



Università
Ca' Foscari
Venezia

Corso di Dottorato di ricerca
in Informatica
ciclo XXXI

Tesi di Ricerca

Secure and Usable QR Codes

SSD: INF/01

Coordinatore del Dottorato

Prof. Riccardo FOCARDI

Supervisore

Prof. Flaminia LUCCIO

Dottorando

Heider Ahmad Mutleq WAHSHEH

Matricola 956262



Università
Ca' Foscari
Venezia

PH.D PROGRAMME
IN COMPUTER SCIENCE

CYCLE 31

PH.D THESIS

Secure and Usable QR Codes

SSD: INF /01

COORDINATOR OF THE PH.D PROGRAMME

Prof. Riccardo FOCARDI

SUPERVISOR

Prof. Flaminia LUCCIO

CANDIDATE

Heider Ahmad Mutleq WAHSHEH

ID: 956262

Dedication

Dedicated to...

My beloved father Ahmad, who has always been supporting me with his love My mother Ahlam, for her loving care, My brothers: Mohammad & Yarub, My Sisters: Majd, Reem, Abeer & Ghadeer, whom I have always had near me.

Acknowledgements

First of all I would like to thank Almighty God for giving me the ability to work hard and carry on even in the difficult times. My deep thanks go to my supervisor Prof. Flaminia Luccio for her encouragement, her precious advice and support. Without her help and supervision this work would not have been possible. I also would like to express my gratitude to Prof. Riccardo Focardi for his continuous help and advice.

Deep thanks go to my thesis reviewers: Prof. Stelvio Cimato, and Prof. Marco Furini for their time and experience. To Prof. Andrea Marin for his help, and to the students at Ca' Foscari University of Venice and Jordan University of Science and Technology for their participation in some parts of the thesis experiments. Finally, thanks so much to Nicola Miotello for his kind help.



Contents

Preface	1
Introduction	2
QR Codes Challenges	2
Thesis Structure	3
Summary of Contributions	4
I 2D Barcodes' Threats and Countermeasures	6
1 Background	7
1.1 Brief Overview of Barcodes	8
1.2 QR Codes	11
1.3 QR Codes Usage	13
1.4 Brief Summary of Barcodes Attacks	15
1.5 Basic Security Terminology	15
2 Security Threats and Solutions for Two-Dimensional Barcodes: A Comparative Study	18
2.1 Introduction	19
2.2 Attack Scenarios for 2D Barcodes	19
2.3 Secure Systems Based on 2D Barcodes	23
2.4 Security Enhanced Barcodes and Readers	24
2.4.1 Security Enhanced Barcodes	25
2.4.2 Security Enhanced Barcode Readers	26
2.5 Summary and Comparison	27
2.6 Conclusion and Future Work	31

II	QR Codes Usability	32
3	Usable Cryptographic QR Codes	33
3.1	Introduction	34
3.2	Security of Digitally Signed QR Codes	34
3.3	Usability of QR Code Scanning	35
3.3.1	Usability Parameters	35
3.3.2	Estimating Usability	36
3.4	Digitally Signed QR codes	42
3.4.1	Time and Space Signature Overhead	43
3.4.2	Format Overhead	44
3.4.3	Usability Evaluation	45
3.5	Conclusion	47
4	Towards Evaluating QR Codes Usability and Cryptographic Solutions	49
4.1	Introduction	50
4.2	Experimental Setup	51
4.3	Usability Factors and Results	54
4.3.1	Scanning Time Analysis	55
4.3.2	Data Size, Image Size and Users' Satisfaction	61
4.4	Usability and Cryptography Trade-off	71
4.5	Conclusion	74
III	QR Code Readers	75
5	QR Code Readers: Security and Usability Analysis	76
5.1	Introduction	77
5.2	Related Work	78
5.3	QR Code Readers	79
5.3.1	URLs Security Applications	81
5.3.2	Crypto-based Security Applications	84
5.3.3	Popular Applications	86
5.3.4	Save-Privacy Applications	87

5.4	Design Recommendation	90
5.5	Users' Experiment and Results of Usability and Security	93
5.6	Conclusion	96
	Conclusion	97
	Bibliography	99
	Appendices	109
	Appendix A <i>BarSec Desktop and BarSec Droid Applications</i>	110

List of Tables

1.1	Different 2D barcodes standards.	10
1.2	Description of different data types maximum capacity in QR code	12
1.3	Error Correction Level and tolerate barcode damage	14
2.1	Summary of the attacking scenarios to 2D barcodes.	23
2.2	Summary of the relevant features of solutions, applications and research proposals.	28
2.3	Cryptographic mechanisms and experimental results.	30
3.1	Usability summary.	42
3.2	Overall size example.	45
3.3	Overall size with 200 bytes of data using JSON.	46
3.4	Usability of the cryptographic solutions.	46
4.1	Descriptive summary of the <i>ST</i> for users' satisfaction levels (seconds).	55
4.2	Back transformed mean and 95% CI of the <i>ST</i> for users' satisfaction levels.	59
4.3	Descriptive summary of the <i>ST</i> for the scanning outcome groups (seconds).	59
4.4	Back transformed mean and 95% CI of the <i>ST</i> for the scanning outcome groups.	61
4.5	Summary of users' comments and their categories.	66
4.6	Summary of usability levels.	70
4.7	Authentication and integrity control data overhead for HMAC (bytes).	71
4.8	Confidentiality control data overhead for AES different modes (bytes).	72
4.9	Summary of the Usability and Cryptography QR Codes Solutions.	73
5.1	Details of tested QR Code readers.	80
5.2	URLs security scanners.	83
5.3	Crypto-based QR code scanners.	85
5.4	Permissions of tested QR Code readers.	89

5.5	<i>BarSec Droid</i> specification.	93
5.6	<i>T</i> -test results for <i>BarSec Droid</i> vs. KasperSky.	95
5.7	<i>T</i> -test results for <i>BarSec Droid</i> vs. QR Droid Private.	95

List of Figures

1.1	1D barcode example.	8
1.2	Popular 2D barcodes.	9
1.3	QR Code Main Structure	12
3.1	Measuring the Readability Range (<i>RR</i>) for 300×300.	37
3.2	Measuring the Readability Range (<i>RR</i>) for 500×500.	38
3.3	Measuring the Barcode Readability (<i>BR</i>) for 300×300.	39
3.4	Measuring the Barcode Readability (<i>BR</i>) for 500×500.	39
3.5	Measuring the Misleading Percentage (<i>MP</i>) for 200×200.	41
3.6	Measuring the Misleading Percentage (<i>MP</i>) for 400×400.	41
3.7	Space overhead, in bytes, for RSA and ECDSA signatures.	43
3.8	Time overhead, in milliseconds, for RSA and ECDSA signature verification.	44
4.1	Flowchart of <i>BarTest</i> Application	52
4.2	Screenshot of <i>BarTest</i> Application.	53
4.3	Overall Outcome Percentage of the Experiments.	54
4.4	Histogram for Right scan (seconds).	55
4.5	Users' satisfaction levels distribution.	56
4.6	Scanning outcome groups distribution.	60
4.7	Outcome percentage for 300×300 pixels image size.	62
4.8	Outcome percentage for 200×200 pixels image size.	62
4.9	Right percentage for all image sizes.	63
4.10	Right percentage details for (400×400) Pixels.	64
4.11	Rescan Impact.	65
4.12	Barcode Usability Score (<i>BarScore</i>) for the tested image and data sizes.	70

5.1	Popular QR code scanners with more than 1 M downloads.	87
5.2	Screenshot of <i>BarSec Droid</i>	91
5.3	Screenshot of <i>BarSec Desktop Application</i>	92
A.1	Generate QR code without cryptographic protection.	111
A.2	Digitally signed QR codes.	112
A.3	QR code with HMAC cryptographic protection.	112
A.4	QR code usability level message.	113
A.5	Encrypt QR code contents (password: Italy).	113
A.6	ACL for QR code contents.	114
A.7	Read QR code contents using <i>BarSec desktop application</i> (password: Italy). . .	115
A.8	Process QR code contents using <i>BarSec Droid</i>	115
A.9	Scan QR code using <i>BarSec Droid</i>	116
A.10	Display QR code contents and security summary using <i>BarSec Droid</i>	116
A.11	Display QR code security details using <i>BarSec Droid</i>	117
A.12	Display <i>BarSec Droid</i> warning message.	118
A.13	Display and process ACL using <i>BarSec Droid</i>	119

Preface

The work presented in this thesis is based on some research papers written during my Ph.D. at the Computer Science Department at Ca Foscari University of Venice from September 2015 to December 2018. Chapter 2 presents a joint work with my supervisor Prof. Flaminia Luccio and Prof. Riccardo Focardi published in 2018 as a chapter in the book *Computer and Network Security Essentials*, Daimi K., editor, pages: 207–219. Springer, 2018 [1].

Chapter 3 presents another joint work with my supervisor Prof. Flaminia Luccio and Prof. Riccardo Focardi published in the proceedings of the 19th *IEEE International Conference on Industrial Technology (ICIT - IEEE 2018)*, IEEE, Lyon, France, February 20-22/2018 [2].

Chapter 4 presents unpublished work that will soon be submitted to a journal, and Chapter 5 presents a work that will appear in the proceedings of the 5th *International Conference on Information Systems Security and Privacy (ICISSP 2019)*, Prague, Czech Republic, February 23-25/2019.

Introduction

A barcode is a machine readable image that represents data in parallel lines (one-dimensional, 1D, barcode), or as dots or lines that are arranged in matrix form (two-dimensional, 2D, barcode). Quick Response (QR) codes are the most widely used 2D barcodes in the marketing world, in education and in public services. Users believe that they are simple to use and useful [3]. They are also the barcodes with higher data capacity [4], and may store different types of data, such as numeric, alphanumeric, binary and Kanji characters [5].

Barcodes are used in various scenarios for different purposes. A typical application is to encode a Uniform Resource Locator (URL) that links to a related Web page containing detailed information about a product or service. However, barcodes can be employed in a malicious way to launch multiple attacks, which target the users' security and privacy [1].

QR Codes Challenges

QR codes are practical tools, easy to use, free and very popular, and are used in various applications [6] but there are several security risks associated to them. The main problem is that barcodes are not human readable, and they can only be read using specific scanning devices or smartphones reader applications. Recent studies show that barcodes can be maliciously used to run different attacks such as: phishing, malware propagation, cross-site scripting (XSS), SQL/command injection and reader applications attacks [3, 7–9]. These attacks may violate the security of the smartphones and the privacy of the users.

Although many of the recent studies provide barcode security solutions, they can still have weak points such as using insecure cryptographic mechanisms, short keys and broken hash functions. In some proposals, cryptographic details are not enough to evaluate the barcode security level [10–12]. In the literature, there is no standard technique for providing authenticity, integrity and confidentiality of barcode contents. In addition, any proposed solution should take into consideration both time and size overhead to avoid lowering QR code usability.

In this thesis we take into consideration all these aspects, we study and critically discuss in detail the limit of current solutions and we try to propose developing guidelines for secure and usable barcodes, we also propose new tools that follow these guidelines.

Thesis Structure

This thesis is divided into three parts; *i.* Presents a comparative study of various attacks to 2D barcodes and evaluates the available protection mechanisms. *ii.* Conducts a systematic study of usable state-of-the-art cryptographic primitives inside QR codes. *iii.* Evaluates QR code secure and useful reader applications.

The following is a brief overview of the contents of each chapter:

- Chapter 1 presents background information regarding barcodes (in particular QR codes), and gives a brief description of security terminology used in this thesis.
- Chapter 2 presents a comparative study of various attacks to 2D barcodes and of the available protection mechanisms. It highlights the limitations and weaknesses of these mechanisms, it explores their security capabilities and it revises potential weaknesses and suggests remedies based on the recommendations from the European Union Agency for Network and Information Security (ENISA).
- Chapter 3 presents the first systematic study of usable state-of-the-art cryptographic primitives inside QR codes. By selecting them based on performance, size and security. Tests were conducted to prove how different usability factors impact the QR code scanning performance and the usability/security trade-off of the considered signature schemes were evaluated.
- Chapter 4 extends the usability analysis of QR codes. Extensive experiments that analyze the impact of Scanning Time, data size, image size and users' feedback were performed. Based on ISO 9241, the Barcode Usability Score (*BarScore*) was defined, as an observable and quantifiable value that represents the overall usability, by calculating the average of effectiveness, efficiency and satisfaction. A barcode usability guidance was built for recommended image and data sizes under different usability levels. The digital signature features and encryption mechanisms are compared based on usability and security.

- Chapter 5 presents a comprehensive systematic review of barcode scanner applications. The features of barcode readers were analyzed and classified into groups and their limitations were highlighted. The recommendations for usable, secure and privacy-guaranteed reader application are presented. Finally, a proof of concept Android application is presented and compared with two popular QR code readers, based on a users' usability and security survey.

Summary of Contributions

In Chapter 1, we explore barcode types, their features and capabilities. We discuss QR code specifications, components and supported data types. After that, we present a brief overview of QR code usage and challenges. The last section presents background information of related security and cryptographic techniques mentioned in the next chapters.

Many previous studies discussed the potential risks in using 2D barcodes, and proposed different security solutions against barcodes threats. Our contributions in Chapter 2 can be summarized as follows: we discuss many different works, and present several potential 2D barcodes attacking scenarios. We summarize the available research studies and applications that aim at protecting 2D barcodes. We find that some of them lack of important detailed information such as: key lengths, encryption algorithms and hash functions. We compare the techniques and evaluate their security level. We show that protecting 2D barcodes using cryptographic methods is still an open issue.

In Chapters 3 and 4, we discuss the context of cryptographic usable QR codes. In Chapter 3 we discuss the potential benefits of enhancing QR codes digital signature. We present the results of extensive experiments to determine the impact of usability factors on QR code scanning. We analyze the time and space overhead of a selected set of digital signature schemes, with various key sizes and formats. We evaluate the digital signature schemes with respect to the usability analysis.

In Chapter 4, we perform extensive experiments that analyze the impact of Scanning Time, data size, image size and users' feedback on the scanning experience. We represent effectiveness, efficiency and satisfaction by observable and quantifiable formulas, and measure *BarScore* that represents the overall barcode usability. We provide barcode usability guidance for recommended image and data sizes under different usability levels. We present our Barcode Security Studio (*BarSec*) tool, the first proof-of-concept implementation of secure/usable QR code generator.

We compare and assess multiple popular digital signature and encryption mechanisms based on usability, performance and security.

In Chapter 5 we provide a comprehensive assessment for 28 barcode scanning applications, from security, usability and privacy perspectives. We analyze the features of these applications and classify them into four groups; URLs security, Crypto-based security, Popular applications and Save-privacy. Through the analysis, we highlight the limitations, and conclude that most of these apps do not cover the users' security and privacy needs. We propose design tips for usable, secure and privacy-guaranteed barcode reader applications, and implement *BarSec Droid*, a proof-of-concept Android app that utilizes other applications' advantages and resolves their weaknesses. In order to evaluate our work, we show users' usability and security survey, which we have conducted to evaluate *BarSec Droid* and two popular QR code readers, KasperSky [13] and QR Droid Private [14]. The results show that when following the design tips, the reader's security and usability will be increased, which will affect user experience and security awareness.

Part I

2D Barcodes' Threats and Countermeasures

Chapter 1

Background

1.1 Brief Overview of Barcodes

In this chapter we present a brief summary of QR codes, their components and applications. In addition we compare 2D barcodes capabilities and present a summary of barcode attacks. The last section presents a brief description of security terminology that is used in this thesis.

Barcode is a universal technology that provides visual data representation using series of lines, squares or dots, organized in a specific standard way. Barcodes are represented as small images that can store data with various languages, data types and lengths, used to identify or describe the object that carries the barcode [15]. In order to extract the encoded data, we need a barcode scanner; which is an optical machine that has imaging and processing capabilities (camera and processor). The barcode scanners can be specific devices or smartphone reader applications, and they require a Line-of-Sight to capture the barcode image and retrieve the stored data [15].

A commercial (linear) or one-dimensional (1D) barcode is represented by varying the widths and spacing of specific horizontal lines. Commercial barcodes are widely used to encode specific identification values, such as: product ID [16] and price. Figure 1.1 shows an example of 1D barcode that is used to store a student identification number.



Figure 1.1: 1D barcode example.

The data type and length varies according to the used standard, and popular linear (1D) barcodes include: Universal Product Code (UPC), European Article Number (EAN), Code 128 [17] and Postal Numeric Encoding Technique (POSTNET) [18]. Both UPC and EAN codes support numeric data with fixed sizes, while Code 128 supports variable data lengths and allows encoding alphanumeric data (all ASCII characters) [17]. In addition, Code 39 type allows encoding uppercase letters A-Z, numbers from 0-9, and number of special characters (Space, ., \$, +, - and %), with variable length [16].

Choosing the suitable barcode type depends on multiple factors such as: the supported

character set, the data type and the country. In some contexts, organizations create their own barcode types (not international standards) that support their needs such as: Australia Post barcode, Intelligent Mail barcode (USA post), PostBar (Canada), Pharmacode (Pharmaceutical packaging) and DotCode (track individual cigarette and pharmaceutical packages). However, these types are not standards and have limited applications [19].

Two dimensional (2D) barcodes are machine readable images that enhance many features of the traditional barcode (1D) such as more data capacity and robustness. In order to create barcodes suitable for industrial and economic purposes, developers have generated 2D barcode [15]. Moreover, developers and companies have investigated incorporating 2D barcodes to hold data and smartphones for image processing to be effective, simple and flexible channel to communicate between physical objects (such as paper-based surface) and digital worlds [20].

2D barcodes are represented by horizontal and vertical modules (squares and dots) organized in a specific grid [15]. Compared to 2D barcodes, 1D barcodes are just keys to a database, while 2D barcodes can act as a portable database themselves. As these can be used to display the data on the screen of smartphones or tablets, or can perform several actions without Internet connection [6, 15]. There are different types of existing 2D barcodes, such as Quick Response (QR) code, Data Matrix , Aztec and PDF 417 [5, 21–23]. These types support encoding different data types and commonly are used to encode a URL, contact information, maps coordinates or a physical object description [24–26]. Figure 1.2 shows examples of popular 2D barcodes that are used to store Ca' Foscari University of Venice website.



Figure 1.2: Popular 2D barcodes.

Table 1.1 presents a comparison between different standards [19], and shows how 2D barcode types differ both in the storage capacity and in the practical applications.

Table 1.1: Different 2D barcodes standards.

	QR Code	Data Matrix	Aztec Code	PDF417
Max Capacity (numeric digits)	7,089	3,116	3,832	2,710
Max Capacity (alphanumeric characters)	4,296	2,355	3,067	1,850
Country	Japan	US	US	US
Notes	Most popular 2D barcode. Used for advertising, government and public services, physical access control and mobile payments	Used for marking small containers	Used for patient identification wristbands and medicines	Used in logistics and in governmental applications

The most popular barcode type is QR code, due to its high-speed readability and its capacity to hold a great amount of information in an area of fixed size [5]. It is used in different applications (discussed in 1.3). All the other listed barcodes have been developed in the USA: the Aztec code has a good storage capacity and it is widely used in patient-safety applications. The Data Matrix barcode is commonly used for item marking and can be printed in a small area size, but it has less data capacity. Finally, the PDF417 barcode has a small data capacity and is commonly used in logistic and governmental applications.

1.2 QR Codes

The QR code is a 2D matrix symbol (ISO/IEC 18004, 2000) which was created by Denso Wave to be easily used by several equipment such as fixed and handy scanners and terminals [15]. Nowadays the smartphone camera characteristics can be utilized to read QR codes. The QR code is a 2D matrix barcode which is designed to allow its contents to be decoded at ultra-high speed. Thus, retrieving the encoded information in a QR code takes place in few seconds [6].

When QR code was first invented by Denso Wave in 1994, its main objective was to allow quick automobiles scanning during manufacturing [15]. QR codes nowadays are widely used in a much broader context, such as commercial tracking and mobile tagging. QR code can be defined as a dissemination tool that creates a link between digital technologies and classical “physical” interaction [6].

Since then, QR code has been extensively used due to the limited technological characteristics of the linear (1D) barcode. However, there has been an increasing demand to store more information than a 1D barcode could provide. Actually, the barcode’s swift usability, readability and innovative practical features have led to fulfill the needs of barcode users and facilitate its acceptance among users worldwide. QR codes have become widespread in several fields, for instance in tracking goods, encoding URLs, composing e-mails and adding vCard contacts to the user’s smartphones [6].

QR codes were confirmed as an international standard in 2000 [27]. The second version of the standard was published in 2006 [28] and the current standard version was published in 2015 [5]. QR codes have 40 versions starting from version 1 of size (21 * 21) modules, to version 40 of size (177 * 177) modules [5], while higher versions have more data capacity [5]. According to [5], there are four major types of QR codes; QR code Model 1, Model 2, Micro QR code and QR code. QR code Model 1 presents the earlier version, Model 2 shows some enhancement. Micro QR code is a limited and small size QR code type that support maximum capability of encoding 15 bytes. Currently the term commonly indicates to QR code as a standard universal version of the QR code.

QR code may store different data types, such as numeric (0 - 9), alphanumeric, binary data and Kanji characters [5]. Table 1.2 presents a brief description for the different data types capacity in QR code.

Table 1.2: Description of different data types maximum capacity in QR code [5].

Data Type	Characters Size
Numeric	7,089
Alphanumeric	4,296
Binary	2,953
Kanji	1,817

The quiet zone is the free area that surrounds QR code image. It is used to separate the barcode image from the neighboring environment, and that allows for accurate barcode reading. Typically a QR code image contains two regions: the encoding region and the function patterns region [5] (see Figure 1.3).

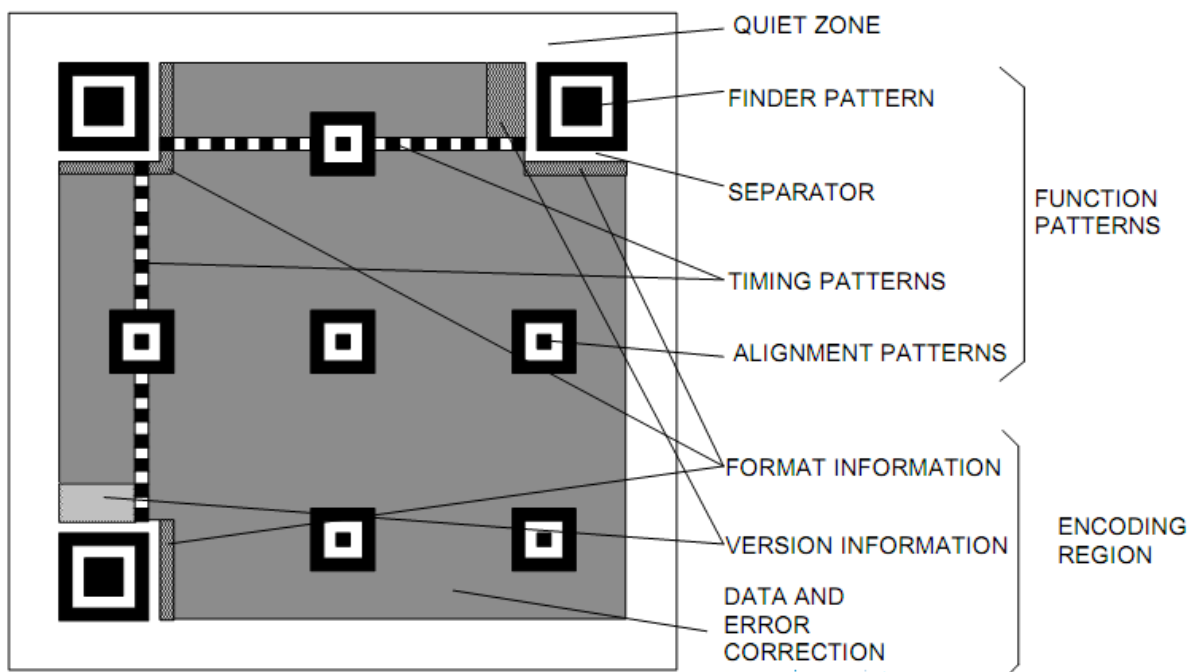


Figure 1.3: QR Code Main Structure [5].

The encoding region contains [5, 29]:

- The encoded data and the error correction: an array of rows and columns, where each cell represents a binary number (1 and 0). This array represents both the encoded data and error correction codes.

- Format information: contains the error correction level and mask pattern (there are 8 masks that controls the distribution of black and white modules).
- Version information: version 1 (21 * 21 modules) to version 40 (177 * 177 modules).

The function patterns region contains the following:

- Finder patterns: three identical squares that appears on the upper left, upper right and lower left corners, which are used by the scanner application to determine the barcode type, and represents a distinctive sign to identify QR codes.
- Separator: blank lines that separate the finder patterns from the rest of the QR code components.
- Timing patterns: two lines, one horizontal and one vertical, help the scanner to determine the position and the width of a single module.
- Alignment patterns: squares that allow the QR scanner to correct distortion when the image is curved.

1.3 QR Codes Usage

QR codes have spread dramatically over the last few years, due to the tablets and smartphones capabilities to scan them easily using various existing applications. They are considered free, simple and effective tools available to all, and capable to store up to 2,953 bytes and retrieve the stored data quickly [4]. Barcodes act as media that provide input data to users or applications. QR codes characteristics establish the way to be employed for various purposes and fields. They represent a communication channel between physical materials and digital world by offering simple and easy information retrieval method. QR codes give the opportunity for their users to navigate among various resources in three main modes; online, offline or combination modes. So when a user reads a QR code, s/he can either reach online website, send an email or read SMS, save contact number, find map coordinates, listen to audio, or watch video [6].

QR codes are adaptable with various environments, by using Reed-Solomon error correction method, which facilitates reading barcodes even if some data blocks were damaged. Reed-Solomon presents a group of redundant bits that try to identify errors, track, and correct them

based on the system itself [5]. QR codes support robust four levels (percentages) of error correction capabilities for restoring destructive data [5].

Table 1.3 shows error correction levels and their capabilities to tolerate possible image damage.

Table 1.3: Error Correction Level and tolerate barcode damage [5].

Error Correction Level	Percentage of tolerate barcode damage
Level L	7%
Level M	15%
Level Q	25%
Level H	30%

The QR code generator can select the suitable level of error correction depending on the type and the importance of encoded data; e.g., higher error correction level (30%) can be used with industrial barcodes that may be severely damaged and distributed in dirty environment. Low level (7%) is preferred with QR codes that are displayed electronically [4, 30]. Medium level (15%) is the most frequently used level for QR Code [15].

QR codes could be attached on any screen, poster or product surface. This enables marketers to establish bridges among advertising products. QR codes can encode the URL of the advertiser to facilitate interaction with users, or retrieve additional product information without typing web address or searching the company name on the Internet [31]. In addition, QR codes can be used to link physical objects to electronic resources [25], which can be effectively used in education, transportation, ticketing services and tourism promotion [6, 32–34]. Furthermore, E-health services can employ QR codes to store patients' information, drugs and medical reports [35, 36]. QR codes allow high-speed component scanning in factories [37]. They have become popular in storing data such as: one-time passwords, Wi-Fi login information [38], bank account information and credit card numbers [39]. QR codes can be used in video games to store game elements and keys [6].

1.4 Brief Summary of Barcodes Attacks

In September 2011 the first malicious usage of QR code was detected by the KasperSky Lab [40]. The attack was performed using a malicious link that was encoded in a QR code: the users were obviously directed to a Web page, where a malware was unconsciously downloaded in the connecting device.

In general, attacks can target barcode scanning devices (e.g., smartphones) by reaching sensitive information such as passwords, contact information, photos, videos, credit card numbers, etc., and can thus violate the users' privacy. Attackers may also take full control of mobile devices by, e.g., accessing E-mail, SMS, etc. [41].

In the last years there has been a large increase in the use of barcodes in everyday life, thus preventing attacks is a necessary and challenging issue. In the literature, there are some proposals and tools to improve QR code security but none of them justifies the architectural choices and the usage of the underlying cryptographic scheme, and often the adopted schemes are vulnerable or deprecated [1]. Attacking scenarios and countermeasures are discussed in Chapter 2.

1.5 Basic Security Terminology

In this section we present a brief summary of cryptographic and information security terms and algorithms. The basic security terminology includes three terms known as the CIA triad: confidentiality, integrity and availability as follows [42, 43]:

- *Confidentiality*: means protecting data from being accessed from unauthorized entities. It is commonly achieved by encrypting data, so that only authorized users who have the key can decrypt and access contents.
- *Data Integrity*: includes assuring that data were not modified by unauthorized entities, and delivered accurately.
- *Availability*: the information system should be available whenever it is needed.

In addition, information security includes the following concepts:

- *Authentication*: in which we aim to verify the identity of users or entities.

- *Non-repudiation*: a security feature which ensures that an entity cannot deny the sending of a message or signing a document.
- *Access Control*: allows to reach data and resources based on roles or specific privileges.

