Addis Ababa University
College of Natural Sciences

A Framework for Verifying Paper Based Document Using Multiple QR Codes

Wondmagegn Abriham

A Thesis Submitted to the Department of Computer Science
in
Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia
30 (September 2019)

Addis Ababa University
College of Natural Sciences

Wondmagegn Abriham

Advisor: *Dagmawi Lemma (PhD)*

This is to certify that the thesis prepared by *Wondmagegn Abriham*, titled: *A Framework for Verifying Paper Based Document Using Multiple QR Codes* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name Signature Date

Advisor: _____

Examiner: _____

Examiner: _____

Abstract

With the growth in popularity of two dimensional (2D) barcodes such as Quick Response (QR) code, automatic verification of printed documents has become possible. 2D barcodes are types of barcode that can store data in two dimension. They have very large storage capacity compared to their one dimensional (1D) counterparts.

In order to verify paper based document using a digital signature, the digital signature must be stored in 2D barcodes like QR code and printed on a document. This digital signature for paper based documents, which we called paper based digital signature (PBDS) in this thesis, not only holds the digital signature, but also a copy of the message from the document. Although QR code has large storage capacity, its storage capacity has a limit. This creates a problem as the size of the document to be signed increases, even with the help of compression algorithms to reduce the size of data to be stored in QR codes.

In this thesis, a framework is developed to make the signing and verification of printed documents automatically verifiable. In the framework, two PBDS architectures are designed, BBDS-A and PBDS-B. These architectures use multiple QR codes for a single PBDS. PBDS-B is an architecture primarily designed to solve the problem of not being able to sign and verify paper based document with larger number of characters than QR code storage capacity. However, this architecture has one problem. The number of QR codes in a PBDS can be too many to track and challenging to verify documents. PBDS-A is also a multi-QR code PBDS designed to solve the problem observed in PBDS-B. In PBDS-A, the number of QR codes used is much smaller than PBDS-B.

Design science research approach is used to conduct the thesis. An iterative framework development was followed between problem identification, solution design, evaluation and literature review to conduct the thesis.

A test is conducted on 15 documents, five of which have less than 1000 characters, another five documents have number of characters between 1000 and 5000 characters, and the final five document have number of characters between 5000 and 10000. The evaluation confirms that multi-QR code PBDS have better capability of verifying large documents. In addition, the test shows that PBDS-A is more accurate during verification and much more easily verifiable than PBDS-B.

Keywords: QR Code, 2D Barcode, Digital Signature, Paper Based Document

Acknowledgements

In life there are times of strength when we work through our problems and challenges with determination, and there are times of weakness when everything gets dark and we don't see a way out. Even though those times seem the end, we pass through them with the help of the almighty. I have been through times of strength and weakness, I have been through times of bright days and dark nights. What I understood after passing through those dark nights is that God carried me all the way. Even when I turned my back on him, he was with me. Thank you God.

Thirty years, thirty freaking years. They taught me, fade me, clothed me, guided me and paid my tuition fees and they still think I'm their five year old child, they still protect me from the cruelty of this world. Who does that? Parents of course. Mr. Abraham Chosha and Mrs. Worknesh Siyoum have been my corner stones all my life. I love you mom and dad, this achievement is not mine alone.

I know we haven't been the loveliest of siblings in the world. We have been through multiple rough patches. What is important at the end of the day is the fact that we don't let the sun go down on our disagreements. Lilina Abraham, your generous heart always reminds me to be generous. Dagmawit Abraham, your strict manners remind me to be well organized. Bishaw Abraham, your brave nature makes me want to be brave in life. Bereket Abraham, as a last child of the house you have taken all the good qualities of all your siblings. You are smarter than me, you are as generous as Lili, you are as well-mannered as Dagu, and you are as brave as Abi. I couldn't have gotten this far without you as my siblings.

You have been and are the best of friends I ever had. When I was happy you were truly happy for me. You have been my conscience in most of what I do. Thanks, Defaru Tefera.

You tolerated my nagging questions and answered them as true Advisors do. You have guided me through this thesis. I couldn't even have known where to start the thesis without your guidance. You even went beyond what is expected of you and helped me with my job. Thanks, Dr. Dagmawi Lemma.

Last but not least, you have instructed and guided me in the three years I have stayed in the University. You have carved a scholar out of me. Thank you all my instructors, I couldn't have done it without you.

Table of Contents

List of Tables

List of Figures

List of Acronyms/Abbreviations

1D	One Dimensional
2D	Two Dimensional
AES	Advanced Encryption Standard
CFHF	Collusion Free Hash Function
DD	Documents Dispatcher Module
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
FPT	Full Plaintext
GUI	Graphical User Interface
HC2D Barcode	High Capacity Two Dimensional Barcode
JCA	Java Cryptographic Architecture
NS	Neutral Subsystem
OCR	Optical Character Recognition
PBDS	Paper Based Digital Signature
PBDS-A	Paper Based Digital Signature Type A
PBDS-B	Paper Based Digital Signature Type B
PDSSF	Printed Digital Signature System Framework
PPT	Partial Plaintext
RSA	Rivest, Shamir, Adelman
SA	System Authorization Module

SC	Signature Collection Module
SGS	Signature Generator Subsystem
SHA	Secure Hash Algorithm
SS	Single Signature Module
SVS	Signature Verification Subsystem
URL	Uniform Resource Locator

Chapter One: Introduction

We understand signatures as a handwritten descriptions of our names or nicknames used to verify documents signed, owned, written or approved by the signatory [23]. They provide signatory authentication, where the verifier of the information makes sure whether the signatory is the right person. The problem with handwritten signatures is that they lack the full authentication, integrity and non-repudiation that digital signatures provide [24]. There is no easy mechanism to make sure that a document is the right document [26]. Handwritten signature verification usually requires a specialist and special equipment such as UV lamp, infrared inspector and magnifying glass [2]. There is no reliable technique to detect tampering with the text in the document or to prevent the signatory from denying that he/she signed the document.

Paper based documents are still being signed with hand and pen. In such documents, a traditional document consists a piece of paper and a text. In an environment where there are no advanced signing and verification tools, handwritten signature can be inspected visually for authenticity. In the case of the link between the signature and the content of the document, there is an indirect and weak link, which weakens the effectiveness of handwritten signatures. The connection is through the medium (paper). It is difficult to detect changes that occur after signing the document, which is a weak proof of integrity of the information being verified [24]. Handwritten signatures can easily be forged in this age of technology. With the invention of digital printers and scanners these days, it is very easy to create a counterfeit document. Further, it takes time to verify handwritten signature [2].

On the other hand, digital signatures provide authentication, integrity and non-repudiation to documents, but they are limited to electronic documents. Hence, it is challenging to use digital signatures on printed documents.

In recent years, there are few of research works conducted addressing the issue of using digital signatures on paper based documents [2, 3, 5]. These researches use 2D barcode to store digital signatures for easy access when they are needed for document verification. One of the most popular 2D barcodes is QR code.

QR code is a technology primarily introduced in the automotive industry in order to track vehicles and parts [6, 7]. The technology spread from there to the food, pharmaceutical and contact lens companies to trace products. Digital signature on the other hand is a combined technology of different cryptographic algorithms used to associate an information/document with the sender [8, 9]. In information exchange between two parties, digital signature assures that an information/document received is from the right person and it is not changed in transit. Even though digital signature is one of the most secure ways to authenticate digital documents [11], the technology is limited to digital or electronic documents.

Documents such as passports, national exam certificates, and university degrees need a high level of security. With the abundant of technologies, those facilitate various document editing tools, the need for a technology that prevents counterfeiting is growing. Some documents such as passports have multiple security features that make them difficult to forge [12, 13]. However, they are also very difficult to verify by an average person [2] and in some cases without the support of technology. This difficulty of verifying documents by an average person encourages the use of counterfeit documents. Therefore, achieving authenticity, integrity and non-repudiation by using digital signatures on paper based documents is one way to go to create documents that are difficult to fake and easily verifiable by anyone.

1.1 Motivation

The growth in information technology in the 21st century has created a revolution in many industries. One of these industries is the printing industry. With the introduction of digital printers and scanner, the quality of printed documents has increased. These days, it is difficult for an average person to differentiate between an original and counterfeit document [2]. As a result document forgery is becoming a huge problem all over the world. Because of document forgery unqualified individuals control and manage businesses. Document forgery affects product quality, economy, and politics and so on directly or indirectly. So finding a solution to this problem will benefit everyone.

In this time change, everything is being digitized. Paperless office is one of the technological improvements expected in the near future [3]. Every document is created, signed, sent, received and stored digitally, making printing documents obsolete. This may explain shortage of research on paper based digital signatures. However, we are not there yet. We are still using

paper based documents for almost every legally binding tasks. On the contrary, digital printers and scanner that make the ability to counterfeit documents are improving with time. The need for research on easy and reliable verification method for paper based documents is growing.

1.2 Statement of the Problem

After the introduction of QR codes for authorization of printed documents by *Lee et al.* [25] in 2003 research in the area resurrected after a decade in 2012 by Warasart and Kuacharoen [2]. They introduced a model for using digital signatures to verify paper based documents, which in this thesis is referred to as PBDS.

The model [2] uses a QR code to store the digital signature and a copy of the text in the document being verified. It follows some fixed steps to generate a QR code and to verify whether the document is authentic using the QR code. The model is efficient, effective, and secure during both generation and verification of paper based documents. However, the model has two weaknesses. First, it is not designed for documents with large texts, which is one of the problems this thesis is addressing. A digital signature program developed based on the model will not be able to generate PBDSs for documents with more than approximately 4,296 characters for alphanumeric character encoding [2, 3]. One of the most important techniques that improves the storage capacity of QR code is the use of data compression algorithms to compress the data being encoded into the QR code [2, 27]. Even using data compression algorithms, it is challenging to sign documents such as corporate contracts with tens to hundreds of pages documents. Lossless data compression algorithms, which are used to compress text, have the capability to compress data down to 50 percent of the original size [28]. Although reducing the size of a document to half its original size seems much, the maximum storage capacity of QR codes is fixed and unchanging. On the other hand, there is no size limit for a document. Considering these two facts a QR code with fixed storage capacity will not be able to sign larger documents. In addition, QR code storage capacity varies depending on the version (the size) of QR code used to store the digital signature. So the length and width of the QR code also affects the number characters stored in the QR code.

Second, according to [2], because of unreliable nature of Optical Character Recognition (OCR) software, it is difficult to be sure that a verification fails because of tampering in the document or because of OCR translation error. So manual verification is needed when

verification fails to be sure that the verification failure is not because of OCR translation error. Manual verification involves manually comparing the text in the document with the text in the QR code. Manual verification of documents takes time depending on the size of text in a document. If a document has thousands of text in a document, the person involved in the verification process needs to read and compare each text in the document with the text in a QR code, which is very challenging. Therefore, a better solution is needed to address the problem observed.

Many research works in the area used the same model discussed above conducted their research works [2, 3, 5]. Since multiple researches used this model without modification, we can conclude that any problem that exists within the model would be inherited by all the researches that used this model. This thesis will address these problems observed in the model and try to resolve them.

1.3 Objectives

General Objective

Providing a framework that can be used to guide the development of a system in generating PBDS and automated paper based document verification.

Specific Objectives

The objectives that add up to the achievement of the general objective are as follows:

- ✓ Refining and stating the problem by reviewing various literatures
- ✓ Analyzing the problem by using logical reasoning
- ✓ Identifying required components/models, refining the models, and developing framework to address the problem
- ✓ Developing a prototype using the framework as guide
- ✓ Evaluating the prototype by testing it and using iterative evaluation

1.4 Methods

This research follows a design science research approach. The reason design science is research approach is chosen in this thesis is because this thesis intends to propose a solution

and repeatedly improve the solution through iterative process. The research will pass through multiple iterations of problem identification, analysis, component/model identification, component specification, prototype development and evaluation. Each iteration will raise a new question and uncover new problems that need to be solved in the following iterations.

After multiple iterations of the framework development, a working framework for paper based digital signature will be developed as an artifact.

Review of literatures such as books, journals and other publications will be used throughout the research process to better understand the problem the research is trying to solve, to explore better methods, techniques and tools that can help achieve the research goals.

Tools

Since the research approach being followed is design science, there is a need for a prototype on which theories and ideas must be tested while developing the right framework. The prototype will be developed using Java programming language and JavaFX is used to create a graphical user interface (GUI). JavaFX is an XML based GUI creator for Java programing language. The text extraction functionality of the program is handled using Apache Poor Obfuscation Implementation (POI) library. Apache POI library is a huge library with a lot of functionalities, a small part being text and image extraction and adding from/to word document. QR code generation and scanning is handled using Zebra Crossing (ZXing) library. The private /public key and certificate generation are provided by KeyStore Explorer. The generation of the private /public key is according to X.509 standard of the certificate authority. A 2048 bit RSA encryption will be used to generate the private/ public key pairs.

An Android barcode scanning app will be developed to perform QR code scanning. Before scanning the QR codes, the mobile phone holding the app will be connected with a computer using Wi-Fi hotspot. When a QR code is scanned, the data will be sent to the prototype program as soon as it is scanned.

1.5 Scope and Limitations

This thesis will focus on developing a framework that will primarily help the verification of paper based documents using PBDSs. Although finding a better way to verify paper based documents is the focus of this thesis, it will not be limited to verification methods. PBDS

signing and verification techniques cannot be seen separately. Studying ways to improve verification methods of PBDSs improves signing techniques of PBDSs as well.

The overall framework we will develop will have different component, PBDS models, and PBDS signing and verification techniques.

The components that will be used in this thesis are part of one of the following specific issues:

- ✓ Digital signatures, specifically on public key cryptography based digital signatures
- ✓ Public key cryptography such as RSA and digital certificates
- ✓ Data compression algorithms
- ✓ Cryptographic hash functions
- ✓ QR codes, theirs storage capacities and encoding methods

1.6 Application of Results

The use of QR code in digital signatures broadens its use, which was limited to digital documents, to paper based documents which are common in developing countries like Ethiopia. Formal documents such as large corporate contracts, deeds, legal documents, college degrees, national exam certificates, passports, business letters, personal letters and notices can be signed with PBDS. They can be applied in governmental, non-governmental or for personal use to verify information/ document received.

