents stored in the safe deposit box because they can have full control over them. The third one may be a more common method because shredding confidential information can help protect personal information and sensitive corporate data [9].

However these methods can not protect the privacy from the source, even the shredding service can not destroy sensitive information completely. In some cases it is technically possible to reassemble the pieces of shredded documents. If the chad is not further randomized, the strips that belonged to the same document tend to come out of the shredder close to each other and remain roughly in that configuration [8]. In DARPA's Shredder Challenge [10], the winning team used original computer algorithms to narrow the search space and then relied on human observation to piece together documents that were shredded into more than 10,000 pieces [11].

Considering the limits of all these methods for handling paper documents, we design a new service "CryptoPaper" which can solve these problems and improve security for physical paper documents.

1.3 Challenge

After considering all the problems above, we know that "CryptoPaper" should be a service which can make sensitive information on physical paper documents only accessible to authorized users. However, there are two main challenges in implementing information security with traditional physical documents:

1. Cost of implementation

2. Limited ability to revoke access

The use of tamper-evident paper, physical access control, proper disposal of sensitive documents, etc. is expensive to implement for businesses. Once documents have been issued, it is difficult to revoke access, which presents a potential for information leaks and insider attacks. An example, where these limitations have resulted in real data breaches, has been the improper disposal of sensitive documents in the trash, where the document content was later revealed to the public [4]. Thus our service should be able to be popularized and revoke access to information at any time.

Based on these requirements, we come up with the idea to replace sensitive information on physical paper documents with machine readable codes which can only be seen by authorized users. The authentication is stored in a cloud database which can be monitored and modified dynamically. So our proposed service can be divided into three parts.

1. CryptoPaper Generation
2. Key management
3. Information Recovery

In the generation part, we design our add-in for Microsoft Word and use our software to replace sensitive information on physical paper documents with QR codes before they are printed out. The sensitive information is first encrypted through AES algorithm [12] [13] and then transformed into QR code. At the same time the encryption key and access rights are uploaded into the Amazon cloud database which is the second part. For the key management part, it stores encryption key and information about different roles of the organization. We use role based access control(RBAC) [14] to restrict access to authorized users. And for the third part, we develop a scanner application to authorize users, scan QR codes on documents, decrypt and display original information. In our "CryptoPaper" service, cost is very low and revoking access is flexible. All the information is on the paper and we don't need to worry about hack of cloud database. Since we have full control over the cloud database, we can change access of all roles conveniently even if the physical paper document is not in our control. Thus we don't need to worry about the breach of sensitive information even if our document is lost.

1.4 Contribution

In this project "CryptoPaper", I act as the leader and coordinate tasks among team members. My contributions are listed below:

- Information Recovery: I'm primarily responsible for the information recovery part. I design and implement the scanner application and deploy it onto the Intel Atom board [15] provided by Intel [16].

- I participated in the system architecture discussion and draw the figure of system architecture.

- I did research on QR codes and analysed the tradeoff of using different size of QR codes for storing meta information.

- I helped in the design of meta-information for QR code and drew the figure.

- I also help in the creation of Microsoft Word add-in and the design of tables in the Amazon cloud database.

1.5 Organization

The remainder of this paper is organized as follows. Chapter 2 introduces related work, the general system architecture of CryptoPaper and application scenarios. The design and implementation of Scanner application for CryptoPaper is discussed in Chapter 3. The evaluation of our prototype system is presented in Chapter 4. Chapter 5 provides a brief discussion of conclusion and future work before defense.

CHAPTER 2

RELATED WORK

2.1 Literature Review

The paper encryption technology has been advocated in the paper "Paper encryption technology" [17]. In this paper, Fujitsu Laboratories pioneered a paper encryption technology to prevent information leaks while allowing documents to be printed. With this technology, only those who know the password can access information that is encrypted and hidden in printed material.

First, this technology encrypts electronic data or printed material. The electronic data are converted into an image with software called a "virtual printer driver" before being encrypted. And the printed material needs to be scanned with a scanner and converted into image data on a computer. Then, the encryption software encrypts the electronic data or printed material which have been converted into image data. In this encryption process, the user selects the information to be hidden with a mouse or similar device. Then, the user enters a password to start the encryption. The image data that has been encrypted may be converted and saved as electronic data again or printed. The encrypted data can be delivered to the recipient by post, fax or by attaching it to an E-mail. The recipient opens the encrypted data with the software (or scans the printed material with a scanner), and then decrypts it with the software. The data can only be decrypted by those who know the password that was set at the encryption stage. Following this idea, we create our own specific paper encryption technology which is more reasonable and efficient.

2.2 CryptoPaper System Architecture

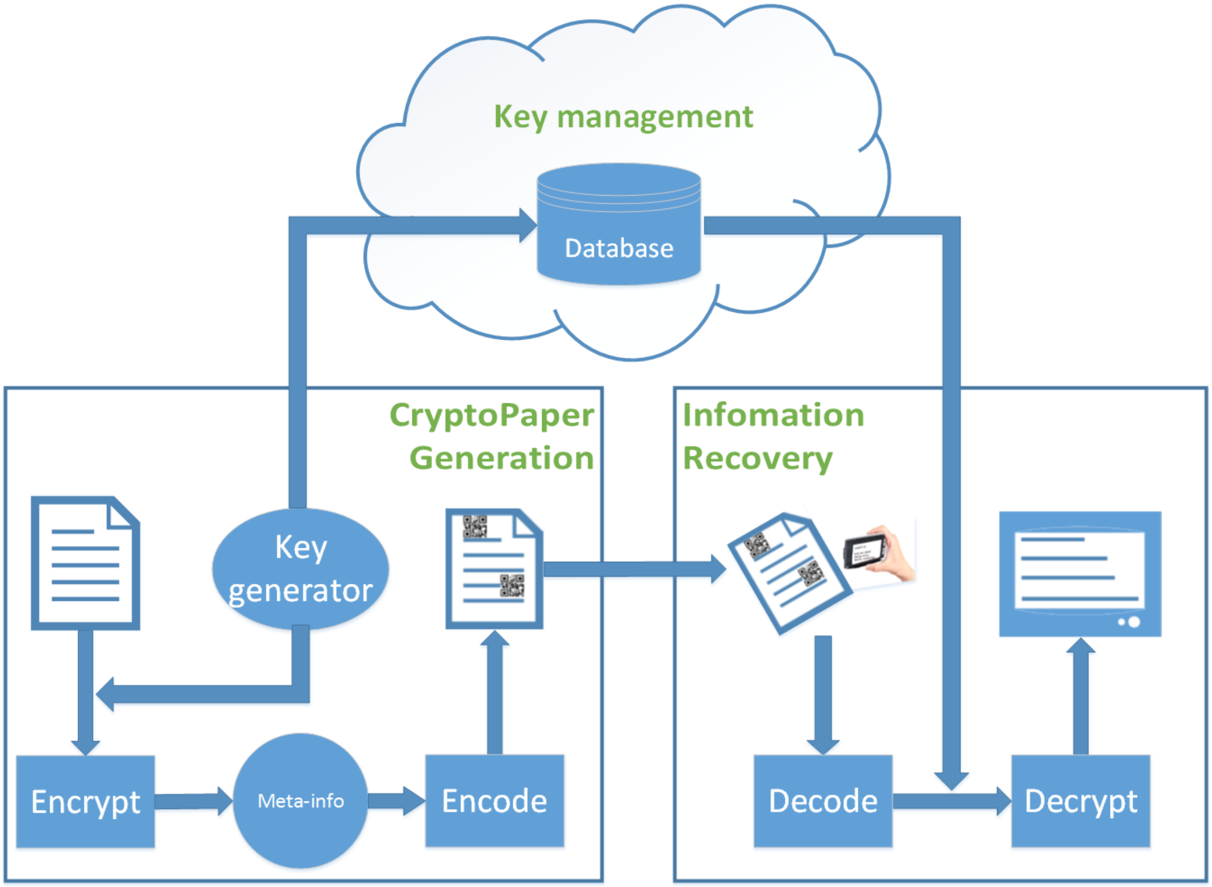


Figure 2.1. CryptoPaper system architecture

CryptoPaper is a service which can help replace sensitive information on paper documents with QR codes which can only be read by authorized users. The general system architecture of CryptoPaper is illustrated in Figure 2.1. There are three major parts in the creation and use of CryptoPaper documents:

1. CryptoPaper generation: 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). The code contains an encrypted version of the data that needs to be protected as well as meta-information that is necessary for decryption. The code is printed on the physical

document instead of the sensitive information and the digital key to access the information is uploaded into the cloud database.

2. Key management: We use a cloud database to store the key and relevant user authority information. 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. What's more, 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. Information recovery: 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. Then we have develop 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).

Next we will discuss the design and implementation of the various components of the CryptoPaper system.