In this thesis, we employ the following cryptographic algorithms and techniques [42–44]:

- *Public key cryptography (asymmetric)*: is a cryptographic system which has two types of keys; public and private. This method is widely used in data encryption and authentication.
- *Digital Signature*: is a security scheme that employs public-key cryptography, to validate authentication, integrity and non-repudiation of a message.
 - *Rivest–Shamir–Adleman (RSA)*: is a public-key cryptographic algorithm that uses mathematical approach based on prime numbers. RSA is widely used for data encrypting and digital signatures.
 - *Elliptic Curve Digital Signature Algorithm (ECDSA)*: is a public-key cryptographic algorithm that is commonly used for digital signatures. It is a variant of Digital Signature Algorithm (DSA), based on the structure of elliptic curves.
- *Symmetric-key encryption*: is a system that uses the same secret key for encryption and decryption. Symmetric-key algorithms are considered simpler and faster than asymmetric, but their drawback is that key exchange should be established in a secure way.
 - *Advanced Encryption Standard (AES)*: is a popular symmetric-key algorithm for encrypting electronic data. AES is a block cipher that was officially published by the National Institute of Standards and Technology (NIST) in 2001. AES supports three key lengths: 128, 192 and 256 bits, while the block size is 128. AES is considered a highly secure algorithm, and it is recommended for substantial applications. AES is used for confidentiality.
- *Digital Certificate*: is used to share public keys that are used in encryption and authentication. It is a signed document from the certificate issuer that proves the ownership of a public key for some entity. Typically, it contains information regarding the issuer, owner's name, owner's public key and related fields.

-
- *Hash Function*: is a function that takes a message as input and return a fixed-size value called “hash”. A hash function is a one-way operation; it is extremely hard to get the original data by processing the hash value. In addition, it is hard to create two messages with the same hash value.
 - *Secure Hash Algorithms (SHA)*: represent a set of cryptographic hash functions with multiple lengths, such as: SHA-2 and SHA-3 (224, 256, 384 and 512 bit variants). SHA hash codes are commonly used for data integrity services.
 - *Hash-based Message Authentication Code (HMAC)*: is a technique that is used to provide authentication and data integrity, based on private key and hash function.
 - *Access Control List (ACL)*: a list that determines which users are authorized to access data and resources.

Chapter 2

Security Threats and Solutions for Two-Dimensional Barcodes: A Comparative Study

2.1 Introduction

The increasing use of 2D barcodes has attracted the attention of cyber attackers trying to break users' privacy by accessing personal information, or to directly compromise users' smartphones and any other connecting device. Thus, understanding possible attacks to barcodes and studying protection techniques is a very challenging and important issue. In 2011, the KasperSky Lab detected the first dangerous attack to a QR code that consisted of encoding a malicious URL inside the barcode, and using phishing and malware propagation to get the users' personal information from the connecting devices [40]. This attack is based on the lack of content authentication and could be mounted, in principle, on all of the most commonly used barcodes.

Previous studies discussed different attacks to 2D barcodes, and proposed various solutions to protect them. The present chapter aims at summarizing all the existing attacks to barcode, and at presenting the available techniques to protect them. All the existing protecting method weaknesses will be highlighted, compared and evaluated based on their security level, and the adopted cryptographic mechanisms. In fact, although many of the available barcode security systems offer cryptographic solutions, they do not always adhere to the latest recommendations and might be still vulnerable due, e.g, to the adoption of deprecated cryptographic hash functions and to the usage of short keys. In some cases, cryptographic solutions do not even provide enough detail to evaluate their effective security. We revise potential weaknesses and suggest remedies based on the recommendations from the European Union Agency for Network and Information Security (ENISA) [44].

The remainder of the chapter is organized as follows: Section 2.2 presents an overview of attack scenarios for 2D barcodes; Section 2.3 revises secure systems in which 2D barcodes are used as a fundamental component; Section 2.4 explores security enhanced barcodes and readers; Section 2.5 summarizes and compares the different studies, and discusses limitations and possible improvements; the last section, presents concluding remarks and future work.

2.2 Attack Scenarios for 2D Barcodes

Barcodes are used in various scenarios for different purposes. A typical application is to encode a URL that links to a related Web page containing detailed information about a product or service. When the barcode is scanned, the link is usually shown to the user that can decide

whether to open it or not in the browser. Barcodes are also used for physical access control, identification and logistics. In these cases, they contain data that are given as input to back-end applications, which interpret them and act consequently.

In general, barcodes are just a way to provide input to users or applications and, since they do not offer any standard way to guarantee content authentication, the input they provide is in fact untrusted. Potential security risks regard the encoding of malicious URLs that look similar to the honest ones and the encoding of data that trigger vulnerabilities in the back-end applications. Moreover, the barcode reader application may become a point of attack since, independently of the use case, the barcode content passes through it and might trigger vulnerabilities directly on the user device.

In the following, we discuss different attack scenarios for 2D barcodes such as phishing, malware propagation, barcode tampering, SQL and command injection, cross-site scripting (XSS) and reader applications attacks.

Phishing. In a barcode phishing attack, the attacker tries to get sensitive information such as the login details and the credit card number of a user, by encoding a malicious Web address inside the barcode that redirects the user to a fake Web page (usually a login Web page). This fake page appears very similar to the legitimate one, thus unintentionally the victim accesses the page and provides the login details to the attacker [7,41]. The study of [3] presents an analysis of QR code phishing, that authors call QRishing. The authors conducted two main experiments, the first one aiming at measuring the proportion of users that scan a QR code and decide to visit the associated Web page, and the second one aiming at understanding the user interaction with QR codes. The results are that the majority (85%) of the users visited the associated Web page, and that the main motivation for scanning QR codes is curiosity or just fun.

Malware propagation. In [8] it is discussed how QR codes can be used by attackers to redirect users to malicious sites that silently install a malware by exploiting vulnerable applications on the device. This is typically done through an exploit kit that fingerprints the device and selects the appropriate exploit and malware. The experiments used crawlers and were run on 14.7 million Web pages over a ten-month period. The crawlers extracted 94,770 QR codes from these Web pages that mainly included marketing products or services. The results showed that 145 out of 94,770 QR codes had a malicious behaviour. They contained attractive words such as: free download and personal/business websites. The authors also found that 94 out of 145 QR codes redirected the users to intermediate sites containing malware that could cause damage to

the users' mobile devices.

Barcode tampering and counterfeiting. Since 2D barcodes are typically used in advertisement and e-commerce to indicate detailed information about the products or to perform the purchase process, an attacker can benefit from the companies reputation by pasting fake 2D barcodes on the real posters. These fake 2D barcodes might advertise false products information or false special offers in which, in fact, the adversary will sell another product to the victims [41]. Interestingly, the study of [4] demonstrates that it is possible to generate 2D barcodes that adhere to multiple standards and that might be decoded, non-deterministically, in multiple ways. One way to achieve this “barcode-in-barcode” is to embed one barcode into another one, so that the decoded content will depend on which of the two is detected by the reader.

The authors show how to embed a QR code, an Aztec Code, and a Data Matrix inside a QR code barcode. The error correction feature of QR codes allow for reconstructing the missing part, so that the hosting barcode is not compromised by embedding another one inside it. The experiments demonstrate that the decoded content depends on the smartphone and reader application used to scan the barcode. This is interesting because it opens the way to stealthy barcode-based attacks that only affect a small number of devices, and are thus harder to detect.

SQL and command injections. The studies of [7, 41] discuss scenarios in which the attacker can encode SQL statements in the barcode in order to attack a database system. The study of [41] refers to automated systems that use the information encoded in the barcodes to access a relational database. If the string in the barcode is appended to the query without a proper input sanitization, the attacker may encode inside the barcodes the SQL commands together with the normal information. For example, this could be done by adding a semicolon ; followed by SQL statements such as: `drop table <tablename>`, causing the destruction of a database table. Similarly the attacker might retrieve or modify sensitive information stored in the database. Both papers also describe possible scenarios in which the content of the barcode is used as a command-line parameter. In this case, it might be possible to directly inject commands and take control of the server host. E.g., in [7] the authors mention how Samsung phones may be attacked by embedding malicious Man-Machine-Interface (MMI) instructions, normally used to change phone settings, into a barcode. Once the barcode is scanned it triggers the execution of these malicious instructions that, e.g., erase all phone data. These attacks happen when developers assume that the information in barcodes cannot be manipulated by attacks and consider it as a trusted input.

Cross-site scripting attacks (XSS). Mobile apps are often based on Web technology and this may allow malicious JavaScript code to be injected into trusted HTML pages, and executed in the app. The simplest case is when the attacker includes JavaScript code into input forms so that, if the server does not sanitize the form data and data are eventually rendered in a page (e.g., as in a blog post), the script would appear and run in the context of a trusted page accessed by the user. This attack is called Cross-Site Scripting (XSS) and can be mounted also using barcodes [45]. The study of [45] discusses risks in HTML5-based mobile applications, in which new forms of XSS attacks using several, unexpected channels, are discussed. For example, authors discuss how the Calendar provider in Android might become a dangerous internal channel in which the attacker inserts malicious JavaScript code that is executed when a vulnerable HTML5-based application displays a Calendar event. The authors show a very interesting example of XSS attack for a barcode reader application. The application reads the QR code and then displays its content to the user. However this is done by putting the content of the barcode in a HTML5 page that is then displayed to the user. This, of course, triggers the attack by executing whatever script is included in the barcode.

Reader applications attacks. During the installation process, many of the 2D barcode reader applications ask for full permissions to access user's smartphone resources such as the device location, the contact list and the photos. If a reader application has a vulnerability that can be triggered by a suitable crafted barcode, this might allow the attacker to get access to private user's data [9]. Table 2.1 summarizes the above attack scenarios for 2D barcodes, classifies the attacks into standard and novel and summarizes the role of the barcode in the attack.

Table 2.1: Summary of the attacking scenarios to 2D barcodes.

Attack scenario	Attack novelty		Role of barcode	
	Standard	Novel	Redirect	Payload
Phishing	✓		✓	
Malware propagation	✓		✓	
Barcode tampering and counterfeiting		✓	✓	✓
SQL and Command Injections	✓			✓
XSS	✓			✓
Reader Applications		✓		✓

In particular, attack novelty indicates to which extent the attack is a novel one, specific for barcodes, or just a variation of a standard attack. The role of the barcode indicates if the barcode is used to redirect to a malicious Web page or if, instead, it contains the attack payload.

2.3 Secure Systems Based on 2D Barcodes

In this section we present some studies that do not focus on how to directly protect a 2D barcode, but on how the barcode can be used as a component of a bigger security system that aims, e.g., at protecting physical documents or operations such as bank transactions. Barcodes may directly enhance security by adding sensitive information into printed documents [46], or may simply provide a human-usable way to implement security protocols, as in the case of [47, 48]. Below, we describe these systems in more detail.

Quick Response - Transaction Authentication Numbers (QR-TAN) [48] is a transaction authentication technique based on QR codes. More precisely, QR-TAN is a challenge-response protocol based on a shared secret and uses QR codes for the transmission of information between the user's computer and the mobile device, which is assumed to be trusted. The protocol works as follows: Transaction data and a nonce (the challenge) from the server are encoded in a QR code which is displayed on the screen of the untrusted computer. The user can use her trusted mobile device to scan it and check that the transaction data are correct. If the user approves, the device secret will be used to authenticate the transaction data together with the nonce through

the generation of an HMAC. The user is required to manually enter the first characters of the (alphanumeric version of the) HMAC into her computer that will send it to the server for the final verification. Since the device secret is shared with the server, the server can recompute the HMAC and check that it is consistent with the fragment inserted by the user.

In [47] it is presented a mobile payment system that is pervasively based on Data Matrix barcodes. Barcodes include product information and merchant URL, so that when a client wants to buy some product, she can scan the barcode and connect to the merchant website. At this point, the client can issue a purchase request which is also encoded as a barcode; the merchant server generates another barcode for the purchase invoice and sends it back to the client; finally, the client sends a barcode payment request to the payment server. All transactions are encoded as barcodes that are digitally signed using Elliptic Curve Digital Signature Algorithm (ECDSA) in order to guarantee authentication.

Authors describe application scenarios for mobile purchasing and payment but no evaluation of the proposed system is provided. CryptoPaper [46] is a system that allows to include secure QR codes in printed documents containing sensitive information. The QR code stores both the encrypted sensitive information and the meta-information which is used for the decryption process. In order to read the QR code, the scanner needs an authorized access to the key which is stored in a cloud database. If the access is granted, the scanner automatically gets the key (through QR code meta-information) and produces the plaintext. Authentication is achieved through a digital signature and confidentiality through AES encryption. Cryptographic keys are stored in the cloud databases. The system allows to include sensitive information in a printed documents and to regulate access through a cloud server. In this way it is possible to dynamically grant or remove access and, at the same time, the cloud server does not have access to sensitive information.

2.4 Security Enhanced Barcodes and Readers

In order to retrieve the encoded data, reader applications are used to scan and process barcode images. We explore some applications that are enhanced to support security features.

We now overview technological solutions and research proposals aiming at improving the security of applications using 2D barcodes. We first revise solutions and studies that extend barcodes through security mechanisms and cryptographic techniques (cf. [subsection 2.4.1](#)).

Then, we describe solutions and research work aiming at preventing attacks directly in the reader applications (cf. [subsection 2.4.2](#)).

2.4.1 Security Enhanced Barcodes

Technology and applications.

Secret-function-equipped QR Code (SQRC) is a type of QR code which can store additional private information, only accessible through a special reader with the correct cryptographic key. One of the features of SQRC is that, when accessed through a standard reader, it is indistinguishable from a normal QR code. There is no publicly available description of SQRC and the official website states that SQRC can only be read by “scanners with the same password (cryptography key) as the one set when the SQRC is generated”. However in a note it is reported that “this function does not provide any security guarantee” [49], which sounds a bit contradictory. However, because of lack of documentation, we cannot evaluate the security of SQRC.

2D Technology Group (2DTG) commercializes a product named Data Matrix Protection/Security Suite (DMPS) [50], based on a patented Barcode Authentication technology [51]. DMPS protects against barcode counterfeiting and data tampering through a symmetric-key based “signature” algorithm.¹ The motivation for adopting this proprietary technology is to overcome the excessive computational load of standard asymmetric key signature schemes. However, as far as we know, there is no security analysis/proof of the patented technology.

Research work.

In [12] it is proposed a tamper detection system for QR codes based on digital signature: a digital signature of the barcode content is embedded into the error correcting area using a steganographic technique. Authors have implemented a prototype and performed experiments finding that the technique could not scale well to QR code version 12. However, they do not give insights about this limitation. Using the steganographic technique they are able to embed just 324 bits of information in the error correcting area. The embedding of actual signatures is left as a future work.

In [52], the author foresees a scenario in which attackers might spam the Internet of Things (IoT) by flooding the physical space with fake or tampered barcodes pointing to unrelated

¹The use of word “signature” for a symmetric-key based algorithm is quite unusual since any entity knowing the symmetric key might provide a valid “signature”.

pages, with the specific purpose of increasing the traffic towards those pages. Independently of the plausibility of the above scenario, the underlying problem is barcode counterfeiting and, more generally, phishing. The proposed solution is to use Elliptic Curve Digital Signature Algorithm (ECDSA) in order to provide authentication, non-repudiation and integrity guarantees of a scanned barcode: the content of the barcode will be trusted only if it contains a valid signature from a recognized content creator. Experimental results on different key lengths and hash functions for ECDSA show a reasonable space overhead but, with respect to our work (chapters 3 and 4), no comparison with other signature schemes is done and there are no considerations about usability. The reported time overhead might also, by itself, break usability. We show that with modern smartphones time is not anymore an issue.

The study of [10] highlights the QR code phishing attack. The researchers embed a fake Google's Web page inside QR codes and perform phishing attack. In their evaluation, they show the possibility of tricking and successfully skipping Google safe browsing service. In other hand, the researchers propose Quick Response Code Secure (QRCS) which is a comprehensive model that uses client-server architecture and depends on using the digital signature. The proposed QRCS model adopts ECDSA algorithm with hash function SHA2 or SHA3 (256 bits) to guarantee QR code generator authentication and data integrity. The proposed model analysis shows the flexibility of implementation and the efficiency against barcodes attacks.

The study of [11] proposes a stereographic scheme to encode Message Authentication Code and digital signature to authenticate data inside QR code. The main advantage of the proposed method is that any barcode reader applications can decode the barcode content. Moreover, the experiments use Universal Message Authentication Code (UMAC) and ECDSA with small key length (160 bits) and results show that proposed scheme performance is better than existing methods.

In [53], a group of students from MIT have performed interesting experiments about enhancing QR codes with cryptography and digital signature. They have also pointed out potential vulnerabilities of two QR code applications: ZXing [54] and SPayD [55].

2.4.2 Security Enhanced Barcode Readers

Technology and applications.

The Norton Snap QR code reader is an Android mobile application which automatically reads the QR code and checks the content to establish the safety of any URL embedded inside

the QR code [56]. The features of the Norton Snap QR code include: identification of safe websites, that are loaded immediately; blocking of malicious, phishing sites, preventing them from being loaded in the browser; expansion of full website address so that users know the final URL before they click it. Norton Snap QR code protects users from phishing, automatic download of malware, and form of frauds where the user is redirected to malicious websites. It does not prevent command/SQL injection, XSS and attacks on the reader application.

Secure QR & barcode reader is an Android mobile application capable of scanning several barcodes [57]. It improves smartphone security by following a simple principle: when installed, it does not ask for permission to access personal information such as user location, contact numbers and photos. This mitigates the consequences of attacks that might leak personal information.

Research work.

The study of [58] investigates the security features of existing QR code scanners for preventing phishing and malware propagation. Authors considered 31 QR code scanner applications. The results showed that 23 out of 31 have a user confirmation feature that gives the user the choice to continue/discontinue visiting the URL, however users typically click on the displayed URL without thinking about the possible consequences. Only two QR code readers out of 31 have security warning features, but authors show that the detection rate is unsatisfactory with too many false negatives. For this reason, authors developed a new scanner, named SafeQR, based on two existing Web services: the Google Safe Browsing API [59] and the Phishtank API [60]. Google Safe Browsing API tests the websites under Google blacklists of phishing and malware URLs, while Phishtank API provides a phishing checking service based on users feedbacks about possible phishing websites. The experiments showed that SafeQR performs a better phishing and malware detection, and has a more effective warning user interface when compared with available QR code readers.

2.5 Summary and Comparison

In this section we summarize, compare and, to some extent, evaluate the various solutions, applications and research proposals discussed in previous sections. In the tables we are going to present, we follow the order in which works have been presented and we refer to each of them through their proper names, when available, or using concise but descriptive ones. We always

include the appropriate citation and a reference to the section in which we have described the work.

Summary of the relevant features. In Table 2.2 we summarize various relevant features of the works: the supported barcodes and whether or not the proposed solution is required to be online; mitigation and prevention of the attacks discussed in section 2.2, grouped by the barcode role (cf. Table 2.1); the provided security properties. Notice that we have grouped authenticity and integrity since solutions that provide one of the two properties also provide the other one.

Table 2.2: Summary of the relevant features of solutions, applications and research proposals.

Paper/Application	Ref.	Barcode		Online	Attack Prevention		Security Properties	
		QR code	Data Matrix		Redirect	Payload	Auth. & Integrity	Confidentiality
QR-TAN [48]	§2.3	✓		✓	N/A	N/A	✓ ^a	✓ ^a
Payment Sys. [47]	§2.3		✓	✓	N/A	N/A	✓ ^a	✓ ^a
CryptoPaper [46]	§2.3	✓		✓	N/A	N/A	✓ ^a	✓ ^a
SQRC [49]	§2.4.1	✓						✓
DMPS [50]	§2.4.1		✓		✓	✓	✓	✓
Enhanced barcode [12]	§2.4.1	✓			✓	✓	✓	
Enhanced barcode [52]	§2.4.1	✓			✓	✓	✓	
Enhanced barcode [10]	§2.4.1	✓			✓	✓	✓	
Enhanced barcode [53]	§2.4.1	✓			✓	✓	✓	✓
Norton Snap [56]	§2.4.2	✓		✓	✓ ^b			
QR&BC Reader [57]	§2.4.2	✓				✓ ^b		
Enhanced reader [58]	§2.4.2	✓		✓	✓ ^b			

^a Properties guaranteed by the system which, in turn, is based on barcodes;

^b Attacks are only mitigated by checking the safety of URLs or by limiting access to resources;

From [Table 2.2](#) we observe that the situation is quite variegated. In particular, some proposals and applications only work if the smartphone/reader is online. This is an important requirement that needs to be taken into account when adopting one of those solutions. Notice that the proposals for enhanced barcodes might use an Internet connection to download missing certificates or to deal with key revocation, however since this does not require a continuous connection we did not mark them as online. The systems proposed in [\[46–48\]](#) do not aim at securing barcodes in general, so attack prevention does not apply here (written N/A). They however give forms for authentication, integrity and confidentiality at the system level (see note *a* in the table). Techniques to enhance barcodes in order to provide authentication and integrity (cf. [subsection 2.4.1](#)) can prevent all the attack scenarios discussed in [section 2.2](#), since the attacker cannot counterfeit or modify barcodes. For these solutions, a tick on Authentication & Integrity implies the two ticks on Attack Prevention. Finally, enhanced barcode readers can only mitigate attacks since, for example, they cannot provide a comprehensive detection of any phishing or malware propagation URL (see note *b* in the table).

Cryptographic mechanisms and ENISA recommendations. [Table 2.3](#) reports the cryptographic algorithms, key lengths, hashes used for digital signatures and the performed experimental results, when available.

Table 2.3: Cryptographic mechanisms and experimental results.

Research paper	Ref.	ECC	EC DSA	RSA	AES	HMAC	key length	signature hash	# tested	delay (ms)
QR-TAN [48]	§2.3			✓		✓	N/A	N/A	N/A	N/A
Payment Sys. [47]	§2.3	✓	✓		✓		256, 128	SHA-2 256	N/A	N/A
CryptoPaper ^a [46]	§2.3				✓ ^b		N/A	N/A	5/test	N/A
Enh. barcode [12]	§2.4.1			✓			N/A	N/A	N/A	N/A
Enh. barcode [52]	§2.4.1		✓				224	SHA-2 224	50	3210
							256	SHA-2 224	50	3290
							384	SHA-2 384	50	7300
							521	SHA-2 512	50	9000
Enh. barcode [10]	§2.4.1		✓				N/A	SHA2 or SHA3 256	N/A	N/A
Enh. barcode [53]	§2.4.1			✓	✓		N/A, 128	N/A	N/A	N/A

^a The proposed system used also asymmetric cryptography but does not provide details;

^b Uses Electronic Codebook (ECB) mode for confidentiality which is insecure [44];

We analyze the results along the European Union Agency for Network and Information Security (ENISA) recommendations about cryptographic algorithms, key size and parameters [44]. In particular, we observe that solutions adopting RSA do not report the key length but it should be noticed that a length of at least 3,072 bits is recommended (cf. [44] section 3.5), which would imply a big space overhead on the barcode. CryptoPaper uses Electronic Codebook

(ECB) mode to encrypt sensitive data bigger than the cipher block, which is considered insecure. Other block cipher modes should be used instead (cf. [44] section 4.1).

2.6 Conclusion and Future Work

In the recent years, the barcode use has spread in most marketing companies around the world. The main aim of these barcodes is to store the information, and to let the customers of the products that contain them to easily read them using smartphones or other scanning devices. There are several types of barcodes with different data capacity storage and this study is dedicated to 2D barcodes. We have discussed many different works, and we have presented several potential attacking scenarios such as: phishing, malware propagation, barcode tampering and counterfeiting, SQL and command injections, XSS and reader application attacks. We have summarized the available research studies and applications that developed and proposed several techniques to protect 2D barcodes. We have found that some of them lack of important detailed information such as: key lengths, encryption algorithms and hash functions. However, other studies provided these details. We have compared the methods, highlighted the limitations and weaknesses of their mechanisms and, to some extent, evaluated their security level. Among other things, our report shows that protecting 2D barcodes against several security threats scenarios using standard state of the art cryptographic techniques is still an open issue. As a future work we plan to investigate new comprehensive solutions for all possible attack scenarios and for different barcodes types, and test them using various cryptographic mechanisms and security parameters, in order to determine the optimal security/feasibility trade-off.

Part II

QR Codes Usability

Chapter 3

Usable Cryptographic QR Codes

3.1 Introduction

In this chapter, we present the first systematic study of usable state-of-the-art cryptographic primitives inside QR codes. We select standard, popular signature schemes and we compare them based on performance, size and security. We conduct tests that show how different usability factors impact on the QR code scanning performance, and we evaluate the usability/security trade-off of the considered signature schemes. Our results show that secure QR codes can be used in practice, but schemes with big size overhead might rise usability issues. Moreover, secure QR codes are denser and cannot be printed on small areas without compromising usability. In particular, we show that in some cases providing a high degree of security breaks usability, and we provide recommendations for the choice of secure and usable signature schemes. We have implemented a proof-of-concept Android app that confirms our findings. In particular, when the scheme and the printing size are chosen appropriately with respect to usability constraints, the QR codes can be scanned without affecting the user experience.

The rest of the chapter is organized as follows: Section 3.2 discusses the security of digitally signed QR codes. Section 3.3 analyzes the performance of QR code scanning with respect to different usability factors, and discusses the results of different usability tests. In Section 3.4, we analyze the overhead of time and space introduced by the addition of cryptographic primitives inside QR codes, using different standard formats, and we evaluate the cryptographic primitive with respect to usability. Section 3.5 draws some concluding remarks.

3.2 Security of Digitally Signed QR Codes

Enhancing QR codes with digital signature prevents the attack scenarios we mentioned in Chapter 2, only when it is not possible for an attacker to perform a legitimate signature. In an open environment this can be hard to achieve, since a Public Key Infrastructure (PKI), similar to the one for the HTTPS protocol, would be vulnerable to the “HTTPS phishing problem”, i.e., attackers that have a valid certificate and use names similar to the one of legitimate entities [61]. However, in a closed/controlled environment, the reader might be configured to only recognize internal certificates and verifying the signature would prove the trustworthiness of the QR code content. For example, a supermarket with its own app might be configured to only use the supermarket’s public key for signature verification.

3.3 Usability of QR Code Scanning

In this section we study the degree of usability of QR code scanning with respect to the size of the payload. This is of ultimate importance in order to establish the maximum space overhead available for cryptographic material. Interestingly, we will see (cf. Section 3.4) that, for some cryptographic schemes, using strong key lengths might compromise usability.

We focus on phone-based readers, as we believe this is the most common use case. Below, we describe the experiments that we performed to measure the average time for scanning barcodes of various sizes which, in turns, can be interpreted in terms of usability.

3.3.1 Usability Parameters

We define QR code usability based on the success and performance of scanning. Along ISO 9241 [62], we consider: satisfaction, the user comfort in terms of simplicity to perform the scanning; efficiency, the time required to perform the scanning; effectiveness, the possibility of successfully scanning a barcode. We have conducted a usability survey among 351 users. Participants were 53.8% males and 46.2% females, 63% were between 18 and 24 years old, 29% between 25-35, 6% more than 35 and 2% less than 18 years old. The online survey proposed some barcodes to users and asked multiple choice questions with free-text answers about their scanning experience such as their awareness of barcode types and trust of contents. Based on the answers we have distilled the following usability parameters.

We first introduce the Readability Range, i.e., the range of distances inside which a barcode is readable.

Definition 1 *The Readability Range (RR) is the difference between the maximum (D_{Max}) and minimum (D_{Min}) distances in centimeters between the scanning device and the barcode, inside which the barcode is readable, i.e., $RR = D_{Max} - D_{Min}$.*

Intuitively, the larger is RR the bigger is the tolerance over the scanning distance, which naturally makes scanning more user-friendly, improving satisfaction. We considered usable a RR of at least 20 centimeters.

For what concerns efficiency, we consider the Scanning Time (ST), i.e., the time, expressed in seconds, required to scan a barcode and extract its content. *Barcode Readability (BR)* classifies the user scanning experience in term of ST . Users of our survey gave the following classification:

Definition 2 Barcode Readability (*BR*) is defined as:

Normal (Excellent) if $ST < 5$;

Reasonable (Good) if $5 \leq ST < 10$;

Hard (Bad) if $10 \leq ST < 15$;

Unreadable if $ST \geq 15$, or failure.

We will call Normally Readable barcodes (*NR*) the barcodes with Normal *BR*. We will consider usable a barcode that is *NR* at least 75% of the times.

We now consider effectiveness in terms of percentage of barcodes that are correctly read. In fact, it might happen that some patterns in a barcode image correspond to another valid barcode. In this case, the application might return the payload from the spare barcode with unexpected consequences [4].

Definition 3 Given a set of barcodes B , let B_M be the set of barcodes that are incorrectly decoded. Then, the Misleading Percentage (*MP*) is defined as $MP = \frac{|B_M|}{|B|}$.