1.7 Organization of the Rest of the Thesis

The rest of the thesis is organized as follows. Chapter Two, which is Literature Review, provides a background for paper based digital signature. It discusses the various components involved in signing and verification PBDS, and the process involved in the signing and verification processes. Chapter Three, Related Work, discusses researches conducted in the area of PBDSs and the gaps identified. Chapter Four, shows the proposed solutions of the thesis. It discusses the artifact designed, which is the PBDS framework. Chapter five evaluates the solution provided in chapter four. The final chapter, chapter six, discusses recommendations and future works that will follow the thesis.

Chapter Two: Literature Review

PBDS is a result of various technologies. Understanding the characteristics, strengths, weaknesses and the inner working principles of these technologies will make the signing and verification of paper based digital signatures easier in addition to contributing to new findings in the area of study. This chapter discusses those components and how they work together in the signing and verification processes of PBDS. In addition it provides a general explanation on areas related to the research being conducted.

2.1 Paper Based Digital Signature (PBDS)

In simple words, paper based digital signature is a digital signature printed on paper based document using 2D barcodes [2, 3, 5]. Additional tools are used in PBDS beyond digital signature to make the verification process easier. PBDS is a technology that combines existing technologies such as OCR, digital signatures and 2D barcodes to make signing and verification process of digital signatures on paper documents possible. If these technologies are well used, the verification process would be easier and the advantages of a digital signature such as authentication, integrity and non-repudiation for paper based documents [2], can be exploited.

2.2 Paper Based Document Signing and Verification Processes

There are certain steps/ processes to follow in order to sign and verify PBDSs. These processes are discussed as follows.

According to [2], the signing and verification process of PBDS involves the following steps as depicted in Figure 2.1:

1. Extract text from a document
2. Convert the message/ plaintext from the document into a hash value.
3. Encrypt the hash value with the signatory's private key. The output is a digital signature.
4. Combine a copy the message with the digital signature and compress.
5. Generate QR code from the compressed data.

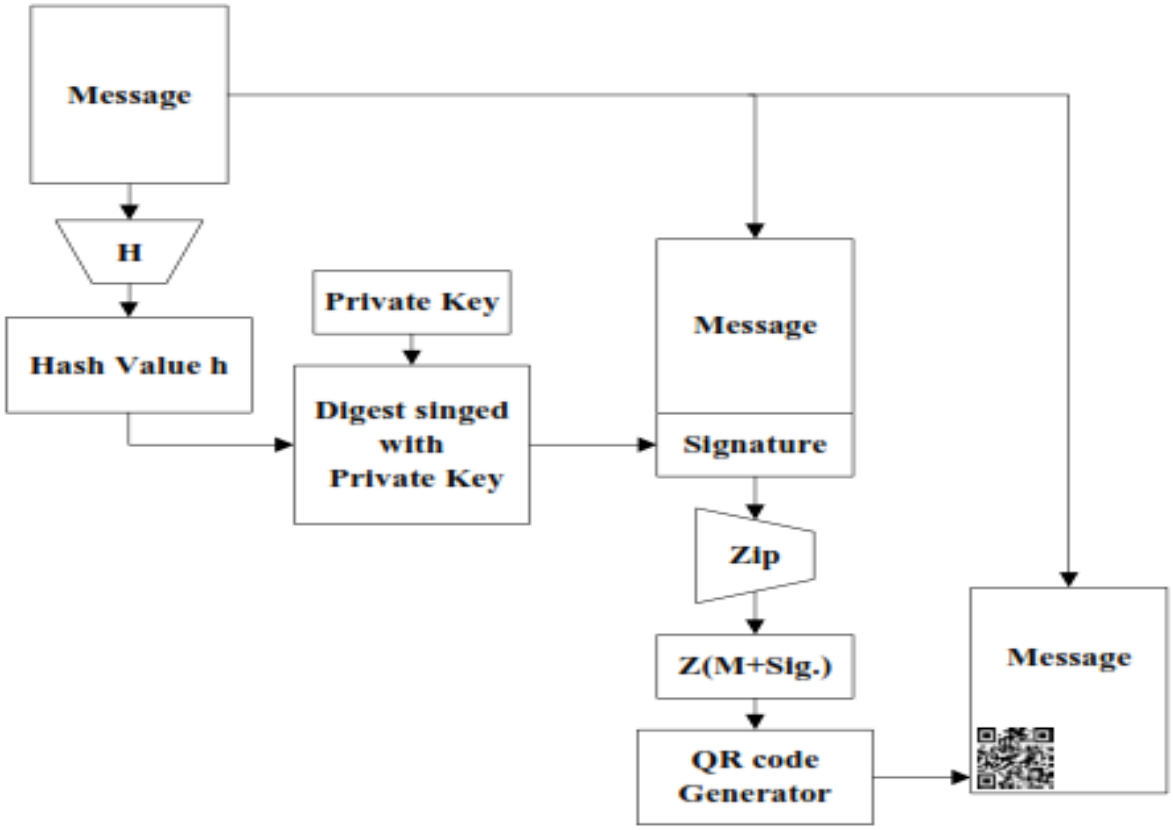

Figure 2.1: The Signing Process of PBDSs

The type 2D barcode used could be QR code, HC2D, Data Matrix or any other 2D barcode. Although the size of 2D barcode used has effect on the document size that can be signed using the 2D barcode [3], it does not affect the signing or verification process used, unless there is a special circumstance that forces us to intentionally change the process as in [3], which is the fact that HC2D barcodes are capable compressing data without the help of external compression algorithm.

The verification process of documents involves the following steps as depicted in Figure 2.2:

1. Scan signed paper based document using scanner.
2. Use OCR to convert scanned document into editable text.
3. Convert the text from the document into a hash value.
4. Scan 2D barcode using barcode scanner.
5. Decompress the output of the scanning process.
6. Convert the plaintext in the barcode into a hash value.

7. Decrypt the digital signature using the signatory's public key.
8. Compare the hash value from 6 and 7. If they are the same compare those with the hash value form 3. If all the three hash values are the same, the document is authentic. The first comparison makes sure that the digital signature is not forged. The second comparison makes sure that the document is authentic and form the right person.

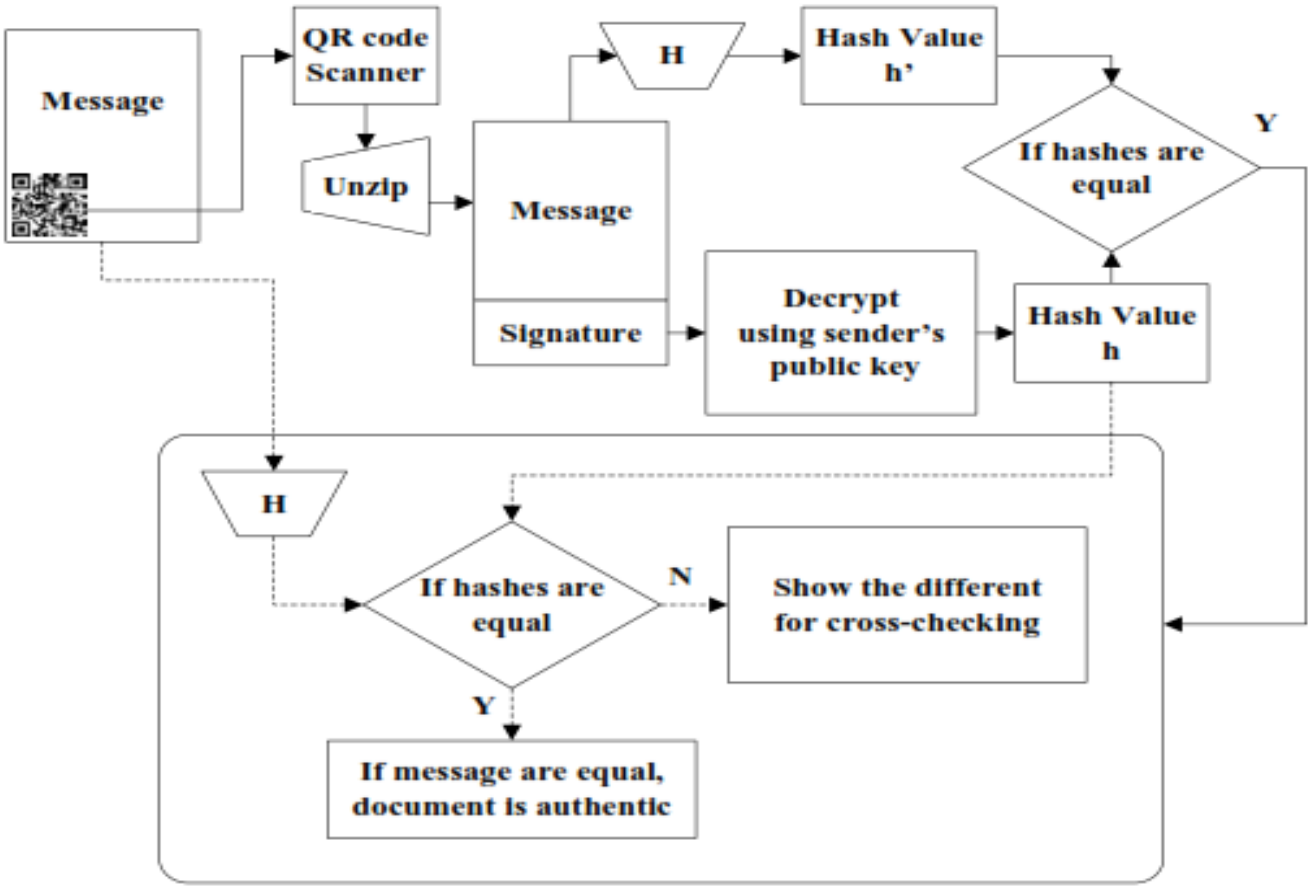

Figure 2.2: The Verification Process of Documents and PBDSs

2.3 Components of Paper Based Digital Signature

The components involved in the signing and verification paper based documents using PBDSs are from different areas of study. However, each component contributes to the signing and verification of paper based documents as follows.

I. Text Hashing Component

A cryptographic hash function is a mathematical algorithm that compresses variable length text into a fixed length output [14, 16, 17]. The output of cryptographic hash functions is called a hash value.

In the signing process of PBDS, the text extracted from documents is converted into hash value.

II. Data Encryption Component

Asymmetric cryptography (public key cryptosystem) is an encryption technique that employs different keys for encryption and decryption process [15]. In public key cryptosystem, confidentiality (secrecy) of a message to be sent to the owner of the private key is kept by encrypting the message with the public key of the owner. The private key owner decrypts the message using his own private key to read the message. On the other hand, authenticity confirms whether the message is the right one and originated from the right source. Authenticity of the sender and document is kept by encrypting the document with the sender's private key. The receiver verifies the sender and the document by decrypting the message using the sender's public key [10].

In our case encryption is being used in the second way. Data Encryption Component is risponsible for signing/encrypting a hash value returned by text hashing component with a private key [2, 3]. The purpose of using a hash value instead of plaintext is because hash value reduces signature size and and emproves execution time [3, 1].

During the document signing process, a hash value of a document is signed with the signer's private key to create a digital signature. Only the owner's public key can decrypt the digital signature to get the hash value again.

III. Data Compression Component

Data compression is the process of representing data in a compact form rather than in its original or uncompressed form [18]. It helps reducing the data length. During PBDS

generation, data compression component gets a string of text from text manipulation component and reduces its size. Finally it returns a block of binary data to text manipulation component to be split into acceptable smaller blocks for QR code generation.

Since QR code storage spaces is limited, the use of data compression to reduce the size of data being encoded into QR code is helps with the storage size limitation of QR code.

IV. QR Code Generation Component

QR code is a one of the most popular types of 2D barcodes developed in Japan by Denso Corporation in 1994 [20]. Its data capacity can be up to 7,089 numeric characters, 4,296 alphanumeric characters, or 2,953 bytes [2].

QR Code Generation Component is used to convert text data, which is the digital signature and other data from the document, into a QR codes, so that they can be printed on a paper based document or embedded in digital document.

V. QR Code Scanner Component

QR Code Scanner is a component / a hardware device that scans QR codes and returns a string of the text stored in QR codes. It serves as text digitization component. It does similar job as scanners and OCR programs by converting paper based text into digital and editable content. QR code scanner reads PBDS which were printed as QR codes from paper based documents.

VI. OCR Component

OCR is the process of identifying and classifying patterns in an optical image that correspond to characters and symbols [21].

The purpose of the OCR component is to convert a text in digital image into an editable string, (i.e., to the corresponding alphanumeric set of characters). It is used along with scanners to convert text in printed documents to editable format.

Paper based document signed with PBDS needs to be converted into a digital document before it is verified. The paper based document must first be scanned with a scanner to convert the paper based document into a digital image of the document. The digital image of the paper

based document then gets converted into actual digital document with the help of OCR programs.

In the verification process, OCR component is used to convert paper based document into editable format before the string returned is sent to hashing component.

VII. Data Decompression Component

Data decompression component is responsible for reversing compressed text into its original form [19]. It expands compressed text to its previous form.

When a paper based document is signed with PBDS, both the copy of the text in the document and digital signature are compressed to reduce size.

During verification, the QR code holding the digital signature and compressed text is scanned with QR code scanner. Then, the compressed text and the digital signature from the QR code are decompressed.

VIII. Data Decryption Component

The purpose of data encryption and decryption components is to make sure only the owner of the documents can sign on documents. The data decryption component specifically assures that the document is signed by a right person [3].

In the verification process, the decryption process involve the use of a public key. The decryption process reveals a hash value previously encrypted by the signatory. This hash value will later be compared with a recreation of the hash value from the re-digitized document to verify the authenticity of the document.

IX. Signature Verification Component

The signature verification component has two main responsibilities. The first responsibility is verification of the PBDS. It makes sure that the plaintext in the 2D barcode is authentic and from the right signatory. The second one is verification of the integrity of the information on the document. It makes sure that the text in the document is the right one and not changed in transit before or after it is printed.

Three hash values are received from other components for comparison. First, the signature verification component compares two hash values generated from information stored in the QR code. The similarity of the two hash values tell us that the signature is not a forgery. Second the component compares the two hash values with a hash value generated from text in the re-digitized document. The similarity of these three hash values tells us that the document is sent from the right sender and there is no tampering with the document

Chapter Three: Related Work

This Chapter discusses research papers directly related to PBDSs. Each paper is discussed as follows.