2.3 CryptoPaper Design and Implementation

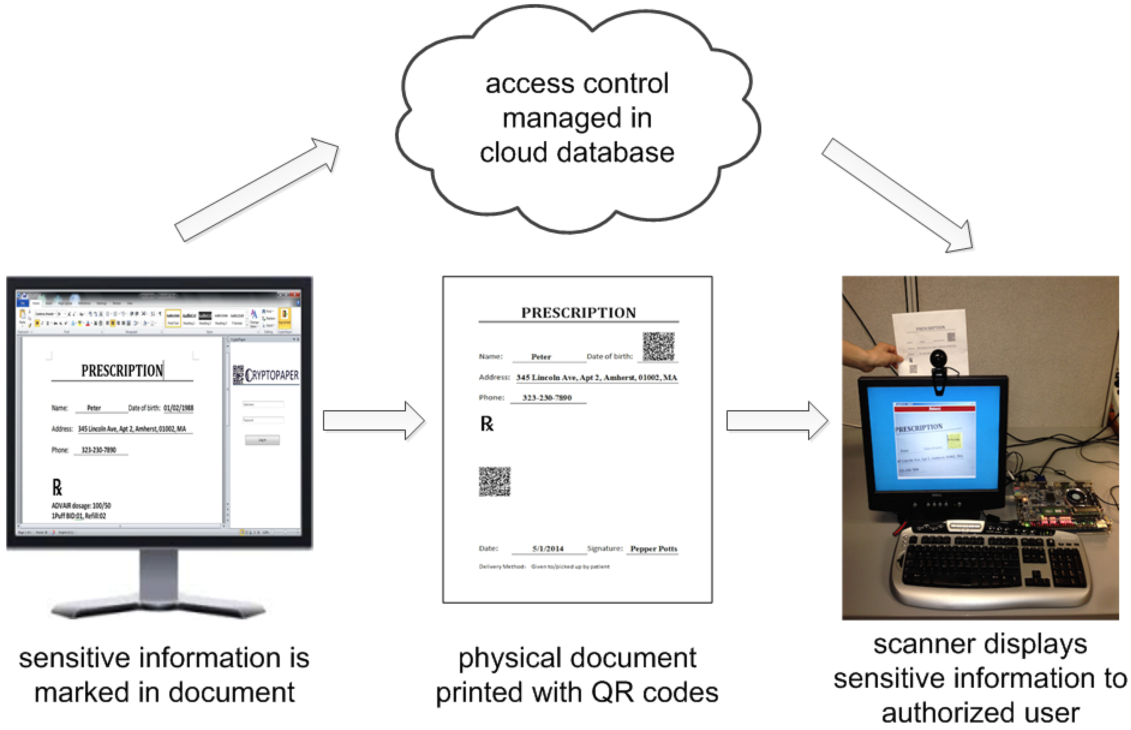

access control managed in cloud database
PRESCRIPTION
CRYPTOPAPER
Name: Peter
Date of birth: 01/02/1988
Address: 345 Lincoln Ave, Apt 2, Amherst, 01002, MA
Phone: 323-230-7890
℞
ADVAIR dosage: 100/50
1Puff BID:01, Refill:02
Date: 5/1/2014
Signature: Pepper Potts
Delivery Method: Given to/picked up by patient
sensitive information is marked in document
physical document printed with QR codes
scanner displays sensitive information to authorized user

Figure 2.2. CryptoPaper system operation

The overall cryptopaper system operation can be seen in Figure 2.2. We develop a Microsoft Word add-in to encrypt and replace the sensitive information with QR codes and upload key to Amazon DynamoDB. For the decryption part, we develop a desktop Java application which can be run on any operating system with JVM installed. This application can communicate with Amazon DynamoDB to authorize users and then open the web cam to decode QR codes and decrypt information. Details are described below.

2.3.1 CryptoPaper Codes

2.3.1.1 QR codes

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. The main advantages are:

- High data capacity
- High speed reading
- High density recording
- High error correction(7% $\sim$ 30%)
- Small printout size
- Various available versions (1 $\sim$ 40)

Based on the advantages above and that QR codes are widely used today, 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 21×21 code and can store 128 bits. When the version increases, the size of the code increases by 4 in each dimension, along with the amount of data that can be stored. For example, version 40 is a 177×177 code that can store 10,208 bits of data (with level L error correction). Example QR codes are shown in Figure 2.3.

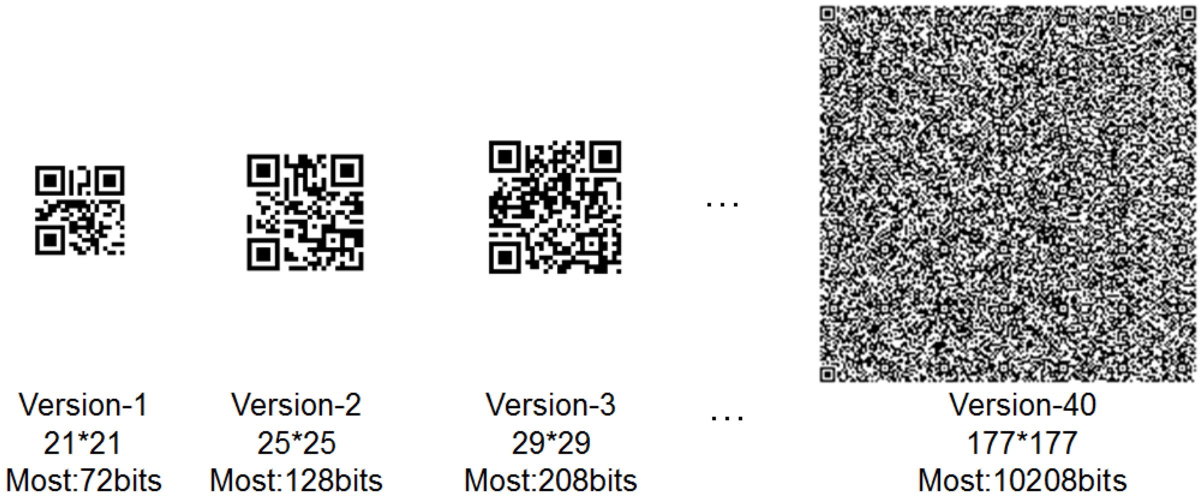


Figure 2.3. Example QR codes

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. The information capacity and size trade off can be seen in Table 2.1. 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 versions, 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.

Data density comparison	Capacity(bits)	Area($inch^2$)	Data density($bits/inch^2$)
Paper document	19400	97.11	200
QR code(version=1)	72	0.4225	170
QR code (version=2)	128	0.5625	228
QR code (version>1)	...	...	>200

Table 2.1. Information capacity and size trade off

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 2.4 shows the maximum bits for each version of QR code. Figure 2.5 shows the number of QR codes that fit into the area of one QR code of version 40. Figure 2.6 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 cannot be read conveniently 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.

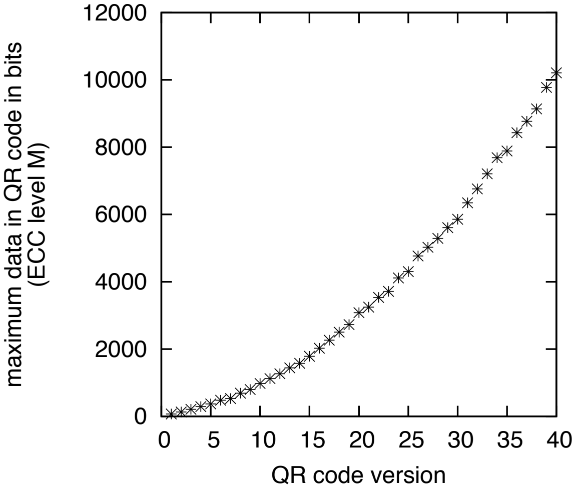


Figure 2.4. Maximum bits for each version of QR code

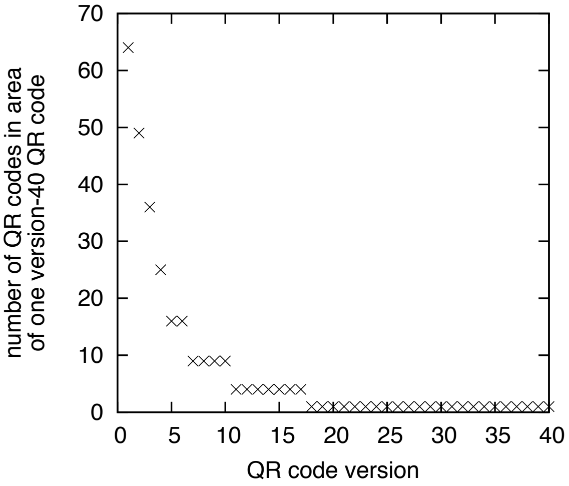


Figure 2.5. Number of QR codes that fit into the area of one QR code of version 40

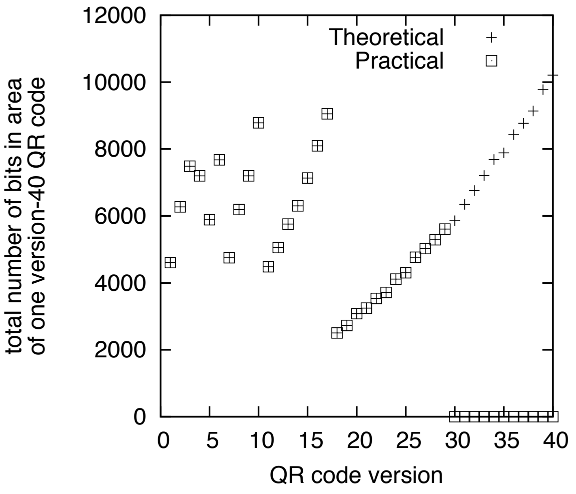


Figure 2.6. Total number of bits that are stored in the area of code version 40

2.3.1.2 Code structure

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.

<table>
<tr><td>CryptoPaper version
(4 bits)</td><td colspan="2">Code header length
(8 bits)</td><td colspan="2">Security attributes
(5 bits)</td><td colspan="2">Total length (TL)
15 bits</td></tr>
<tr><td colspan="7">Identification (Code ID)
(128 bits)</td></tr>
<tr><td colspan="2">Encryption method (EM)
(8 bits)</td><td colspan="4">Organization
(16 bits)</td><td>Cyclic redundancy check for header
(8 bits)</td></tr>
<tr><td colspan="7">Data (Ciphertext)
(variable size)</td></tr>
<tr><td colspan="7">Digital signature
(1024 bits)</td></tr>
<tr><td colspan="4">Cyclic redundancy check for data
(16 bits)</td><td colspan="3">Padding
(variable size)</td></tr>
</table>

Figure 2.7. Meta-information in QR code.

Our QR codes have the structure shown in the Figure 2.7, 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 unique 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 are 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

2.3.1.3 Security Properties

We can achieve the three main security properties for the data stored in the QR code:

- Confidentiality: We achieve confidentiality by encrypting the sensitive data in the document using AES. 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. The key is stored in the cloud database that manages access control.

- Integrity: To ensure integrity of the data stored in the QR code, we use a cyclic redundancy code (CRC).

- Authenticity: To achieve authenticity, we use a digital signature mechanism in our CryptoPaper. We add a digital signature at the end of the QR code. When reading the code, the signature is compared to the originator's public-key hash of the decrypted data.