Misleading confuses users and is source of attacks. Our users considered tolerable a Misleading Percentage of at most 5%.

3.3.2 Estimating Usability

In order to evaluate how the various parameters discussed in Section 3.3.1 may affect the usability of barcode scanning, extensive experiments were conducted with different image and data sizes. For the experiments, we have used an Android smartphone (version 5.0.2) with a 1.2 GHz dual-core CPU, 1 GB of RAM and a 8 MP Camera. An application running on a laptop computer generated and displayed 480 different barcodes containing random data that were automatically scanned by the smartphone. In order to simulate human hand shaking, the barcode images were moved and zoomed. To confirm results, more experiments were conducted by humans, so to have real hand shaking in place.

The generated barcodes were of four different sizes: 200×200 , 300×300 , 400×400 , 500×500 pixels that, visualized on a 96 DPI screen, correspond to 5.29×5.29 , 7.93×7.93 , 10.58×10.58 , 13.22×13.22 centimeters. The idea is to cover both barcodes that can be printed on small areas, e.g., in products, and bigger ones that might be printed on advertisements, bus stops, etc. For readability, in the following we will always refer to the size in pixel (assuming

implicitly 96 DPI) but, of course, what matters in the experiment is the actual size. The generated barcodes contain random data of various sizes from 100 to 2000 bytes.

The first set of experiments measures the Readability Range (RR). Figure 3.1 and Figure 3.2 show the measured RR for (300 and 500 pixels) barcodes, where the X-axis represents the data size in bytes and the Y-axis represents the distance between the scanning device and the barcode image.

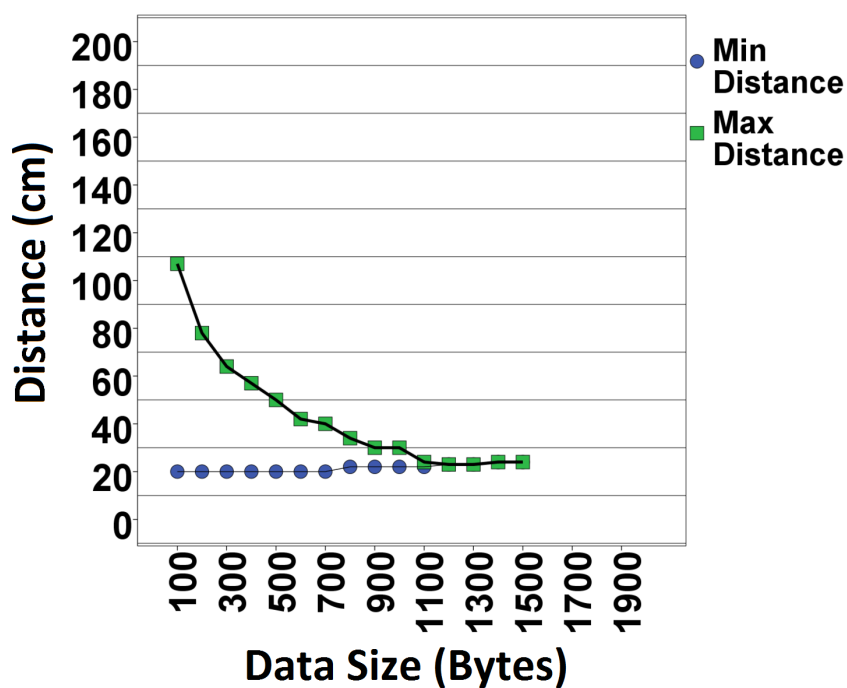


Figure 3.1: Measuring the Readability Range (RR) for 300×300 .

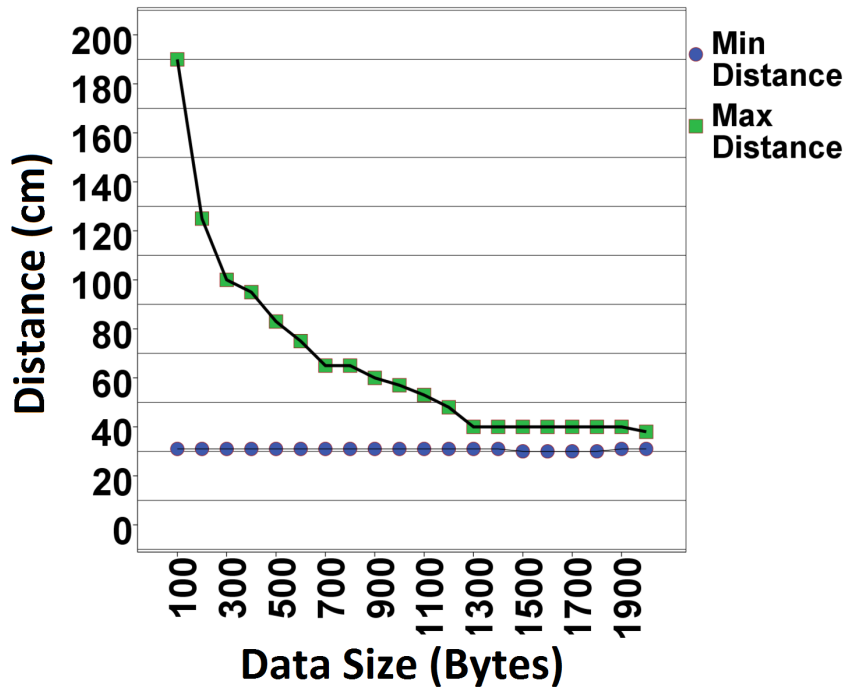


Figure 3.2: Measuring the Readability Range (RR) for 500×500 .

We observe that when the data size increases, the RR becomes narrower. For small data size, barcodes can be read with wide RR , for example the barcode with 100 bytes data size is readable from a distance of 31 to 190 cm ($RR = 159$ cm) for 500×500 barcodes, while the same barcode size with 900 bytes data size is readable from 31 to 60 cm ($RR = 29$ cm), reducing usability. Results also show that the image size plays important role. When the image size is larger, the RR becomes wider for the same data size. For example, if we compare the RR for two barcode images with the same data size (500 bytes) and different image sizes of 300 and 500 pixels, we have that RR is 30 cm and 52 cm, respectively.

The second set of experiments evaluates the relation between barcode data size, image size and the Scanning Time (ST) in terms of Barcode Readability (BR). Figure 3.3 and Figure 3.4 show the BR measurements with 300×300 and 500×500 pixel barcodes.

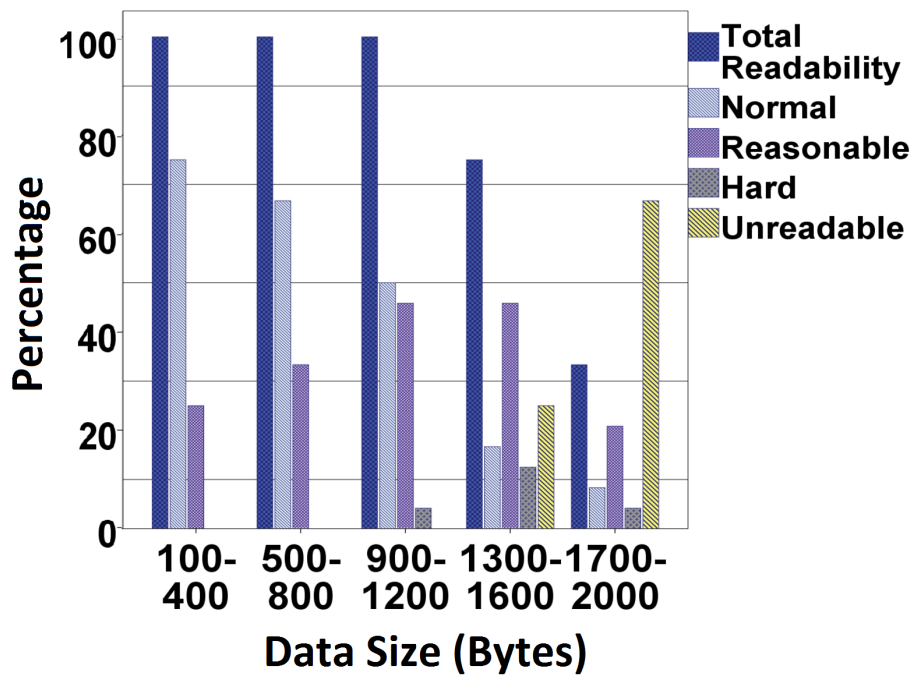


Figure 3.3: Measuring the Barcode Readability (BR) for 300×300 .

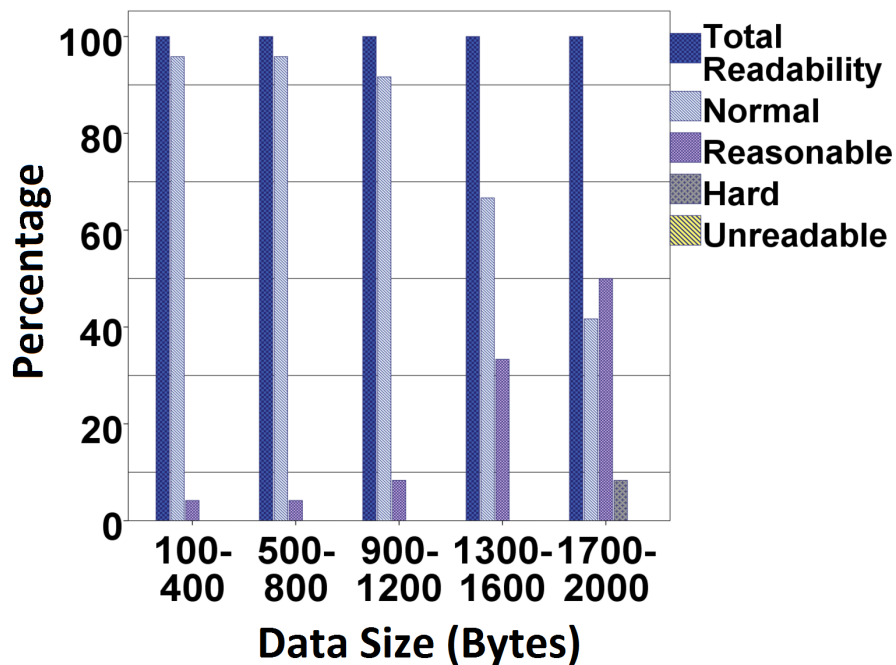


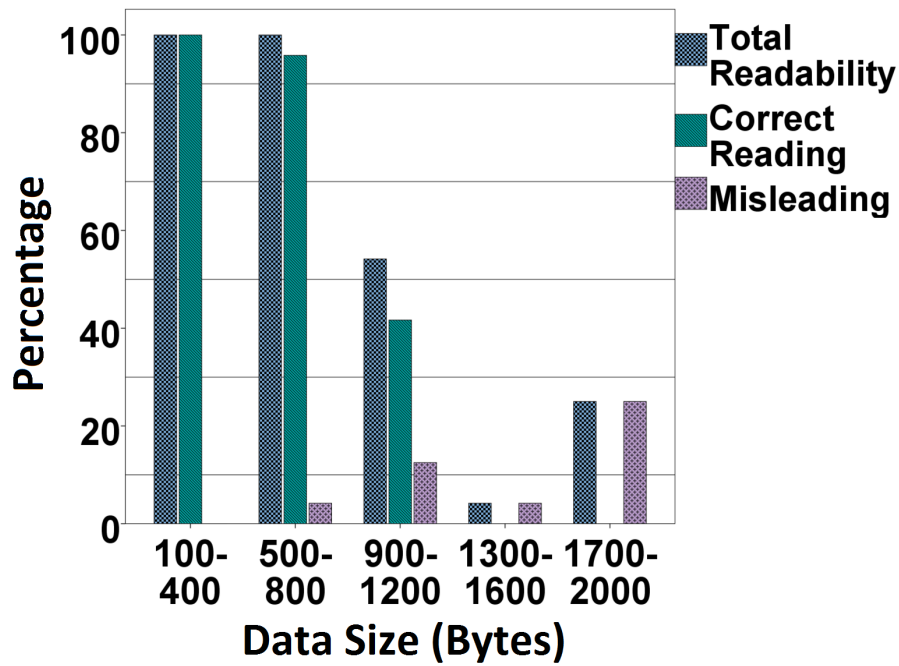
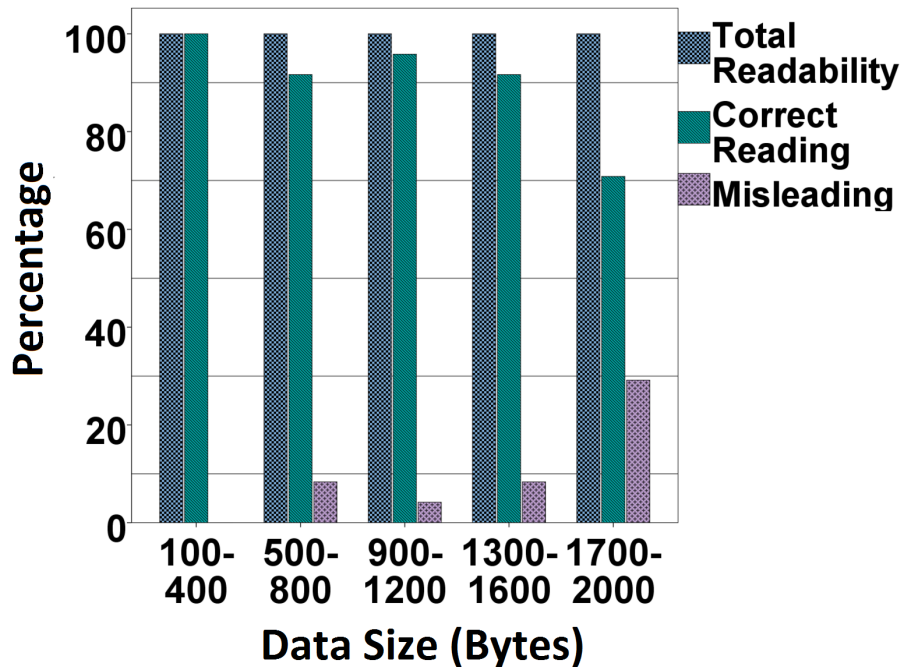
Figure 3.4: Measuring the Barcode Readability (BR) for 500×500 .

The data sizes were grouped into five ranges of 100-400, 500-800, 900-1200, 1300-1600 and 1700-2000 bytes. The X-axis represents the data size groups, the Y-axis represents the measured

BR. E.g., in the first group of 100-400 bytes, the first blue bar represents the total Barcode Readability with a value of 100%. Note that, the total Barcode Readability is the summation of normal (75%), reasonable (25%) and hard (0%), in other words, we include the barcodes with Scanning Time less than 15 seconds (cf. Definition 2). The yellow bar represents the percentage of unreadable barcodes. The summation of the percentage of total Barcode Readability and of unreadable barcodes represents the whole set of barcodes (100%).

We observe that, when the data size increases, readability becomes harder and requires more time, i.e., *ST* increases. For example, barcodes of size 500×500 with 100-400 bytes data size are all readable, and 95.8% of them with normal Barcode Readability (i.e., *ST* less than 5 seconds, cf. Definition 2). When data size increases to 1700-2000 bytes, barcodes are all readable: 41.7% with normal readability, 50% with reasonable readability (5-10 seconds) and 8.3% with hard readability (10-15 seconds). We also notice that larger printing image size gives smaller *ST*'s. For example, comparing the groups 900-1200 bytes in Figures 3.3 and 3.4, we notice that the *ST* has lower values in larger images: 91.6% normal, 8.4% reasonable and 0% hard for 500×500 images with respect to 50% normal, 45.8% reasonable and 4.2% hard for 300×300 images.

The third set of experiments evaluates the relation between barcode data size, image size and the correct barcode decoding. Figure 3.5 and Figure 3.6 show the percentage of correct reading versus the Misleading Percentage (*MP*) for 200×200 and 400×400 pixels barcodes, respectively. The X-axis represents data size groups while the Y-axis represents the fraction of barcodes that are correctly and incorrectly decoded, among the ones that are readable.

Figure 3.5: Measuring the Misleading Percentage (MP) for 200×200 .Figure 3.6: Measuring the Misleading Percentage (MP) for 400×400 .

We notice that MP increases when the data size increases. In fact, barcodes become denser and the probability of misleading barcode decoding increases. E.g., for 400×400 barcodes MP

is 0% for the group of 100-400 bytes and becomes 29.2% for 1700-2000 bytes. Interestingly, *MP* generally decreases when the image size increases, since larger image sizes allow for a more accurate scanning of barcodes with large amount of data. For example, comparing *MP* values for two barcode images of 400×400 and 200×200 pixels with the same data size of 900-1200 bytes, we observe that *MP* is 12.5% for 200×200 barcodes and 4.2% for 400×400 barcodes.

Table 3.1 summarizes the results of our experiments on the usability of QR code scanning. For each of the tested barcode sizes we report the maximum data size, in bytes, that can be included in the code providing a good level of usability.

Table 3.1: Usability summary.

Size	$RR \geq 20$ (bytes)	$NR \geq 75\%$ (bytes)	$MP < 5\%$ (bytes)	Max size (bytes)
200×200	400	400	800	400
300×300	700	400	800	400
400×400	1100	1200	1200	1100
500×500	1100	1200	1200	1100

As we have previously indicated, we require that the Readability Range (*RR*) is at least 20 cm that the percentage of Normally Readable barcodes (*NR*) is more than 75% and the Misleading Percentage (*MP*) is less than 5%. Interestingly, we get similar values for the various parameters. In the last column we pick the lowest one, i.e., the maximum size that is compatible with all the selected parameters.

3.4 Digitally Signed QR codes

In the following we discuss the time and space overhead of selected cryptographic primitives implemented in standard Android smartphone libraries. In particular, in Section 3.4.1 we study time and space overhead of digital signature standard mechanisms; in Section 3.4.2 we discuss time overhead of data formats, which are necessary to embed the cryptographic data together with the QR code payload; finally, in Section 3.4.3 we summarize the usability of QR codes with the various digital signature algorithms and key sizes.

In order to compute the size of digital signatures and the average needed time to verify a signature, we have considered the two most commonly used digital signature algorithms: RSA with key lengths 1,024 bits, 2,048 bits and 3,072 bits and Elliptic Curve Digital Signature

Algorithm (ECDSA) with key length of 256 bits. We use SHA-256 as hash function. For new applications, ENISA [44] recommends a key length of 3,072 bits for RSA and of 256 bits for Elliptic Curve, so we will consider RSA 1,024 as low-secure, 2,048 as medium-secure, and RSA 3,072 together with ECDSA 256 as high-secure. However, it is worth noticing that ENISA recommends to adopt only certain variants of RSA and ECDSA for new applications, i.e., the ones provided with a security proof in a strong computational model. We believe that size and performance are not significantly affected by picking a specific variant. We thus report on results achieved using the default implementations offered by Java 8 (security and cryptographic libraries), and we leave as future work a comparison between the different variants of the signature algorithms. The interested reader can refer to [44] for more detail.

3.4.1 Time and Space Signature Overhead

Figure 3.7 presents the signature lengths for different key lengths of RSA and ECDSA. Notice that, ECDSA signature length is twice the size of key length, i.e., 512 bits = 64 bytes, while RSA signature length is equal to the key length.

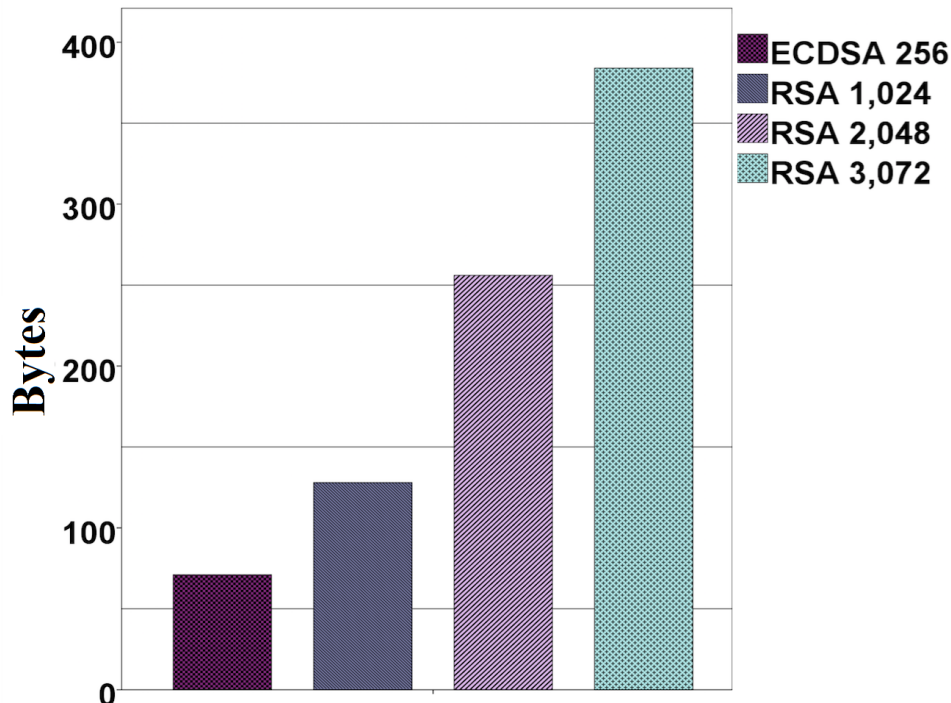


Figure 3.7: Space overhead, in bytes, for RSA and ECDSA signatures.

However, adding a digital signature will require more control data than just the digital signature itself as we will discuss in Section 3.4.2. We have developed an Android mobile

application to test signature's verification overhead. The average signature verification delay is shown in Figure 3.8.

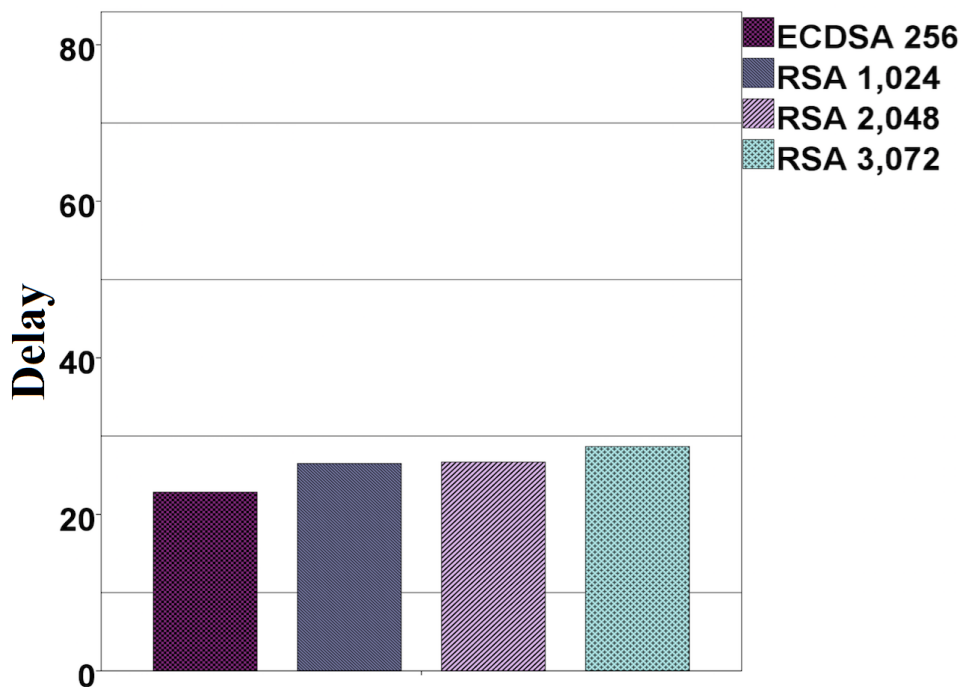


Figure 3.8: Time overhead, in milliseconds, for RSA and ECDSA signature verification.

The tests has been performed on an Android smartphone (version 5.0.2) with 1.2 GHz dual-core CPU, 1 GB of RAM. We notice that, digital signature verification consumes only a small time interval in milliseconds for the various key lengths and algorithms. Thus, the considered digital signature algorithms and key lengths can in principle be used in QR codes in terms of signature size and verification delay. However, size can be critical with respect to usability in some cases, as we will discuss in Section 3.4.3.

3.4.2 Format Overhead

Since we have seen that size influences usability, we will aim at the most concise possible format providing reliable encoding and decoding. We aim at embedding the following:

Payload: The actual data that we want to load in the QR code. It can be offline information requiring no Internet connection, or an URL referencing to an external resource;

Generator: the identity of the QR code generator;

Algorithm: the cryptographic mechanisms adopted;

Signature: the digital signature;

Certificate: the certificate of the QR code generator. This can be included in the barcode or referenced through an URL.

We consider two possible standard formats: JavaScript Object Notation (JSON) [63], and Abstract Syntax Notation One (ASN.1) [64]. JSON is more verbose than ASN.1. For the same data structure we have observed that ASN.1 requires about 75% of the space required by JSON. However, JSON has the advantage of being human readable which might be useful when the QR code is scanned using standard readers, as it would provide a meaningful result anyway. Our best encoding of all the required fields (including the certificate) requires 83 and 104 bytes for ASN.1 and JSON, respectively. Without certificate the overhead reduces to 39 and 54 bytes, respectively. We have used very short but human readable tags. In principle it would be possible to adopt ad-hoc, less verbose, formats but at the price of a less reliable encoding and decoding. So, even if there is margin for improvement, we preferred to adopt standard formats in our study.

Table 3.2 presents an example of the overall size of a signed QR code with and without the certificate. The payload is a 33 byte long URL, and the certificate contains test data.

Table 3.2: Overall size example.

Algorithm	Key length (bits)	with certificates		without certificates	
		JSON (bytes)	ASN.1 (bytes)	JSON (bytes)	ASN.1 (bytes)
ECDSA	256	398	377	151	136
RSA	1,024	588	567	213	198
RSA	2,048	976	955	341	326
RSA	3,072	1,360	1,339	469	454

We notice that including the certificate significantly increases the data size which, in turns, reduces usability.

3.4.3 Usability Evaluation

Table 3.3 summarizes an estimation of the data size for the various algorithms and key size, up to two certificates, using the most verbose format (JSON), and assuming 100 bytes of data for the payload, ID's, etc. and 100 bytes for meta-data of each extra certificate. Without certificates,

we just sum up the size of signature with the overhead for JSON structure plus the 100 bytes of data. For certificates, we have to add one more signature, one public key, JSON overhead and 100 more bytes, and so on. We consider more than one certificate in order to evaluate usability with certificate chains.

Table 3.3: Overall size with 200 bytes of data using JSON.

Algorithm	Key length (bits)	Signature (bytes)	JSON (bytes)	JSON 1 cert. (bytes)	JSON 2 cert. (bytes)
ECDSA	256	64	218	464	710
RSA	1,024	128	282	688	1094
RSA	2,048	256	410	1072	1734
RSA	3,072	384	538	1456	2374

Crossing Table 3.1 with Table 3.3, we obtain Table 3.4.

Table 3.4: Usability of the cryptographic solutions.

Algorithm	Key length (bits)	Signature (bytes)	JSON (bytes)	JSON 1 cert. (bytes)	JSON 2 cert. (bytes)
ECDSA	256	64	✓	✓ ^a	✓ ^b
RSA	1,024	128	✓	✓ ^b	✓ ^b
RSA	2,048	256	✓ ^a	✓ ^b	✗
RSA	3,072	384	✓ ^b	✗	✗

^a Requires at least 400×400 size; fits smaller sizes with small payloads;

^b Requires at least 400×400 size;

ECDSA and RSA 1,024 without certificates are the only ones that fit small and big QR codes. ECDSA with one certificate and RSA 2,048 without certificates are borderline (slightly bigger than 400 bytes), so for small payloads they might provide usable QR codes. All other algorithms except RSA 3,072 with one and two certificates and RSA 2,048 with two certificates

are fine with big QR codes (at least 400×400). RSA 3,072 with one certificate and 2,048 with two certificates are too big and might result in poor usability. Notice that with two certificates ECDSA and RSA 1,024 require bit QR codes. The only algorithm that scales up to 3 certificates is ECDSA. We have implemented a proof-of-concept secure QR code reader for Android based in the Zebra crossing (*ZXing*) library [54]. The reader supports the various digital signature schemes and key lengths discussed in Section 3.4.1. We have tested our implementation on 480 different barcodes confirming our usability evaluation on the various algorithms and key sizes: the app is usable in all of the cases pointed out in Table 3.4.

3.5 Conclusion

QR codes may be subject to attacks in which malicious content is embedded in the barcodes in order to break user's privacy, steal credentials, redirect to malicious websites or install malware. In fact, a QR code is just a medium that provides input and, as such, might easily become source of attacks. Digital signature is a standard effective way to authenticate the barcode content and prevent most of the attacks on QR codes when adopted in closed environments, i.e., when the public keys of trustworthy entities are clearly established. However, it is rarely adopted in this setting since QR codes have limited space and are usually scanned by smartphones that do not generally offer the same performance as personal computers or laptops. This motivated us to perform a systematic study of usable digitally signed QR codes.