3.1 Paper Based Digital Signature

A research by Warasart and Kuacharoen [2] presents a paper-based document authentication model that uses digital signature and QR code to verify the sender of the document and the integrity of the text in a document.

During document signing, the sender converts the message in a document into hash value. The hash value is encrypted with the sender's private key and combined with a plain version of the message. The combination of both the plain message and the hash value is then compressed with a compression algorithm. Finally, compressed digital signature and message is encoded into a QR code. The QR code is inserted into the document and printed.

In the receiver's/verifier's side, the receiver of the document scans the document and converts the text into digital format. The QR code is scanned with QR code scanner and unzipped. The message from the QR code is hashed and compared with the hash value from the QR code. Finally, the message in the main body of the document is converted into hash value and compared with the hash values from the QR code. If they are the same, the signature is valid and authentic.

Java cryptographic architecture (JCA) is used to get the digital signature and digital certificate functionalities. Java Keytool is used to generate 1024-bit RSA public/private key pairs and certificates. Zxing library is used to encode the digital signature into QR code. The digital certificates were created in compliance with X.509 standard of the Certificate Authority. MS Word document is used to in the implementation and Apache POI library is used extract text form Word document. After the extraction of the message from document, SHA-256 is used to generate a hash value from the message.

There are two problems with identified in the research. First, the technique used in the paper works for small documents with less than approximately 4,296 characters, which is the limit of storage capacity of QR codes for alphanumeric characters when the text is not compressed. However the research doesn't consider larger documents. QR codes have limitations. They

cannot hold more than 4,296 alphanumeric characters. Data compression also has its limits. Second, failure of automatic verification doesn't guarantee that the document is not authentic. Rather there is a difference between the original document and the document being verified. The difference could be caused by OCR translation error. So it has to be supported by manual inspection of the two messages. Manual inspection could be a tiring process for large documents.

A research paper by Subpratatsavee and Kuacharoen [3] contributed a work with the main goal of creating a digitally verifiable signature for paper based documents using HC2D barcode. HC2D is a barcode with the largest storage capacity, which is 7,250 alphanumeric characters. The use of HC2D barcode reduces the need of compression.

During document signing, a hash value is generated from the message. Then, a digital signature is generated from the hash with sender's private key, and after the digital signature is generated, a copy of the message and the digital signature is encoded into an HC2D barcode, which is then attached to the message. A new document which consists of the original text and the HC2D barcode is generated and can be printed on paper.

At the receiver end, HC2D barcode is scanned using HC2D scanner and the digital signature is decrypted with the public key of the signatory. The document received is scanned. The message in the document received is converted into a hash value. The plaintext message in the barcode is hashed and compared with the hash value sent form the signatory. If the hash values are the same, they are compared with hashed values of the message in the main document's body.

For the implementation, the researchers used Java Cryptography Architecture (JCA) and Java programming language. The digital signature the research used complies with X.509 standard. SHA=256 algorithm, which provides 256-bit output is used along with RSA algorithm which is used to sign and verify the digital signature. Document are created using Word Processor.

The evaluation result shows authenticity of paper based documents can be achieve by using digital signatures and 2D barcode without the need for database. The verification process can be done automatically if the OCR programs are 100% accurate. Otherwise, manual inspection is required.

The model used in this paper has the same problem as in [2] with large documents with the exception of data compression. Because of data compression capability of HC2D barcodes the data compression step is skipped in this research. Even though HC2D has larger storage capacity than QR codes, as the document size to be signed increases, the occurrence of the same storage problem we have discussed in previously is inevitable.

A research by Singhal and Pavithr [5] uses the same techniques as [2] to authenticate a university degree using smartphones.

SHA-256 algorithm is used to produce a fixed length input. The input mode for the QR code is set to Byte mode which can hold up to 2,953 characters.

Since the model used in [2] is used without modification, problems with the technique discussed previously exist in this research as well.

3.2 Summary

All the researches discussed above used the same paper based document signing and verification technique with slight change in [3]. First, the problems identified in the researches are the same in all the three researches. The 2D barcodes used in all the researches have storage limitation. Although QR code and HC2D barcode have a huge storage capacity difference, both have fixed storage capacity. That means they cannot store more than their maximum storage capacities without the help of compression. The PBDSs generated using QR code and HC2D are not flexible to incorporate document size as it grows. Second, since manual verification involves inspection texts manually, manual verification would take too much time and effort in large documents.

Table 3.1: List of Related Work

Related Work	Method	Gaps
Warasart and Kuacharoen [2]	JCA Java Keytool SHA-256 algorithm X.509 standard.	QR code storage capacity is limited and PBDS not flexible to incorporate large documents Manual verification of documents is time consuming and tiring
Subpratatsavee and Kuacharoen [3]	JCA SHA-256 algorithm, X.509 standard.	HC2D barcode storage capacity is limited and PBDS not flexible to incorporate large documents Manual verification of documents is time consuming and tiring
Singhal and Pavithr [5]	JCA Java Keytool X.509 standard.	QR code storage capacity is limited and PBDS not flexible to incorporate large documents

Chapter Four: A Framework for Verifying Paper Based Document Using Multiple QR Codes

In this Chapter, a framework is designed in order to address problems regarding verification of paper based documents using digital signature. This framework resolves issues with verification of paper based documents by creating PBDS architectures, and modifying how PBDS are signed and verified to make verification of PBDS possible for any text based documents. Various mechanisms are devised to tackle different problems based on the advantages and disadvantages of the designed solutions.

This chapter discusses the details of the PBDS framework, its components and the architecture of the PBDS.

4.1 Basic Terminologies and Concepts

Some of the terms coined or borrowed from other researches to describe ideas in this thesis are explained as follows.

To clarify the concepts discussed more clearly, we will use a simple example. Let us assume there is a document *D* ready to be signed.

A. Full Plain Text (FPT)

An FPT is a plaintext extracted from a document where the text extraction aims encoding the manual document in its entirety. The term is used to differentiate it from partial plain text that will be defined next. For example, during the signing process, the complete text of the document is extracted from the document. This complete text of the document is the FPT.

B. Partial Plain Text (PPT)

Partial plain text (PPT) is a term used to describe a text extracted from part of the complete text in the document or FPT. For example, during the signing process, the part of complete text of the document *D* is extracted from the document. This part of complete text of the document *D* is the PPT. Furthermore, the size of the PPT S_{PPT} should be less than the size of FPT $S_{FPT.}$

$$S_{PPT} < S_{FPT}$$

C. Full Hashed Text (FHT)

Full hashed Text is a term used to describe the FPT converted into a hash value using a cryptographic hash function *f()*. Hence;

$$FHT=f(FPT)$$

D. Partial Hashed (PHT)

PHT is a term used to describe a hash value generated by converting the partial plain text (PPT) into a hash value using a cryptographic hash function *f()*. Hence;

$$PHT=f(PPT)$$

E. Automatic Verification

Automatic verification is the process of verifying paper based documents automatically with the help of a program. The process employs technologies such as Scanners and OCR programs to digitize paper based documents. Since document and QR code scanners need manual support during digitization of information, the term automatic verification is only used to describe processes involved after digitization of printed documents. Thus, after the re-digitization of the printed document, the document is automatically verified with the system used for verification [2]. Further, the term semi-automatic verification can be used to describe the whole process from digitization to verification.

F. Manual Verification

On the contrary to automatic verification, manual verification, which is a process of verifying document with human intervention, uses no scanner or OCR programs. This technique is used when automatic verification fails [2]. During manual verification, the QR code in the paper based document is scanned and the text in the document is compared with text in the QR code manually.

G. Signature Verification

Signature verification is a term used to describe comparison of two hash values from QR code to verify whether the PBDS is authentic or fake. As shown in Figure 4.1, the first hash value is a result of hashing the partial plaintext PPT_Q and full plaintext $FPT_{Q,}$ which are partial plaintext and full plaintext form the QR code, $PHT=h(PPT_Q)$, $FHT= h(HPT_Q)$. The second hash value *PHT'* is an output of decrypting a digital signature in the QR code, $PHT' = D_{EK}(E_{KE}(PHT))$, $FHT' = D_{EK}(E_{KE}(FHT))$, (Figure 4.1). The similarity of these two hash values

during comparison indicates that the PBDS is authentic. Being able to decrypt the digital signature with the signer's public key to get an authentic hash value can be achieved only if the hash value is encrypted with the right private key in the first place. This show that only the right sender of the document could have signed the document. Hence;

If PHT=PHT' or/and FHT=FHT', the PBDS is authentic, otherwise it is forged.

H. Document Verification

Document verification is the process of verifying whether a document is authentic and tamper free after its signing. For a document to be considered authentic it must pass two criteria. First, the PBDS on the document need to be authentic, that is *PHT=PHT'* or *FHT=FHT'*. Second, the hash value of the re-digitized document *PHT''*, which is a result of hashing the plaintext of the document *h(D)*, $PHT''=h(PPT_D)$, *or* $FHT''=h(FPT_D)$ needs to be the same as the hash values of the PBDS, Hence;

If PHT=PHT'=PHT'' or/and FHT=FHT'=FHT'' the document is authentic, otherwise there is tampering or change on the original text after the document is signed.

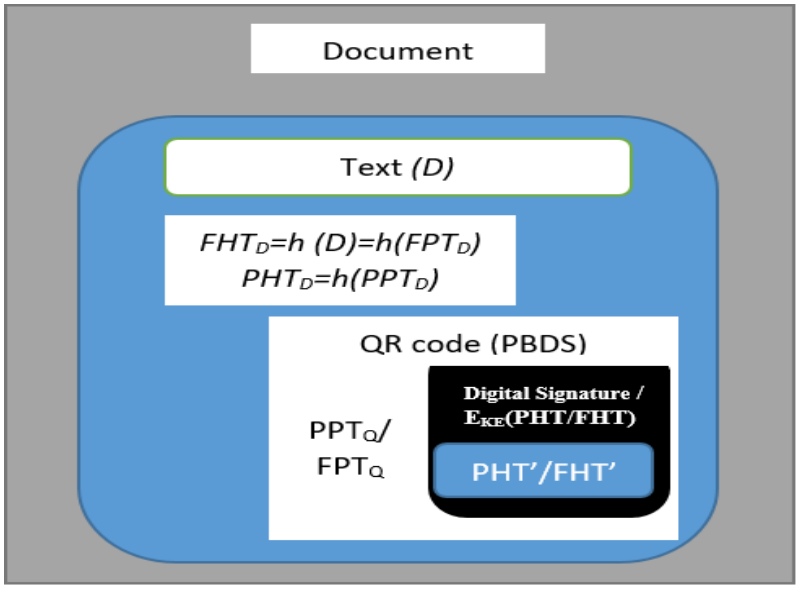

Figure 4.1: Paper Based Digitals Signature in a Document

4.2 Paper Based Digital Signatures Architectures

The modified digital signature is not only an encryption of *FHT* as the standard digital signature, it can be an encryption of *FHT* and *PHT* or just an encryption of only *FHT* as shown in Figure 4.5.

There are two parts in paper based document, the text in the document (Figure 4.2, (A)) and the QR code/signature (Figure 4.2, (B)). Before document verification, the text in the document needs to be converted into an editable text and the QR code needs to be scanned. First, an image of the document is taken using a scanner or camera and converted into editable text using OCR. Second, the QR code needs to be converted into a text using QR code scanner.

(B) 

(A)

Massive University entrance exam scandal compels ministry to reject results for several subject matters

Ethiopian government shut down the internet in the first week of June 2019 so as to prevent foul play as high school students in the preparatory class (grade 12) were writing the university entrance exam. As it turns out, the results of exams administered in the last two days demonstrate that there was an exam scandal, and the ministry of education is rejecting the results of those exams.

Ministry of education authorities during the press conference.

Ethiopia's Ministry of Education announced on Wednesday that it will only accept results of only four of Ethiopian Higher Education Entrance Examination (EHEEE) that were administered during the first two days of the exam sometime in June of 2019.

The ministry was joined by education offices of regional states as well as NEAEA (National Educational Assessment and Examinations Agency) authorities during the press conference.

Figure 4.2: An example of a signed document ready for printing

In the verification process of PBDSs, the digital signature inside the QR code needs to be decrypted with the signer's private key. After the decryption, the hash value is used to verify documents automatically. In automatic document verification, the verification program verifies the document. An image of a document is converted into digital text by using OCR

program and the verification program verifies the authenticity of the document. Automatic verification is used when the text from document image is converted into digital text without any errors. However, sometimes OCR program may not convert images into text without error. When automatic verification fails for any reason, manual verification of documents is needed to verify whether the paper based document is authentic. Manual verification involves manually comparing the plaintext from the document with the plaintext from QR code.

There are two types of PBDS models identified based on the context in which they will be used, PBDS type A (PBDS-A), PBDS type-B (PBDS-B).

4.2.1 Paper Based Digital Signature Type B (PBDS-B)

PBDS-B is a variation of PBDS proposed by Warasart and Kuacharoen [2], modified in this thesis to be signed and verified using multiple QR codes. The architecture of a single QR code PBDS is already been covered in [2]. The single QR code PBDS targets small document such as small notes, notices, letters, and so on. It is an ideal solution to sign documents with small amount of text. PBDS-B on the other hand, removes the paper based document limit that exists because of QR code storage limitation and flexibility problem in the single QR code PBDS. Because of its capability to grow as the document size grows in the signing process, it can include FPT plus a digital signature generated by signing FHT of any document. Since the FPT and FHT are generated from the whole document, the signature provides all the three properties of digital signature.

The use of multiple QR codes rather than a single QR code in PBDS-B, increases the storage capacity of PBDS-B *S*, to the number of QR codes *n*, times storage capacity of a single QR code *s*, which is $S=n(s)$. While one QR code can be used to store only the digital signature, another QR code can be used to store the FPT. An additional QR code can be added to store the FPT if one is not enough. The FPT is broken into multiple chunks and each chunk is stored in a different QR code as shown in Figure 4.3.

During verification, the FPT is broken into multiple chunks that are collected from multiple QR codes and the verification process continues.

PBDS-B has the following characteristics:

1. All the three properties of digital signature authentication, integrity and non-repudiation both in automatic and manual verification of documents.
2. The size of the document that can be verified by PBDS-B can be increased by increasing the number of QR codes used in the PBDS.
3. Depending on the size of the document to be signed, the PBDS can be very large and bulky.