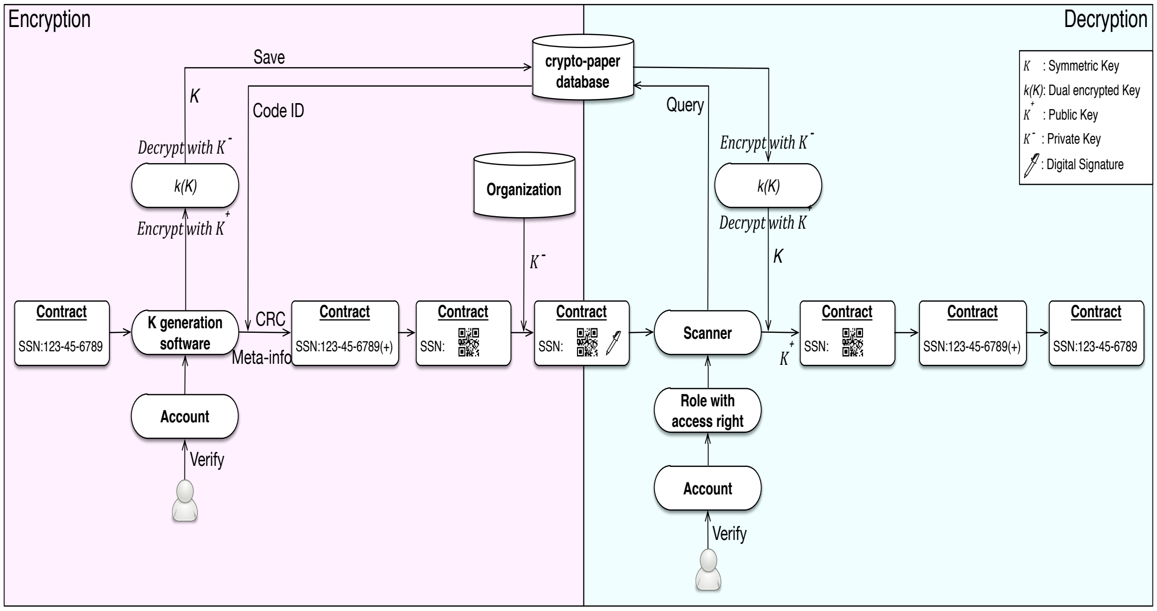
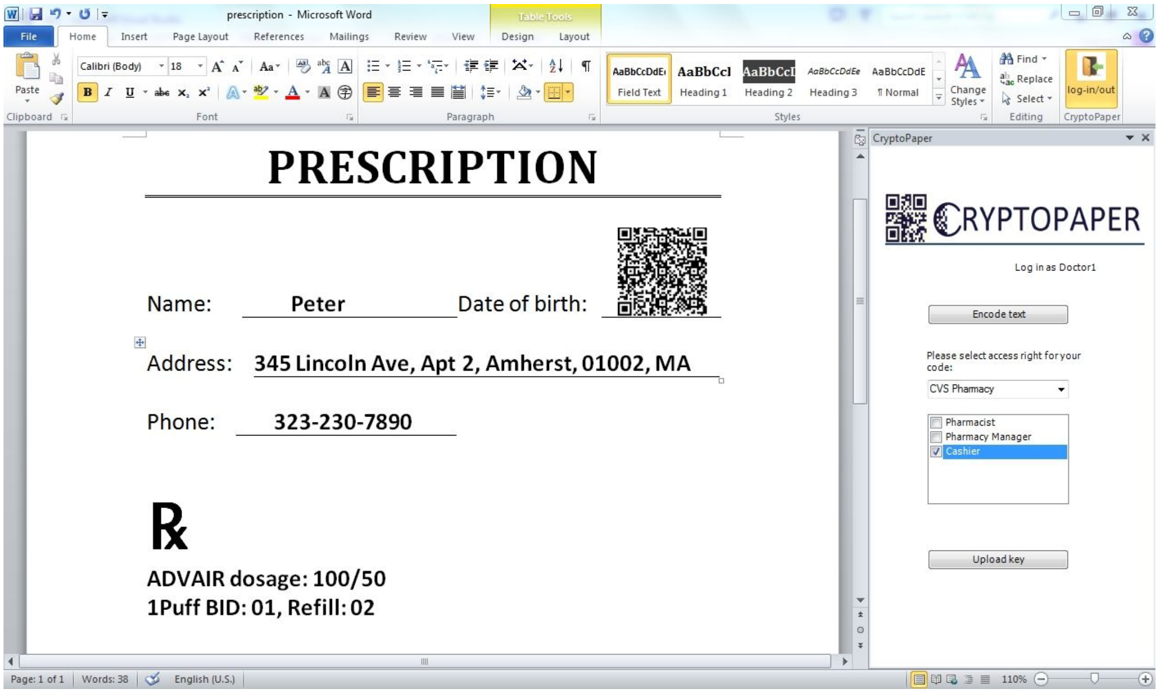


Figure 2.8. Process of creating and validating meta-information in CryptoPaper. The example provides confidentiality, integrity, and authenticity for an SSN on a contract.

The overall process of creating the meta-information used in CryptoPaper is shown in Figure 2.8. The left shows the process that is used during QR code generation. The right shows the process used by the scanner for validation.

2.3.1.4 Generation of Codes



PRESCRIPTION

Name: Peter Date of birth:

Address: 345 Lincoln Ave, Apt 2, Amherst, 01002, MA

Phone: 323-230-7890

℞

ADVAIR dosage: 100/50
1Puff BID: 01, Refill: 02

CryptoPaper

CRYPTOPAPER

Log in as Doctor1

Encode text

Please select access right for your code:

CVS Phamacy

□ Pharmacist
□ Pharmacy Manager
☑ Cashier

Upload key

Figure 2.9. Word plug-in for generating CryptoPaper codes in document.

To generate the QR codes using CryptoPaper in a convenient fashion, we developed a Microsoft Word plug-in (Figure 2.9) that manages the process described above. The plug-in enables users to mark sensitive text that needs to be replaced by a QR code. 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.

2.3.2 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 generation process and the scanning system that attempts to interpret a QR code. The main functions of this database are to implement key management and role-based access control.

2.3.2.1 Key Management

The database 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.

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.

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.

2.3.2.2 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.

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).

2.3.2.3 Database Structure

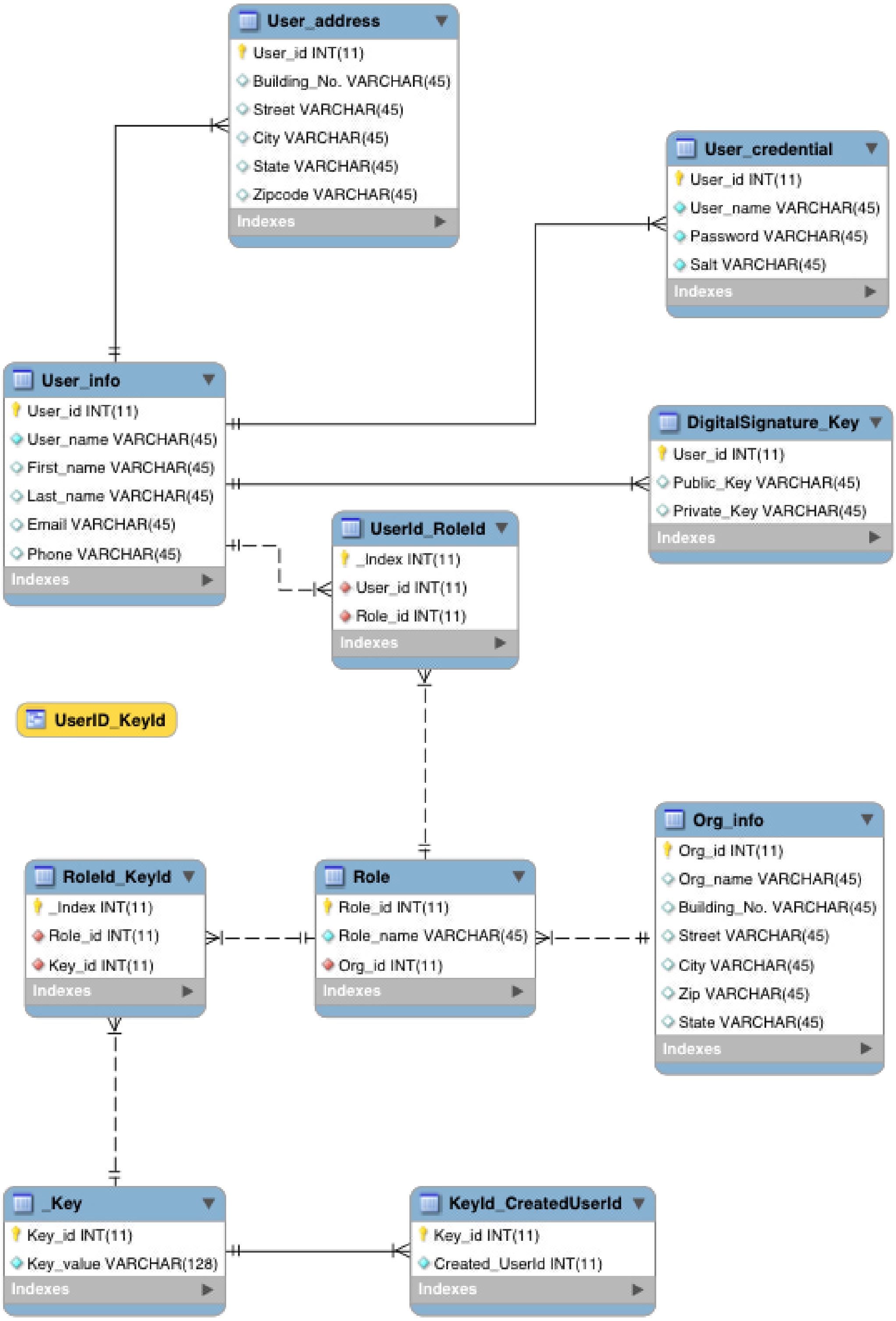


Figure 2.10. CryptoPaper access control database structure.