First of all we have tested that modern smartphones do not have performance issues for what concerns signature verification. Notice that, this was not the case a few years ago [52]. Then we have considered size issues. QR codes can potentially embed up to about 3 Kbytes which would allow for easily embedding digital signature and certificates. However, we have performed a series of experiments to check in which extent such "big" QR codes can be efficiently scanned, with a reasonable user experience. We have considered the Scanning Time, the distance range tolerated while scanning, and the possibility of spuriously scanning other (simpler) barcodes that appear, by chance, in the QR code.

Our results show that ECDSA and RSA with small keys are usable on QR codes even when printed in small sizes (for example on supermarket products). Bigger RSA keys requires bigger print sizes, and RSA with 3,072 bit keys give usability problems when one certificate is included in the QR code. In fact, when certificates are included, we have pointed out potential usability

issues for all of the experimented signature schemes. Despite these limitations, our results are promising and we have implemented a proof-of-concept Android app that performs the scan of cryptography enhanced barcodes, confirming our findings. We have used standard algorithms and formats so we are confident that our solution might be employed in practice.

As a future work, we intend to study less popular signature schemes to look for potential secure-and-usable alternatives to the popular ones, and we want to extend our solution to also provide confidentiality through encryption.

Chapter 4

Towards Evaluating QR Codes Usability and Cryptographic Solutions

4.1 Introduction

The aim of this chapter is to extend the usability analysis of QR codes that was preliminary conducted in the previous chapter. The main extended features are mentioned in the following points:

- We need further analysis to represent the usability factors (effectiveness, efficiency and satisfaction) by observable and quantifiable formulas. In this chapter we define Barcode Usability Score (*BarScore*); a single value that represents the overall usability, by calculating the average percentage of effectiveness, efficiency and satisfaction.
- We need to extend the analysis of cryptographic primitives in QR codes, such as applying encryption techniques and HMAC.

In this chapter, we perform extensive experiments that analyze the factors that affect the barcodes usability, by developing Barcode Usability Tester (*BarTest*), an Android mobile application that embeds the barcode reader [65]. *BarTest* poses different questions to the users and collects their feedback in order to evaluate the barcode scanning experience [65]. We have analyzed the impact of Scanning Time (*ST*), data size, image size and users' feedback.

We have represented effectiveness, efficiency and satisfaction by observable and quantifiable formulas, and measured *BarScore* that represents the overall barcode usability. We build a barcode usability guidance for recommended image and data sizes under different usability levels, and implement Barcode Security Studio (*BarSec*) tool, the first systematic secure QR code generator that follows our usability recommendations. Then, we compare the digital signature and encryption mechanisms supported by *BarSec* based on usability and security. The obtained results show that QR codes can support powerful, usable and secure solutions.

The rest of this chapter is organized as follows: In section 4.2, we explore the details regarding the design of our experiments. Section 4.3 analyzes the impact of data size, image size, *ST* and users' feedback on barcodes usability. In section 4.4, we compare the digital signature and encryption mechanisms based on their usability and security levels. Section 4.5 draws some concluding remarks and areas for future research.

4.2 Experimental Setup

In order to evaluate the usability of QR codes, we have performed extensive experiments that analyze the users' satisfaction and try to determine the factors that affect the scanning experience, started from the previous chapter [2]. Here we have developed Barcode Usability Tester (*BarTest*) [65], an Android mobile application that embeds the ZXing library [54]. *BarTest* poses different questions to the users and collects feedbacks in order to evaluate the barcode scanning experience [65].

More precisely, *BarTest* records anonymous information such as:

- Experiment ID: random number that identifies the experiment.
- Barcode Number: 4 barcodes for every user.
- Barcode Type: i.e. QR code, Universal Product Code (UPC), European Article Number (EAN), etc.
- Scanning Time: the time required to decode a barcode image.
- Error Correction level: QR codes have four error correction levels that can tolerate image damage.
- User Answer.

Figure 4.1 shows the flowchart of our experiments. Reading a barcode can lead to one of three possible scanning outcomes:

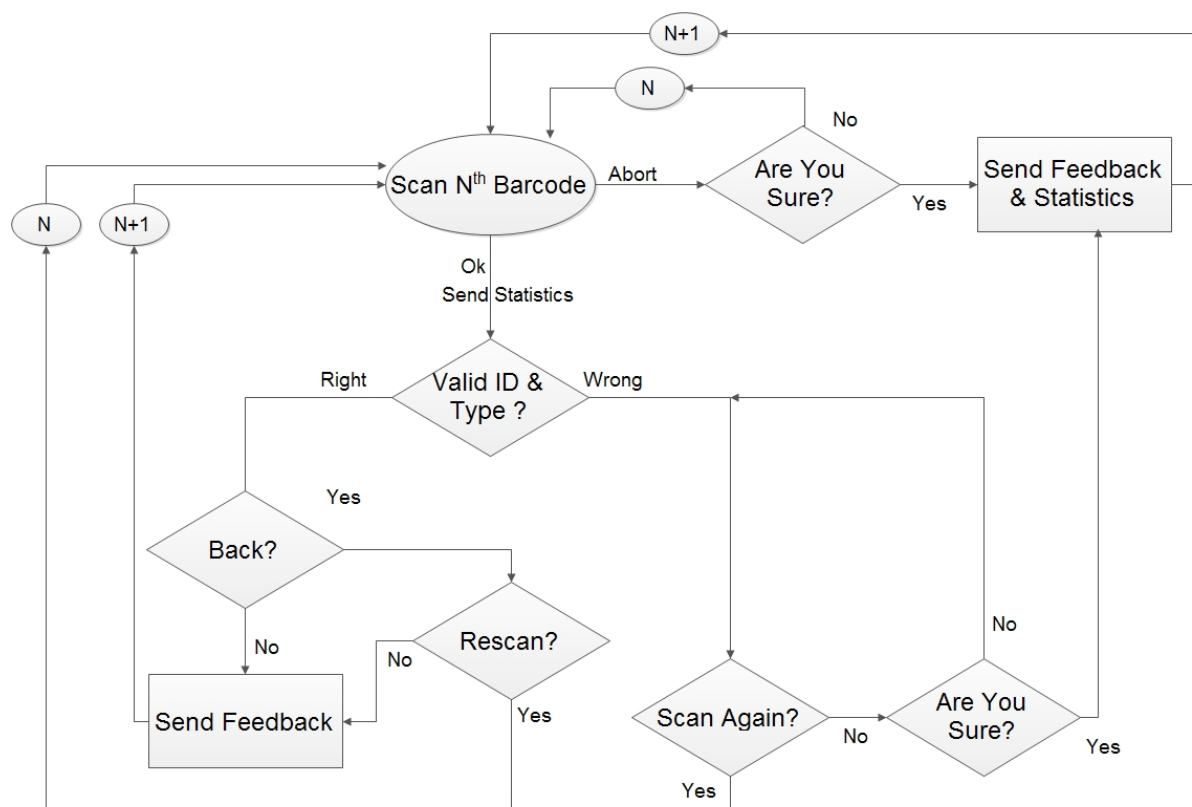


Figure 4.1: Flowchart of *BarTest* Application [65].

- Right Scan: the barcode is successfully decoded.
- Wrong Scan: the barcode is incorrectly decoded as the reader finds some patterns of another barcode type inside a QR code image.
- Aborted Scan: the user cancels the scan.

Note that, *BarTest* application alerts the users about Wrong and Aborted scans, and offers a rescan choice before asking for users' feedback [65]. After each Right scan the user is asked about her/his satisfaction level. The positive answer is defined in a scale of three levels and might be: Excellent, Good or Bad. In case the answer is different from Excellent, the user is also asked to elaborate the negative answer more with one or more of the following replies:

- I had to move the phone many times.
- It took too much time.
- I had to rescan the barcode many times.
- Other (filled by the user).

After each Wrong and Aborted scans, the user will have the opportunity to rescan the barcode, and in case she/he does not accept to do it, the application will ask the same questions of Good

or Bad scans. Figure 4.2 presents a screenshot of the *BarTest* application.

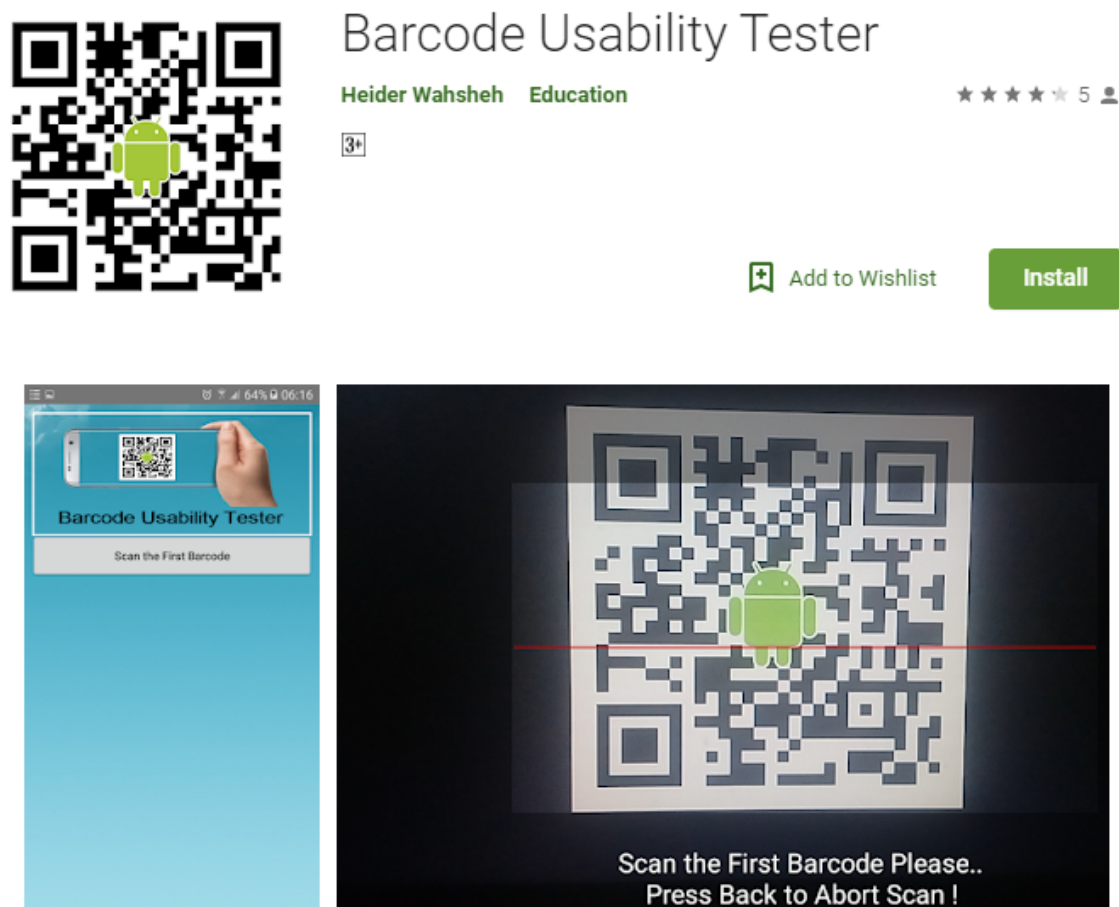


Figure 4.2: Screenshot of *BarTest* Application.

QR codes. To run the experiments, we have printed different QR code images, divided into five groups of data sizes: 100-400, 500-800, 900-1200, 1300-1600 and 1700-2000 bytes. The generated barcodes were printed with four image sizes: 200×200 , 300×300 , 400×400 , 500×500 pixels that, visualized on a 96 DPI screen, correspond to 5.29×5.29 , 7.93×7.93 , 10.58×10.58 , 13.22×13.22 centimeters.

Instructions: Users were instructed as follows:

- Open Play Store and type: barcode usability tester.
- Download the application.
- Allow camera permission.
- Push scan button when you are ready to scan.
- You will get warning from the app if some barcodes are incorrectly decoded.

4.3 Usability Factors and Results

The experiments were conducted with the help of 149 undergraduate computer science students (volunteers). They used their modern smartphones, where various device models and capabilities were tested for comprehensive usability evaluation. For example, the camera resolution for most of the devices was around 8-16 MP, which provides variety to our sample.

Scanning outcome: The overall outcome percentages of scanning are shown in Fig. 4.3, in which we can observe that:

- Totally, the users aborted only 8.5% of the scanning operations, while they were able to read barcodes in 91.5% of the cases, divided into 65.9% Right and 25.6% Wrong scans.
- The right percentage represents about two thirds of the scans and is divided into: Excellent (38.8%), Good (18.4%) and Bad (8.7%) scans.
- About one third of total scans failed to decode barcodes correctly (Wrong plus Aborted scans).
- Wrong scans recorded 25.6% of total scans, and this high percentage highlights the possible risk of the Misleading problem that affects barcodes usability.

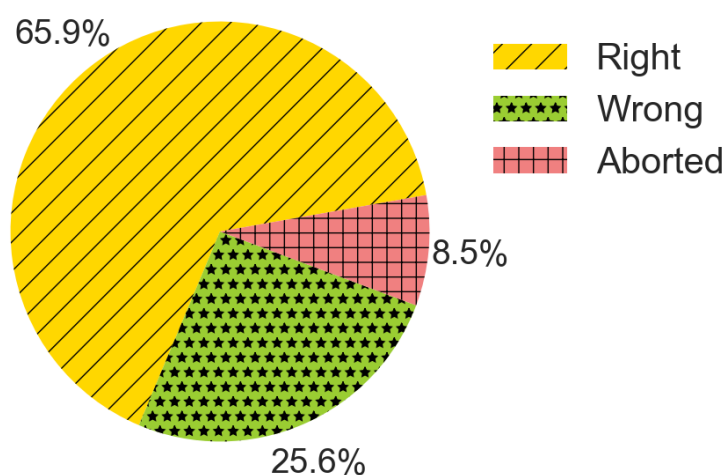


Figure 4.3: Overall Outcome Percentage of the Experiments.

4.3.1 Scanning Time Analysis

The data distributions of the Scanning Time are shown in Fig. 4.4 for Right outcome. The histograms reveal skewed to the right distributions for all the 6 cases (Right, Wrong, Aborted, Excellent, Good and Bad). This implies that the median and InterQuartile Range (IQR) are more suitable descriptive measurements than the conventional mean and standard deviation. Since the distribution is skewed, then the mean is usually not in the middle. Better measure of the center for this distribution is the median [66].

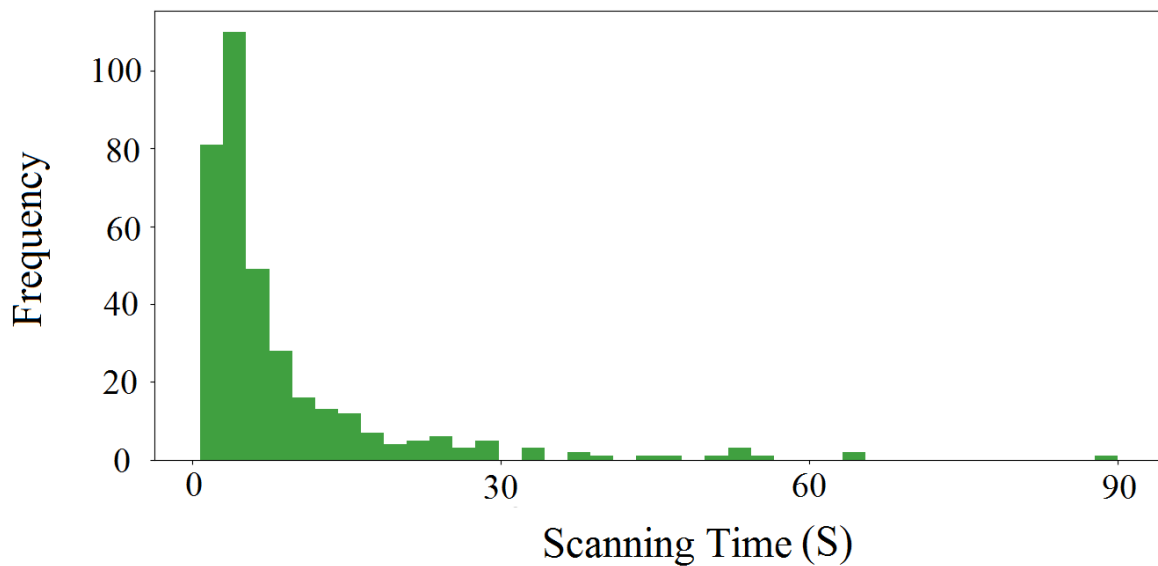


Figure 4.4: Histogram for Right scan (seconds).

We have measured the median Scanning Time, the first and third quartiles Q1, Q3, the IQR (Q3–Q1), the minimum non-outlier and the maximum non-outlier and compared these measures for the users' three satisfaction levels: Excellent, Good, and Bad as shown in Table 4.1.

Table 4.1: Descriptive summary of the *ST* for users' satisfaction levels (seconds).

Satisfaction Levels	Minimum non-outlier	Q1	Median	Q3	IQR	Maximum non-outlier
Excellent	0.85	2.7	4.1	5.9	3.1	10
Good	1.4	4.7	7.3	14.4	9.7	26.7
Bad	0.78	5.6	14.4	27.4	21.8	56

Visually, the relationship between these satisfaction levels and the *ST* is shown in the

boxplots in Fig. 4.5.

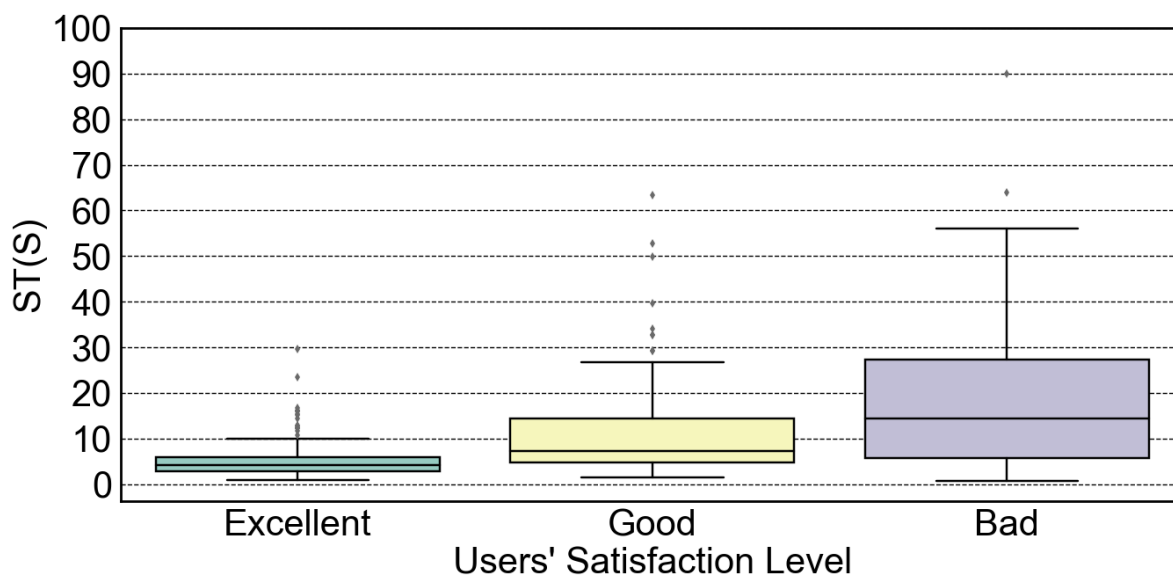


Figure 4.5: Users' satisfaction levels distribution.

By analyzing Fig. 4.5, we can notice that the Excellent readings were more centered around their median point 4.1 seconds and the majority of them recorded less than 10.5 seconds delay, with some outliers (16 scans out of 209) that exceeds 10.5 seconds. Good and Bad readings show Minimum values close to the Excellent scans, but with wider range of points distribution. The majority of Good scans recorded less than 28.9 seconds with some outliers (9 scans out of 99). The Bad maximum non-outlier reached 56 seconds with few outliers (2 scans out of 47).

The question we asked at this point, are these observed differences between the satisfaction levels significant? Or is it that the nature of these observed differences is only due to chance variation? The answer for these questions comes from conducting a test for comparing population means that is the Analysis of Variance (ANOVA) test.

ANOVA is statistical method used to test the differences between population means. In describing an ANOVA design, statisticians use the term factor or treatment as a synonym of independent variable, which is the explanatory, categorical variable that is tested for difference between its levels. The dependent variable is called response which represents the quantitative continuous variable that is measured for each level of the factor variable [67].

Although it may seem odd that the technique is called "Analysis of Variance" rather than "Analysis of Means", the name is appropriate because inferences about means are made by analyzing variance [68]. The essence of the test is based on testing whether several populations have the same mean by comparing how far apart the sample means are with how much variation

there is within the samples [67]. The sources of variation in the response variable ANOVA studies is based on two estimates of the population variance (σ^2), one estimate is the mean square error (MSE) and it is based on differences among scores within the treatment levels. MSE estimates σ^2 regardless of whether the null hypothesis is true (the population means are equal). The second estimate is called the mean square between (MSB) and is based on differences among the sample means. MSB only estimates σ^2 if the population means are equal. Thus, if the population means are not equal, MSB estimates a quantity larger than σ^2 . Therefore, if the MSB is much larger than the MSE, then the population means are unlikely to be equal. On the other hand, if the MSB is about the same as MSE, then the data are consistent with the hypothesis that the population means are equal [68].

Mathematically, the analysis of variance F statistic for ANOVA has the form of: $F = \frac{MSB}{MSE}$ with a corresponding p -value that indicates the probability of occurrence [67]. P -value are used to determine whether a null hypothesis will be accepted or rejected, the most studies refer to statistically significant as p -value < 0.05 [69].

For our problem, we conducted a one-way ANOVA (a one-way ANOVA is a design in which there is only one factor; in our case the users' satisfaction level) to compare the distributions of the response variable Scanning Time for the three factor levels we have: Excellent, Good, and Bad. The aim is to determine whether the differences of the population means among the three levels are statistically significant, or not [67].

Like all inference procedures, ANOVA is valid only in some circumstances. The conditions under which ANOVA can be used were tested for our data, and corrective actions were taken to guarantee that the ANOVA results are not biased. The ANOVA conditions are:

- The samples are independent Simple Random Samples (SRSs).
- The populations have the same variance (homogeneity of variance assumption).
- The populations are normally distributed.

Due to the way we designed the experiment and collected the data, we can claim that the first condition is not violated. The second condition assumes that the variability of observations is the same in all populations. ANOVA is not too sensitive to violation of this condition, and if the samples estimates of the population variance is similar in all the factor levels, the condition is validated [67]. For the third condition, fortunately, the ANOVA test (like other procedures for comparing means (μ)) is robust to the populations' normality condition, however, it requires

us to test for the normality of the samples as an estimate to the populations. Because our data showed skewed to the right distributions with some outliers, we conducted a data transformation (the logarithmic transformation, Log base 10) to better fit the normality condition [70]. The logarithmic transformation was applied on each instance of the data, then, the distributions were evaluated and showed less skewed, and more normal distributions. At this stage we applied the ANOVA on the Log-transformed data hypothesizing these null and alternative hypotheses:

- The null hypothesis: the population means of all levels under consideration are equal.

Mathematically, it can be represented as:

$$H_0 : \text{Log}(\mu_{\text{Excellent}}) = \text{Log}(\mu_{\text{Good}}) = \text{Log}(\mu_{\text{Bad}})$$

- The alternative hypothesis: At least one of the population means is different.

$$H_a : \text{not all } \text{Log}(\mu_{\text{Excellent}}), \text{Log}(\mu_{\text{Good}}), \text{Log}(\mu_{\text{Bad}}) \text{ are equal}$$

A one-way analysis of variance (ANOVA) was conducted to test the hypothesis of whether the means of the Log-transformed Scanning Time (*ST*) of the three users' satisfaction levels would considerably vary or not. Note that *F* parameters are (K-1, N-K), where K is the number of groups and N is the total number of scans. The analysis result showed a significant difference $F(2, 352) = 52.23, p\text{-value} = 0.000$. Even though we have done the ANOVA test on the Log-transformed variable, the results are back transformed (raised 10 to the power of each number) and reported in the original units for better interpretation, as recommended in [70]. Table 4.2 shows the results of the one way ANOVA test we conducted, where N is the number of scans and C% is the Confidence Interval. CI is the probability that the interval will capture the true population value in repeated samples. That is, the confidence level is the success rate of the method. As we are estimating the value of population parameters, the statistical inference (measured by the Confidence Interval) provides a method of drawing conclusions about the population from the sample data. C% value is user-defined, usually a 90% or higher is chosen. The most common confidence level used is 95% [67]. A 95% CI was chosen here for our ANOVA test. So, we can be confident 95% of the time that the population mean of the Excellent users' satisfaction level lies within this interval (3.7, 4.6).

Table 4.2: Back transformed mean and 95% CI of the *ST* for users' satisfaction levels.

Satisfaction Levels	N	Mean	95% CI
Excellent	209	4.1	(3.7, 4.6)
Good	99	8	(6.9, 9.4)
Bad	47	12.4	(9.9, 15.4)

The p -value corresponding to the F statistics showed significant difference 0.000 when compared to $\alpha = 0.05$ (α is the error level we chose along with the 95% CI). This leads us to reject the null hypothesis and conclude that there is a strong evidence that the three population means of the users' satisfaction levels are significantly different. We can notice that the Confidence Interval of the mean of each of the three levels of users' satisfaction do not overlap with the Confidence Interval of the other groups. This indicates that the user's satisfaction levels are distinct, and can be differentiated from each others.

Table 4.3 shows quartile values (Q1, median, Q3 and IQR), minimum and maximum non-outlier for scanning outcome groups and Fig. 4.6 presents the boxplot of scanning outcome distribution.

Table 4.3: Descriptive summary of the *ST* for the scanning outcome groups (seconds).

Outcome	Minimum non-outlier	Q1	Median	Q3	IQR	Maximum non-outlier
Right	0.78	3.2	5	9.3	6.1	18.4
Wrong	1.9	8.4	15.8	29.7	21.3	61.1
Aborted	2.75	22.8	34.4	53.4	30.6	95.6

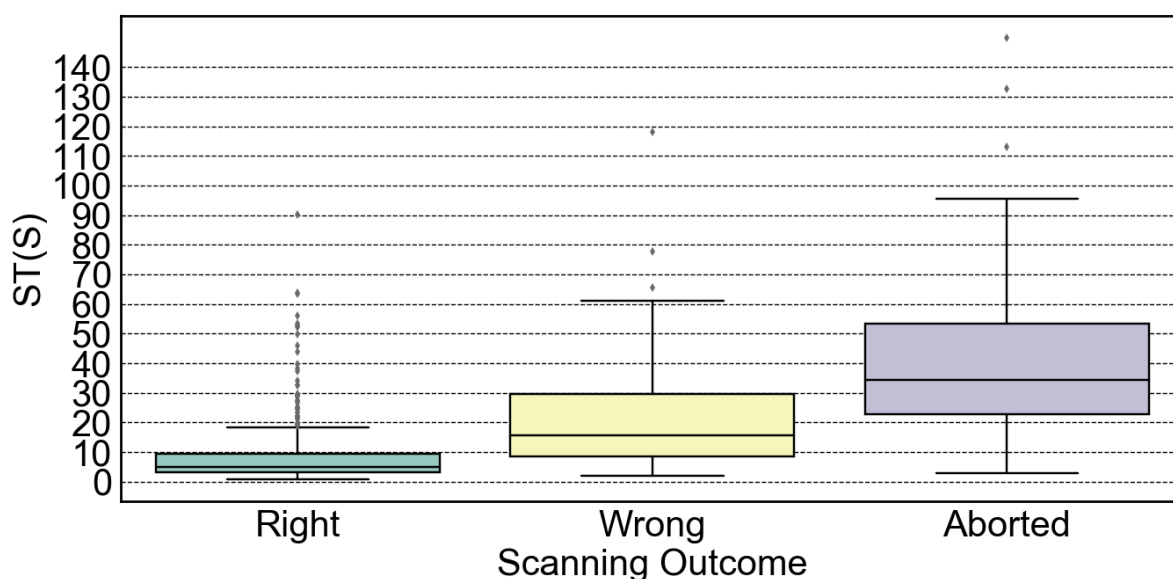


Figure 4.6: Scanning outcome groups distribution.