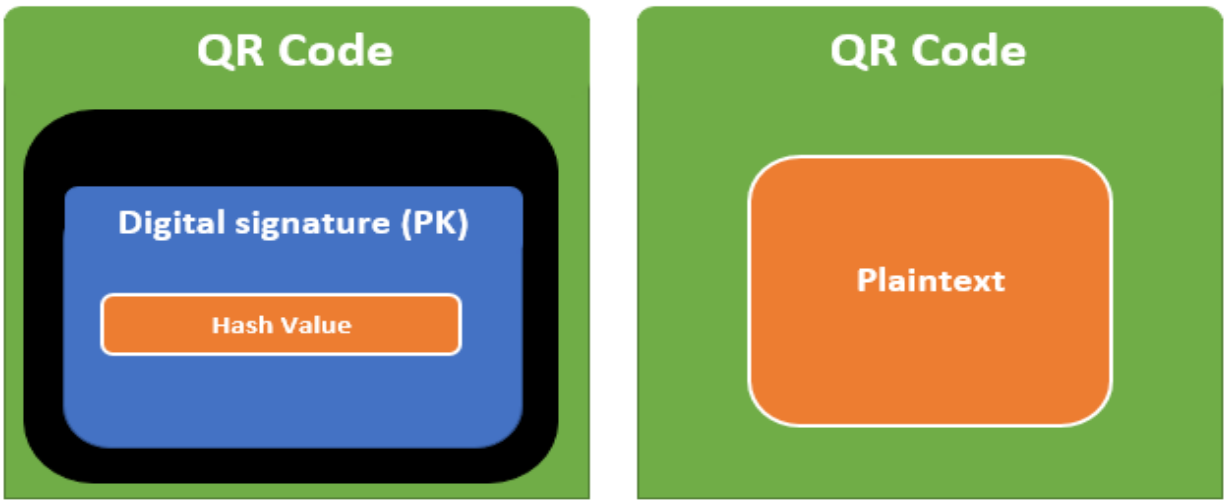

Figure 4.3: Architecture of PBDS-B using multiple QR Codes

4.2.2 Paper Based Digital Signature Type A (PBDS-A)

PBDS-A is a PBDS designed to reduce the number QR code used in PBDS-B. However, on the contrary to PBDS-B, a digital signature is PBDS-A does not include only one hash value. The first QR code includes a digital signature that contains FHT and PHT. The rest of the QR codes store a compressed chunks of PPT.

There are two ways to verify PBDS-A, quick verification and full verification. As shown in Figure 4.1, the Quick Verification process includes converting the partial plaintext from the digitalized document $PPT_{D,}$ into a hash value PHT_D, decrypting the digital signature and comparing the resulting hash value PHT'. Manual verification includes comparing the PPT_Q from the QR code with the PPT_D from the digitalized document manually.

Full document verification allows authentication, integrity and non-repudiation of documents. However to do a full document verification, we need a copy of the full text of the document (FPT) in the PBDS, as discussed in PBDS-B. QR code storage limitation doesn't allow storing large amount of data in a single QR code, even compressed. In multi-QR code PBDS on the other hand, the number of QR codes can grow very large and become very challenging to verify. Thus, there must be a way to get only the core ideas of a document rather than every important and non-important text in a document.

I. Selection Criteria for Partial Plain Text (PPT)

The use of PPT text eliminates the need to store FPT in a QR code. When selecting PPT for documents signing and verification, focusing the following point will reduce the risk of forged documents slipping verification.

Table 4.1: Sample PPT Selection Criteria

Document Type	**Content that Must be included in the PPT and PHT**
College certificates, grade reports, school cards or any document with results on it	Date of issue, name of the institution, grade/mark/point scored/GPA, name of department, special conditions of the issuing.
Personal documents such as letters, resumes, notices and so on	Date of signing, name of the signer, name of the receiver, title of the document, part of the content that convey the core point of the document.
Business documents such as letters, financial documents, and human resource hiring documents and so on.	Date of signing or approval, name of the signer, receiver of the document, title of the document, part of the content that convey the core point of the document. Any numerical data in the document must be part of the PP.

Asking the following questions may show us what important in the document:

- ✓ When did the document signed?
- ✓ Who singed it?
- ✓ Who is the recipient of the document?
- ✓ What is the document approving/ granting/ acknowledging/ telling?
- ✓ What is the effect period of the document, if there is one?

The problem with using PPT instead of FPT is that only parts of the document can be verified for integrity. So it is very important to make sure there is no opening for tampering not to be detected when selecting the PPT. If every bit of text in the document is important, it must be included in the PPT. Although the whole document cannot be verified against tampering when using PPT and PHT, part of the document that is important can be selected and used for tamper detection. If the PPT is selected properly, the inconsistencies can easily be detected if there is tampering in the document.

II. New Document Content Organization

OCR programs these days are not as smart as we like [22]. For an OCR program to work as expected, the document format before it is printed is important. Especially, this is true if only part of the document needs to be scanned and converted into digital text. Selection of PPT employs such a process to convert image into text. Therefore, in order to make OCR conversion of part of a document effective, the part of the document that need to be selected (the PPT) must be separate from the main content of the document. A distinct separation between the PPT and FPT must exist in the document. This can be achieved by allocating a separate position for the PPT as shown in Figure 4.4.

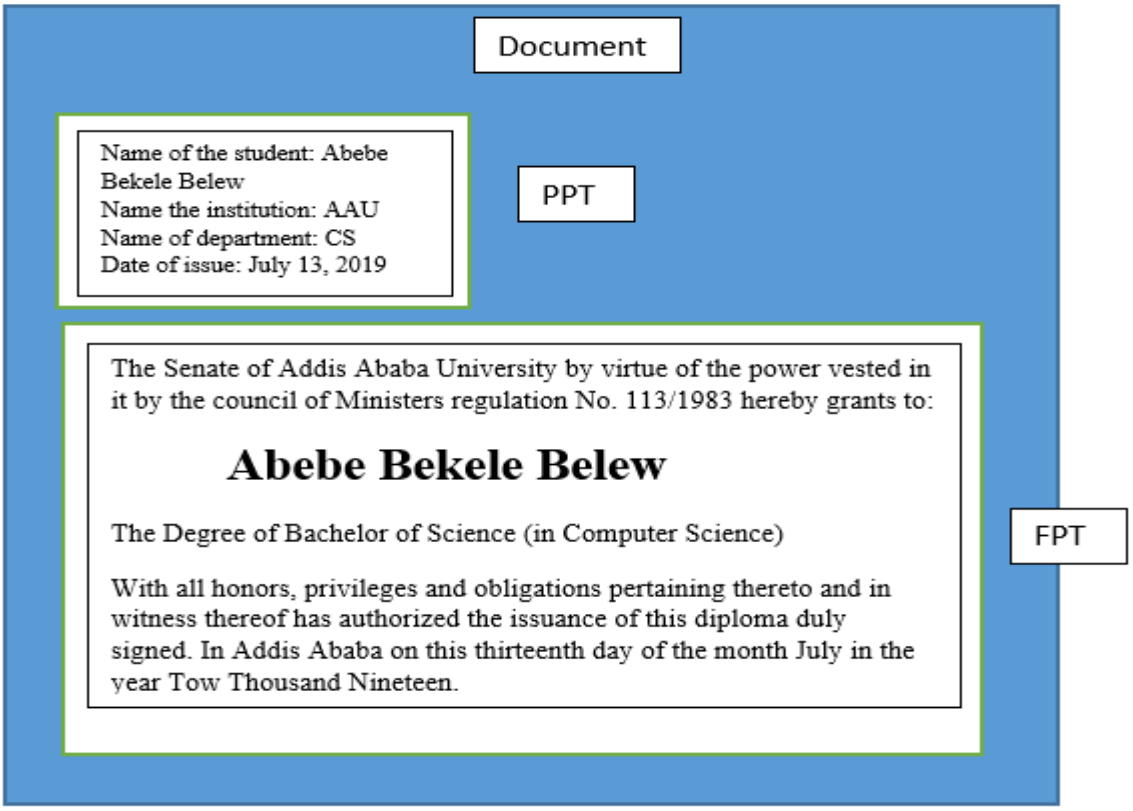

Figure 4.4: New Document Format for Separation of Partial Plain Text and Full Plain Text

A well selected PPT can be as strong as FPT in forgery detection. If a document has a separate location for the PPT, only the PPT section of the document needs to be scanned. Since, only the OCR section in the documents is scanned, we can easily identify the section need to be converted into digital text and to an OCR program. Otherwise, the whole document needs to be scanned and converted into digital text, which is not efficient. Next, the PPT needs to be manually identified and selected, which is time consuming and tiresome process. Therefore, providing a separate location for PPT will make verification process very easy.

The use of PPT and PHT rather than FPT and FHT makes verification of large documents easier and reduces the number of QR codes. However, if not enough PPT is selected to cover the core points of the document, PBDS's ability to secure the integrity of the document gets compromised. Only the part of the document covered by the PPT is effective in detecting modification of the document after the signing. That is why it is important to include the important parts of the document, which hold the core meaning of the information in the document, into the PPT.

If every part of the document is important and needs to be verified by the document, the whole documents needs to be scanned, digitized with OCR programs and converted into a hash value to be compared with the FHT' in the QR code. This is called full verification. Even though the verification process is the same as verifying documents with PPT and PHT discussed earlier, the difficulty of converting documents into a format ready for verification in full verification makes this verification technique challenging. During verification of document with FHT and FPT the whole document is verified, not just part of the document. If the document has thousand pages, each page needs to be scanned, digitized and hashed. As the size of the document increases the number of translation errors made by OCR program increases. Therefore, Full Verification is used only when Quick Verification not enough or when thorough verification of a document is needed.

III. Architecture of PBDS Type A

With the use of multiple QR codes, the storage capacity of PBDS-A can be increased. In this architecture the first QR code is used to store only the digital signature. Another QR code can be used to store the PPT. An additional QR code can be added to store the PPT if one QR code is not enough. The PPT is broken into multiple chunks and each chunk is stored in different QR codes (Figure 4.5). Finally, the generated QR codes are added to the document and the document is printed.

PBDS-A has the following characteristics:

1. Provides a full authentication, and non- repudiation during full verification using FHT.
2. During automatic verification, the PHT provides document integrity verification to the parts of the document covered in the PHT.
3. For manual verification, the PPT provides document integrity verification to important parts of the document covered in the PHT.
4. If there is a need for securing integrity of the whole document, the FHT of the PBDS needs to be compared with hash value of the whole document. The process is time consuming and challenging. However, it provides the full authentication, integrity and non-repudiation.

5. Other than for quick verification, the PPT and PHT of PBDS-A can be used to verify the authenticity of the signature.

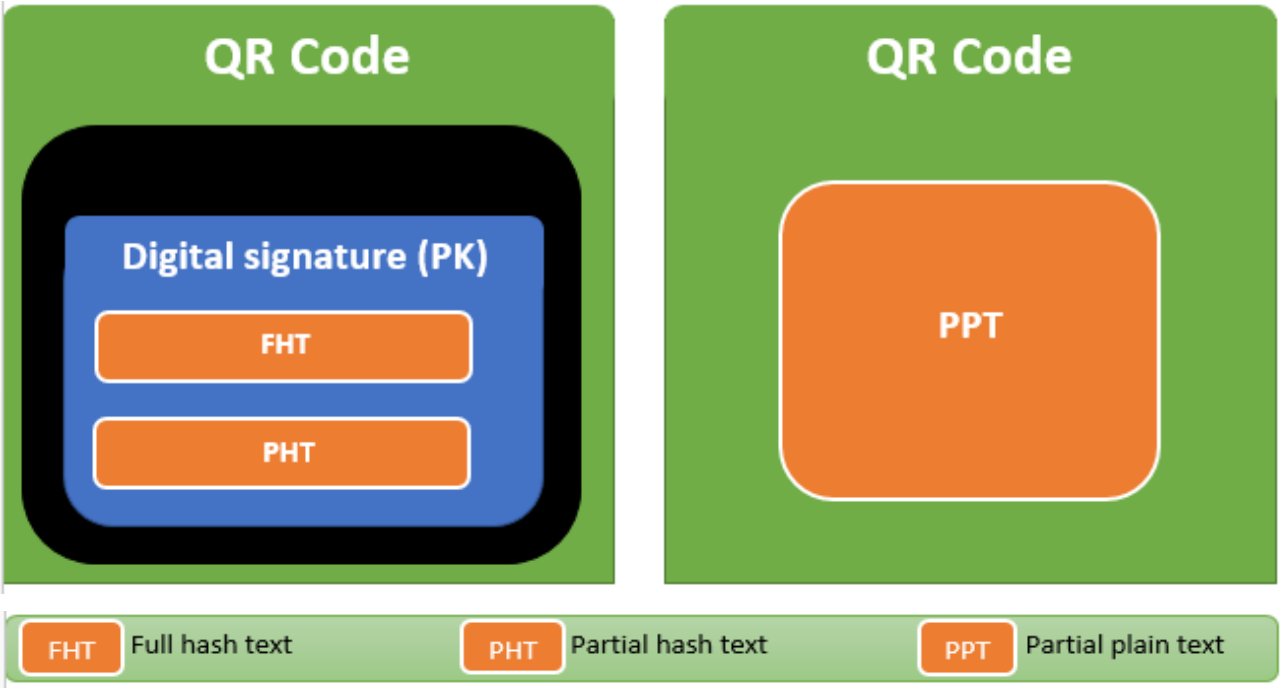

Figure 4.5: Architecture of PBDS-A Using Multiple QR Codes

4.3 Proposed PBDS Generation and Verification Processes

The use of multiple QR codes rather than a single QR code to generate and verify PBDS has changed how PBDS are signed and varified as shown in Figures 4.6 and 4.7 respectively.