We have implemented a CryptoPaper database in the scalable cloud database Amazon DynamoDB [18]. Figure 2.10 provides an overview of this database. The tables used in our implementation are:

- The `User_info` table contains basic user information. The `DS_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.

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

- The `Role` table saves every identified role in every identified organization.

- 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.

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.

2.3.3 Document Scanner

The scanner part is what I mainly focus on in this project. I design a scanning application which 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. First I create an desktop Java application and deploy it into the Intel Atom board to achieve this goal. But after that I find it quite inconvenient to carry all the heavy equipments for use. As a result, I create an android application and deploy it into the Android tablets or phones. Thus users can use our products more conveniently to achieve the security goal. Details about the design and implementation can be seen in Chapter 3.

2.4 Application Scenarios

Based on this process and the properties that it achieves, CryptoPaper can be used in a range of application scenarios. To highlight the versatility of the system and the usefulness of the features it provides, we briefly describe a few potential application scenarios:

- Healthcare industry: An example from healthcare is prescriptions. While some prescriptions are handled fully electronically, there are still many that are processed with paper forms. These paper prescriptions have the convenient property that they are easy to understand and do not require the setup of a common information technology system between the doctor's office and the pharmacy. However, paper prescriptions may carry a lot of personal information about the patient. To protect the privacy of the patient, CryptoPaper can be used to ensure that different entities interacting with the paper prescription only have access to the information that is relevant to their task. For example, the clerk in the doctor's office may need to access a patient's name and address to ensure that he or she get the prescription, but not the details of the medicine prescribed. A pharmacist filling the prescription may need to know the details of the medication, but not the address of the patient. Using CryptoPaper, sensitive information (name, address, medication) can be encoded, and users with different roles can be given different access.

- Human resource departments: Human resource departments in organizations have access to a lot of personal information of their employees. These organizations also often use paper documents. To balance the convenience of paper files with protecting employee's private information, CryptoPaper can be used to encode different portions of personnel files. Thus, a clerk filing documents may be able to see the name on a file (which is necessary to determine where to file the document), but not sensitive details (e.g., social security number).

- Student records: An example, where dynamic management of access and authentication is useful, is that of student records, specifically transcripts. A student may apply for a job and thus needs to provide an organization with a transcript. However, once the student has found a job, she may want to ensure that the organizations who did not hire her no longer have access to her transcript. When using CryptoPaper to replace scores with QR codes, the student can dynamically manage access rights and thus revoke access after the interview cycle has completed.

It is important to note that CryptoPaper is particularly useful in cases where *accidental* disclosure of information is a concern. If authorized users *maliciously* try to disclose information (even after later revocation of access) by recording scanner output (e.g., taking photos of decoded QR codes), then CryptoPaper (nor any other practical system) can prevent them from doing that.

CHAPTER 3

SCANNER APPLICATION

The scanner application consists of authorizing user identification, detecting QR codes on physical paper documents, decrypting information inside QR codes and then display cleartext as a substitute for QR code in the video streaming.

3.1 Java Desktop Application

3.1.1 User Authentication

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. For our information recovery part, we create a login interface as part of the application (see Figure 3.1) in Java for user authentication. This Java application has a graphical user interface which provides verification for user identities and allows them to log into the system. Once they login to our system successfully, our application will open the web cam and then they will be able to read the encrypted QR codes through it.

Java uses the Composite Design Pattern to create GUI components that can also serve as containers to hold more GUI components. To create the GUI, we use the "AWT" and "Swing" code libraries in Java. With these two libraries, we implement the graphical user interface for the login system, and use them in the Java application development.

The user is prompted to provide the username and password. Then our application will search the CryptoPaper database for the user credentials and check whether the username and password combination is valid and present in the database. If the user login is successful, the software then opens the webcam and displays another frame to display the content being scanned by the webcam. Next the webcam will check whether there are any QR codes in the content scanned from the document. If there are, it will decode the QR codes and obtain the meta-information from them.

The integrity of the decrypted information is checked by the CRC mechanism and the message is authenticated through the digital signature. Connecting to the CryptoPaper database and we check the access level of the user trying to view the information whether this user is authenticated to read the QR code. If he is authenticated, it will retrieve key from the database and then use it to decrypt the cipher text. After getting the plaintext, it will cover the QR codes in the video with semi-transparent rectangles and display plaintext on these areas. Thus the whole login and detection flow is completed. If the user wants to get back and login with another account, he can use the return button to get back to login interface.

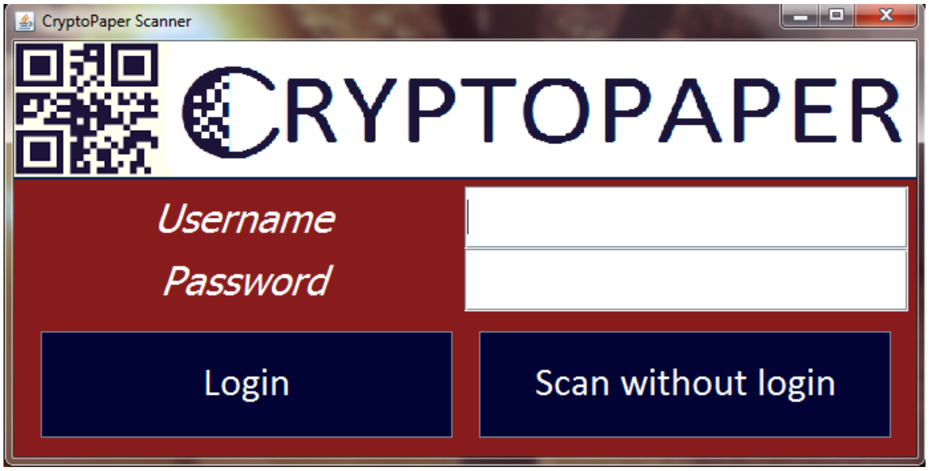


Figure 3.1. Scanner login interface.

3.1.1.1 Password Hashing

The most important aspect of a user account system is how user passwords are protected. User account databases are hacked frequently, so we employ salted password hashing [19] to protect the users' passwords if our Amazon Cloud database is ever breached. The general workflow for account authentication in a hash-based account system is as follows:

1. The user creates an account.
2. Their password is hashed and stored in the database. At no point is the plain-text (unencrypted) password ever written to the hard drive.
3. When the user attempts to login, the hash of the password they entered is checked against the hash of their real password (retrieved from the database).
4. If the hashes match, the user is granted access. If not, the user is told they entered invalid login credentials.
5. Steps 3 and 4 repeat everytime someone tries to login to their account.

In our CryptoPaper system, the accounts for users of a company or an organization are created in advance by administrators of this company or organization. Thus each user owns the username and password from administrator and don't need to create his own account. When using our scanner application, what he only needs to do is typing correct username and password.

3.1.1.2 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. After the user logs into the scanner system (see Figure 3.1), the login system will connect securely over the Internet using SSL to the CryptoPaper database.

3.1.2 QR Code Detection and Processing

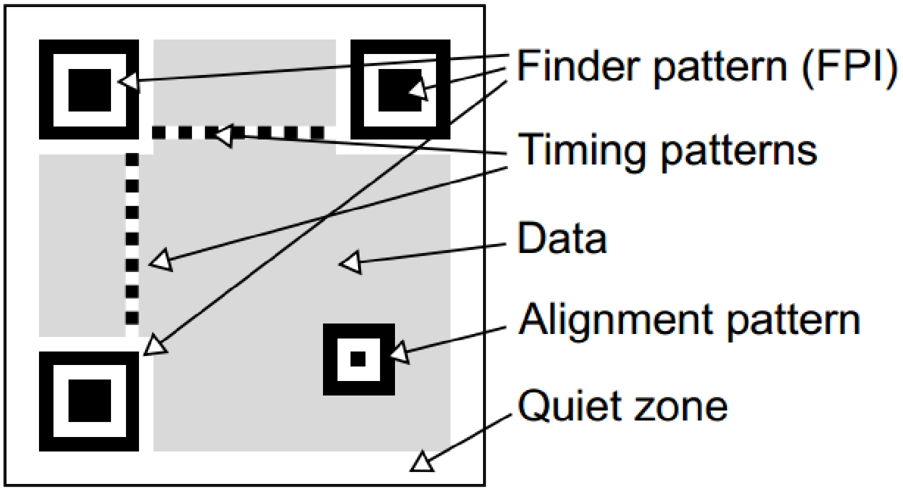


Figure 3.2. QR code finder pattern

The first step in the process is detecting QR codes in the recorded image. Standards for QR codes are described in [20]. Techniques for detecting QR codes in images are described in [21]. Our software for detecting QR codes is based on [22]. The design of the QR code enables easy identification of such codes in images based on the finder and alignment patterns (see Figure 3.2 as described in [21]).

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 [22]. 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 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 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. The steps are:

1. Get coordinates of three FPI $(x_1, y_1), (x_2, y_2), (x_3, y_3)$
2. Calculate center of QR code $(x_c, y_c) = (\frac{x_1+x_3}{2}, \frac{y_1+y_3}{2})$
3. Calculate length of the side of QR code $length = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
4. Define the start point of square which covers QR code $(x_0, y_0) = (x_c - \frac{2 \times length}{3}, y_c - \frac{2 \times length}{3})$
5. Choose side length of square $length' = \frac{4}{3} \times length$
6. Use painter interface in Java to draw yellow square and information on QR codes

3.1.3 Information Recovery

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.

3.2 Android Application

To make our products more portable and popular, I create the Android version of our scanner application to achieve the goal of scanning QR codes, decrypting ciphertext and displaying cleartext.

3.2.1 Problems

One main difference is the connection to Amazon relational database on Android device. Although it seems simpler and faster to do it with JDBC, but this method has lots of disadvantages [23]. In the real world operating environment of phones and portable devices, the unstable connectivity through crowded traffic is a big problem. Result sets sent from the Database to android will consume a lot of bandwidth and battery power. If users frequently use large queries, it can bog down our system quickly and needlessly. And exposing the database directly to the client has security implications. If our app connects directly to the database server I have to hardcode username/password which is very insecure. With some tools an attacker can decompile the apk and can access username/password in this way and can connect to our database with read (+write) access without using our app.