By analyzing Table 4.3 and Fig. 4.6 we can observe that:

- The Aborted scans recorded the highest ST values (median 34.4 seconds), compared to Right (5) and Wrong (15.8). This indicates that the users were patient and gave the scanner the enough time to read barcodes. The Aborted maximum non-outlier reached (95.6 seconds) with few outliers (3 scans out of 46).
- The Wrong scans recorded higher ST than the Right ones. A remarkable note is that when trying to read a QR code, the ZXing [54] focuses on the finder patterns and tries to decode the image as a QR code. If the scanner after a period of time cannot successfully decode it, it searches for other barcode types and may decode a spare barcode (Wrong case). This note highlights the effect of scanner's algorithm in detecting the right types of barcodes. The Wrong maximum non-outlier recorded 61.1 seconds with few outliers (3 scans out of 138).
- The median of the Right scans (5 seconds) was close to the median of Excellent (4.1) and Good (7.3) (as shown in Table 4.1), due to the small number of Bad scans. This affects the Right points distribution, where the majority of them recorded less than (18.4) seconds delay (maximum non-outlier of Right), with more outliers (39 scans out of 355) that represent the Bad scans and part of the Good scans.

ANOVA was conducted to test the hypothesis of whether the means of the Log-transformed Scanning Time of the three scanning outcome groups would considerably vary or not. The analysis result was significant, $F(2, 536) = 118.56, p = 0.000$.

Table 4.4 presents the back transformed mean and 95% CI for the *ST* for the different scanning outcome groups. We can notice that the Confidence Intervals for the three groups are distinct and scanning outcome groups did not overlap.

Table 4.4: Back transformed mean and 95% CI of the *ST* for the scanning outcome groups.

Outcome	N	Mean	95% CI
Right	355	5.8	(5.2, 6.3)
Wrong	138	15.5	(13.4, 18)
Aborted	46	30.5	(23.6, 39.3)

4.3.2 Data Size, Image Size and Users' Satisfaction

In order to evaluate the scanning outcome, we have considered each image size and each data size. Fig. 4.7 presents the detailed outcome of the 300×300 pixels image size. The data sizes were grouped into five ranges of 100-400, 500-800, 900-1200, 1300-1600 and 1700-2000 bytes. The X-axis represents the data size groups, while the Y-axis represents the outcome percentage. The summation of Right, Wrong and Aborted percentages is 100%.

1. Data Size Impact

Interestingly, when we analyzed the data size impact, we noticed that the Right percentage decreases when the data size increases. On the other hand, the Wrong and the Aborted percentages increase for higher data sizes. For example in Fig. 4.7, the Right outcome percentage is the highest (100%) for the minimum data size (barcodes with 100-400 bytes), while the highest data size (barcodes with 1700-2000 bytes) yielded 18.2% as Right, 59.1% as Wrong and 22.7% as Aborted outcome. Moreover, we can notice that the Wrong scans recorded higher percentage than the Aborted, which means that the users gave the scanner the enough time to read barcodes.

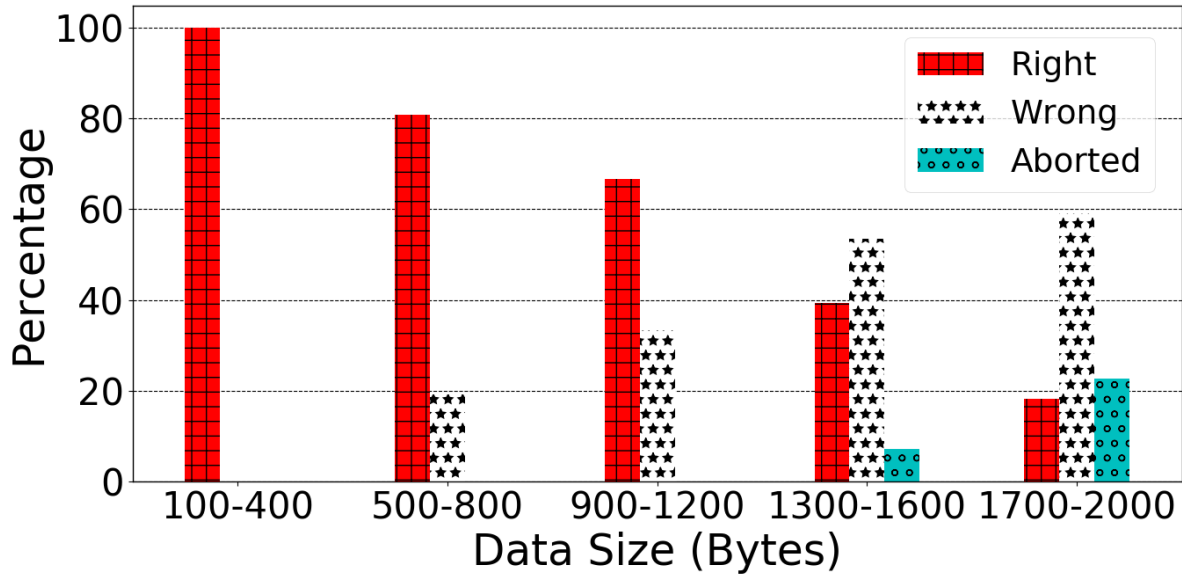


Figure 4.7: Outcome percentage for 300×300 pixels image size.

However, there are cases in which the Aborted percentage overtakes the Wrong one, see, e.g., Fig. 4.8 that shows the outcome percentage for 200×200 pixels barcodes. Here, the largest group of data sizes (1700-2000) recorded a higher Aborted percentage (48%), compared to the Wrong one (36%). Small QR code images with a large amount of data have very dense modules, giving to the users the impression that the barcode is unreadable. In addition, dense QR codes make the scanner's task in finding barcode patterns (even the Misleading one) harder.

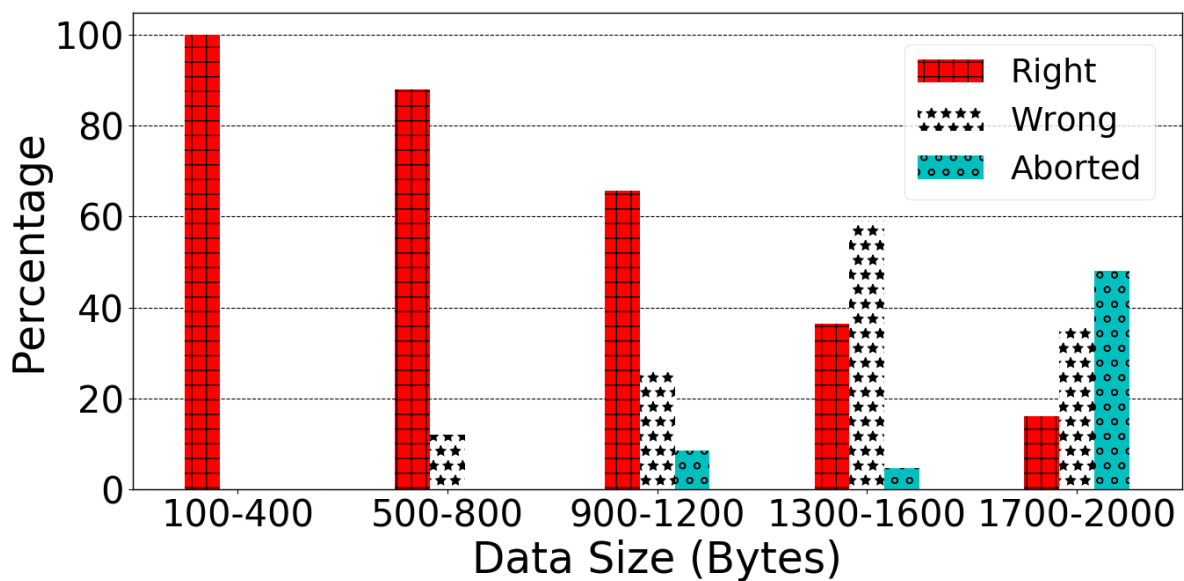


Figure 4.8: Outcome percentage for 200×200 pixels image size.

2. Image Size Impact

Fig. 4.9 shows how the image size affects the readability of barcodes. We can notice that for the same group of data size, the Right percentage increases for larger image sizes, and decreases for smaller image sizes. For example, in the group data size of 900-1200 bytes the Right percentage for 200×200 pixels is 65.7%, 66.7% for 300×300 pixels, 72% for 400×400 pixels and 80.6% for 500×500 pixels. This is more evident in larger data sizes, where the Right scans increases for larger image sizes. However, some slight differences exist in smaller data sizes (500-800 bytes) due to multiple factors such as: Misleading patterns, distance, light, etc (see Table 4.5).

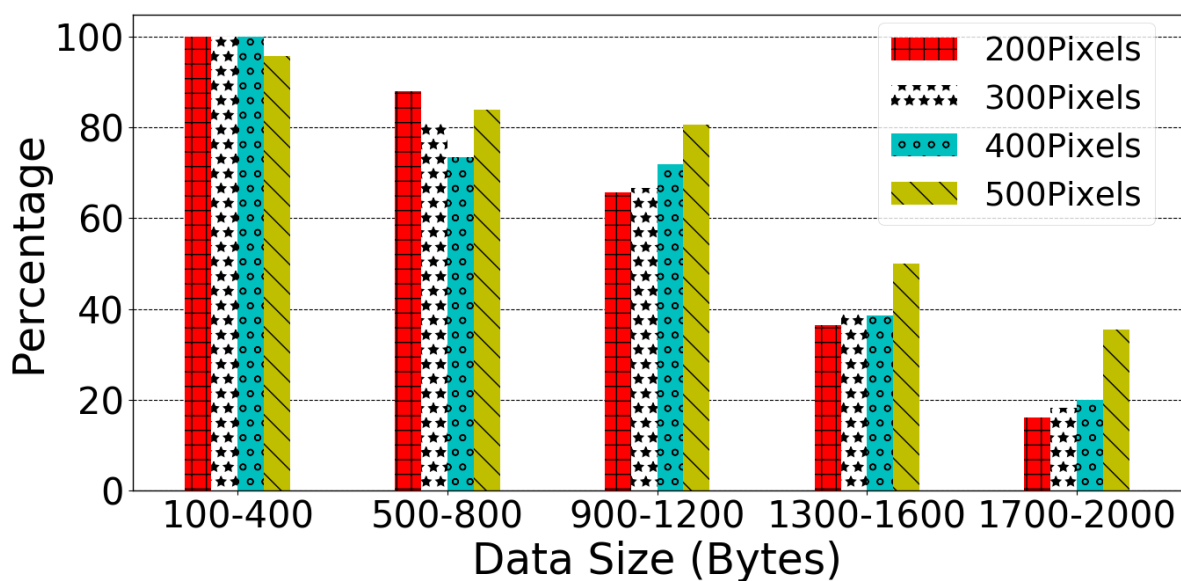


Figure 4.9: Right percentage for all image sizes.

3. Users' Satisfaction

- (a) **Users' Satisfaction vs. Data Size:** Comparing users' satisfaction with data sizes, it emerges that the users' satisfaction is higher for lower data sizes. Fig. 4.10 shows the users' satisfaction level for barcodes with 400×400 pixels. The X-axis represents the data size groups, while the Y-axis represents the percentage. Note that, the Right percentage (red bar) is divided into three sub-groups; Excellent, Good and Bad.

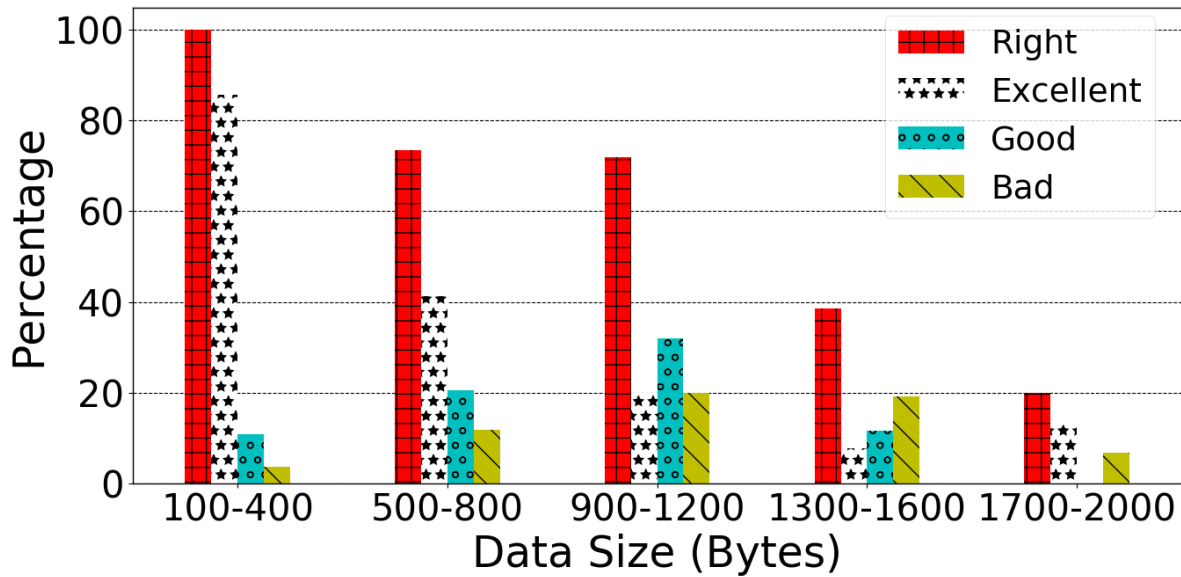


Figure 4.10: Right percentage details for (400×400) Pixels.

For example, in Fig. 4.10, the group of 100-400 bytes recorded Excellent percentage as 85.7%, Good represents 10.7% and Bad only 3.5%. In the group of 900-1200 bytes the percentage was 20%, 32% and 20% for Excellent, Good and Bad respectively (out of 72% which is the Right percentage).

(b) **Users' Feedback:**

The users were asked some questions about their experience (the questions are mentioned in the 4.2), in this part we aim to analyze their answers. The users who answered the questions regarding challenges in scanning barcodes (non-excellent scans) reported some points that we analyzed and found that: 26.4% of the users reported that they had to move the phone many times to be able to read the barcodes. This highlights the impact of distance in reading QR codes; there are Min and Max distances in which the QR code is readable, expressed by the Readability Range (*RR*) [2]. Note that in Chapter's 3 experiment we measured the *RR* in closed environment, and evaluated its effect on barcodes' usability. Smaller image sizes can be decoded from closer distance, because their finding patterns will fit inside the scanning area. The larger images will need more distance, so that their finding patterns fit inside the scanning area. 39.1% of the users reported that non-excellent scans occurred because they had to wait too much time. The *ST* impact was discussed in 4.3.1. 50.6% mentioned that they had to rescan the barcode many times. We have analyzed the rescan results as shown in Fig. 4.11.

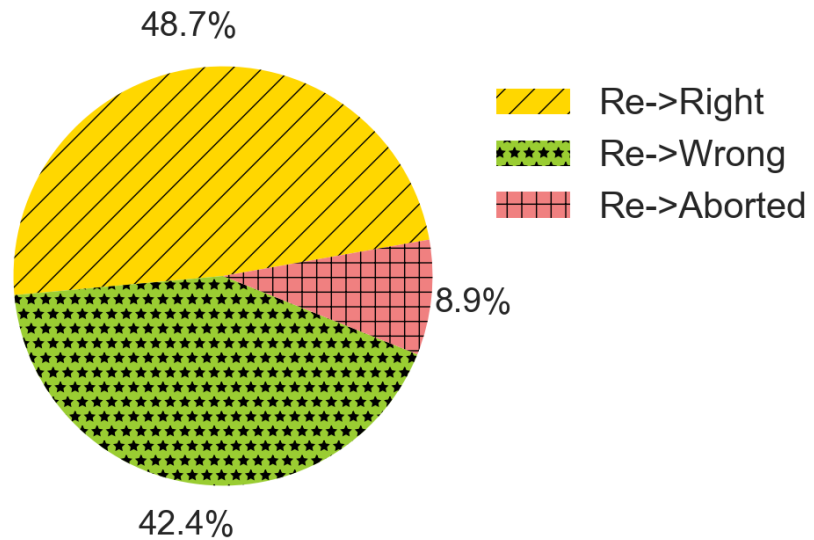


Figure 4.11: Rescan Impact.

In total, 48.7% of rescan attempts ended with Right decoding (even with several rescans), divided into Excellent (12.9%), Good (17.5%) and Bad (18.2%) scans. The remaining rescan attempts could not decode barcode contents successfully, 42.4% ended with Wrong contents and 8.9% reported as Unreadable. This is a remarkable challenge of the Misleading issue, and shows the importance of providing the rescan feature.

The “Other” field was filled by few users, only 6.9% mentioned their notes regarding their challenges. Table 4.5 presents the users’ comments, in which they were grouped into the following categories:

- Reporting Status: notes that report the outcome of barcodes, without giving useful information.
- Unknown Barcode Type: Even with the previous instructions, some users reported that they could not understand why the retrieved contents were wrong.
- Scanning Environment (Scanning Env.): notes that discuss effects of light and surface where the QR code is attached.
- Image Size: notes that discuss image size effects on the scanning outcome.
- Distance: notes that discuss distance effects on the scanning outcome.
- Other: notes that do not belong to any of the previous categories.

Table 4.5: Summary of users' comments and their categories.

User Comment	Reporting Status	Unknown Barcode Type	Scanning Env.	Image Size	Distance	Other
Cannot read.	✓					
Could not scan.	✓					
Does not work.	✓					
I cannot read it.	✓					
It could not resolve the barcode.	✓					
Nothing goes! ^a .	✓					
The first scan was wrong.	✓					
The scan were all wrong.	✓					
Wrong content.	✓					
Why wrong.		✓				
"Wrong barcode" even if it is correct.		✓				
I do not know why it is wrong.		✓				
Could not read it even with flash light.			✓			
Dense image, I used my PC ^b .			✓	✓		

Continued on next page

Table 4.5 – Continued from previous page

User Comment	Reporting Status	Unknown Barcode Type	Scanning Env.	Image Size	Distance	Other
I had to put the barcode on a flat surface ^a .			✓			
I could not read it from paper, I used the computer and zoomed the image ^b .			✓	✓		
Too small.				✓		
I had to move a step away, could not read itso close.					✓	
I used an old phone.						✓
Slightly slower than the first and second. known that holding the phone vertically is more accurate than keeping it horizontally ^a .						✓

^a Reported in Italian Language and translated to English;

^b In two cases, the instructor who printed the sheets read barcodes from PC;

According to the users' feedback and behaviour, we reported the following points:

- The light in the room (scanning environment) can affect the scanning task. The study of [71] claims that shining light with less reflection and less dust lead to better scanning experience.

- The media or surface reflectance where the QR code is attached or displayed can affect the scanning outcome [71] i.e. digital screens vs. papers.
- The distance between the barcode image and the scanner device affects the scanning outcome.
- The awareness of barcode types and misleading problem is important, so that the users will have the ability to determine types and rescan Wrong barcodes to get the right contents.

Choosing larger image size is not always better, and we should take the usage and the surface (where the QR code is attached) into consideration. Note that the previous experiments were conducted with a single error correction level M (Medium), since the Medium level is the most frequently selected [15]. However, there are four levels of error correction: H, Q, M and L that tolerate 30%, 25%, 15% and 7% of image damage respectively [5].

The error correction level can play important role affecting barcodes' availability. In theory, we believe that higher error correction level will lead to better scanning outcome (Right scan) and reduce the Wrong percentage.

4. Barcode Usability Score (*BarScore*)

In this study, we have performed comprehensive usability analysis; for every scan we recorded the scanning outcome, user's opinion and Scanning Time. We define QR code usability based on the success and performance of scanning. Along ISO 9241 [62], we consider the following parameters:

- Effectiveness (Effect.): the possibility of successfully scanning a barcode, the scanning outcome determine whether the scan attempt was successful (Right) or not successful (Wrong or Aborted).

$$Effect. = \frac{Successful\ tasks}{Total\ tasks\ undertaken} \times 100\% \quad [72] \quad (4.1)$$

Where:

- Number of successful tasks represents Right scans.
- Total number of tasks undertaken represents the summation of Right, Wrong and Aborted scans.

- Efficiency (Effi.): represents the time required to perform the scanning.

$$Effi. = \frac{\sum_{j=1}^U \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^U \sum_{i=1}^N t_{ij}} \times 100\% \quad [72] \quad (4.2)$$

Where:

- U: represents number of users.
 - N: represents total number of tasks.
 - n_{ij} : the result of task i by user j ; if the user successfully completes the task, then $n_{ij} = 1$, if not, then $n_{ij} = 0$.
 - t_{ij} : the time spent by user j to complete task i .
- Satisfaction (Sat.): the user comfort in terms of simplicity to perform the scanning; determined by users' answers

$$Sat. = \left(\frac{\sum_{j=1}^U \sum_{i=1}^{Q_+} P_{ij}^+ + \sum_{j=1}^U \sum_{i=1}^{Q_-} P_{ij}^-}{(Q_+ \times Q_- \times U)} \right) \times 100\% \quad [73] \quad (4.3)$$

. Where:

- U: represents number of users.
- Q_+ : the number of positive answers.
- Q_- : the number of negative answers.
- P_{ij}^+ : positive weight for the answer to a positive question (4.2) (Excellent 3/3, Good 2/3 and Bad 1/3).
- P_{ij}^- : negative weight for the answer to a negative question (4.2) (4 negative replies 4/4, 3 negative replies 3/4, 2 negative replies 2/4 and 1 negative reply 1/4).

Fig. 4.12 shows the *BarScore* values for the tested image and data sizes. The X-axis represents the data size groups, the Y-axis represents *BarScore* and each line represents a specific image size.

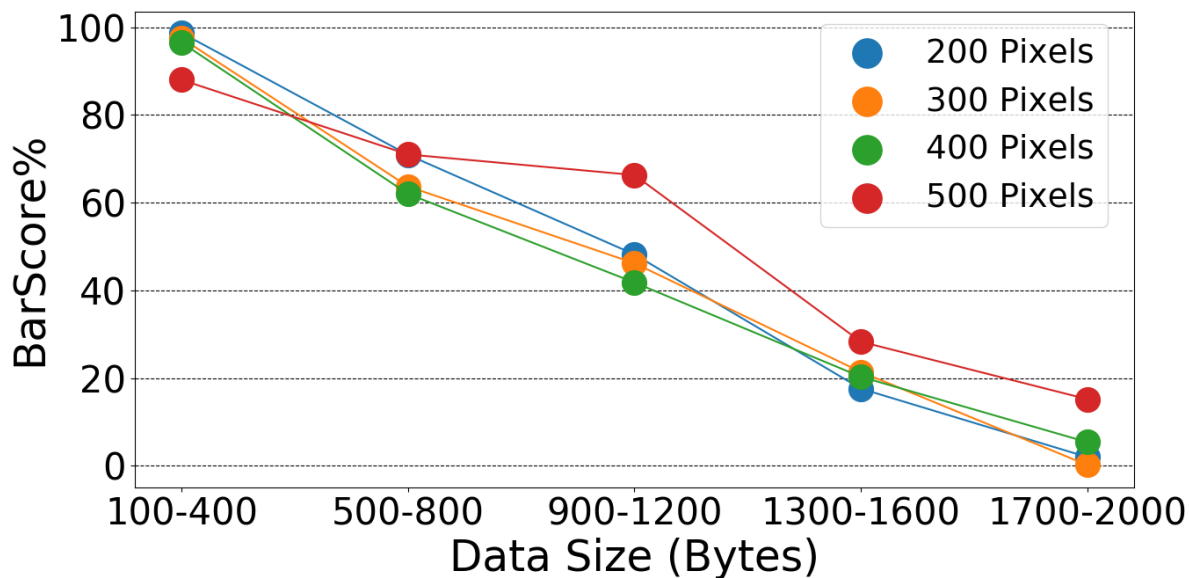


Figure 4.12: Barcode Usability Score (*BarScore*) for the tested image and data sizes.

Based on *BarScore*, we consider three usability levels:

- High Usable level (H): $BarScore \geq 80\%$.
- Low Usable level (L): $60\% \leq BarScore < 80\%$.
- Unusable level (U): $BarScore < 60\%$.

According to Fig. 4.12, we present a barcode usability guide for recommended image and data sizes shown in Table 4.6. We require that the Readability Range (*RR*) is at least 20 cm [2].

Table 4.6: Summary of usability levels.

Data Size (Bytes)	200 Pixel	300 Pixel	400 Pixel	500 Pixel
100-400	H	H	H	H
500-800	L	L	L	L
900-1200	U	U	U	L ^a
1300-1600	U	U	U	U
1700-2000	U	U	U	U

^a $RR > 15$ cm;

4.4 Usability and Cryptography Trade-off

We proposed a proof-of-concept usable barcode security tool, *BarSec*, a desktop tool that adopts symmetric and asymmetric cryptographic mechanisms in order to generate safe and usable QR codes. *BarSec* offers various promising choices of security objectives that include: barcodes authentication, data integrity, access control and confidentiality. *BarSec* provides usability warning messages based on our usability guide, and can be used for both generating and reading QR codes. In addition, *BarSec* provides detailed information about the used algorithms, usability level, Scanning Time and size overhead. In our previous work [2], we noticed that, smartphones are capable for applying cryptographic mechanisms, which consumes only a small time interval in milliseconds for the various key lengths and algorithms. However, size can be critical with respect to usability in some cases, in the following we discuss the space overhead of selected cryptographic primitives, and evaluate their usage, based on *BarScore*. We used the JSON structure proposed in [2]. Our preliminary results in the previous chapter measured the overhead of the two most commonly used digital signature algorithms: RSA with key lengths 1,024 bits, 2,048 bits and 3,072 bits and Elliptic Curve Digital Signature Algorithm (ECDSA) with key length of 256 bits. In which we use SHA-256 as hash function.

Authentication and integrity can be achieved by employing hash-based message authentication code (HMAC) with secret key. Table 4.7 shows authentication and integrity control data overhead for HMAC with three different key lengths; 256, 384 and 512 bits.

Table 4.7: Authentication and integrity control data overhead for HMAC (bytes).

Auth. & integrity	HMAC Length	Overhead
HMAC 256	32	109
HMAC 384	48	125
HMAC 512	64	141

For the data encryption, we have considered Advanced Encryption Standard (AES) with four modes; Cipher Block Chaining (CBC), Output Feedback (OFB), Cipher Feedback (CFB) and Galois/Counter Mode (GCM) as shown in Table 4.8.

Table 4.8: Confidentiality control data overhead for AES different modes (bytes).

AES	Overhead
CBC	69 + block overhead ^a
OFB	69 + block overhead ^a
CFB	69 + block overhead ^a
GCM ^b	85

^a Padding bytes could be added;

^b GCM guarantees authentication and data integrity;

In Table 4.8, the used block size is 16 bytes, padding bytes might be added to the encrypted payload to reach the block size limit. For example if the payload size is 6 bytes, then 10 padding bytes will be added.

Crossing Table 3.2, Table 4.6, Table 4.7 and Table 4.8, we obtain Table 4.9. Table 4.9 summarizes the usability/cryptography QR codes solutions by assuming 200 bytes for the payload.

Table 4.9: Summary of the Usability and Cryptography QR Codes Solutions.

Solution	Key length (bits)	High usable	Low usable	Unusable
ECDSA	256	✓		
RSA	1,024	✓ ^a		
	2,048		✓	
	3,072		✓	
ECDSA (cert.)	256		✓	
RSA (cert.)	1,024		✓ ^a	
	2,048			✓
	3,072			✓
HMAC	256	✓		
	384	✓		
	512	✓		
AES (CBC)	128	✓		
AES (OFB)		✓		
AES (CFB)		✓		
AES (GCM)		✓		

^a Preferred 500 pixels;

[44] recommends a key length of 3,072 bits for RSA and of 256 bits for Elliptic Curve, so we will consider RSA 1,024 as low-secure, 2,048 as medium secure, and RSA 3,072 together with ECDSA 256 as high secure. According to Table 4.9, we recommend ECDSA for high usable/secure digital signature scheme. AES with 128 bits key length is considered as high secure encryption algorithm [74], which can be used to achieve confidentiality and used in access control (*BarSec* supports Access Control List with encrypted payload). We recommend choosing GCM mode; since it provides additional features of authentication and integrity. HMAC [74] provides suitable alternative to digital signature scheme, by providing authentication and integrity with high usable/secure level. However, since HMAC uses shared secret, we need more effort to address the key management issues.

4.5 Conclusion

QR codes may be subject to multiple types of attacks such as: phishing, fraud malware, SQL and command injections. Actually, QR codes provide a medium that can store information or provide input for automated systems, where malicious content can be embedded in order to break users' privacy or steal their credentials.

Standard cryptographic schemes provide authentication and encryption, which can be used to prevent most of the attacks on QR codes. In this context, we assume that we have closed environment, so that the public and shared keys of trustworthy entities are clearly established. Employing cryptographic schemes leads to size overhead that may break the QR codes' usability. This motivated us to perform a systematic comprehensive usability study of QR codes. We conducted set of experiments with the help of undergraduate university students, who used specific application that collects usability metrics and users' feedback. We analyzed how Scanning Time, barcode data size, image size and users' feedback can affect the barcode usability score (*BarScore*); represented by effectiveness, efficiency and users' satisfaction. We provided a barcode usability guideline for recommended image and data sizes, and evaluated the usage of cryptographic mechanisms according to their usability levels. In addition, we implemented *BarSec*, a proof of concept tool that supports generating secure and usable QR codes. The obtained results showed that QR codes can support powerful, usable and secure solutions.