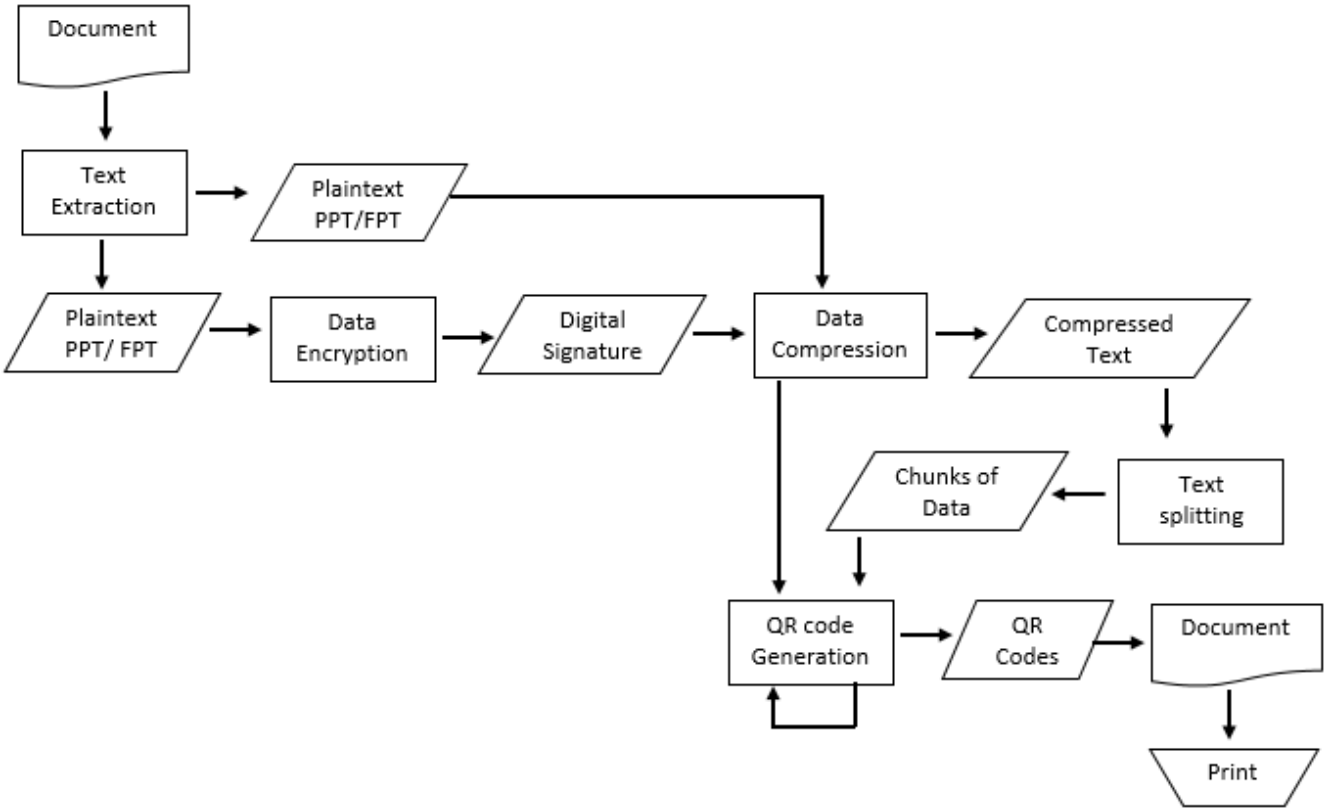

Figure 4.6: Proposed PBDS Generation Process

The signing process of PBDS starts with extracting text from document such as Ms Word. According to [2], the next step is in the signing process is hashing the extracted text. The output of the hashing process is a fixed length hash value. So the hash value is next encrypted with the signer's private key. The product of the encryption process is a digital signature. The digital signature is combined with a plaintext and compressed. Finally, the product of the compression is converted into QR code.

As dipicted in Figure 4.6, the proposed signing process is different. The size of plaintext we use to convert into a hash value depend on the size of the text and the type of PBDS architecture used.

1. In the case of PBDS-A, two types of hash values are created.
    a. Some percent of the text in the document is used to convert into a hash value (PHT).
    b. The whole text is converted into a hash value (FHT).
2. In the case of PBDS-B, only one hash value is generated from the whole document (FHT).

Following the hashing, the hash values are encrypted with the signer's private key. If two hash values are created, they are combined together before encryption. The product of the encryption process is a digital signature. Next, the digital singature and the plaintext (PPT/FPT) are compressed. The data being compressed depends the type of the PBDS archtecture.

1. In the case of PBDS-A, PPT is selected based on some selection criteria or pattern and extracted from a document. The PPT is next combined with the digital signature before compression.
2. In the case of PBDS-B, the whole text in the document (FPT) is extracted and combined with the digital signature.

Next, the PPT/FPT is split into chunks of data after compression. Each chunk of compressed data is finally encoded into a different QR code.

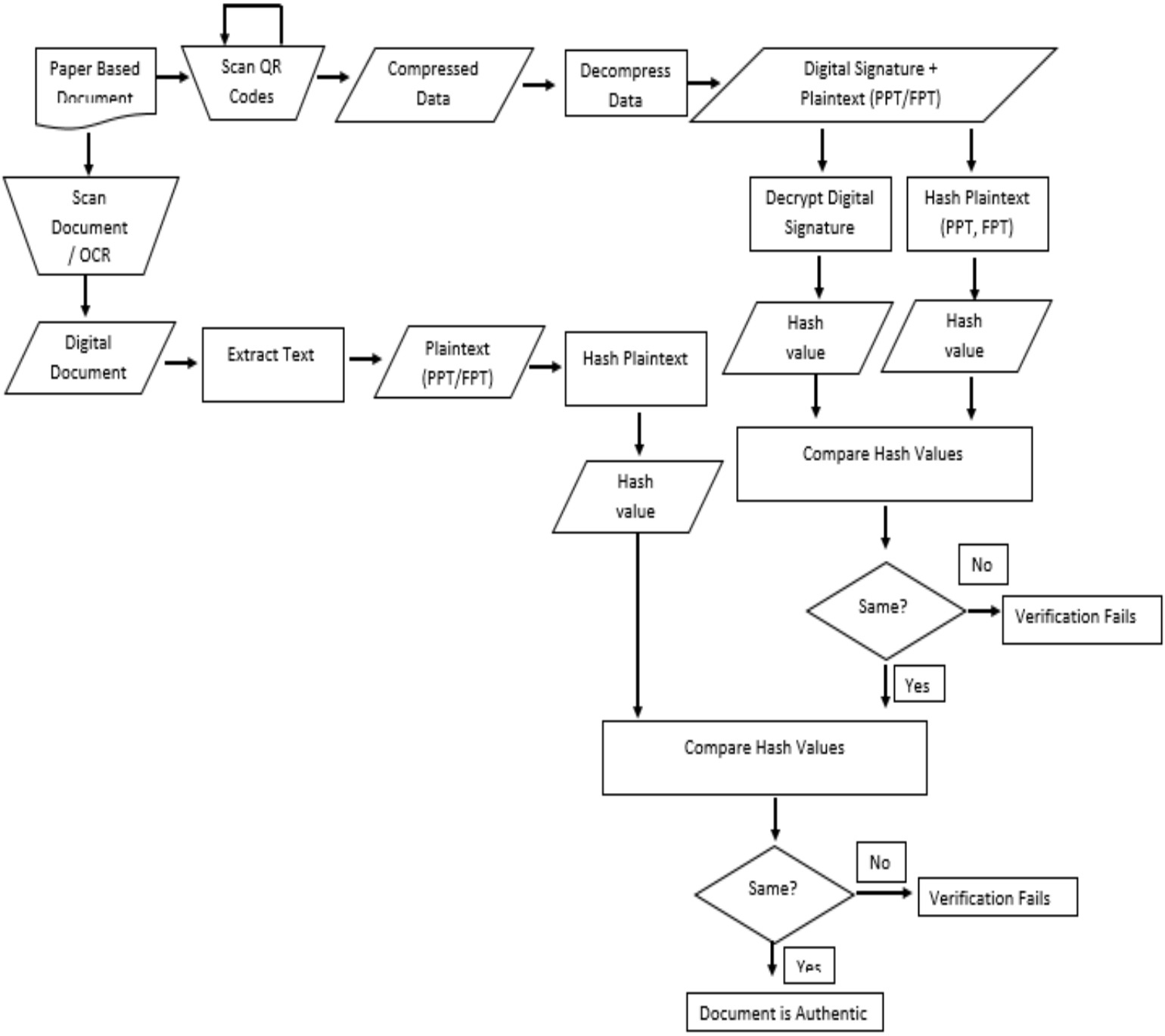

Figure 4.7: Proposed PBDS Verification Process

In the verification of paper based digital signature, three hash values are compared to verify whether a document is authentic. These three hash values are generated from three different sources in the paper based document. The first two hash values are from the PBDS/QR code.

The third is generated from the main content of the document we digitized using scanners and OCR programs.

As shown in Figure 4.7, the verification process of PBDS starts with scanning the paper document with a scanner and scanning the QR code with QR code scanner simultiniously. The output of the QR code is a compressed data that needs to be decompressed. After the decompression we get two types of data, plaintext and a digital signature. The plaintext can be PPT or FPT. The plaintext is split into chunks. So before converting it into a hash value the split chunks from different QR codes need to be combined together to create the original PPT of FPT.

During verification, the plaintext (PPT/ FPT) from the QR code is converted into a hash value to generate the first hash value, *H1*. The digital signature from the QR code is decrypted to reveal the second hash value *H2*. The output of scanning the printed doument is an image of the document. The image is then converted into a digital text using an OCR program. The hashing process of the plaintext from the paper document is as follows:

1. If the PBDS is type A, The PPT in the docuement is selected and extracted for quick verification. FPT of the document is extracted from the document for full verification. Next, the FPT/ PPT is converted it into a hash value to generate FHT/ PHT respectively.
2. If the PBDS is type B, FPT is converted into a hash value. This generate full hash (FHT).

The above two options generate the third hash value *H3*. Finally, the three hash values are compared to verify document's authenticity as follows:

1. *H1* is first compared with *H2*. If the both hash values are the same, they are compared with H3. A document must pass both comparisons of hash values to be authentic.
2. In automatic verification, full hashes are not compared with partial hashes for verification. Similarly in manual verification, full plains are not compared with partial plains for verification.

3. If manual verification is needed, in the case PBDS type A, the PPT in the document is compared with the PPT in the QR code. If the singature is type B, the FPT in the document is compared with FPT of the QR code manually.

4.4 Printed Digital Signature Framework (PBDSF)

There are two main components for the PDSF.

- ✓ Paper Based Digital Signature Generator (PBDS Generator), and
- ✓ Paper Based Digital Signature Verifier (PBDS Verifier)

Each component works independently of each other. The purpose of PBDS generator component is to generate a digital signature that can be printed, which later can be verified by anyone using PBDS verifier.

PBDS verifier reads PDSF digitally readable digital signature from paper based documents and automatically verifies the document or facilitates access to user to manually compare the content on the paper with the FPT or PPT decoded from the QR code.

As Figure 4.8 shows, there are two components in the framework. PBDS-A and PBDS-B generator and verifier components. The areas shown as gray boxes are areas in which this thesis has added new knowledge. This includes new PBDS architectures using multiple QR codes, modified PBDS signing and verification techniques, and new document content organization/ format to make paper based document verification easier. The architectures and processes identified in this thesis make up the overall framework shown in Figure 4.8. The areas in which this thesis contributed are show in detail in Figure 4.9 and 4.10. The framework shows how each component, the architectures and processes, work together to verify documents.