After considering these limitations, I decide to create a web service to act as a middleware which provides a RESTful API for our android app. It will deal with the request from android app, retrieve data from database and then send back the results. The key benefit of a web service is that it has short-lived connections with minimal state, so it's easy to get back to where you were when the device switches WiFi networks, to/from cellular, loses connectivity briefly. It can also provide additional features on top of the JDBC connection like authentication, quality of service, authorization, conditional GET requests and error handling. In addition to these, it can pass through all but the most awful and draconian web proxies.

3.2.2 Architecture

The architecture for the system is shown in Figure 3.3. It mainly consists of three parts: Android app, server and database.

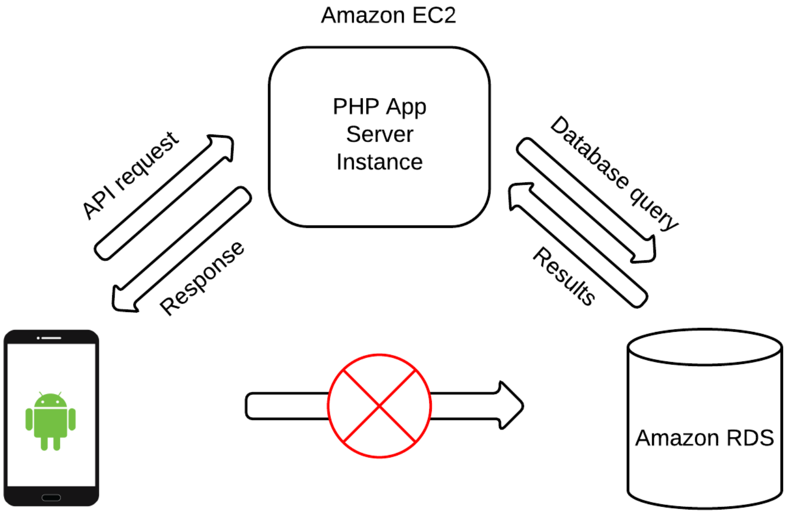


Figure 3.3. Android app system architecture

3.2.2.1 Server

For this server, I use a standard LAMP stack which is deployed to Amazon EC2. The external database is what we used before - Amazon RDS for MySQL. The details for our server are shown below.

- Name: cryptopaperserver-env
- Instance ID: i-55bdf4a4
- Public DNS: ec2-52-0-215-252.compute-1.amazonaws.com
- Instance type: t1.micro
- Availability zone: us-east-1c

3.2.2.2 API

I use PHP to create the API. The link for the API is:

http://cryptopaperserver-env.elasticbeanstalk.com/scripts.php

The PHP file will write and read to the MySQL database based on requests received from the android app. It will also send the results of the requests back to the app. The requests and responses will be in JSON as both parties understand this format. Currently, the PHP file only need to deal with two requests.

- Authenticate: First the user will enter username and password from the Android application. Then it will send request to cloud server and server will send query to database retrieve right hash of password. After comparing the stored hash of password and hash of current password, we will know whether current user is authorized.

- Retrieve decryption key: It will send request to server with current QR code id and user id. Based on this information, server will send query to database and return the decryption key if current user is permitted to see current QR code.

3.2.3 Android App design

This android application mainly consists of two activities. The first one is the "Login" activity for authentication of user's identity and the second one is "Scan" activity for scanning QR codes.

3.2.3.1 Login

The login interface can be seen in Figure 3.4. Users can input their username and password through this interface and then click on the "Sign in" button to login our system.

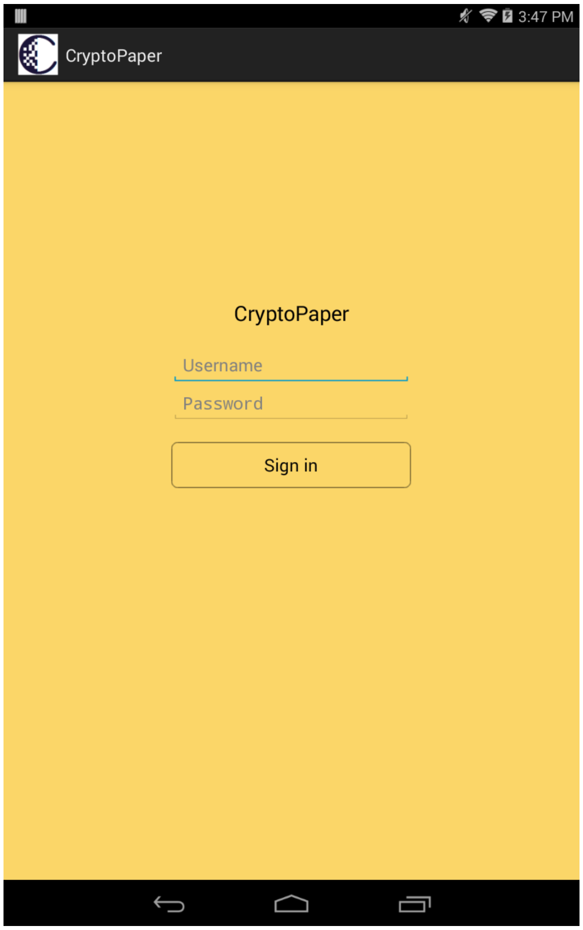
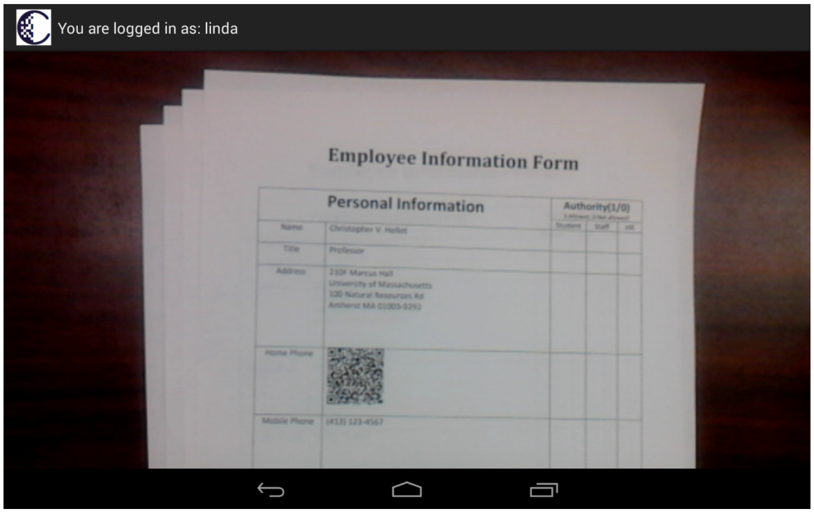


Figure 3.4. Android App login interface

After user has successfully logged into our system, the android app will switch to the second activity “Scan”.

3.2.3.2 Scan

The scan interface can be seen in Figure 3.5. In the “Scan” interface, our app will open the camera on android device and get ready for scanning any QR code which may appear in the camera.



You are logged in as: linda

Employee Information Form

Personal Information		Authority(1/0) 1:Allowed, 0:Not allowed		
		Student	Staff	HR
Name	Christopher V. Hollot			
Title	Professor			
Address	210F Marcus Hall University of Massachusetts 100 Natural Resourses Rd Amherst MA 01003-9292			
Home Phone				
Mobile Phone	(413) 123-4567			

Figure 3.5. Android App scan interface

CHAPTER 4

PERFORMANCE EVALUATION

In this Chapter, we describe the various functionalities implemented in information recovery module and different behaviours under different circumstances. First we will describe the performance for Java desktop application. Then we will evaluate the performance of android application.

4.1 Java Desktop Application

4.1.1 Authentication

We have implemented a prototype system based on the design of CryptoPaper described above. The QR code generation is implemented in a Microsoft Word plug-in using a conventional desktop computer. The current prototype implements the confidentiality and authenticity security properties using a simplified meta-information structure. The access control database is implemented in the Amazon DynamoDB. The scanner is implemented on a Terasic DE2i-150 board, which uses an Intel Atom Processor N2600. The use of such an embedded processor type shows that CryptoPaper can be implemented on lower-performance processing systems (e.g., such as cell phones and tablets).

First the scanner application is 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. The operational prototype is shown in Figure 4.1.