As a future work, we intend to study the usability capabilities for other error correction levels; L, Q and H. Furthermore, we plan to evaluate more cryptographic schemes to look for potential secure-and-usable alternatives to the popular ones.

Part III

QR Code Readers

Chapter 5

QR Code Readers: Security and Usability Analysis

5.1 Introduction

Typically, a barcode scanner is an optical machine that has imaging and processing capabilities (camera and processor), and is used to extract data from a barcode image [15]. The widespread usage of smartphone devices with high resolution cameras, motivates developers to create mobile applications that can decode barcode images, and provide additional features such as: sharing contacts, messages and URLs. Quick Response (QR) codes are the most popular barcode types, with the highest data capacity.

Multiple studies were dedicated to address QR codes threats and solutions, while the practical side needs more analysis [2, 58]. Hundreds of barcode scanners are available for smartphone devices, some of them claim of being "secure" or "privacy friendly" [75, 76]. These applications need to be investigated and evaluated from security, usability and privacy perspectives.

In this chapter we present a comprehensive systematic review of barcode scanner applications. We analyze the features of barcode readers, classify them into groups and highlight their limitations. Then we present a set of design tips and recommendations for usable, secure and privacy-guaranteed reader application. Depending on our recommendations, we have implemented *BarSec Driod*, a proof-of-concept Android application that exploits some features of other applications and at the same time overcomes their limitations [77]. We have performed a user usability and security survey, for *BarSec Droid* [77] and two popular secure QR code readers; KasperSky and QR Droid [13, 14]. Our results show that applying the design tips will increase the user's security trust, improve the user's attitude towards applying security solutions, and increase the awareness of possible attacks.

The rest of this chapter is organized as follows: section 5.2 presents a brief summary of the related work. Section 5.3 explores QR code reader applications and classifies them based on their features. In section 5.4 we present our design tips and recommendations for secure, usable and privacy-friendly QR code reader, then we present *BarSec Droid*, our recommended reader application. Section 5.5 explores the users' experiment and results of usability and security. Finally, in Section 5.6 we present the conclusion and future work.

5.2 Related Work

In this section we recall the studies that evaluate some of the available QR code readers. However, w.r.t. our work, these studies take into consideration a limited number of applications (we check 28 of them) and none of them focuses on all the possible features, i.e., security, privacy and usability.

The study of [75] explores the available Android secure QR code readers, highlights their security properties and evaluates their capabilities in detecting malicious QR codes. According to the conducted analysis, many QR code readers claim to be secure scanners, however, they do not provide the basic security aspects and require enhancements. E.g., several of these readers display the encoded URLs without checking against possible attacks. Furthermore, some scanners open URLs automatically without asking for user's confirmation, which may lead to dangerous consequences. The results show that 8 out of 14 analysed applications provide some protection mechanisms that are capable of detecting phishing better than malware attacks. Among the tested applications, only KasperSky scanner [13] and G Data QR code [78] check the full URL. The study addresses the potential weaknesses and limitations of secure scanners applications, and provides recommendations to improve the security level. However, the analysis does not take into consideration the usability of barcode scanners.

The study of [58] explores the security capabilities for the 31 most popular QR code Android readers. The study focuses on detecting the potential phishing and malware attacks. Results show that 23 out of 31 applications adopt user confirmation features, which allow users to determine either to continue or to quit visiting the embedded URLs. Only 2 out of 31 applications have security warning capabilities, but extended experiments show that the threat detection mechanisms for phishing and malware attacks are very weak. Thus, the researchers propose a new QR code reader called SafeQR, which depends on two existing security APIs: Google safe browsing and phishtank [59, 60]. SafeQR results show the efficiency of visual warning interface techniques, compared to the other existing QR scanners applications. However, the authors do not give any empirical evidence that the scanner is able to enhance the detection rates of malicious URLs. Moreover, the study does not take into consideration QR codes' offline threats such as SQL and command injections, privacy and usability features.

In [79] the authors present a comprehensive analysis of QR code security and privacy issues. The study contains two phases: QR code scanners' evaluation and users' behaviour study. In the

first phase the study analyses 12 top downloaded QR code scanners, determines their security features and tests them with a set of malicious QR codes. The obtained results show that most of these applications cannot detect malicious URLs. In addition, these applications violate the users' privacy, by getting extra permissions and accessing users' personal information. The second phase includes a survey to evaluate the users' behaviour and knowledge regarding QR codes. The survey results highlight the need for readers' security improvements, which makes scanning QR codes a secure user experience. The study presents design recommendations for usable and secure reader applications, and proposes a prototype that employs Base64 digital signatures and URL checking. Results show that following the design recommendations can effectively protect users from malicious QR codes. However, the study is limited to 12 applications and does not discuss the size and time overhead of applying digital signature protection and the implemented algorithms.

5.3 QR Code Readers

Searching Google Play Store [76] for the word "QR Code" gives more than 240 applications. The majority of these apps provide the scanning service without security features, and some of them claim of being "secure". According to [79] most of the QR code reader applications are not able to protect users from malicious URLs, and they significantly violate the user's privacy by getting permissions to access and transmit personal information to third parties. We now analyze different scanner applications from security, privacy and usability perspectives. We have found that these apps employ poor security techniques, with weak algorithms and inappropriate key lengths. In addition they use non-standard structures and non-optimal encoding schemes. Our proposed app, proposed in Section 5.4 overcomes these problems.

Our analysis includes 28 reader applications that are popular or claim to provide security and privacy features, Table 5.1 shows the details of the tested applications. We classify these readers into four groups as follows:

- URLs security applications;
- Crypto-based security applications;
- Popular applications;
- Save-privacy applications.

Table 5.1: Details of tested QR Code readers.

App Developer	Version	Installs	Category	Rate	1D/2D	Type
[78]	1.0.2.0643c6ef	10K+	URLs	3.3	QR	
[13]	1.2.4.51	1M+	URLs	4.4	QR	
[56]	2.0.0.71	1M+	URLs	4.2	✓	
[80]	1.0.0	10K+	URLs	4.8	✓	
[81]	1.1	10+	URLs	5	✓	
[82]	1.2	100+	Crypto	5	✓	
[83]	1.0.17	1K+	URLs	4.1	✓	<i>a</i>
[84]	1.0	5K+	URLs	4.4	✓	✓
[14]	7.0.4	50M+	Crypto	4.2	✓	
[85]	Free ^b	100+	Crypto	5	QR	
[86]	2.5.0	100K+	URLs	4.3	✓	✓
[87]	1.0.0	100K+	URLs	4.3	QR	
[88]	1.6.1	10K+	Save-Privacy	4.4	✓	✓
[89]	2.4.3	500+	URLs	4.1	✓	
[90]	1.1	5+	URLs		✓	
[57]	1.03	500K+	Save-Privacy	3.8	✓	✓
[91]	1.1.7	5K+	Save-Privacy	4.2	✓	
[92]	2.1.6	5M+	Save-Privacy	4.5	✓	
[93]	1.3.1-L	1M+	URLs	4.6	✓	✓
[94]	1.0.2	1+	Crypto		✓	
[95]	1.7.6	10M+	Save-Privacy	4.7	✓	
[96]	2.33	50M+	Save-Privacy	4	✓	

Continued on next page

Table 5.1 – Continued from previous page

App Developer	Version	Installs	Category	Rate	1D/2D	Type
[97]	Varies with device	100M+	Popular	4.1	✓	✓
[98]	1.2.91	10M+	Popular	4.6	✓	✓
[99]	Varies with device	50M+	Popular	4.4	✓	
[100]	1.25	5M+	Popular	4.4	✓	✓
[101]	0.92	10M+	Popular	4.6	✓	
[102]	1.0.5	1K+	Crypto	5	✓	

^a Always display QR code;

^b Free version to test functionality;

Where in Table 5.1:

- App developer: the identity of developer or company name.
- Version: current app's version.
- Installs: number of app's installations from Google Play.
- Category: App's category.
- Rate: a 5-point scale users' evaluation of an application from Google Play.
- 1D/2D: ability to read 1D and 2D barcodes.
- Type: display barcode type.

5.3.1 URLs Security Applications

Security is a major issue that can affect the users' experience since QR codes can be used to attack the scanning devices (smartphones), e.g., by embedding malicious URLs. Online protection includes checking URLs that are encoded inside QR codes, which can be used to

launch phishing, malware and XSS attacks. The protection technique is simple and aims at alerting users about malicious links.

G data QR code scanner [78] is a simple, free Android application, designed for QR code protection. This app checks the encoded URLs, in order to detect phishing and malicious links, it retrieves the full destination web address, even if it was embedded as a shortened URL [75]. Unlike other QR codes readers, this app prevents the users from opening suspected URLs in their browsers [75].

KasperSky QR Scanner is a free app that validates the QR code links against malware and phishing Web pages [13]. The main limitation of KasperSky QR Scanner is that it opens URLs directly in the browser without user confirmation, if detected as safe URLs [75].

The Norton Snap QR code scanner [56] is an application that automates the online QR code checking against online attacks. This application provides different protecting features: It alerts users against unsafe/untrusted URLs, shows the full expansion of the website address and blocks the malicious online content before being loaded on the user's browser.

Other URLs security applications such as: Trend Micro [80], FANSec [81], Dennings [83], Avira [86], iTechSo [90], KidControl [84] and X & C Hi-Tech Inc [89] offer URL checking service. However, they do not get the full expanded URL. If the encoded URLs are shortened or redirected, the users will not be able to view the final URL destination.

Barcode Reader for CM browser [87] is a lightweight QR code scanner that requires CM browser and automatically opens URLs. The CM browser performs the security protection that includes: Advertising blocker, malicious Web pages checking and download protection.

TeaCapps [93] barcode scanner offers URL checking by using Chrome Custom Tabs, which employs Google Safe Browsing technology [59].

Table 5.2 presents a comparison of URLs security scanners.

Table 5.2: URLs security scanners.

App Developer	Check URL	Display URL	Get full URL	Direct Open	URL checking technique
[78]	✓	✓	✓		N/A
[13]	✓			✓ ^a	KasperSky Virusdesk
[56]	✓	✓	✓	✓ ^a	Norton Safe Web
[80]	✓	✓			N/A
[81]	✓	✓			N/A
[83]	✓	✓			Google Safe Browsing
[84]	✓	✓			N/A
[86]	✓	✓			N/A
[87]	✓			✓	CM browser
[93]	✓	✓			Google Safe Browsing
[90]	✓	✓			N/A
[89]	✓			✓ ^a	N/A

^a Open URL directly if it is safe;

The main limitation of these apps is that URL-checking readers can protect users from malicious URLs only, while other offline attacks such as SQL and command injection can still be performed without detection. In addition, these applications require Internet connection to check URLs. Note that in this work, we evaluate the applications features, not the database/model that checks the URL.

5.3.2 Crypto-based Security Applications

Cryptographic techniques can be used to encrypt, sign and control the access to QR code contents. Choosing the suitable algorithm, key size and structure are discussed in multiple studies [1, 44, 74], but the key factor is the size overhead on barcode usability [2]. However, there are few applications that offer generating and reading cryptographic QR codes.

Madiff Net [82] scanner application offers reading and creating password-protected QR codes, and the contents are encrypted using a shared password-based key between the generator and the barcode reader. The encryption algorithm is not mentioned, the key length is 6 bytes (48 bits), and the ciphertext is shown as a Base64 string. The application is free, contains advertisements and is available with three languages: English, Vietnamese and Chinese. As a limitation, we highlight that being the algorithm unavailable we cannot evaluate its strength. Moreover, as we will mention later, using Base64 for encoding causes size overhead.

QR Droid Private [14] is a full-featured and multi-language application, which offers reading and creating QR codes. In addition, the application supports URL shortening, QR code sharing and contents encryption. Data Encryption Standard (DES) is used with 56 bits symmetric key and keyword structure, in which the ciphertext is encoded using Base64. The application is free and does not include advertisements. QR Droid has two versions; private and full. The private version requires fewer permissions while the full version requires more to allow users to create QR codes directly from the device without typing anything. QR Droid is well adopted and available in 29 languages. The app employs the DES algorithm with 56 bits key length which is considered weak and breakable, and using Base64 for encoding causes size overhead.

Crypto Message [85] is a security application that offers encrypting text messages, and allows encoding ciphertexts in QR codes. The application provides the encoding of QR codes in the free version, while the decoding requires the paid version. Advanced Encryption Standard (AES) is available with four modes: Electronic Codebook (ECB), Cipher Block Chaining (CBC), Counter (CTR), and Output Feedback (OFB) modes. The supported key lengths are 128, 192 and 256 bits, and ciphertexts are encoded as a hexadecimal or Base64 string. Note that, the embedded scanner cannot decode ordinary QR codes (generated by other applications). Crypto Message usage is not straightforward, the users should be aware of cryptography concepts in order to use it.

EC QR [94] is a QR code reader and generator application. The algorithm, the key size and

the structure of this app are not available and cannot be evaluated. In addition, this app may read any type of barcode, wrongly displaying it as a QR code image (even if it is something else).

Observe that, all the above mentioned applications have some limitations: 1) They assume no standard way of encoding cryptographic data in QR codes, i.e., each application uses its own structure. Thus, in order to decode a crypto-barcode, the user will need to use the same generating application, while, on the other hand, the study of [2] proposes the use of the standard JavaScript Object Notation (JSON) as a general structure to be used with crypto-QR codes. 2) All these applications adopt weak cryptographic mechanisms such as: DES and AES-ECB. 3) These applications employ Base64 and hexadecimal strings to represent ciphertexts, which leads to size overhead since Base64 represents each 6 bits with one digit, and hexadecimal represents each 4 bits as one digit.

The password-protected QR codes achieve confidentiality and access control, where only authorized users (who have the password) can retrieve the encoded data. However, encrypting the contents is not enough to protect users who scan the QR code, since even encrypted data can contain malicious links or offline attacks. Generator authentication, data integrity and non-repudiation can be useful in protecting the users, and can be achieved using digital signatures [1]. Table 5.3 presents a summary of crypto-based QR code scanners and it includes the app developer, encryption, digital signature (DS), algorithm (Alg), Key length (KL), encoding scheme (EncS) and structure (Str).

Table 5.3: Crypto-based QR code scanners.

App Developer	Encryption	DS	Alg	KL (bits)	EncS	Str
[82]	✓		N/A	48	Base64	N/A
[14]	✓		DES	56	Base64	Keyword
[85]	✓		AES	128,192 & 256	Base64 & hex	N/A
[102]	✓		N/A	N/A	Base64	N/A
[94]	✓		N/A	N/A	Base64	N/A

Note that, these applications offer a single access control mechanism, the encoded data may

either be public (plaintext) or private (ciphertext). We cannot have a single QR code that has two parts (encrypted and plain) at the same time. Note that applying Access Control List (ACL) allows the generator to have multiple layers of data; i.e multiple users who have controlled access to data. Let's assume this example: a QR code with ACL may include these tags: public, student and teacher, where:

- Public tag: contains plaintext data;
- Student tag: contains ciphertext that is encrypted with students' key.
- Teachers tag: contains ciphertext that is encrypted with teachers' key.

Each tag has authorized users who can access its contents, where a student can read the public tag, student tag but not the teachers' tag (since the student do not have the teachers' key).

5.3.3 Popular Applications

In this section we present popular QR code reader applications that have been downloaded by more than 1 million users (see Figure 5.1). Note that popular apps may be included in the previous groups, for example Norton Snap is included in URL-security as well as popular apps groups.

ZXing Barcode Scanner [97] is one of the most popular applications, with more than 100 million downloads. It is compatible with various 1D and 2D barcodes, displays the barcode type and retrieves additional information about URLs such as title and redirections. The ZXing library [54] is a core Java source for multiple popular applications such as Barcode Scanner Pro (10M downloads) [98], and Barcode Scanner [100] (5M downloads).

Other applications have nearly the same functionalities, and are able to read 1D and 2D barcodes are QR & Barcode Scanner by [99] which recorded more than 50M downloads, and All-in-one QR + Barcode Scanner: QR Scanner/QR Reader [101] that recorded more than 10M downloads.

Figure 5.1 shows the tested applications that have more than one million downloads. Note that, being popular is not enough to be usable and secure, so we have investigated these applications also from security and privacy perspectives (see Table 5.4).

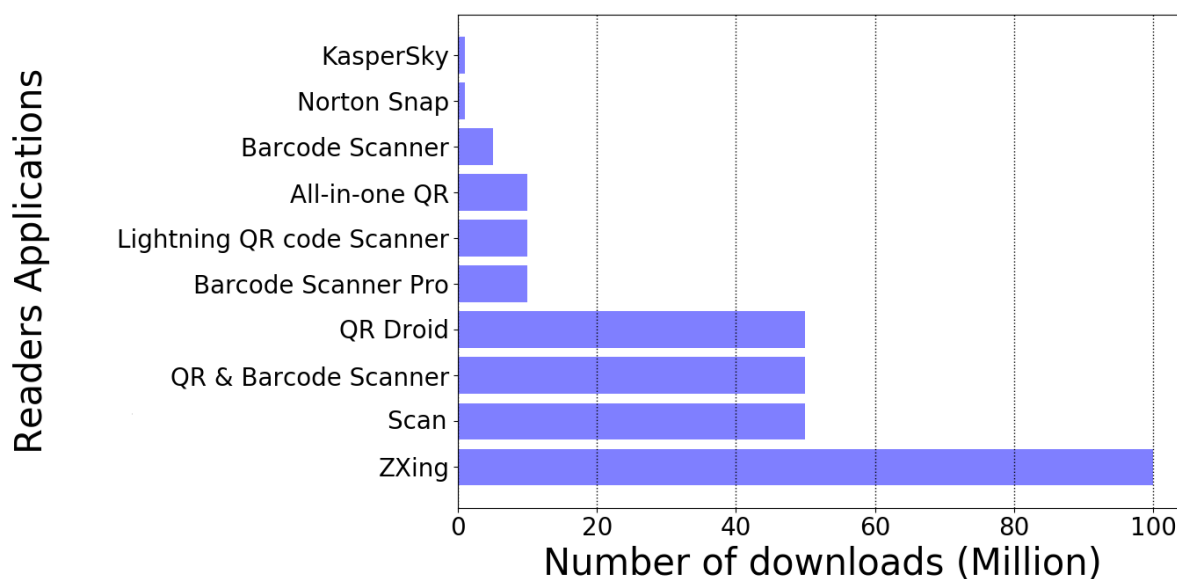


Figure 5.1: Popular QR code scanners with more than 1 M downloads.

5.3.4 Save-Privacy Applications

The applications we will illustrate in this section claim to protect the users' privacy, as they do not require access to personal information details. Obtaining permissions enables direct access to the information without users' interaction, which means easier and faster tasks. On the other hand, some applications may exploit these permissions and send user's private data to a third party [79].

A privacy-violating QR code scanner may access private images, videos, contacts, call history and user's location. Therefore, we need to balance the ease-of-use (getting the permissions) and protecting the users' privacy. Usually, minimal permissions include accessing the camera (to scan the barcode) and network (if there is a need to check URLs). Obtaining other permissions can be extremely dangerous, and may result in information leakage attacks.

Red Dodo [57] is a "secure" QR and barcode reader application. The application does not provide any protection against malicious links or offline attacks, but it claims that it saves the user's privacy. Although the application description says it does not require personal information details, we checked the app's permissions list and found that it accesses the storage, photos, Wi-Fi details, media and files.

The same happens for QR Code Reader Extreme [92] that it claims to require few permissions, whereas we checked the app's permissions list and found that it accesses the storage, photos, Wi-Fi details, media and files.

Some applications offer security QR code's online contents check (URLs), alongside with less permissions. An example is TeaCapps Scanner [93], which requires camera and Internet permissions but not access to storage or files.

On the good side, Tokoware [91] a simple 1D and 2D barcode reader application, was developed based on the ZXing [54] library and Lightning QR code Scanner [95], require access to the camera and network, QR Scanner (Privacy Friendly) [88] only require access to the camera. Thus, all these applications are suitable for users who aim at protecting their privacy.

Users' reviews showed dissatisfaction to applications that require unneeded permissions. For example a user review for [96] mentioned that "Why do you now need access to my location, photos, media and files? It worked perfectly in the past without these permissions and I see no reason to have them". This application does not provide any security protection and it requests permissions for location, camera, photos, media, storage, files and network.

Since users' privacy is important, we have evaluated all the 28 apps in terms of granted permissions. Table 5.4 shows the requested permissions for all our 28 applications. These permissions [76] include get access to :

- Device & app history (DevHis): read sensitive log data;
- Contacts (Cont): read contact list;
- Location (Loc): approximate location (network-based) and precise location (GPS and network-based);
- Phone (Phn): directly call phone numbers;
- Photos/media/files (Files): read, modify or delete the photos/media/files;
- Storage (Stg): read, modify or delete the contents of USB storage;
- Camera (Cam): take pictures and videos;
- Wi-Fi info (wi-fi): view Wi-Fi connections;
- Device ID & call info (DevInf): read phone status and identity;
- Network (Net): full network access and view network connections;

Table 5.4: Permissions of tested QR Code readers.

App Developer	DevHis	Cont	Loc	Phn	Files	Stg	Cam	Wi-Fi	DevInfo	Net
[78]			✓		✓	✓	✓	✓		✓
[13]				✓	✓	✓	✓	✓	✓	✓
[56]				✓	✓	✓	✓		✓	✓
[80]	✓				✓	✓	✓	✓		✓
[81]	✓			✓	✓	✓	✓	✓	✓	✓
[82]	✓	✓	✓	✓	✓	✓	✓	✓		✓
[83]		✓	✓		✓	✓	✓			✓
[84]					✓	✓	✓			✓
[14]					✓	✓	✓			✓
[85]		✓		✓	✓	✓	✓			
[86]				✓	✓	✓	✓	✓	✓	✓
[87]					✓	✓	✓	✓		✓
[88]							✓			
[89]		✓	✓		✓	✓	✓	✓		✓
[90]							✓			✓
[57]					✓	✓	✓	✓		✓
[91]							✓			✓
[92]					✓	✓	✓	✓		✓
[93]							✓			✓
[94]					✓	✓	✓			✓
[95]							✓			✓
[96]			✓		✓	✓	✓	✓		✓

Continued on next page

Table 5.4 – Continued from previous page

App Developer	DevHis	Cont	Loc	Phn	Files	Stg	Cam	Wi-Fi	DevInfo	Net
[97]	✓	✓		✓	✓	✓	✓	✓		✓
[98]	✓	✓			✓	✓	✓	✓		✓
[99]							✓	✓		✓
[100]	✓	✓			✓	✓	✓	✓		✓
[101]	✓				✓	✓	✓			✓
[102]		✓	✓		✓	✓	✓			✓

5.4 Design Recommendation

Based on our analysis for the available barcode readers, and based on suggestions provided in other works [2, 103] we present design tips for secure, usable, and privacy friendly barcode reader applications. The reader applications should provide the following services:

- Support different barcode types, so to be used in various contexts;
- Display the barcode type, in order to avoid wrong barcode type decoding;
- Check any URL to avoid phishing and malware attacks;
- Use security warning such as: browser warning against malicious URLs;
- Apply digital signature services, to authenticate the barcode generator and guarantee data integrity;
- Adopt encrypted contents, to achieve confidentiality and access control;
- Save the users' privacy, by requesting minimum set of permissions and prevent accessing private files. We recommend obtaining permissions for camera (to scan the image) and Internet (to check URLs) only;

- Provide default basic functionalities with simple interface, so that non-expert users can use the app easily;
- Prevent the execution of any encoded codes or commands in users' smartphones;
- Provide manuals and resources for users to learn how to use secure reader applications.

Based on these recommendations, we have implemented *BarSec Droid*, an Android mobile application that employs ZXing library [54], and follows our design tips to provide secure barcode scanning service [77]. Figure 5.2 shows a screenshot of *BarSec Droid* (for more details, check Appendix A), while *BarSec Droid* specifications are included in Table 5.5.

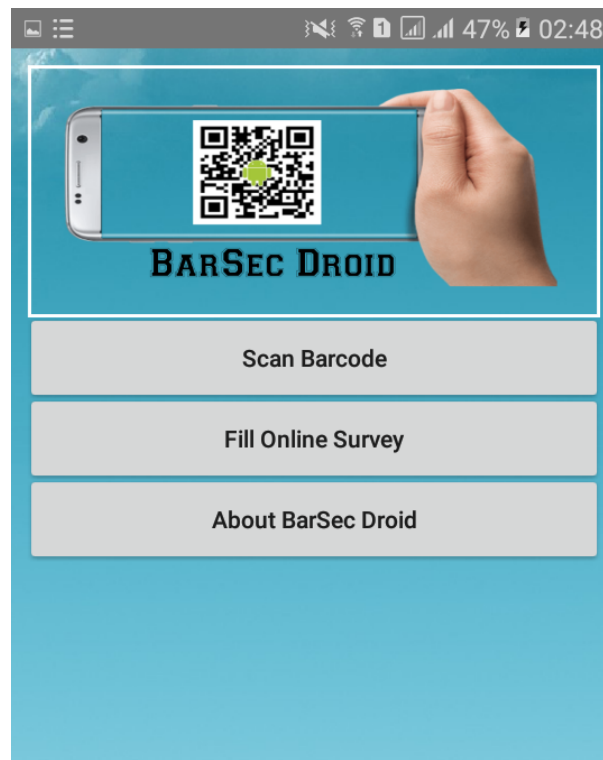


Figure 5.2: Screenshot of *BarSec Droid* [77].

We have implemented Barcode Security Studio (*BarSec*) a Java desktop application, which adopts symmetric and asymmetric cryptographic mechanisms in order to generate and read secure and usable barcodes. Figure 5.3 presents *BarSec* Desktop application (for more details, check Appendix A).

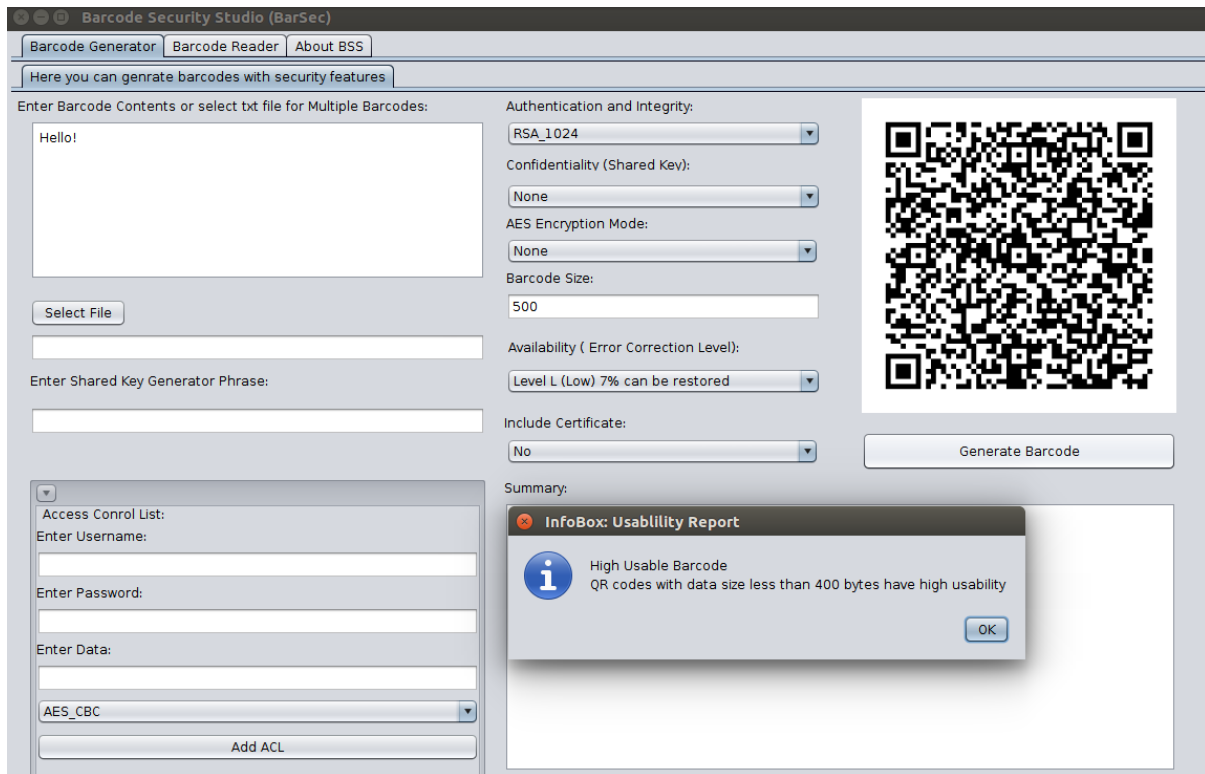


Figure 5.3: Screenshot of *BarSec* Desktop Application.