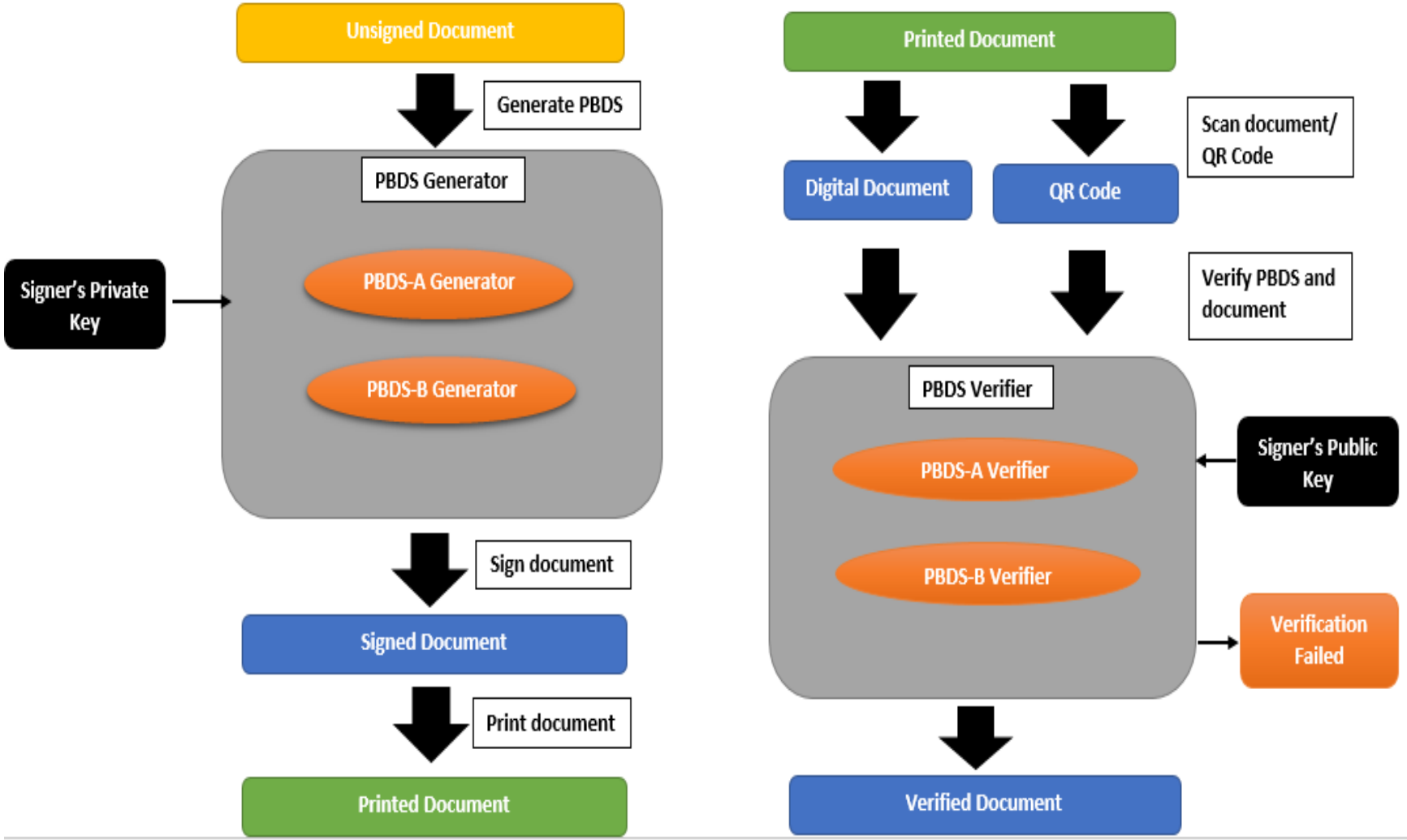

Figure 4.8: QR Code Based Digital Signature Framework from Process Point of View

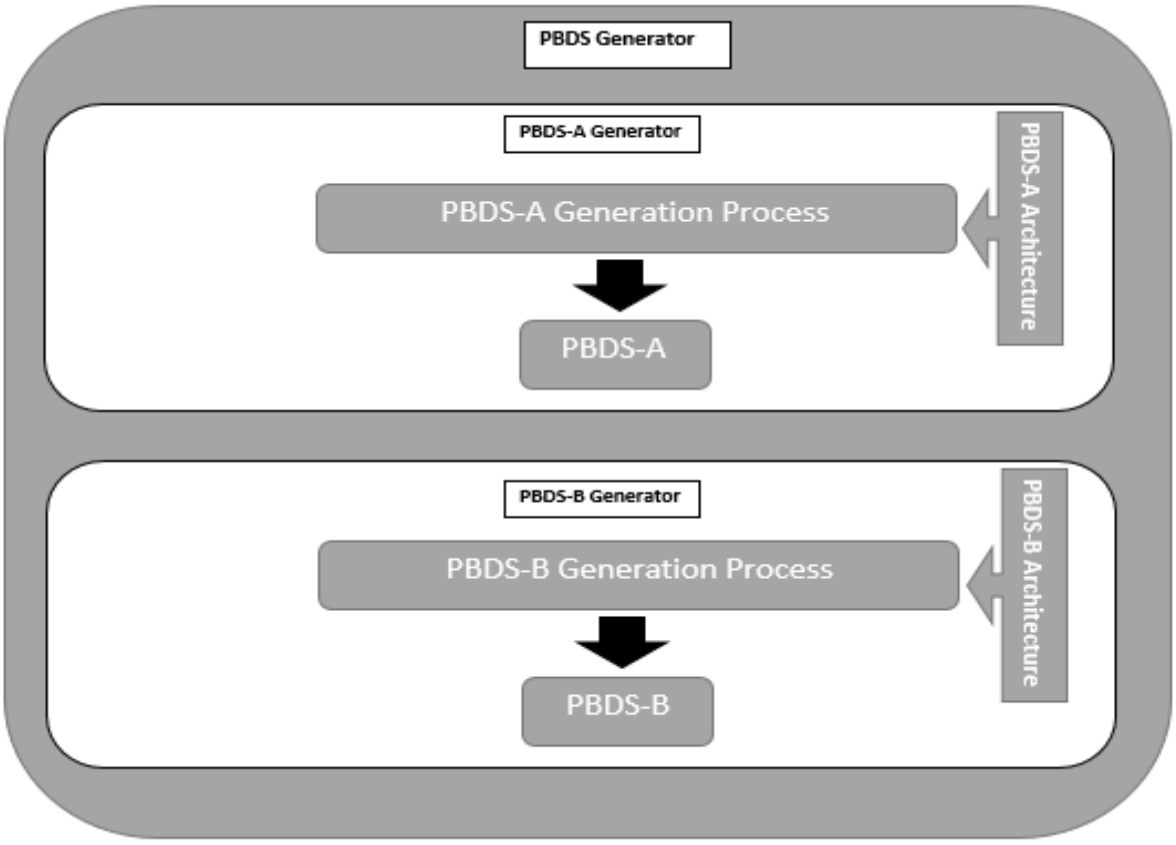

Figure 4.9: PBDS Architecture and Process in Paper Based Document Signing

Figure 4.9 show how the new PBDS architectures work with PBDS signing process. In the PBDS generator, the selection of which architecture to use will depend on user preference. If the architecture selected however, the signing process specific to that architecture is used to sign a document. PBDS-A can only be signed using PBDS-A signing process. The same is true for PBDS-B.

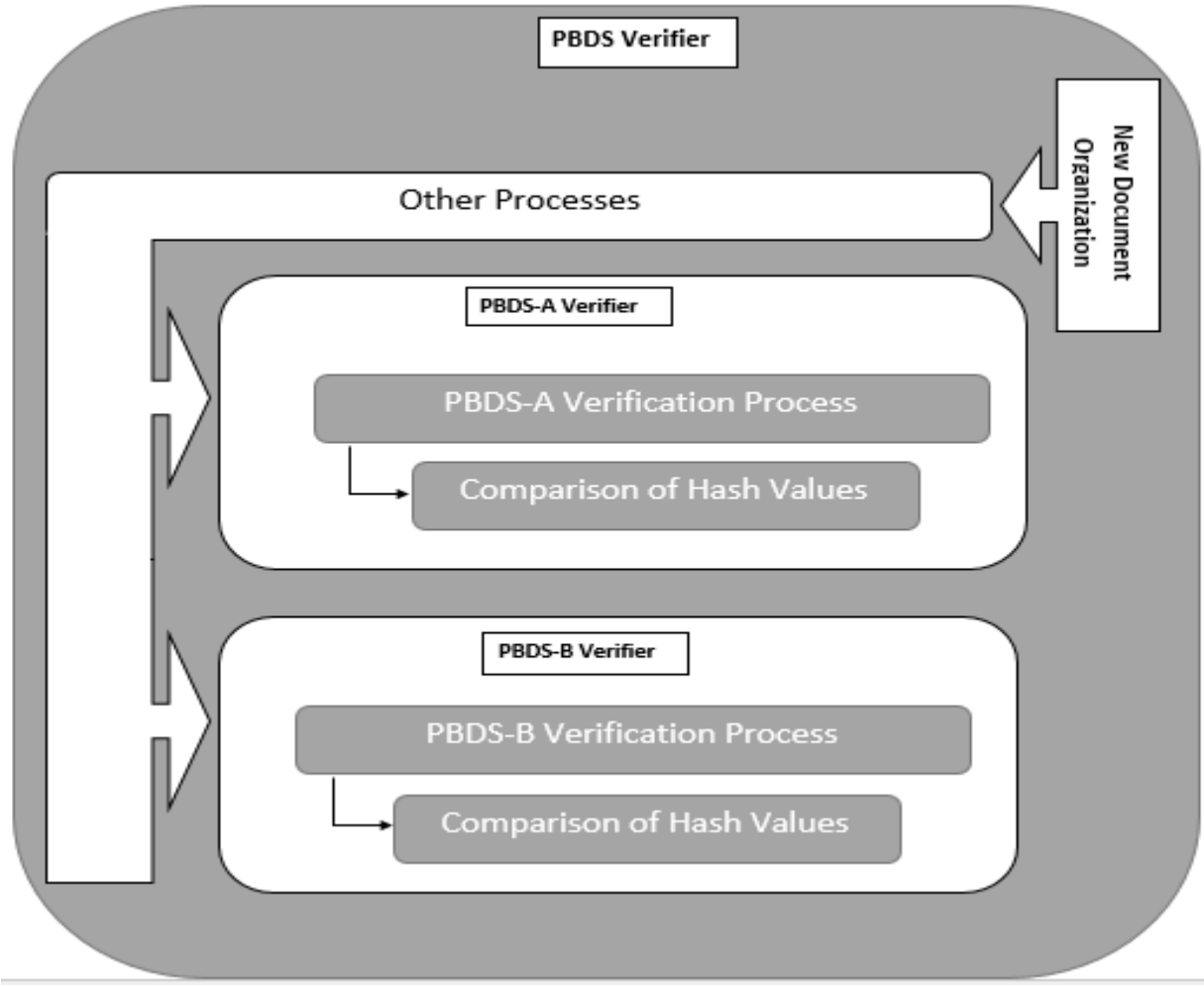

Figure 4.10: New Paper Based Document Content Organization in Paper Based Document Verification

As discussed in section 4.2.2, the document content organization can make verification PBDS easier or hard. The selection and separation of the PPT from the main content of the document makes scanning and OCR conversion less time taking and easier. Depending on the type of architecture used to sign the document, the verification process takes place. The processes described as "Other Processes" in Figure 4.10 are Text Extraction, Text Hashing, Decryption, Decompression and OCR, which are process that need to be finished before the verification. These other processes get paper based document as an input and produce three hash value. The Verification Process then compares the three hash values to make sure that the document is authentic.

The components involved in the singing and verification PBDS are common component used in different areas of study. Most of the components are covered in Chapter Two. One component is modified in this thesis as follows.

Text Splitting Component

Documents can have larger text size than QR codes can hold. The text in the document must be split into multiple texts that can fit into QR codes. The text splitting component splits text into chunks based on a given text length and returns a list of chunks of text.

Chapter Five: Experimentation/Prototype and Evaluation

In the previous chapter, a framework for paper based digital signature is discussed. The architecture, the framework, components involved in the framework, and the communications between components are explained in detail. In this Chapter, the implantation of the framework using a prototype, evaluation and result of the evaluation will be discussed.

5.1 Prototype PBDS Program

The prototype was designed by grouping all the components into three subsystems, signature generator subsystem (SGS), signature verifier subsystem (SVS) and neutral subsystem (NS). Those components involved in signature generation only are grouped under SGS. Those involved in signature verification only are grouped under SVS. Components that communicate with both SGS and SVS are grouped under NS. Although NS is not discussed as an independent component, the components in this sub-system were part of either SGS or SVS in chapter 4. Figure 5.1 shows the components of each subsystem and how they communicate with each other.

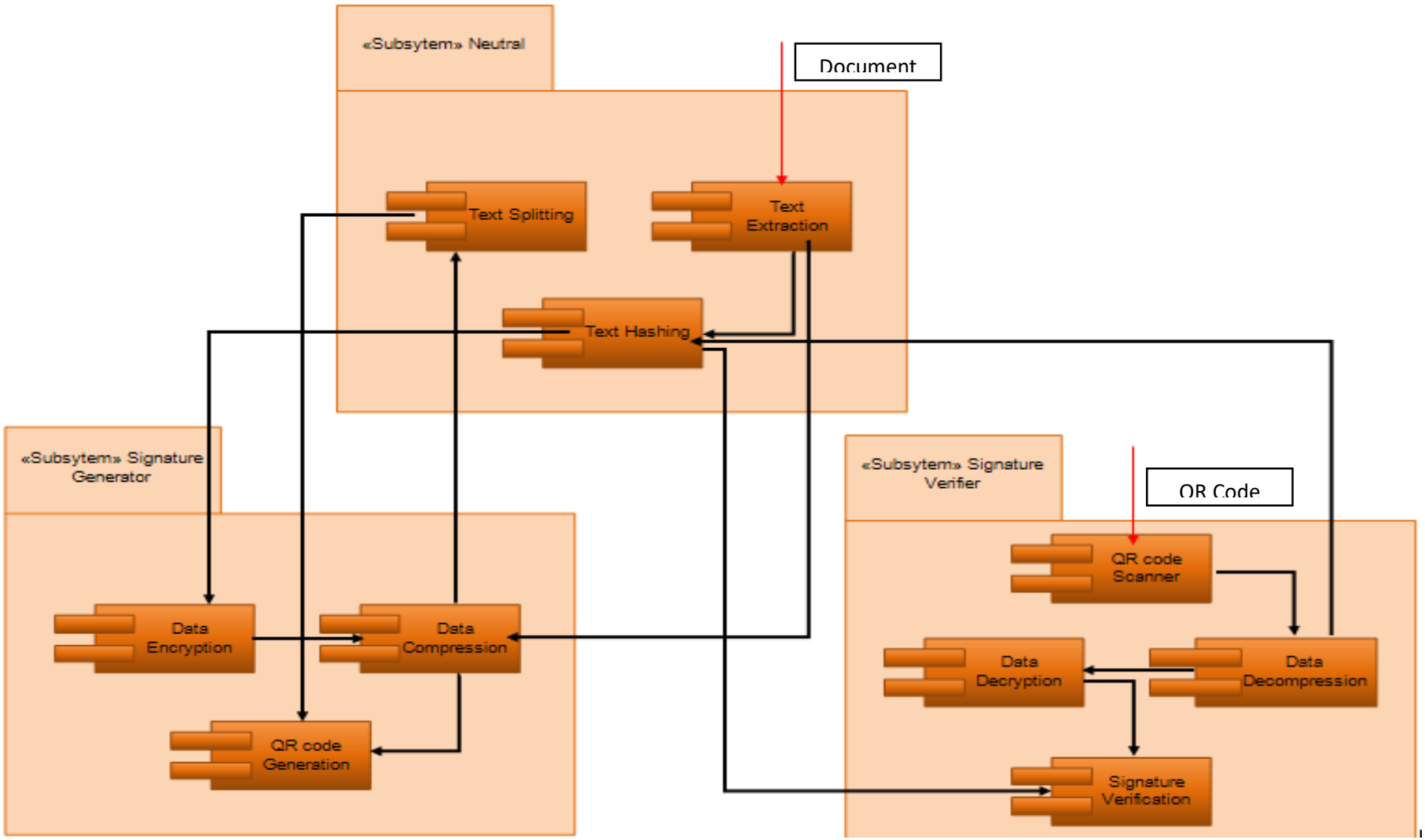

Figure 5.1: Subsystems of the Prototype PBDS System

Figure 5.1 shows how the prototype is structured and designed rather than how the signing and verification process PBDS works. As it can be seen from the Figure, there is no OCR or Scanner Component added to the sub-systems. These components are considered external components to the prototype. Even though these components are part of the PBDS verification process, they are not part of the prototype. This is because, adding them in the prototype would not add anything to the research. Adding them to the prototype would be reinventing the wheel.

The signature generator subsystem contains various components used in generating PBDSs such as data compression, data encryption and QR code generation. These components get input from text manipulation and text hashing components of neutral subsystem in the signature generation process.

The signature verification subsystem on the other hand contains components involved in the signature verification process such as QR code scanner, OCR, data decompression, data decryption, and signature verification components. Similar to the signature generation subsystem, the components from signature verifier subsystem communicate with components from neutral subsystems for the signature verification.

As discussed in the previous chapter, most of the components in each subsystem are support components that provide input/output data to the main components. In the signature generation subsystem the main components are text hashing and data encryption component. These component are key in creating digital signature. In SVS the main component is signature verification component. This component gets three different hash value from various components in the SVS and NS for comparison. However, the purpose of this thesis is not improving the digital signature. Rather it is to modify digital signatures in a way they can easily be used to verify digital signatures from paper based documents.