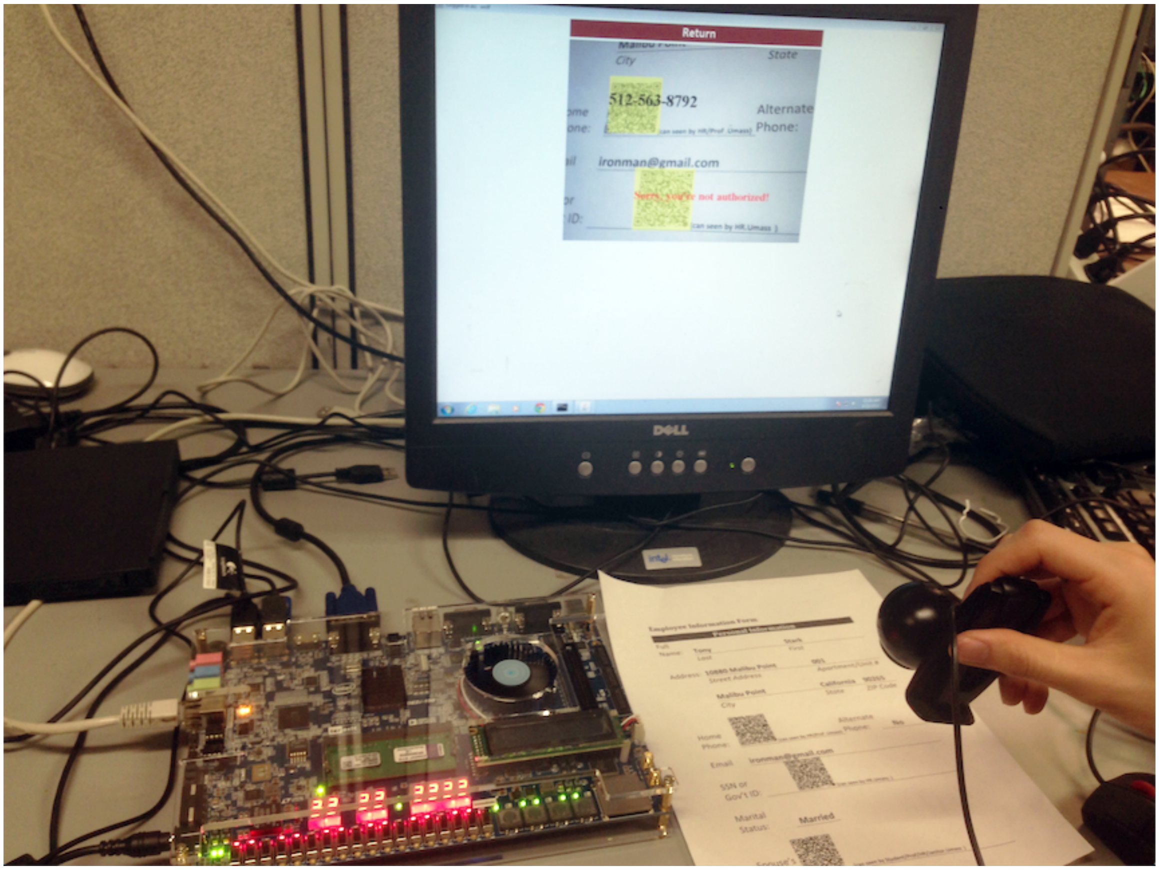
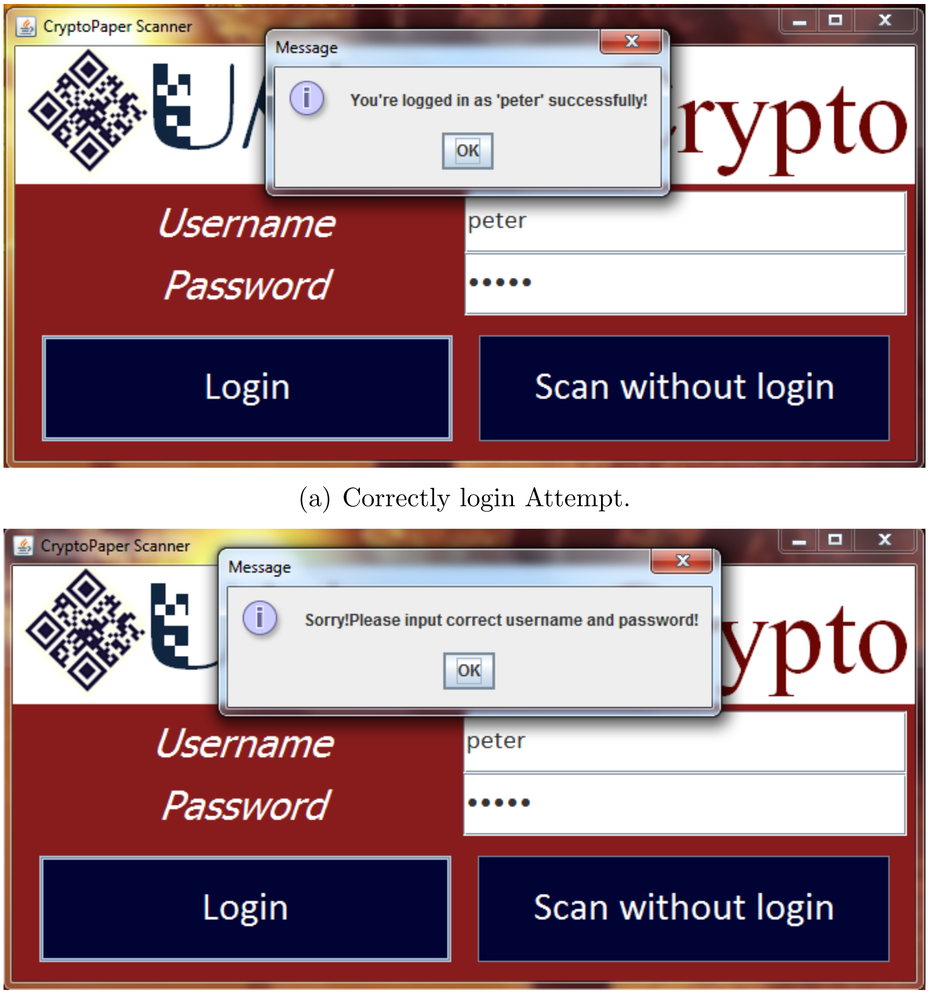


Figure 4.1. CryptoPaper prototype system.

The output of the scanner for both authorized and unauthorized access is shown in Figure 4.2, indicating the correct operation of the prototype implementation.



You're logged in as 'peter' successfully!

Username peter

Password •••••

(a) Correctly login Attempt.

Sorry!Please input correct username and password!

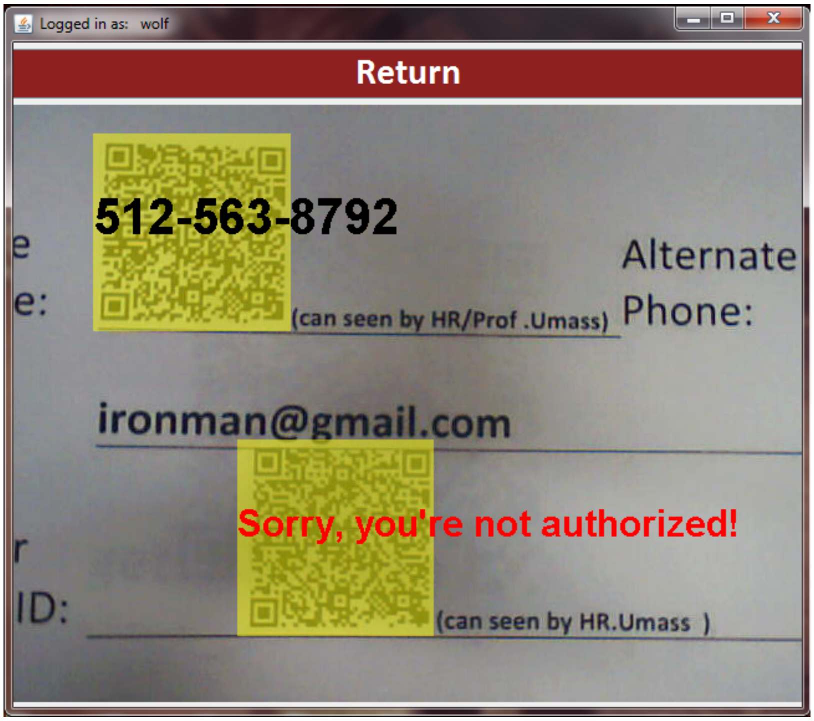
Username peter

Password •••••

(b) Unauthorized login Attempt.

Figure 4.2. Login access approved for authorized users and denied for unauthorized users.

We also test our application for different access levels set by a user. Assume there are two QR codes on the same paper, we define the first code to be accessed by HR or Professor Wolf in organization named UMASS and the second QR code only accessible to the HR who belongs to the organization named UMASS. If the Professor login, then he can see information inside the first QR code(phone number) but won't be able to see the second one(ID). The result is shown in Figure 4.3.



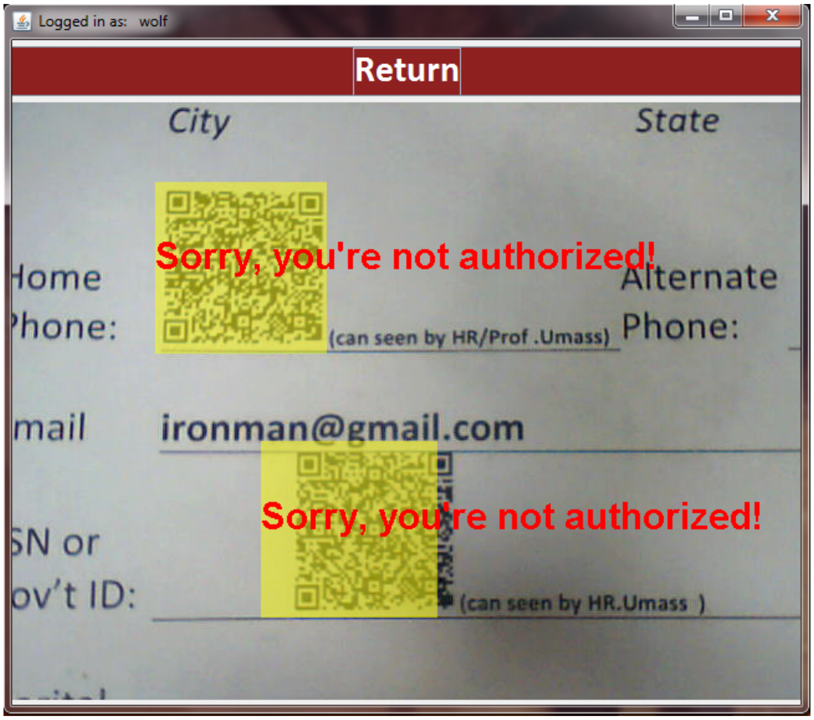
512-563-8792

ironman@gmail.com

Sorry, you're not authorized!

Figure 4.3. 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.

One functionality of our application should be able to be kept synchronized with cloud database dynamically. So we need to check if the updates in a database are reflected in the scanning application. To do this we revoke the access rights of anyone belonging to organization "UMASS" to view the code. For example, assume the user "wolf" who is a professor at "UMASS" is now logging in the system and can see phone number as described in Figure 4.3. If we revoke his access to the phone number, then he should not be able to see the cleartext any more. The expected result was confirmed in Figure 4.4.



Sorry, you're not authorized!

ironman@gmail.com

Sorry, you're not authorized!

Figure 4.4. The user "wolf" who is a professor at UMASS won't be able to see cleartext if his access is revoked.