Table 5.5: *BarSec Droid* specification.

Feature	Supported	Key length (bits)
Encryption	AES ^a	128
Digital Signature	ECDSA	256
	RSA	1,024
		2,048
		3,072
Encoding Scheme	ISO-8859-1	-
Structure	JSON	-
URL Checking	✓ ^b	-
Compatibility	✓ ^c	-

^a CBC, OFB, CFB and GCM;

^b Norton safe web;

^c Supports legacy QR code;

As shown in Table 5.5, *BarSec Droid* considers JSON structure proposed in [2]. It can read barcodes generated by *BarSec Desktop* application which supports Advanced Encryption Standard (AES) with four modes; Cipher Block Chaining (CBC), Output Feedback (OFB), Cipher Feedback (CFB) and Galois/Counter Mode (GCM). In addition, it supports digital signature using ECDSA and RSA. Furthermore, *BarSec Droid* support reading barcodes that has Access Control List (ACL). Note that *BarSec* generator supports encoding ACL in QR codes. *BarSec Droid* can read legacy (non cryptographic) QR codes that do not follow [2] structure by getting the full URL and checking their online content using Norton Safe Web service [104].

5.5 Users' Experiment and Results of Usability and Security

We have conducted a users' survey to get the users' reactions about the *BarSec Droid* [77] usage, and the level of trust for the provided security information. In order to compare the results with other popular security apps, we choose KasperSky [13] that belongs to the URL protection group,

and QR Droid Private [14] that belongs to the Crypto-based protection group. We conducted our survey with the help of 30 undergraduate students (volunteers), who were asked to scan two QR codes for each reader (6 barcodes per user). Then, the users completed a survey that was built following the lines of [105], a very popular usability questionnaire, and [106], a usability survey on secure mobile applications. Our survey includes the following six points:

- Overall, I am satisfied with the ease of completing the tasks.
- Overall, I am satisfied with the amount of time it took to complete the tasks.
- Overall, I am satisfied with the support information (warnings and details messages).
- How much do you trust the security information in the application?
- Overall, I would like to use the application.
- How visually appealing is the application?

Each point have five-point scale, described as: (1: very unsatisfied to 5: very satisfied). We have followed the answers evaluation method used on [106] by using paired *t*-test; which is a statistical method that compares the mean values of two groups [68]. Paired *t*-test was used because the survey asked the subject to evaluate the 2 apps.

Table 5.6 shows the Means (the value before \pm), Mean Standard Error (the value after \pm) and *p*-value results from participants' feedback for *BarSec Droid* [77] and *KasperSky* [13]. Table 5.7 shows the same results for *BarSec Droid* and *QR Droid Private* [14]. Note that in *t*-test, when the *p*-value is less than 0.05, this means that there is a statistically significant difference between two groups [69]. So, when there is a statistically significant the mean and the mean standard error values are marked in bold.

Table 5.6: *T*-test results for *BarSec Droid* vs. KasperSky.

	Easy to use	Time Satisfaction	Support info satisfaction	Security of app	Likely to use	Visually appealing
<i>BarSec Droid</i>	4.0±0.2	3.7±0.2	3.9±0.2	4.6±0.1	3.6±0.1	3.6±0.2
[13]	3.4±0.1	3.8±0.2	2.3±0.2	3.8±0.2	3.0±0.2	2.2±0.2
<i>p</i> -value	0.001	0.895	0.000	0.000	0.012	0.000

According to Table 5.6, it is clear that the users' opinions recorded better results for *BarSec Droid*. For all questions the *BarSec Droid* means recorded higher values with statistical significant, which reflects the advantages of *BarSec Droid* over KasperSky. The time of tasks satisfaction recorded converged values (i.e., similar satisfaction).

Table 5.7: *T*-test results for *BarSec Droid* vs. QR Droid Private.

	Easy to use	Time Satisfaction	Support info satisfaction	Security of app	Likely to use	Visually appealing
<i>BarSec Droid</i>	4.0±0.2	3.7±0.2	3.9±0.2	4.6±0.1	3.6±0.1	3.6±0.2
[14]	3.3±0.2	3.4±0.2	3.1±0.2	1.9 ±0.2	4.5±0.1	3.8±0.2
<i>p</i> -value	0.002	0.169	0.004	0.000	0.000	0.393

According to Table 5.7, *BarSec Droid* recorded better answers for easiness of use, support information satisfaction and security trust. On the other hand, QR Droid Private [14] recorded higher level of being likely to use, which reflects the application excellent design and options such as supporting multiple languages (29 languages). The time of tasks recorded converged values, which reflects that *BarSec Droid* [77], KasperSky [13] and QR Droid Private [14] have acceptable time delay according to the users' feedback.

5.6 Conclusion

This study provides a comprehensive assessment for 28 barcode scanning applications, from security, usability and privacy perspectives. We have analyzed the features of these applications and classified them into four groups; URLs security, Crypto-based security, Popular applications and Save-privacy.

Through the analysis, we have highlighted the limitations, and concluded that: most of these apps do not cover the users' security and privacy needs. We proposed design tips for usable, secure and privacy-guaranteed barcode reader applications, and implemented *BarSec Droid*, a proof-of-concept Android app that utilizes other applications' advantages and resolves their weaknesses.

In order to evaluate our work, we have conducted a users' usability and security survey, for *BarSec Droid* and two popular QR code readers, i.e., KasperSky and QR Droid Private. The results show that when following the proposed design tips the user's security trust increases, as well as the ease of use (the mean values for our app are higher than the other tested apps with statistical differences). Adding to that, it will enhance the user's satisfaction towards using security applications. As a future work, we plan to extend our analysis to cover more applications, and evaluate the security techniques that check QR code contents (URLs) such as: Google safe browsing and Norton Safe Web.

Conclusion

In this thesis we presented a comprehensive security, usability and privacy analysis of QR codes. We surveyed the threats and attacking scenarios that can be performed to target the scanning devices (smartphones), and explored the available protection methods with concentration on cryptographic solutions. We found that the available protection systems do not fulfill the users' security requirements, and still have weak points such as: using deprecated algorithms, broken key lengths and weak hash functions. We suggested to follow the ENISA [44] recommendations for the algorithms and security parameters.

Although cryptographic solutions can provide strong guaranteed authenticity, data integrity and confidentiality for QR codes' contents, the time and size overhead can break the usability and downgrade the users' satisfaction. In order to establish common rules for encoding cryptographic data in QR codes, we suggested using JSON structure format. Extensive experiments were conducted to measure the feasibility of applying cryptographic methods on mobile devices, as well as the feasibility of encoding ciphertexts and digital signatures inside QR codes. Our experiments' results showed that: mobile devices are capable to perform the cryptographic operations with acceptable time delay, and with a variable size overhead according to the used structure and algorithm.

In order to evaluate the usability of cryptographic QR codes, we analyzed the data size, image size and users' satisfaction impacts on barcodes' usability. We followed ISO 9241 usability standard to express usability parameters: effectiveness, efficiency and satisfaction as quantifiable quantities using mathematical formulas, and calculate their average (*BarScore*). Based on *BarScore* we built a usability table for QR codes with the preferred image and data sizes for different usability levels, and evaluated the usability of applying multiple cryptographic algorithms. Then we presented a proof-of-concept implementation of secure/usable barcode reader and generator.

We analyzed the available QR code reader applications and evaluated their security, privacy

and usability levels. We classified these applications into groups based on their protection mechanisms, and explored their popularity and users' rate. We found that these applications can target the users' privacy by requesting wide set of unneeded permissions, which allows the reader app to spy and leak users' private data. In addition, although there are security applications that offer cryptographic protection and URL checking services, they still have weak points and vulnerabilities such as using weak algorithms and direct URL visiting. Since we found a potential need for a comprehensive solution, we presented a guideline and recommendations for secure, usable and privacy-friendly barcode readers. Furthermore, we implemented a reader app that follows our recommendations and evaluated its usability in comparison with two popular security applications. We found that applying our design tips will increase the users' satisfaction, likely-to-use, easy-to-use and security trust of barcode reader application.

As a future work, we plan to investigate more cryptographic algorithms and techniques. In addition, we intend to evaluate the impact of error correction level on QR codes' security and usability. Furthermore, we plan to explore other 2D barcode types and evaluate their usability.

Bibliography

- [1] R. Focardi, F. L. Luccio, and H. A. M. Wahsheh. Security Threats and Solutions for Two Dimensional Barcodes: A Comparative Study. In Daimi K., editor, *Computer and Network Security Essentials*, pages 207–219. Springer, Cham, 2018.
- [2] R. Focardi, F. L. Luccio, and H. A. M. Wahsheh. Usable Cryptographic QR Codes. In *Proceedings of the 19th IEEE International Conference on Industrial Technology (ICIT - IEEE 2018)*, pages 1664–1669. IEEE, 2018.
- [3] T. Vidas, E. Owusu, S. Wang, C. Zeng, L. Cranor, and N. Christin. QRishing : The Susceptibility of Smartphone Users to QR Code Phishing Attacks. In *17th International Conference on Financial Cryptology and Data Security (FC'13), Okinawa, Japan, April 1, LNCS, Springer, 7862*, pages 52–69, 2013.
- [4] A. Dabrowski, K. Krombholz, J. Ullrich, and E. Weippl. QR Inception: Barcode-in-Barcode Attacks. In *Proceedings of the 4th ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM'14), November 7, Scottsdale, Arizona, USA*, pages 3–10, 2014.
- [5] ISO/IEC Standard. ISO/IEC 18004:2015, Information technology – Automatic identification and data capture techniques – QR code 2005 Bar code Symbology Specification, 2015.
- [6] C. Akta. *The Evolution and Emergence of QR Codes*. Cambridge Scholars Publishing, United Kingdom, 1st edition, 2017.
- [7] K. Krombholz, P. Fruhwirt, P. Kieseberg, I. Kapsalis, M. Huber, and E. Weippl. QR Code Security: A Survey of Attacks and Challenges for Usable Security. In *Proceedings of the Second International Conference on Human Aspects of Information Security, Privacy, and Trust (HAS'14), 8533*, pages 79–90, 2014.

- [8] A. Kharraz, E. Kirda, W. Robertson, D. Balzarotti, and A. Francillon. Optical Delusions: A Study of Malicious QR Codes in the Wild. In *44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'14)*, 23-26 June, Atlanta, GA, USA, pages 192–203, 2014.
- [9] P. Kieseberg, M. Leithner, M. Mulazzani, L. Munroe, S. Schrittwieser, M. Sinha, and E. Weippl. QR Code Security. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia (MoMM'10)*, Paris, France, November 8-10, pages 430–435, 2010.
- [10] V. Mavroeidis and M. Nicho. Quick Response Code Secure: A Cryptographically Secure Anti-phishing Tool for QR Code Attacks. In *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*, pages 313–324. Springer, 2017.
- [11] C. Chen. QR Code Authentication with Embedded Message Authentication Code. *Mobile Networks and Applications*, 22(3):383–394, 2017.
- [12] T. Ishihara and M. Niimi. Compatible 2D-code Having Tamper Detection System with QR-code. In *Proceedings of the Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'14)*, Kitakyushu, Japan, August 27-29, pages 493–496. IEEE, 2014.
- [13] Kaspersky Lab. QR Code Reader and Scanner: App for Android, 2018. <https://free.kaspersky.com/?cid=acq-gplay-lnk#mobile>.
- [14] DroidLa. QR Droid Private., 2016. <http://qrdroid.com/>.
- [15] Denso Wave. QRcode.com DENSO WAVE, 2017. <http://www.qrcode.com/en>.
- [16] Bar Code Graphics, Inc. . Barcode 101: Guide To Barcode Symbologies, 2017. <http://www.gtin.info/barcode-101/>.
- [17] A. Cobb. The Comprehensive Guide to 1D and 2 D Barcode Types, 2016. <https://www.dynamsoft.com/blog/barcode-reader/the-comprehensive-guide-to-1d-and-2d-barcodes/>.

- [18] TechnoRiver. Technoriver Barcode Software, Components, and Font, 2018. <https://www.technoriversoft.com/>.
- [19] Tec-it. EC-IT Barcode Software Overview, 2018. http://www.tec-it.com/download/PDF/Barcode_Reference_EN.pdf.
- [20] Z. Rizwan. Do People Use QR Codes in 2017? The Answer Will Definitely Surprise You, 2017. <https://scanova.io/blog/blog/2017/08/04/do-people-use-qr-codes/>.
- [21] ISO/IEC Standard. ISO/IEC 16022:2006, Information technology – Automatic identification and data capture techniques – Data Matrix Bar code Symbology Specification, 2006.
- [22] ISO/IEC Standard. ISO/IEC 16022:2008, Information technology – Automatic identification and data capture techniques – Aztec Bar code Symbology Specification, 2008.
- [23] ISO/IEC Standard. ISO/IEC 15438:2015, Information technology – Automatic identification and data capture techniques – PDF417 Bar code Symbology Specification, 2015.
- [24] D. Lorenzi, J. Vaidya, S. Chun, B. Shafiq, and V. Atluri. Enhancing the Government Service Experience through QR Codes on Mobile Platforms. *Government Information Quarterly*, 31(1):6–16, 2014.
- [25] R. Want, B. N. Schilit, and S. Jenson. Enabling the Internet of Things. *Computer*, 48(1):28–35, 2015.
- [26] T. J. Soon. QR Code. *Synthesis Journal*, pages 59–78, 2008.
- [27] ISO/IEC Standard. ISO/IEC 18004:2000, Information technology – Automatic identification and data capture techniques – Bar code Symbology QR code, 2000.
- [28] ISO/IEC Standard. ISO/IEC 18004:2006, Information technology – Automatic identification and data capture techniques – QR code 2005 Bar code Symbology Specification, 2006.
- [29] About QR Codes Esponce.com , 2018. <http://www.esponce.com/resources/about-qr-codes>.

- [30] QR codes. QR Codes What Is A QR Code, 2017. <http://qrcode.meetheed.com>.
- [31] S. Demir, R. Kaynak, and K. A. Demir. Usage Level and Future Intent of Use of Quick response (QR) Codes for Mobile Marketing among College Students in Turkey. *Procedia-Social and Behavioral Sciences*, 181:405–413, 2015.
- [32] J. Palazón and A. Giráldez. QR Codes for Instrumental Performance in the Music Classroom. *International Journal of Music Education*, page 0255761418771992, 2018.
- [33] N. Gammer, T. Cherrett, and C. Gutteridge. Disseminating Real-time Bus Arrival Information via QR code Tagged Bus Stops: a Case Study of User Take-up and Reaction in Southampton, UK. *Journal of Transport Geography*, 34:254–261, 2014.
- [34] M. Pérez-Sanagustín, D. Parra, R. Verdugo, G. García-Galleguillos, and M. Nussbaum. Using QR Codes to Increase User Engagement in Museum-like Spaces. *Computers in Human Behavior*, 60:73–85, 2016.
- [35] J. Mira, M. Guilabert, I. Carrillo, C. Fernández, M. A. Vicente, D. Orozco-Beltrán, and V. F. Gil-Guillen. Use of QR and EAN-13 Codes by Older Patients Taking Multiple Medications for a Safer Use of Medication. *International journal of medical informatics*, 84(6):406–412, 2015.
- [36] V. Uzun and S. Bilgin. Evaluation and Implementation of QR Code Identity Tag System for Healthcare in Turkey. *SpringerPlus*, 5(1):1454, 2016.
- [37] C. E. H. Ventura, R. V. Aroca, A. Í. S. Antonialli, A. M. Abrão, J. C. Campos Rubiob, and MA M. A. Câmara. Towards Part Lifetime Traceability Using Machined Quick Response Codes. *Procedia Technology*, 26:89–96, 2016.
- [38] Admin. QR Codes For WiFi Network Sharing , 2018. <https://blog.qrstuff.com/2018/01/17/wifi-qr-codes>.
- [39] X. Zhu, Z. Hou, D. Hu, and J. Zhang. Secure and Efficient Mobile Payment Using QR Code in an Environment with Dishonest Authority. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pages 452–465. Springer, 2016.

- [40] Kaspersky Lab. Malicious QR Codes: Attack Methods & Techniques Infographic. https://usa.kaspersky.com/about/press-releases/2011_malicious-qr-codes-attack-methods--techniques-infographic, 2011.
- [41] P. Kieseberg, S. Schrittwieser, M. Leithner, M. Mulazzani, E. Weippl, L. Munroe, and M. Sinha. Malicious Pixels Using QR Codes as Attack Vector. *Trustworthy Ubiquitous Computing, series Atlantis Ambient and Pervasive Intelligence*, 6:21–38, 2012.
- [42] B. A. Forouzan. *Cryptography & Network Security*. McGraw-Hill, Inc., 2007.
- [43] I. Alsmadi, R. Burdwell, A. Aleroud, A. Wahbeh, M. Al-Qudah, and A. Al-Omari. *Practical Information Security: A Competency-Based Education Course*. Springer, 2018.
- [44] European Union Agency for Network and Information Security (ENISA). Algorithms, Key Size and Parameters Report – 2014, 2014. <https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014>.
- [45] X. Jin, X. Hu, K. Ying, W. Du, H. Yin, and G. Peri. Code Injection Attacks on HTML5-based Mobile for Apps: Characterization, Detection and Mitigation. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS'14)*, pages 66–77, 2014.
- [46] P. Wang, X. Yu, S. Chen, P. Duggisetty, S. Guo, and T. Wolf. CryptoPaper: Digital Information Security for Physical Documents. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC'15), April 13-17, Salamanca, Spain*, pages 2157–2164, 2015.
- [47] J. Gao, V. Kulkarni, H. Ranavat, L. Chang, and H. Mei. A 2D Barcode-based Mobile Payment System. In *Third International Conference on Multimedia and Ubiquitous Engineering (MUE'09), Qingdao, China, June 4-6*, pages 320–329, 2009.
- [48] G. Starnberger, L. Frohofer, and K. Goschka. QR-TAN: Secure Mobile Transaction Authentication. In *International Conference on Availability, Reliability and Security (ARES '09), March, 16th - 19th, Fukuoka, Japan*, pages 16–19, 2009.

- [49] Denso Wave Inc. SQRC® Secret-Function-Equipped QR Code, 2017. <https://www.denso-wave.com/en/adcd/product/software/sqrc/sqrc.html>.
- [50] 2D Technology Group Inc. Barcode Security Suite, 2016. <http://www.2dtg.com/node/74>.
- [51] V. Yakshtes and A. Shishkin. Mathematical Method of 2-D Barcode Authentication and Protection for Embedded Processing, 2012. <https://www.google.com/patents/US8297510>.
- [52] F. Razzak. Spamming the Internet of Things: A Possibility and its Probable Solution. In *Proceeding of the 9th International Conference on Mobile Web Information Systems (MobiWIS'12), Niagara Falls, Canada, August 27-29*, pages 658–665, 2012.
- [53] K. Peng, H. Sanabria, D. Wu, and C. Zhu. Security Overview of QR Codes, 2014. MIT Student Project: <https://courses.csail.mit.edu/6.857/2014/files/12-peng-sanabria-wu-zhu-qr-codes.pdf>.
- [54] GitHub. ZXing Project Home, 2018. <https://github.com/zxing/zxing/>.
- [55] GitHub. Short Payment Descriptor Project Home, 2018. <https://github.com/spayd/spayd-java>.
- [56] NortonMobile. Norton Snap QR Code Reader., 2016. https://support.norton.com/sp/en/us/home/current/solutions/v64691018_EndUserProfile_en_us?client=norton&site=nrt_n_US.
- [57] Red Dodo. QR & Barcode Reader (Secure), 2014. <http://reddodo.com/qr-barcode-scanner.php>.
- [58] H. Yao and D. Shin. Towards Preventing QR Code Based for Detecting QR Code Based Attacks on Android Phone Using Security Warnings. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security (ASIA CCS'13), Hangzhou, China, May 8-10*, pages 341–346, 2013.
- [59] Google. Google Safe Browsing API, website. <https://developers.google.com/safe-browsing/>.
- [60] Phishtank. Phishtank API, website. <https://www.phishtank.com/>.

- [61] Wired. Sneaky Exploit Allows Phishing Attacks from Sites that Look Secure, 2017. <https://www.wired.com/2017/04/sneaky-exploit-allows-phishing-attacks-sites-look-secure/>.
- [62] SFSEN ISO. 9241-11 (1998), 1998.
- [63] JSON, 2016. <http://www.json.org>.
- [64] ASN.1, 2017. <http://www.itu.int/en/ITU-T/asn1>.
- [65] Heider Wahsheh. Barcode Usability Tester, 2018. <https://play.google.com/store/apps/details?id=heider.barcodetesterv1>.
- [66] M. K. Smith. Summary Statistics for Skewed Distributions, 2016. <https://web.ma.utexas.edu/users/mks/statmistakes/skeweddistributions.html>.
- [67] D. S. Moore and S. Kirkland. *The Basic Practice of Statistics*, volume 2. WH Freeman New York, 2007.
- [68] D. M. Lane. *Hyperstat Online Statistics Textbook*. David M. Lane., 1993.
- [69] StatsDirect Limited. P-value, 2018. https://www.statsdirect.com/help/basics/p_values.htm.
- [70] J. H. McDonald. *Handbook of Biological Statistics*, volume 3. sparky house publishing Baltimore, MD, 2014.
- [71] A. Grover, P. Braeckel, K. Lindgren, H. Berghel, and D. Cobb. Parameters Effecting 2D Barcode Scanning Reliability. *Advances in Computers*, 80:209–235, 2010.
- [72] R. Alturki and V. Gay. Usability Testing of Fitness Mobile Application : Case Study Aded Surat App. *International Journal of Computer Science and Information Technology (IJCSIT)*, 9:107–127, 2017.
- [73] A. Sergeev . UI Designer - ISO-9241 Satisfaction Metrics - Theory of Usability, 2010. <http://ui-designer.net/usability/satisfaction.htm>.
- [74] D. Giry. Keylength - Cryptographic Key Length Recommendation, 2018. <https://www.keylength.com/>.

- [75] R. Dudheria. Evaluating Features and Effectiveness of Secure QR Code Scanners. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2017 International Conference on*, pages 40–49. IEEE, 2017.
- [76] Google Inc. Google Play Store., 2018. <https://play.google.com/store?hl=en>.
- [77] Heider Wahsheh. BarSec Droid, 2018. https://play.google.com/store/apps/details?id=barcode_security.heider.bsr.
- [78] G Data Software AG. G DATA QR Code Scanner, 2018. <https://www.gdata.de/>.
- [79] K. Krombholz, P. Frühwirt, T. Rieder, I. Kapsalis, J. Ullrich, and E. Weippl. QR Code Security—How Secure and Usable Apps Can Protect Users Against Malicious QR Codes. In *Availability, Reliability and Security (ARES), 2015 10th International Conference on*, pages 230–237. IEEE, 2015.
- [80] Trend Micro. QR Scanner - Free, Safe QR Code Reader, Zero Ads., 2018. https://www.trendmicro.com/en_us/business.html.
- [81] FANSec Lab Apps. Secure QR Code Scanner., 2018. <https://play.google.com/store/apps/details?id=com.fansec.lab.security.secureqrscodescanner>.
- [82] Madiff Net. QR & Barcode Security., 2017. <https://play.google.com/store/apps/details?id=com.trustbookin.qrcodebarcodesecurity>.
- [83] Dennings. Safe QR - Scanner & Generato., 2018. <http://www.dennings.org/>.
- [84] KidControl Dev. Safe Geotag QR Scanner., 2018. https://web.facebook.com/GeoTagQR?_rdc=1&_rdr.
- [85] Daniel Tengler. Crypto Message., 2018. https://play.google.com/store/apps/details?id=cz.crypto_message_free.apk.
- [86] Avira. Free QR Scanner., 2018. <https://www.avira.com/>.
- [87] Browser Extension. QR Code Scanner & Barcode Reader for CM Browser., 2018. <http://www.cmcm.com/en-us/>.

- [88] SECUSO Research Group. QR Scanner (Privacy Friendly), 2016. <https://secuso.aifb.kit.edu/index.php>.
- [89] X and C Hi-Tech Inc. Scan 2d Social QR Code Scanner., 2016. <http://www.scan2d.com/static/index.html>.
- [90] iTechSol. Secure QR Barcode Scanner., 2018. <https://play.google.com/store/apps/details?id=com.scanner.qr.barcode.reader.codes>.
- [91] Tokoware. Private QR Reader Free., 2016. <http://www.tokoware.com/>.
- [92] FancyApp. QR Code Reader Extreme., 2018. <https://play.google.com/store/apps/details?id=com.fancyapp.qrcode.barcode.scanner.reader>.
- [93] TeaCapps. QR & Barcode Reader., 2018. <https://play.google.com/store/apps/details?id=com.teacapps.barcodescanner>.
- [94] Ecrubit Consultancy Service. EC QR., 2018. <http://www.ecrubit.com/>.
- [95] Application4u. Lightning QR code Scanner., 2018. <http://ww7.application-4u.com/>.
- [96] Scan. QR Code Reader., 2016. <https://www.scan.me/>.
- [97] ZXing Team. Barcode Scanner ., 2017. <https://play.google.com/store/apps/details?id=com.google.zxing.client.android&hl=en>.
- [98] Geeks.Lab.2015. Barcode Scanner Pro., 2018. <https://play.google.com/store/apps/details?id=com.geekslab.qrbarcodescanner.pro>.
- [99] Gamma Play. QR & Barcode Scanner., 2018. <https://play.google.com/store/apps/details?id=com.gamma.scan>.
- [100] Barcode Scanner. Barcode Scanner., 2018. <https://barcodescannerblog.wordpress.com/>.
- [101] EZ to Use. All-in-one QR+Barcode Scanner: QR Scanner QR Reader., 2018. <https://play.google.com/store/apps/details?id=app.qrcode>.

- [102] I-Plex Technology. Secure Barcode Reader & QR Code Generator, 2018. <https://play.google.com/store/apps/details?id=com.iplextech.barcode.scanner>.
- [103] R. W. Reeder, A. P. Felt, S. Consolvo, N. Malkin, C. Thompson, and S. Egelman. An Experience Sampling Study of User Reactions to Browser Warnings in the Field. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 512. ACM, 2018.
- [104] Symantec Corporation. Norton Safe Web., 2018. <https://safeweb.norton.com/>.
- [105] G. Perlman. After Scenario Questionnaire, 2015. <http://garyperlman.com/quest/quest.cgi?form=ASQ>.
- [106] M. Farb, Y. Lin, T. Kim, J. McCune, and A. Perrig. Safeslinger: Easy-to-Use and Secure Public-Key Exchange. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 417–428. ACM, 2013.

Appendices

Appendix A

BarSec Desktop and BarSec Droid Applications

BarSec desktop application is a security tool that adds security features to protect two-dimensional barcodes, using cryptography and URL checking against Web attacks. The current version 1.0. *BarSec Droid* is a barcode reader application that is designed for Android smartphones. *BarSec Droid* supports reading and processing secure QR codes generated by (*BarSec*) as well as checking online contents.

1. **Generate QR Code without Cryptographic Protection**

BarSec desktop application is used to generate and read QR codes. To generate QR code without any cryptographic protection, users should follow these steps (Figure A.1):

- (a) Enter barcode contents or select file.
- (b) Determine barcode image size (pixels).
- (c) Choose error correction level.
- (d) Generate barcode.

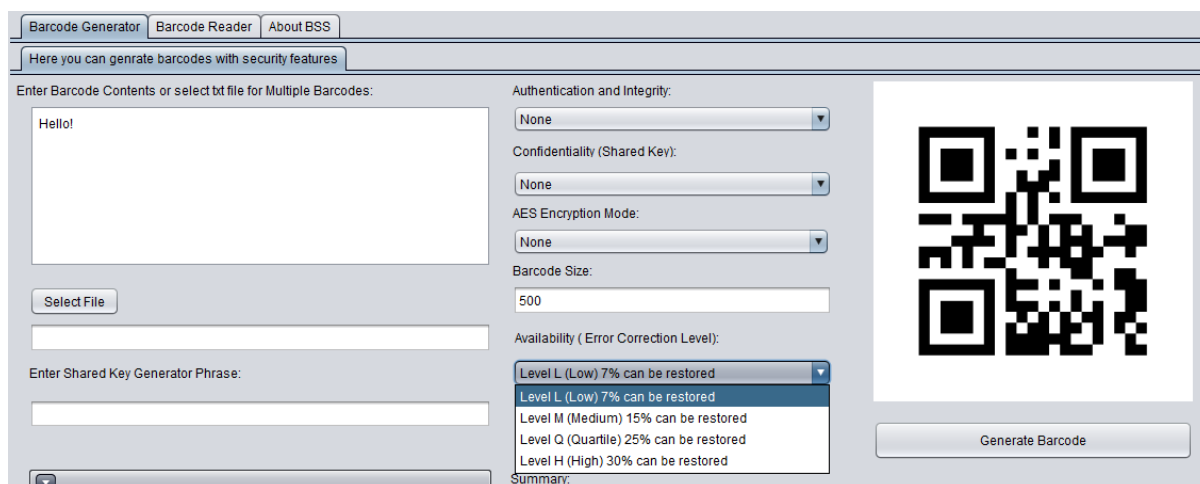


Figure A.1: Generate QR code without cryptographic protection.