In order to improve digital signatures according to our objective we needed to change the nature of digital signatures, how they are stored, compressed, read and verified. So this affects many components from each subsystem such as data compression and encryption from SGS, data decompression, decryption and signature verification components from SVS and text splitting from NS. These components are considered main component in terms of determining the nature of the signature to be generated and the type verification process involved.

5.1.1 Signature Generation

PBDS-A

The first step in generating PBDSs is extracting text from documents. The extraction process generates plaintext. Next, processes such as text hashing, text splitting, data encryption, data compression and QR code generation follow depending on what is needed as shown in Figure 5.2. Based on the type PBDS being generated, the number of components involved in the generation process may vary.

The signing process of PBDS-A starts with PPT selection. As show in Figure 5.3. The signatory carefully selects what is important in the document to be included in the PPT. Next, plaintext (PPT/FPT) is extracted from the document. After the extraction of plaintext, the PPT and FPT extracted from the document are converted into hash values to become PHT and FHT respectively. Next, the PPT is split into chunks that can fit into QR codes, in our case that is 500 characters. Following the splitting process, the PHT and FHT are combined together and encrypted to generate a digital signature. Then, the digital signature and chunks of FPT are compressed separately and encoded in QR codes.

```
This is an algorithm for signing a document using PBDS
IF selected architecture is PBDS-A THEN
    Select PPT
    Extract text from document
    Convert PPT to hash value
    Convert FPT to hash value
    Split PPT into 500 character chunks
    Encrypt PHT+FHT with a private key
    Compress digital signature
    Compress each chunk of FPT
    Encode compressed digital signature into QR code
    Encode compressed PPT into QR codes
END IF
```

Figure 5.2: Pseudo Code for PBDS-A Generation Process

PBDS-B

First, text is extracted form document (Figure 5.4). After the extraction, the extracted text is converted into a hash value. The hash value is then encrypted with a private key to produce a digital signature. On the contrary to PBDS-A, a digital signature inside PBDS-B contains only one hash value, FHT. Next, the FPT is split into chunks that can fit into QR codes, in our case it is 500 characters. Following the splitting, the digital signature and each chunk that we get from the splitting of FPT is compressed and encoded into a separate QR code.

```
14  This is an algorithm for signing a document using PBDS
15  ELSE IF selected architecture is PBDS-B
16      Extract text from document
17      Convert FPT into hash value
18      Encrypt FHT using private key
19      Split FPT in 500 character chunks
20      Compress digital signature
21      Compress each chunk of FPT
22      Encode compressed digital signature into QR code
23      Encode compressed chunks of FPT into QR codes
24  END IF
```

Figure 5.3: Pseudo Code for PBDS-B Generation Process

Finally, the PBDS generated is added into the digital document such as Ms. Word, the document is printed.

5.1.2 Signature Verification

PBDS-A

The signature verification process mostly involves doing the signature generation process in the reverse order. The process include re-digitizing document and signature using a scanner, OCR program and QR code scanner, hashing re-digitized document, decompression and decrypting digital signature in the QR code as shown in Figure 5.5. After the reversing process is over, the final step is comparing three different hash values resulted in the previous steps, which are Figure 5.5 line 30, 31, 35.

```
25  This is an algorithm for verifying a document using PBDS
26  IF the architecture used is PBDS-A THEN
27      Scan QR codes
28      Combine data
29      Decompress ditial signature and PPT
30      Decrypt digital signature
31      Convert PPT from QR code into hash value
32      Separate PHT and FHT
33      Extract text from redigitalized document
34      Convert redigitalized text (PPT) into hash value
35      Compare hash values
36  END IF
```

Figure 5.4: Pseudo Code for Digital Signature Verification Using PBDS-A

In the verification process of paper based document, the PPT from the first QR code is converted into a hash value. Next, the digital signature is decrypted with a public key and the partial hash and full hash inside the digital signature are separated. Then, a PPT is selected from the re-digitized document and converted into a hash value.

In the final step of the verification process, the two partial hash values from the digital signature and the hash value generated from PPT, from QR codes are compared, as shown in Figure 5.6. The similarity of the two hash values shows the authenticity of the PBDS, Figure 5.6 line 47-54. Later, the hash values are compared with a third hash value generated from the PPT from the re-digitized document, as shown in Figure 5.6 line 56-58. If more detailed verification is needed integrity wise, the full hash from the digital signature is compared with the full text of the document converted into a hash value.

```
46  This algorithm compares three hash values
47  function(Argument hashValue1, Argument hashValue2)
48  {
49      IF hashValue1 and hashValue2 are the same THEN
50          return true
51      ELSE
52          return false
53      END IF
54  }
55
56  If CALL function(hashValue1, hashValue2) is true THEN
57      CALL function(hashValue1, hashValue3)
58  END IF
```

Figure 5.5: Pseudo Code for Comparison of Hash Values for Document Verification

PBDS-B

In the verification process of PBDS-B, compressed text data and digital signature are received from QR code. The compressed text data is combined together to create the original data while the digital signature is stored separately, as shown in Figure 5.7, line 38.

```
36  ELSE IF the architecture used is PBDS-B THEN
37      Scan QR codes
38      Combine data
39      Decompress ditial signature and FPT
40      Decrypt digital signature
41      Convert FPT from QR code into hash value
42      Extract text from redigitalized document
43      Convert redigitalized text (PPT) into hash value
44      Compare hash values
45  END IF
```

Figure 5.6: Pseudo Code for Document Verification Using PBDS-B

The text extracted from a document and text retrieved from different QR codes are converted into a hash value to produce the two hash value needed for verification, as shown in Figure 5.7, line 41 and line 43 respectively. The final hash values is produced by decrypting the digital signature retrieved from QR code, as shown in Figure 5.7, line 40.

The signature verification from now on, follow similar steps as PBDS-A discussed above. As shown in Figure 5.6, the digital signature is verified by comparing the two hash values from the QR code. The comparison of the hash value from the QR code with the hash value from QR code verifies the document. Verification of the document passes if the verifications prove both the signature and the document are authentic.

5.2 Comparative Evaluation

A test is conducted to identify the effectiveness and practicality of the single QR code PBDS model proposed by Warasart and Kuacharoen [2] and multi-QR code based signatures. Fifteen document are created for the experiment. Five of these documents have less than 1000 characters. Another documents five have 1000 to 5000 characters and the final five have 5000 to 10000 characters. When a test is done on the documents there is not tempering/ change after the document is signed. Based on the results of the experiment, the following result is registered.

Table 5.1: Test Result of Different PBDSs

	Single QR code PBDS	Multi-QR code PBDS	
No. of Characters	Single QR Code	PBDS-B	PBDS-A
< 1000	5/5	5/5	5/5
1000-5000	2/5	3/5	5/5
5000 - 10000	0/5	0/5	5/5

5.2.1 A Single QR code Versus Multi-QR code

As shown in Figure 5.1, the verification capability of single QR code PBDS decreases as the document size increases. There are two reasons for this. The first reason is that as the text size increases, the readability of a QR code decreases. Second, when document size is larger than QR code storage capacity even compressed, single QR code PBDS is not able to sign document. Thus, a document that is not signed cannot be verified, that is what we see in the third row of single QR PBDS.

In multi-QR code PBDS on the other hand there is no such readability problem because of document size increase. As we can see from the PBDS-A column, all the documents verified using PBDS-A passed verification.

5.2.2 PBDS-A Versus PBDS-B

The test show that the rate of failure of PBDS-B increases as document size increases (Figure 5.1). On the contrary to Single QR PBDS the increase in the rate of failure of verification is not caused by the decrease in the readability QR codes. Rather caused by human error. The number of QR codes in PBDS-B becomes too large to track during verification as document size increases. This leads to repeated scans or skipping of QR codes, which leads to verification failure.

5.3 User Interface

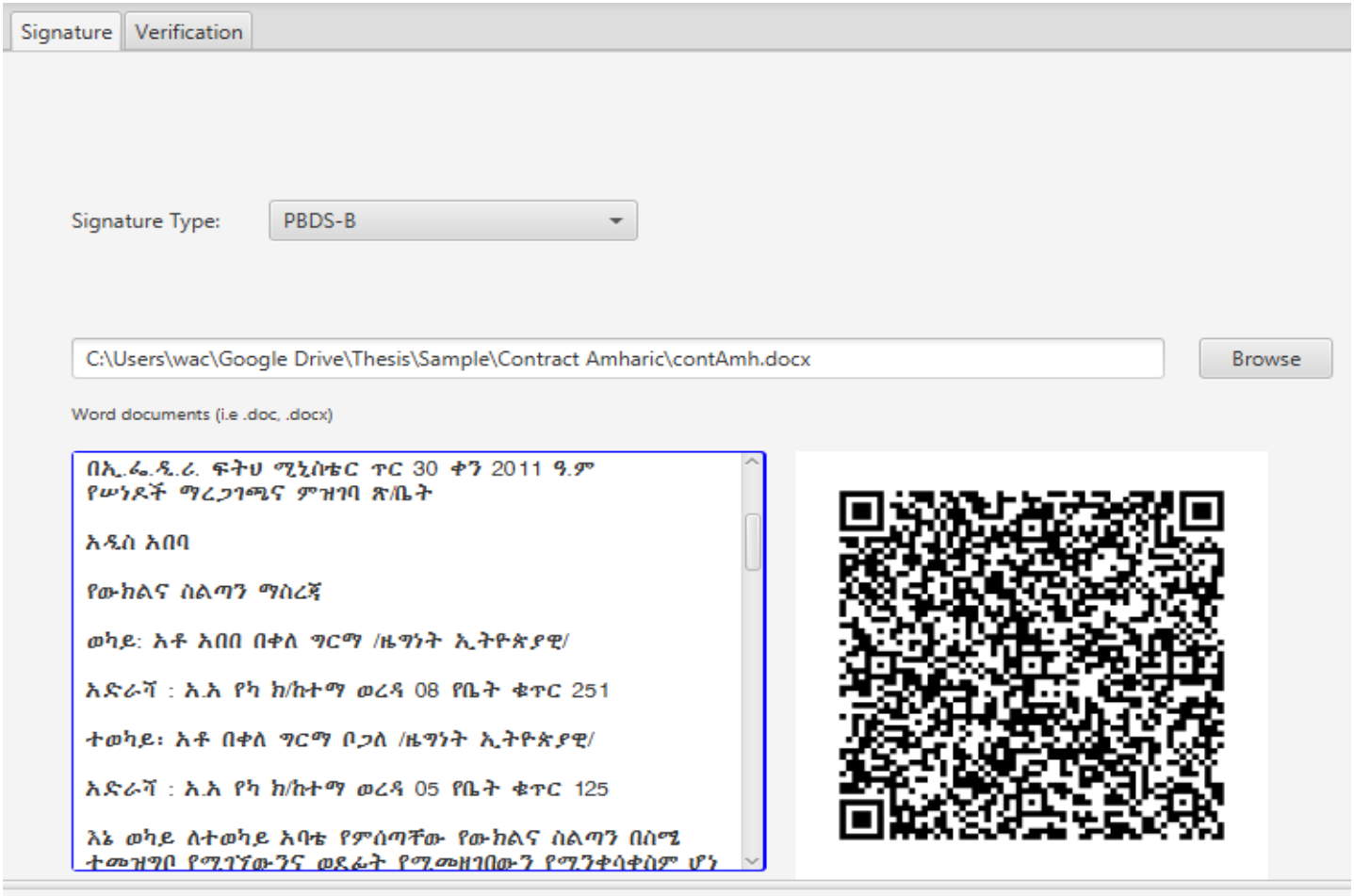

Figure 5.7: Paper Based Digital Signature Generation Tab

As shown in Figure 5.8, there are two tabs in the window being displayed, the signature and verification tabs. The signature tab makes PBDS generation possible. There is a dropdown button to select between PBDS-A and PBDS-B PBDS types. The “Browse” button allows a user to select the document that needs to be signed. The program automatically generates a PBDS when a document is selected and displays the QR code that holds the digital signature on the right and the text of the document on the left side. QR codes generated are then added to documents and printed.

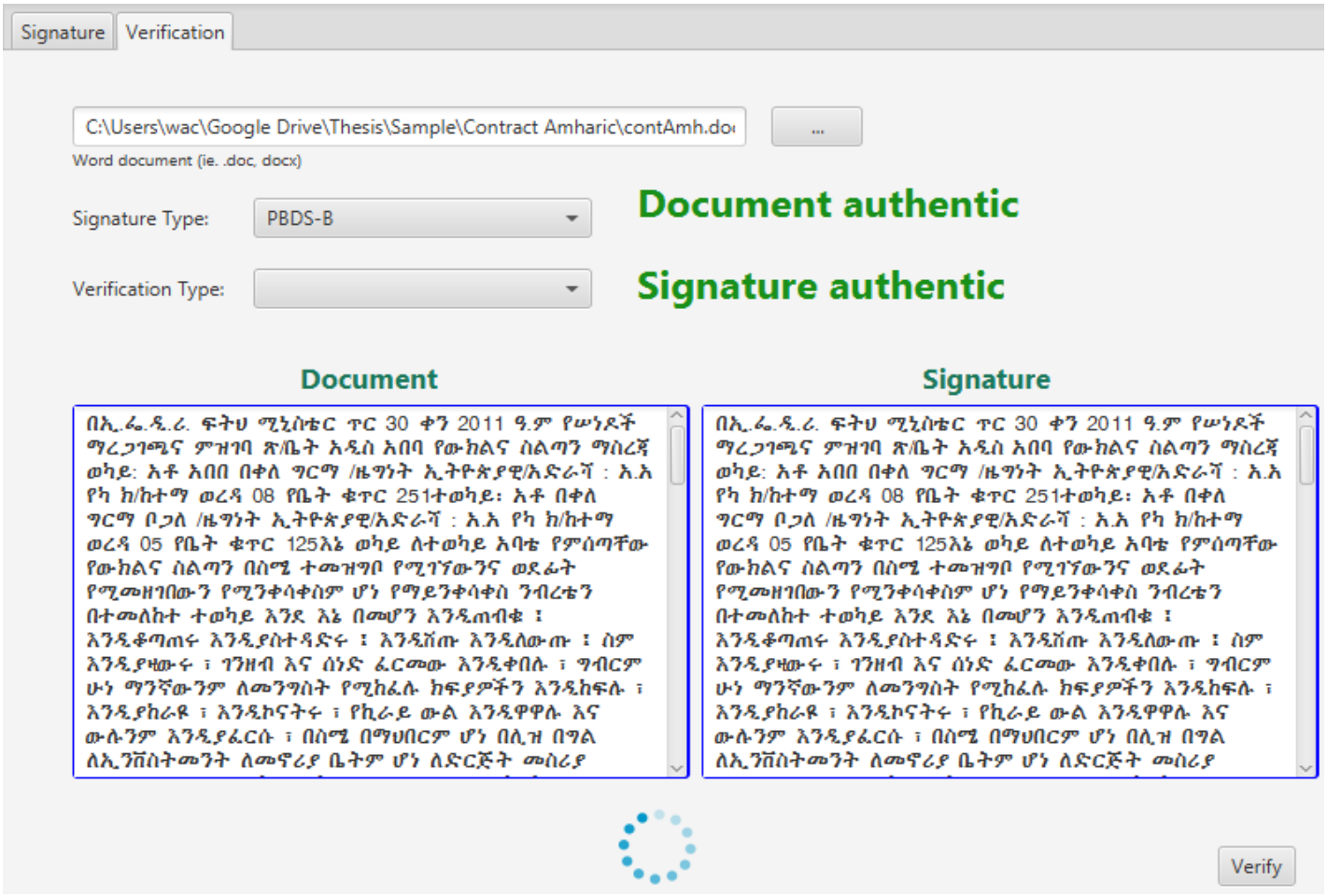

Figure 5.8: Paper Based Digital Signature Verification Tab

As discussed in the Methods section of Chapter one, the QR code scanner is an android mobile application that connects with the prototype via WI-FI hotspot to send data (Figure 5.10).