4.1.2 Time expense

We have also evaluated the performance of our system to provide a context for the complexity of each operation:

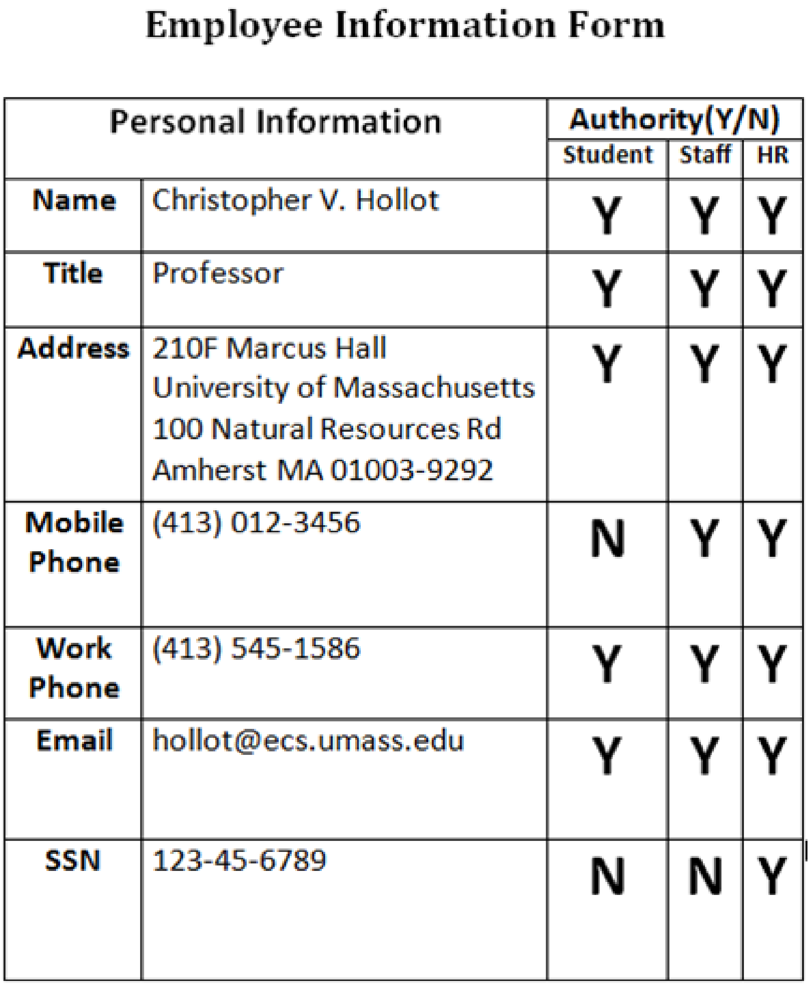
- Time taken to create a CryptoPaper code with 30 lines of data: 3 seconds.
- Time taken for an update in database to reflect in the CryptoPaper generation and recovery modules: 5 seconds.
- Time taken to recover the information on the CryptoPaper by the scanner with 5 QR codes: 8 seconds.

To provide fluid video display of decoded QR codes, the decoded information can be cached on the scanner and decoding operation (which includes a remote database access) is only necessary when a new code is detected. This can help reduce duplicate connections to cloud database and improve decoding speed. Next I will show the performance of Android App.

4.2 Android App

4.2.1 Scenario

We test our android app based on the scenario of an employee information form (Figure 4.5) at ECE Department of UMass Amherst. On the left part of this form is personal information about this employee and on the right side is the authority for each role. “Y” stands for “Allowed” and “N” stands for “Not Allowed”.



Employee Information Form

Personal Information		Authority(Y/N)		
		Student	Staff	HR
Name	Christopher V. Hollot	Y	Y	Y
Title	Professor	Y	Y	Y
Address	210F Marcus Hall University of Massachusetts 100 Natural Resources Rd Amherst MA 01003-9292	Y	Y	Y
Mobile Phone	(413) 012-3456	N	Y	Y
Work Phone	(413) 545-1586	Y	Y	Y
Email	hollot@ecs.umass.edu	Y	Y	Y
SSN	123-45-6789	N	N	Y

Figure 4.5. Part of the Employee Information Form at ECE of UMass Amherst

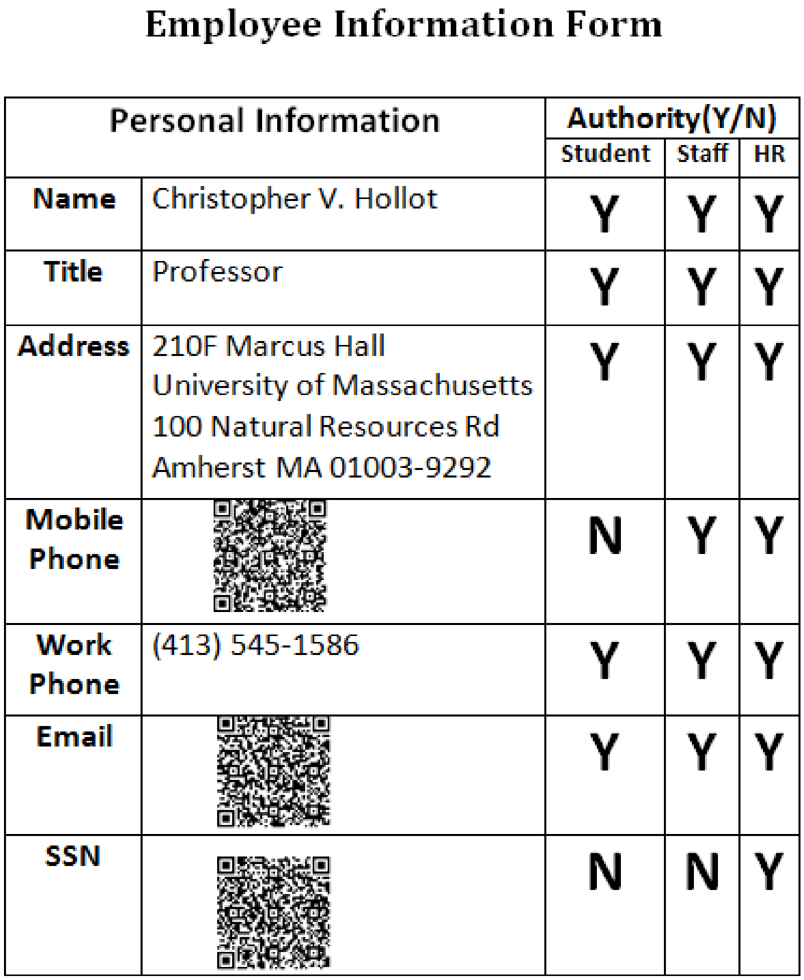
Currently, we set three kinds of roles in our database in the group of UMass.

- Student: Full-time student at UMass.
- Staff: Including professors, administrators, etc.
- HR: Responsible for human resource management.

And we set three users in our system for now. Each has a different role.

- Pengcheng Wang (USERNAME: pengcheng, PASSWORD: pengcheng): Student
- Tilman Wolf (USERNAME: tilman, PASSWORD: tilman): Professor
- Linda Klemyk (USERNAME: linda, PASSWORD: linda): HR

If we set the authority and encode encrypted information as shown in Figure 4.6, then we should expect that mobile phone of Christopher V. Holletcan can only be seen by tilman and linda. Email address of Christopher V. Holletcan can be seen by pengcheng, tilman and linda. And SSN of Christopher V. Holletcan can only be seen by linda. Next we will test whether or not our app will act as expected.



Employee Information Form

Personal Information		Authority(Y/N)		
		Student	Staff	HR
Name	Christopher V. Hollot	Y	Y	Y
Title	Professor	Y	Y	Y
Address	210F Marcus Hall University of Massachusetts 100 Natural Resources Rd Amherst MA 01003-9292	Y	Y	Y
Mobile Phone		N	Y	Y
Work Phone	(413) 545-1586	Y	Y	Y
Email		Y	Y	Y
SSN		N	N	Y

Figure 4.6. Part of the Employee Information Form at ECE of UMass Amherst

The process for scanning QR codes on document can be seen in Figure 4.7.

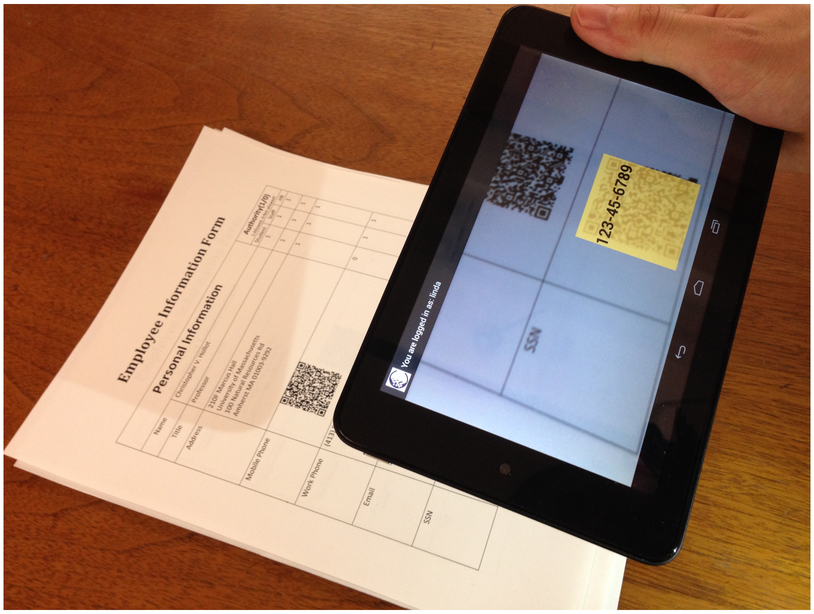
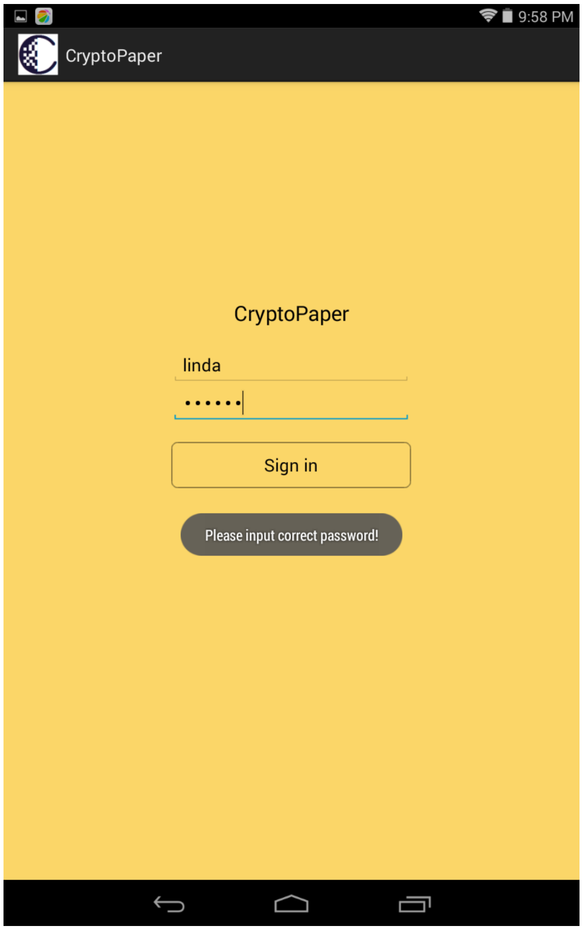