2. Authentication and Integrity for QR Code

In order to provide authentication and integrity for QR code contents, users should follow these steps:

- (a) Enter barcode contents or select file.
- (b) Determine barcode image size (pixels).
- (c) Choose error correction level.
- (d) Choose the appropriate mechanism:
 - For digitally signed QR codes, the user will have the choice to include the certificate inside QR code (Figure A.2).



Figure A.2: Digitally signed QR codes.

- For HMAC, the user should enter shared key generator phrase (Figure A.3).

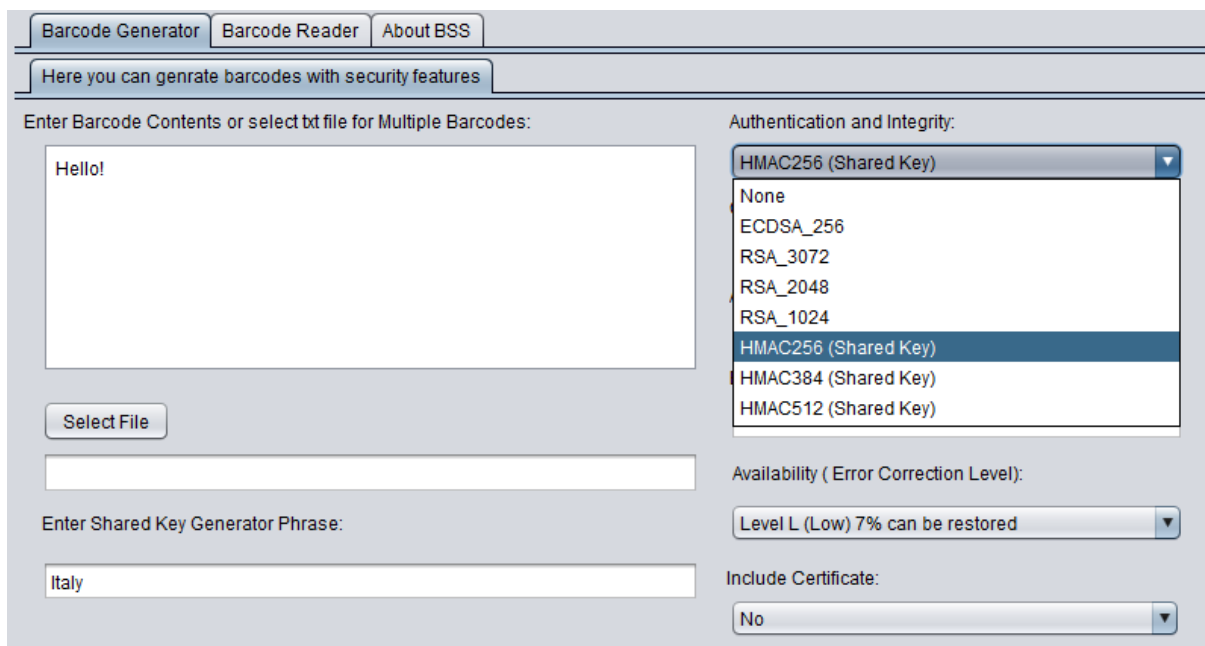


Figure A.3: QR code with HMAC cryptographic protection.

- (e) Generate barcode: the user will determine a directory to store QR code image, and *BarSec* will display usability level message.

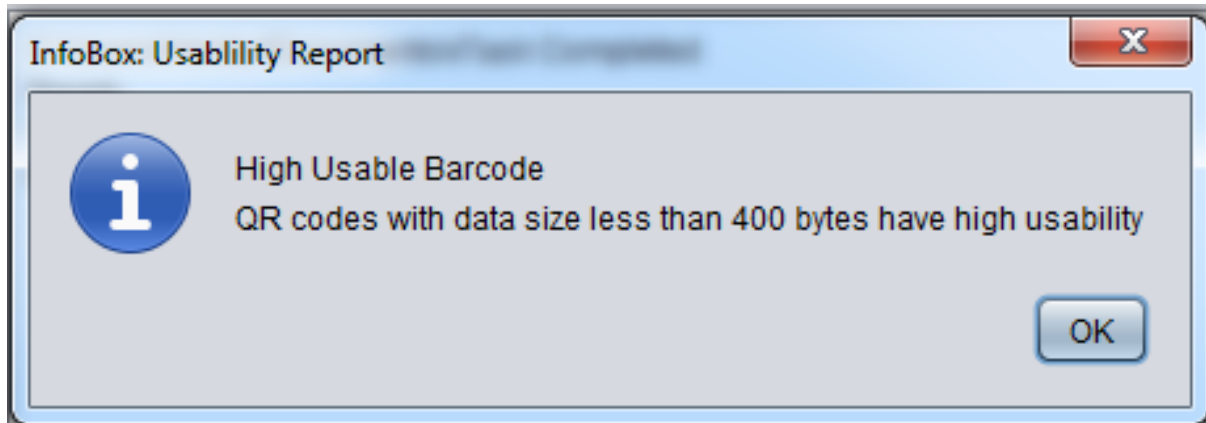


Figure A.4: QR code usability level message.

3. Encrypt QR Code Contents

In order to provide confidentiality for QR code contents, users should follow these steps:

- (a) Enter barcode contents.
- (b) Determine barcode image size (pixels).
- (c) Choose error correction level.
- (d) Choose the appropriate AES mode and enter shared key generator phrase.



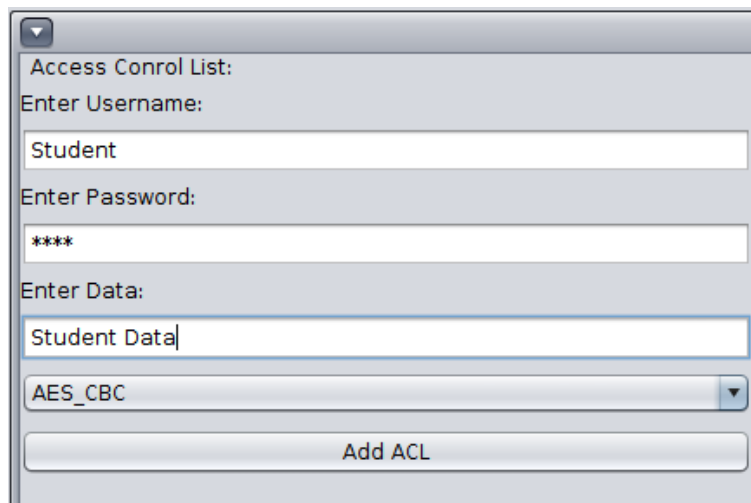
Figure A.5: Encrypt QR code contents (password: Italy).

- (e) Generate barcode.

4. Access Control List for QR Code Contents

QR code generator can add Access Control List (ACL) in special frame as the following:

- (a) Enter username.
- (b) Enter password.
- (c) Enter data.
- (d) Choose encryption mode.
- (e) Add ACL, note that users can add more than one ACL in the same QR code.



The screenshot shows a software interface for adding an Access Control List (ACL). It features a title bar with a close button. Below the title bar, the text "Access Control List:" is displayed. The form contains four input fields: "Enter Username:" with the text "Student", "Enter Password:" with "****", "Enter Data:" with "Student Data", and a dropdown menu for encryption mode currently set to "AES_CBC". At the bottom of the form is a button labeled "Add ACL".

Figure A.6: ACL for QR code contents.

- (f) Generate barcode.

5. Read QR Code Contents

To read QR code contents, users can use either *BarSec* desktop application or *BarSec Droid*. In both applications users can read barcode contents, read ACL, process data and check online contents (Figure A.7 and Figure A.8).

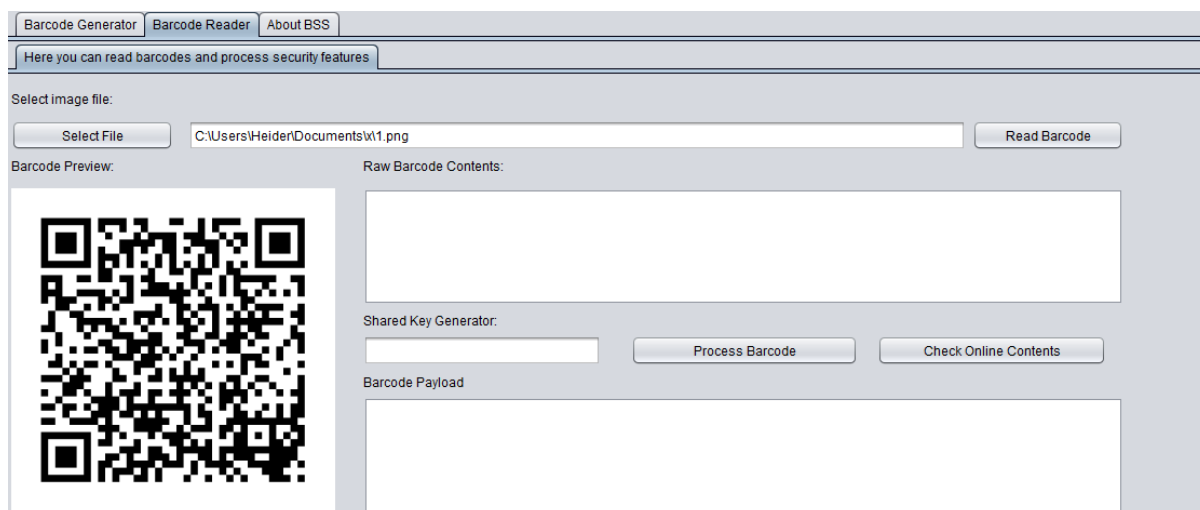


Figure A.7: Read QR code contents using *BarSec* desktop application (password: Italy).

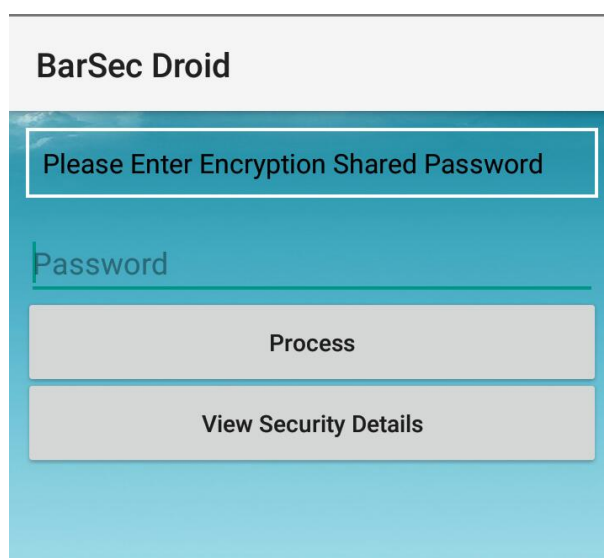


Figure A.8: Process QR code contents using *BarSec Droid*.

Reading a QR Code with *BarSec Droid* include the following steps:

- (a) Scan QR code (Figure A.9).

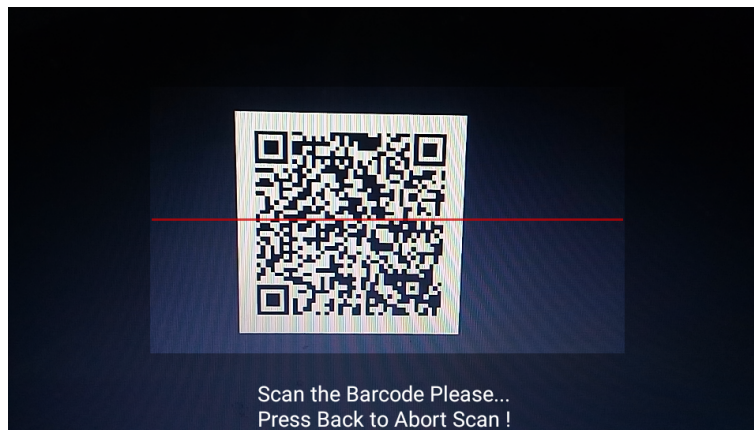


Figure A.9: Scan QR code using *BarSec Droid*.

(b) *BarSec Droid* will display the contents, and provide security summary (Figure A.10).

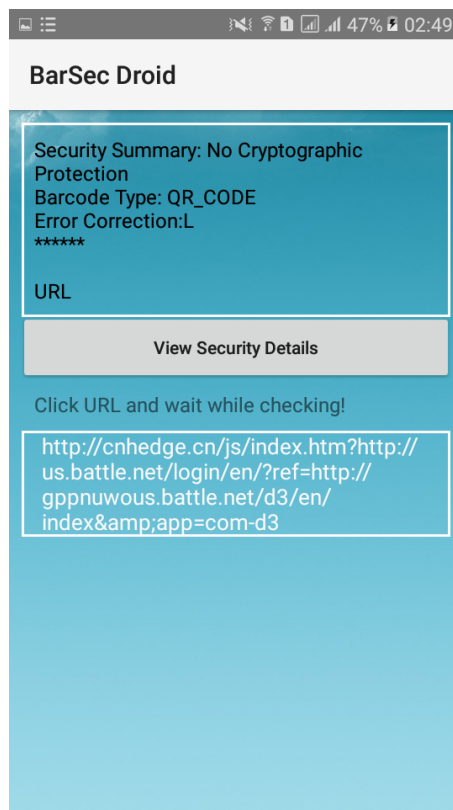


Figure A.10: Display QR code contents and security summary using *BarSec Droid*.

(c) For more information, the user can display security details (Figure A.11).

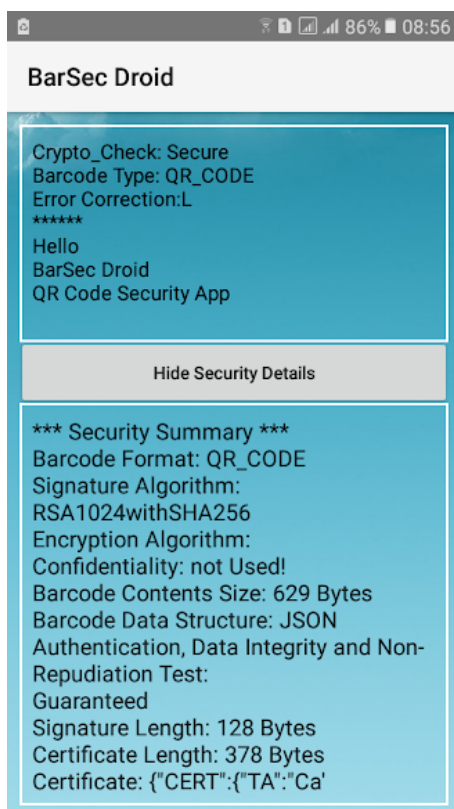


Figure A.11: Display QR code security details using *BarSec Droid*.

- (d) *BarSec Droid* will display warning message if the user tries to visit malicious URL (Figure A.12).

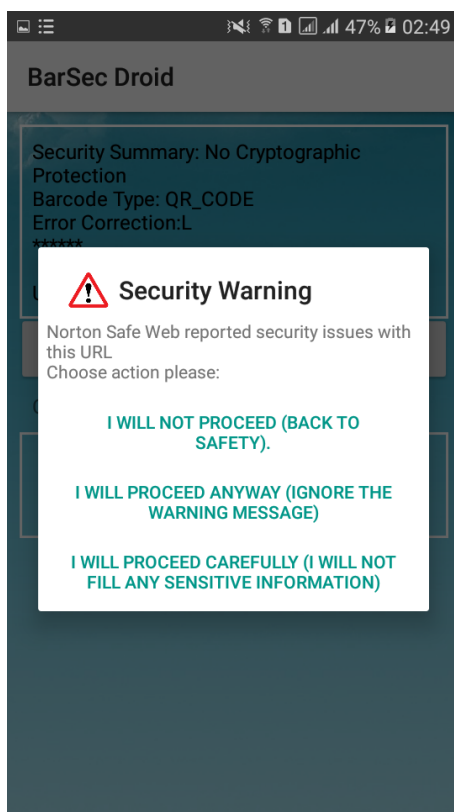


Figure A.12: Display *BarSec Droid* warning message.

(e) *BarSec Droid* allows users to display and process ACL (Figure A.13).

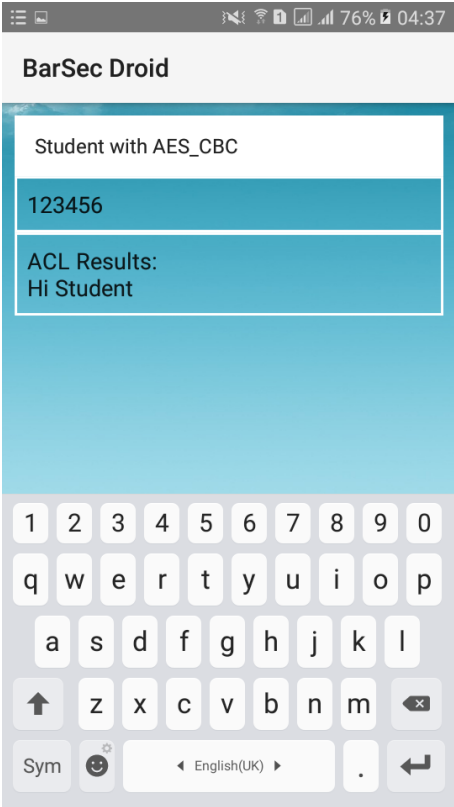


Figure A.13: Display and process ACL using *BarSec Droid*.

Estratto per riassunto della tesi di dottorato

Studente: Heider Ahmad Mutleq WAHSHEH **matricola:** 956262

Dottorato: Informatica

Ciclo: 31°

Titolo della tesi¹: **Secure and Usable QR Codes**

Riassunto: Il codice a barre è una tecnologia universale che fornisce una rappresentazione visiva dei dati utilizzando serie di linee orizzontali (1D), o matrici di riquadri e punti (2D), organizzati in strutture particolari. I codici a barre vengono rappresentati come immagini che possono immagazzinare dati di varia natura e dimensioni, usati per identificare l'oggetto che li contiene. Nella letteratura scientifica non vi è traccia di meccanismi standard in grado di garantire l'autenticità e la confidenzialità del contenuto del codice a barre. Attacchi come la codifica dei link malevoli sono realistici.


In questo lavoro presenteremo uno studio comparativo delle minacce ai codici a barre 2D e dei meccanismi di protezione disponibili. Sottolineeremo i limiti di questi meccanismi ed analizzeremo le loro potenzialità a livello di sicurezza. Inoltre proporremo soluzioni pratiche basate sulle raccomandazioni dell'ENISA (Ente Europeo per la Sicurezza dei Network e le Tecnologie Informatiche).

Presenteremo inoltre il primo studio sistematico sulle primitive crittografiche utilizzabili dentro i codici QR. Discuteremo numerose sperimentazioni condotte per analizzare i fattori che influenzano l'utilizzo dei codici a barre. Sulla base delle normative ISO 9241, abbiamo definito un "Barcode Usability Score" (BarScore), un valore osservabile e misurabile che rappresenta l'usabilità globale dei codici a barre, calcolato su una media dell'efficacia, efficienza e soddisfazione. Abbiamo redatto una guida all'utilizzo dei codici a barre basata sulle dimensioni dei dati. Abbiamo inoltre realizzato un generatore di codici QR sicuri e utilizzabili e abbiamo messo a confronto la firma digitale e i meccanismi di crittografia basati sull'utilizzo e la sicurezza. I risultati ottenuti mostrano che i codici QR possono fornire soluzioni applicabili e sicure.

Abbiamo infine presentato un'ampia panoramica sulle applicazioni per la lettura dei codici a barre analizzando le loro caratteristiche. Abbiamo suddiviso queste applicazioni in quattro gruppi: sicurezza degli URL, sicurezza basata sulla crittografia, applicazioni per la difesa della privacy e altre applicazioni popolari. Abbiamo anche evidenziato le loro debolezze e fornito suggerimenti per applicazioni efficaci, sicure e in grado di garantire la sicurezza e la protezione. Abbiamo sviluppato di codici a barre con sistema operativo lettore Android che segue i nostri consigli e abbiamo condotto un'indagine sulla sua usabilità e sicurezza. I risultati mostrano che quando si seguono i suggerimenti di progettazione, la consapevolezza della sicurezza dell'utente e l'usabilità aumentano.

Parole chiave: QR Codes, Cryptography, QR Code Security, QR Code Usability, QR Code Privacy, Barcode Scanners, Android Security.

Firma dello studente


Heider Wahsheh

¹ Il titolo deve essere quello definitivo, uguale a quello che risulta stampato sulla copertina dell'elaborato consegnato.

English**The thesis title is: Secure and Usable QR Codes**

Abstract: Barcode is a universal technology that provides visual data representation using series of horizontal lines (1D), or matrix of squares and dots (2D), organized in a specific standard way. Barcodes are represented as images that can store data with various data types and sizes, used to identify objects. In the literature, there is no standard mechanism for providing authenticity and confidentiality of the barcode content. Attacks such as the malicious links encoding are realistic and feasible in practice.

In this work, we present a comparative study of 2D barcodes' threats and the available protection mechanisms. We highlight the limitations of these mechanisms, and explore their security capabilities. Moreover, we suggest practical solutions based on the recommendations from the European Union Agency for Network and Information Security (ENISA).

For what concerns usability, we present the first systematic study of usable cryptographic primitives inside QR codes. We have performed extensive experiments to analyze the factors that affect the barcodes usability, by developing a barcode reader application that collects the users' feedback. We have analyzed Scanning Time, data size, image size and users' feedback. Based on ISO 9241, we have defined Barcode Usability Score (*BarScore*) an observable and quantifiable value that represents the overall usability, by calculating the average of effectiveness, efficiency and satisfaction. We have built a barcode usability guidance for recommended image and data sizes under different usability levels. Then, we have implemented a systematic secure/usable QR code generator and compared the digital signature and encryption mechanisms based on usability and security. The obtained results show that QR codes can support powerful, usable and secure solutions.

Finally, we present a comprehensive review of barcode reader applications by analyzing their properties. We categorize these apps into four groups; URLs security, Crypto-based security, Popular applications and Save-privacy. We also highlight their weaknesses and present design recommendations for usable, secure and privacy-guaranteed scanner applications. We have developed a proof-of-concept Android reader app that follows our recommendations, and performed a user usability and security survey. The results show that when following the design tips, user's security awareness and usability increase.

Keywords: QR Codes, Cryptography, QR Code Security, QR Code Usability, QR Code Privacy, Barcode Scanners, Android Security.

Student's Signature



Heider Wahsheh



DEPOSITO ELETTRONICO DELLA TESI DI DOTTORATO

DICHIARAZIONE SOSTITUTIVA DELL'ATTO DI NOTORIETA'

(Art. 47 D.P.R. 445 del 28/12/2000 e relative modifiche)

Io sottoscritto Heider Ahmad Mutleq WAHSHEH

nato.. a Irbid - Jordan (prov.) il 22/08/1987

residente a Italy in Dorsoduro 2408 Venezia n.

Matricola (se posseduta) 956262 Autore della tesi di dottorato dal titolo:
Secure and Usable QR Codes

Dottorato di ricerca in Informatica (Computer Science)

(in cotutela con

Ciclo 31

Anno di conseguimento del titolo 2019

DICHIARO

di essere a conoscenza:

- 1) del fatto che in caso di dichiarazioni mendaci, oltre alle sanzioni previste dal codice penale e dalle Leggi speciali per l'ipotesi di falsità in atti ed uso di atti falsi, decado fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
- 2) dell'obbligo per l'Università di provvedere, per via telematica, al deposito di legge delle tesi di dottorato presso le Biblioteche Nazionali Centrali di Roma e di Firenze al fine di assicurarne la conservazione e la consultabilità da parte di terzi;
- 3) che l'Università si riserva i diritti di riproduzione per scopi didattici, con citazione della fonte;
- 4) del fatto che il testo integrale della tesi di dottorato di cui alla presente dichiarazione viene archiviato e reso consultabile via Internet attraverso l'Archivio Istituzionale ad Accesso Aperto dell'Università Ca' Foscari, oltre che attraverso i cataloghi delle Biblioteche Nazionali Centrali di Roma e Firenze;
- 5) del fatto che, ai sensi e per gli effetti di cui al D.Lgs. n. 196/2003, i dati personali raccolti saranno trattati, anche con strumenti informatici, esclusivamente nell'ambito del procedimento per il quale la presentazione viene resa;
- 6) del fatto che la copia della tesi in formato elettronico depositato nell'Archivio Istituzionale ad Accesso Aperto è del tutto corrispondente alla tesi in formato cartaceo, controfirmata dal tutor, consegnata presso la segreteria didattica del dipartimento di riferimento del corso di dottorato ai fini del deposito presso l'Archivio di Ateneo, e che di conseguenza va esclusa qualsiasi responsabilità dell'Ateneo stesso per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
- 7) del fatto che la copia consegnata in formato cartaceo, controfirmata dal tutor, depositata nell'Archivio di Ateneo, è l'unica alla quale farà riferimento l'Università per rilasciare, a richiesta, la dichiarazione di conformità di eventuali copie;

Data 6/12/2018

Firma Heider wahsheh

NON AUTORIZZO

l'Università a riprodurre ai fini dell'immissione in rete e a comunicare al pubblico tramite servizio on line entro l'Archivio Istituzionale ad Accesso Aperto la tesi depositata per un periodo di 12 (dodici) mesi a partire dalla data di conseguimento del titolo di dottore di ricerca.

DICHIARO

- 1) che la tesi, in quanto caratterizzata da vincoli di segretezza, non dovrà essere consultabile on line da terzi per un periodo di 12 (dodici) mesi a partire dalla data di conseguimento del titolo di dottore di ricerca;
- 2) di essere a conoscenza del fatto che la versione elettronica della tesi dovrà altresì essere depositata a cura dell'Ateneo presso le Biblioteche Nazionali Centrali di Roma e Firenze dove sarà comunque consultabile su PC privi di periferiche; la tesi sarà inoltre consultabile in formato cartaceo presso l'Archivio Tesi di Ateneo;
- 3) di essere a conoscenza che allo scadere del dodicesimo mese a partire dalla data di conseguimento del titolo di dottore di ricerca la tesi sarà immessa in rete e comunicata al pubblico tramite servizio on line entro l'Archivio Istituzionale ad Accesso Aperto.

Specificare la motivazione:

- motivi di segretezza e/o di proprietà dei risultati e/o informazioni sensibili dell'Università Ca' Foscari di Venezia.
- motivi di segretezza e/o di proprietà dei risultati e informazioni di enti esterni o aziende private che hanno partecipato alla realizzazione del lavoro di ricerca relativo alla tesi di dottorato.
- dichiaro che la tesi di dottorato presenta elementi di innovazione per i quali è già stata attivata / si intende attivare la seguente procedura di tutela:

.....;

- Altro (specificare):

Involved external authors and papers from thesis are under-review

.....

.....

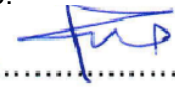
.....

A tal fine:

- dichiaro di aver consegnato la copia integrale della tesi in formato elettronico tramite auto-archiviazione (upload) nel sito dell'Università; la tesi in formato elettronico sarà caricata automaticamente nell'Archivio Istituzionale ad Accesso Aperto dell'Università Ca' Foscari, dove rimarrà non accessibile fino allo scadere dell'embargo, e verrà consegnata mediante procedura telematica per il deposito legale presso la Biblioteca Nazionale Centrale di Firenze;

- consegno la copia integrale della tesi in formato cartaceo presso la segreteria didattica del dipartimento di riferimento del corso di dottorato ai fini del deposito presso l'Archivio di Ateneo.

Data 6/12/2018

Firma Heider Wahsheh 

La presente dichiarazione è sottoscritta dall'interessato in presenza del dipendente addetto, ovvero sottoscritta e inviata, unitamente a copia fotostatica non autenticata di un documento di identità del dichiarante, all'ufficio competente via fax, ovvero tramite un incaricato, oppure a mezzo posta.

Firma del dipendente addetto

Ai sensi dell'art. 13 del D.Lgs. n. 196/03 si informa che il titolare del trattamento dei dati forniti è l'Università Ca' Foscari - Venezia.

I dati sono acquisiti e trattati esclusivamente per l'espletamento delle finalità istituzionali d'Ateneo; l'eventuale rifiuto di fornire i propri dati personali potrebbe comportare il mancato espletamento degli adempimenti necessari e delle procedure amministrative di gestione delle carriere studenti. Sono comunque riconosciuti i diritti di cui all'art. 7 D. Lgs. n. 196/03.