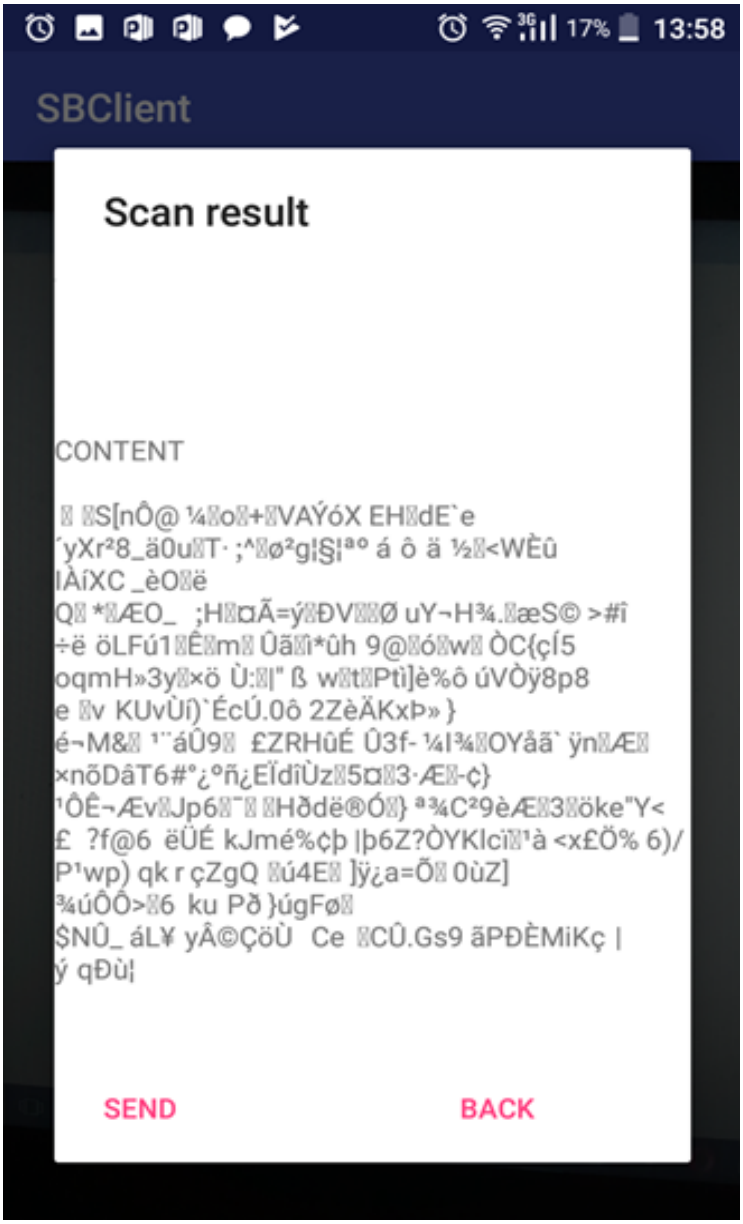

Figure 5.9: Scan Result of QR Code Scanner App Showing Compressed Text

The PBDS verification tab allows user to verify a document. First, the re-digitized document is selected by a user (Figure 5.9). The program automatically converts the document into a hash value. Second, the program accepts compressed data from QR code scanner app via WI-FI hotspot that connects both the computer and a mobile phone that contains the QR scanner app (Figure 5.13). The compressed chunks are combine together, decompressed and finally the plaintext we get from the decompression is converted into a hash value. On another hand, the digital signature that we get after decompression is

decrypted to get another hash value. Finally, when verify button is clicked, the three hash values are compared. The similarity of the two hash values form the QR codes tell us that the PBDS is authentic and signed by a right person. The similarity of all the three hash values further tells us that the document is authentic.

Chapter Six: Conclusions and Future Work

6.1 Summary

This thesis provides a number of solutions for the problems discussed in the statement of the problem section in Chapter one and other problems identified while iteratively working on the solutions. The first solution is a new PBDS multi-QR code architecture called PBDS-B designed to solve PBDS flexibility problem. Existing PBDSs don't change their storage size to incorporate large document. The second solution is a multi-QR code PBDS architecture called PBDS-A, designed to solve problem with manual verification and to reduce the number of QR codes used in PBDS-B. The discovery of these two has modified how paper based documents are signed and verified. Previously, the PBDS signing and verification process involves only single QR code. When it comes to multi-QR code PBDS, the signing and verification processes had to change to adapt to the multi-QR code nature of PBDSs. Finally the multi- QR code architectures and the PBDS signing and verification techniques are incorporated into an overall framework of PBDS.

6.2 Contributions

There are three important things that make this thesis unique. First, it is the first research on PBDSs at framework level. The components involved and the reasoning that binds the components together in the framework make the thesis more detailed and reusable for developing software repeatedly. Second, this thesis improves already existing model by resolving problems observed in the previous model. Third, new PBDS architectures are designed. Because of this new architectures, a modification is made on how PBDS are signed and verified.

Since this thesis is the first on paper based digital signature framework, it covers a number of areas. The followings are the achievements of the thesis:

- ✓ Designed a framework that can be adapted for PBDS system development.
- ✓ Developed a prototype that can sign and verify both PBDS-A and PBDS-B.
- ✓ Identified two types of PBDS, PBDS-A and PBDS-B that can be used in different situations for signing printed documents.
- ✓ Created a selection criteria for PPTs that can be useful when selecting PPTs.

- ✓ Designed a new document format that can ease PPTs detection by OCR programs during document verification.

6.3 Future Work

As any research work, this thesis has its own limitations. The first limitation of this thesis is that every document that passes through automatic verification and identified as fake or tampered, needs to be verified manually. Because of the unreliable nature of OCR programs [21], it is challenging to be sure whether the difference in text in the document and the signature during verification is a result of wrong OCR conversion or the document is actually changed after it is signed. So the only time automatic verification doesn't need the support of manual verification is when the document is authentic.

The second limitation of the thesis is that as the size of the document being signed increases the number of QR codes used as part of the signature increases. Sometimes the signature alone can take multiple pages in the case of PBDS-B.

The third limitation of this thesis is inability of the PBDSs to sign diagrams, pictures and graphics printed on paper based document. Although pictures and graphics in digital documents can be signed using a regular digital signature such as RSA, when the graphics is printed on a paper it is impossible to get the original binary data after digitization of the graphics.

The final limitation is that the thesis doesn't cover multi-signatory digital signature. On other research works multi-signatory digital signatures are considered a completely different technology to single signatory digital signature. There are low efforts to combine single and multi-signatory digital signatures. Therefore, this thesis also continues with the trend considering single signatory digital signatures are different from multi-signatory digital signatures.

Further researches can be conducted to improve PBDSs based on the limitation discussed. A research on a fixed length multi-QR code paper based digital is one approach to reduce the bulky nature of PBDSs in large documents. A research on lossless data compression algorithms and QR codes could lead to another way to reduce the size of PBDSs.

Verification of images and graphics on paper based documents is currently being conducted with the aid of computer vision and machine learning [4]. A research on combining PBDSs, OCR technology and computer vision may result in a new discovery that can help in verification of graphics in printed documents.

Rather than doing a separate research on single and multi- signatory digital signatures, a research can be conducted on a digital signature that can be used by both single and multi-signatories.

References

[1] H. Shaker, G. Jumaa, "Digital Signature Based on Hash Functions", International Journal of Advancement in Engineering Technology, Management and Applied Science (IJAETMAS), Vol. 04, No.1, Jan 2017

[2] M. Warasart, P. Kuacharoen, "Paper-based Document Authentication using Digital Signature and QR Code, *4th International Conference on Computer Engineering and Technology (ICCET 2012)*, 2012

[3] P. Subpratatsavee, P. Kuacharoen, "An Implementation of a Paper Based Authentication Using HC2D Barcode and Digital Signature, *13th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM)*, November 2014

[4] U. Garain, B. Halder, "On Automatic Authenticity Verification of Printed Security Documents", *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, December 2008

[5] A. Singhal, R.S. Pavithr, Degree Certificate Authentication using QR Code and Smartphone, *International Journal of Computer Applications (0975–8887)*, Vol.120–No.16, June 2015

[6] History of QR codes (December 14, 2018). Retrieved from http://www.mobile-qr-codes.org/history-of-qr-codes.html

[7] History of QR codes (December 14, 2018). Retrieved from https://www.qrcode.com/en/history/

[8] J. Lee, T. Kwon, S. Song, J. Song, "A Model for embedding and Authorizing Digital Signatures in Printed Documents", International Conference on Information Security and Cryptology, 2002

[9] Digital Signature: Market Analysis and Trend, *Cygnet Infotech Publication*, 2018

[10] P. Saha, “A comprehensive study on digital signature for internet security”, *ACCENTS Transactions on Information Security*, Vol. 1, No.2, 2016

[11] What are digital signatures? (May 8, 2019), https://acrobat.adobe.com/si/en/sign/capabilities/digital-signatures-faq.html

[12] R. Gupta, N. Ravi**,** “Passport Forgery and Forensic Examination of Indian Passport”, *Journal of Forensic Science*, Vol. 5, No. 5, Sep. 2017

[13] S. Gupta, K. Gupta, A. Singla, **“**Advancement in Indian Passport- A Forensic Perspective”, *International Research Journal of Engineering and Technology (IRJET),* Vo1. 3, No. 11, Nov. 2016

[14] P. Gauravaram, “Cryptographic Hash Functions: Cryptoanalysis, Design and Application”, Ph.D dissertation, Queenland University of Technology, Brisbane City, Astralia, 2007

[15] R. Tripathi, S. Agrawal, “Comparative Study of Symmetric and Asymmetric Cryptography Techniques”, International Journal of Advance Foundation and Research in Computer (IJAFRC) Vol.1, No. 6, Jun. 2014

[16] B. Preneel, “Analysis and Design of Cryptographic Hash Functions”, Ph.D dissertation, University of Leuven, Belgium, Feb. 2003

[17] W. Kotas, "A Brief History of Cryptography" (2000). University of Tennessee Honors Thesis Projects. https://trace.tennessee.edu/utk_chanhonoproj/398

[18] S.R. Kodituwakku, U. S. Amarasinghe, “Comparison of Lossless Data Compression Algorithms for Text Data”, *Indian Journal of Computer Science and Engineering*, Vol. 1, No. 4, pp. 416-425, Dec. 2010

[19] H. Kaur, “A Review of Data Compression Techniques and Data Compression Symmetry”, Journal of Computer Science and Technology, Vol. 4, No. 2, June 2013

[20] J. Lin, C. Fuh, " 2D Barcode Image Decoding", *Hindawi Publishing Corporation Mathematical Problems in Engineering,* Vol. 2013, pp 10, 16 Nov. 2013

[21] A. Chaudhuri, K. Mandaviya, P. Badelia K. Ghosh, "Optical Character Recognition Systems for Different Languages with Soft Computing", *Studies in Fuzziness and Soft Computing,* Springer International Publishing, Sep. 2017

[22] N. Islam, Z. Islam, N. Noor, "A Survey on Optical Character Recognition System", *Journal of Information and Communication Technology-JICT,* Vol. 10 No. 2, Dec. 2016

[23] Signature (6 June, 2019), Retrieved from https://en.wikipedia.org/wiki/Signature

[24] G. Lax, F. Buccafurri, G. Caminiti, "Digital Document Signing: Vulnerabilities and Solutions", *Information Security Journal: A Global Perspective,* Vol. 24, No. 1-3, 2015

[25] J. Lee, T. Kwon, S. Song, L. Song, "A Model for Embedding and Authorizing Digital Signatures in Printed Documents*", International Conference on Information Security and Cryptology, pp 465-477,* Mar. 2003

[26] D. Impedovo, G. Pirlo, R. Plamondon, "Handwritten Signature Verification: New Advancements and Open Issues", *International Conference on Frontiers in Handwriting Recognition,* pp. 367-372, Sep. 2012

[27] N. Victor, "Enhancing the Data Capacity of QR Codes by Compressing the Data before Generation", *International Journal of Computer Applications*, Vol.60, No. 2, Dec. 2012

[28] D. Kleiman, "The Official CHFI Study Guide (Exam 312-49): For Computer Hacking Forensics", 1st ed, Syngress, D. Kleiman, K. Cardwell, T. Clinton, M. Cross, M. Gregg, J. Varsalone, C. Wright, Oct.8, 2007

Signed Declaration Sheet

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

Declared by:

Name: _____

Signature: _____

Date: _____

Confirmed by advisor:

Name: _____

Signature: _____

Date: _____