Figure 4.7. Operation of using android device to scann QR codes

4.2.2 Performance

4.2.2.1 Login

To test the authentication, we can use Linda's account to login our system with correct username and wrong password. The login result can be seen in Figure 4.7. Our app will check the combination and remind user of how to correctly login.



linda

••••••

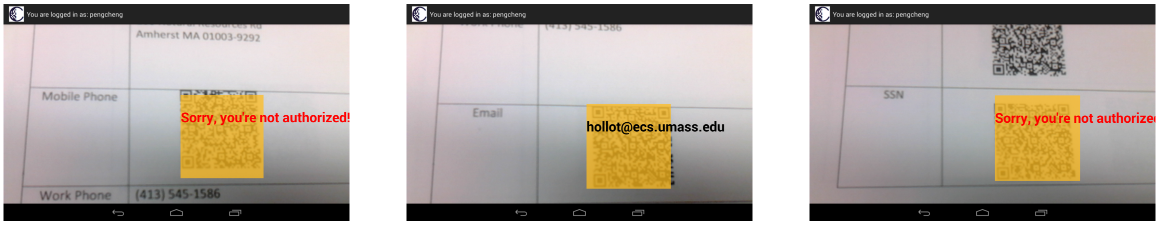
Please input correct password!

Figure 4.8. Result of using incorrect password to login

4.2.2.2 Scan

If we use each of the three users' account to login our system then we will see the scanning results for "Pengcheng Wang" shown in Figure 4.9. The results for "Tilman Wolf" are shown in Figure 4.10. And the results for "Linda Klemyk" are shown in Figure 4.11.

As we can see from the results for different users, our android app works correctly as the authority part of the employee information form.

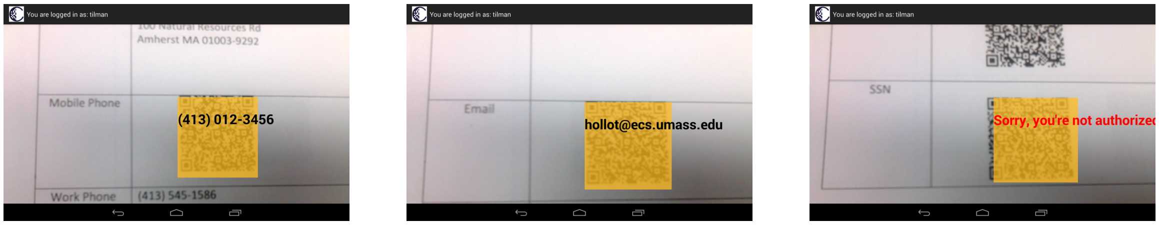


Sorry, you're not authorized!

hollot@ecs.umass.edu

Sorry, you're not authorize

Figure 4.9. Scanning results for user "Pengcheng Wang". From the results we can find that "Pengcheng Wang" is only authorized to see "Email". He can't see information inside the QR code for "Mobile Phone" and "SSN". This is the same as the authority we set for "Student" in Figure 4.6: "N" for "Mobile Phone", "Y" for "Email" and "N" for "SSN".

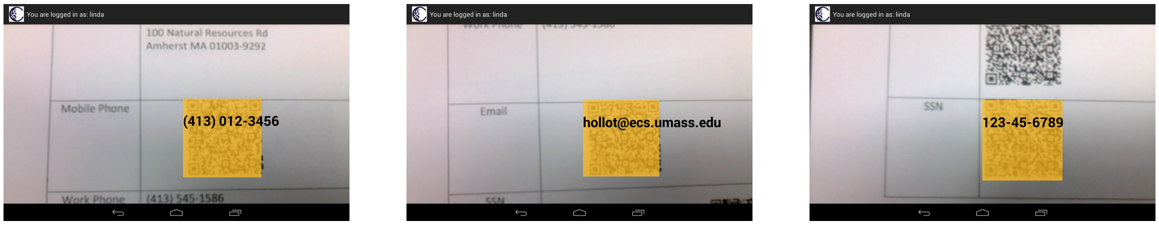


(413) 012-3456

hollot@ecs.umass.edu

Sorry, you're not authorized

Figure 4.10. Scanning results for user "Tilman Wolf". From the results we can find that "Tilman Wolf" is only authorized to see "Mobile Phone" and "Email". He can't see information inside the QR code for "SSN". This is the same as the authority we set for "Staff" in Figure 4.6: "Y" for "Mobile Phone", "Y" for "Email" and "N" for "SSN".



(413) 012-3456

hollot@ecs.umass.edu

123-45-6789

Figure 4.11. Scanning results for user "Linda Klemyk". From the results we can find that "Linda Klemyk" is authorized to see all the information. This is the same as the authority we set for "HR" in Figure 4.6: "Y" for "Mobile Phone", "Y" for "Email" and "Y" for "SSN".

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Managing sensitive information on physical documents poses a considerable challenge in business environments. We present the design, implementation, and prototype evaluation of a system called CryptoPaper, which replaces sensitive information with 2-dimensional QR codes that can only be read by authorized users.

Based on the scanning application designed in Chapter 3, we can achieve the goal that sensitive data on paper documents be read only by authorized users through a scanner. Our system can also provide access revocation on these codes while still providing the convenience of handling conventional paper.

5.2 Future work

As far as now, the android application can only be used for scanning QR codes and recover information. If it can be used to do more, it would certainly act much more impressively. I'm considering adding more functions shown below to our android app and make it more popular.

- Creat new account: Now new accounts can only be created by our administrator through directly adding credentials to cloud database which is not efficient. If I can make "Creat new account" available on android device, it will be more reasonable and acceptable.

- Forget password: Since people can't avoid forgetting password some day, we need the "Forget password" option on android device to help people finding existing password or setting new password.

- History: This can help users find information of previously scanned QR codes.

One more possible improvement to our system is that we can create an website to help people manage accounts and manage permissions for previous QR codes. Surely, there are still a lot which can be improved, hope our "Cryptopaper" service can help users improve the security of physical paper documents and live a more secure life!

BIBLIOGRAPHY

[1] TAPPI. Paper industry statistics. http://www.tappi.org/paperu/all_about_paper/faq.htm.

[2] Wikipedia. Physical information security. http://en.wikipedia.org/wiki/Physical_information_security, 2015.

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

[4] Ben Rothke. Why information must be destroyed. http://www.csoonline.com/article/2123705/privacy/why-information-must-be-destroyed.html, 2009.

[5] Wikipedia. 2014 celebrity photo hack. http://en.wikipedia.org/wiki/2014_celebrity_photo_hack, 2015.

[6] K. Jain. Increased-security identification card system, February 8 1994. US Patent 5,284,364.

[7] Wikipedia. Filing cabinet. http://en.wikipedia.org/wiki/Filing_cabinet, 2015.

[8] Wikipedia. Paper shredder. http://en.wikipedia.org/wiki/Paper_shredder, 2015.

[9] Shredit. Why shred? http://www.shredit.com/en-us/why-shred, 2014.

[10] DARPA. Darpas shredder challenge solved. http://www.darpa.mil/NewsEvents/Releases/2011/12/02_.aspx, 2011.

[11] Tom Geller. Darpa shredder challenge solved. *Commun. ACM*, 55(8):16–17, August 2012.

[12] Announcing the advanced encryption standard (aes). *Federal Information Processing Standards Publication 197. United States National Institute of Standards and Technology (NIST).*, November 26, 2001. Retrieved October 2, 2012.

[13] Wikipedia. Advanced encryption standard. http://en.wikipedia.org/wiki/Advanced_Encryption_Standard, 2015.

[14] Wikipedia. Role-based access control. http://en.wikipedia.org/wiki/Role-based_access_control, 2015.

[15] Intel atom board. http://www.systemseng.cornell.edu/se/intel/team/upload/CornellCupRecommendedBoardTipsResources.pdf, 2014.

[16] Intel-cornell cup. http://www.systemseng.cornell.edu/intel/, 2014.

[17] Taizo Anan, Kensuke Kuraki, and Jun Takahashi. Paper encryption technology. *FUJITSU Sci. Tech. J.*, 46:87–94, 2010.

[18] D. Giuseppe, H. Deniz, J. Madan, K. Gunavardhan, L. Avinash, P. Alex, S. Swaminathan, V. Peter, and V. Werner. Dynamo: Amazon's highly available key-value store. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles–SOSP'07*, pages 1–8, 2007.

[19] Defuse Security. Salted password hashing - doing it right. https://crackstation.net/hashing-security.htm.

[20] INTERNATIONAL STANDARD ISO/IEC 18004. *Information technology-Automatic identification and data capture techniques-QR Code 2005 bar code symbology specification*, 2005.

[21] István Szentandrási, Adam Herout, and Markéta Dubská. Fast detection and recognition of qr codes in high-resolution images. In *Proceedings of 28th Spring conference on Computer Graphics.ACM*, pages 1–8. Comenius University in Bratislava, 2012.

[22] S. Owen. Zxing. https://github.com/zxing/zxing, 2013.

[23] Jdbc vs web service for android. http://stackoverflow.com/questions/15853367/jdbc-vs-web-service-for-android, 2013.