|  |
| --- |
| University of Missouri – St. Louis |
| **Beyond Fobs: A Secure QR Code-Based Access System** |
|  |

|  |
| --- |
| Patrick Jennewein  5-1-2025 |

Table of Contents

[Abstract 2](#_Toc195011624)

[Introduction 2](#_Toc195011625)

[The Structure of QR Codes 2](#_Toc195011626)

[Data Masking and Reversing 4](#_Toc195011627)

[Comparing Traditional Barcodes to QR Codes 6](#_Toc195011628)

[The Rising Popularity of QR Codes 6](#_Toc195011629)

[QR Codes for Access Control 7](#_Toc195011630)

[Problem Statement 7](#_Toc195011631)

[Objective 7](#_Toc195011632)

[Literature Review 7](#_Toc195011633)

[Techniques in QR-Based Authentication 7](#_Toc195011634)

[Security Vulnerabilities in QR-Based Authentication 7](#_Toc195011635)

[Time-Based Authentication with QR Codes 8](#_Toc195011636)

[Methodology and Implementation 8](#_Toc195011637)

[System Overview 8](#_Toc195011638)

[Technologies Used 8](#_Toc195011639)

[Enhanced QR Code Detection 8](#_Toc195011640)

[Time-Based QR Code Refresh System 8](#_Toc195011641)

[Challenges and Solutions 8](#_Toc195011642)

[Experimental Results and Evaluation 8](#_Toc195011643)

[Test Setup 8](#_Toc195011644)

[Performance Analysis 8](#_Toc195011645)

[Security Analysis 8](#_Toc195011646)

[User Experience 9](#_Toc195011647)

[Discussion 9](#_Toc195011648)

[Conclusion 9](#_Toc195011649)

[Works Cited 10](#_Toc195011650)

## Abstract

In this project, I propose a secure QR code authentication system that enhances traditional authentication methods.

While QR codes are widely used for authentication, conventional systems simply attempt to match a displayed QR code to a database. In the best case, these standard QR code detection methods are simply unreliable in challenging environments such as in low-light conditions with user-induced motion blur. In the worst case, standard QR code detection methods are subject to cybersecurity threats.

To address these issues, this project integrates two key components: (1) Enhanced QR code detection using adaptive thresholding, homography transformations, and deblurring techniques to improve robustness in real-world conditions; and (2) A time-based system that ensures QR codes dynamically refresh to prevent reuse or replay.

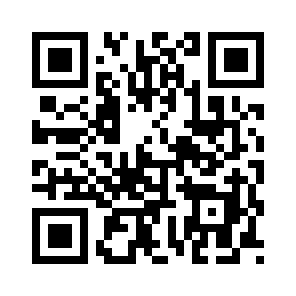
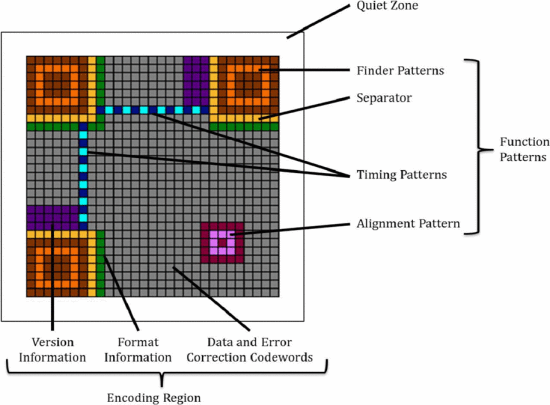
This approach provides a secure and user-friendly authentication solution that can be deployed in various environments, offering improved protection while maintaining ease of use.

## Introduction

“Quick Response” codes, otherwise known as “QR” codes, give users a two-dimensional matrix of white and black pixels (*Figure*). These codes – unlike more standardized one-dimensional barcodes – are capable of storing large amounts of data, offer a much faster recognition, and can be read omni-directionally.

### The Structure of QR Codes

#### QR Code Patterns

QR code functionality is simple, but usually effective. These codes contain four types of patterns, giving it structural integrity and enabling accurate data retrieval [1]. The *finder pattern* encompasses three of the four corners of the QR code and correctly orients the decoder with its perfect ratio of a 7 x 7 outer dark square, 5 x 5 inner light square, and 3 x 3 inner dark square. *Separators* encompass the white space around each of the three finder patterns. The *timing patterns* are alternating black and white pixels that form horizontal and vertical lines in 6th row and 6th column, respectively, to ensure proper alignment. Finally, *alignment patterns* consist of a 5 x 5 dark module surrounding a 3 x 3 light module, and 1-pixel dark module. See *Figure right* for a visual description [1]. These patterns distinguish QR codes from other encoding methods.

#### The Basics of Reading QR Code Data

To understand how the error correction works one must understand how the QR is read from the camera:

1. The QR code image is scanned by the camera.
2. Each pixel[[1]](#footnote-1) of the QR code is interpreted as a bit. A black pixel is interpreted as a 1 and a white pixel is interpreted as a 0.
3. Bits are read following a zigzag pattern and grouped into 8-bit chunks – or bytes. For example, if the camera scans in an alternating set of black and white bits (10101010), this data is converted as the byte 170.
4. These bytes are then interpreted as a message. If some data is corrupted or missing, *Reed-Solomon* algorithms (discussed later) use the mathematical relationship between bytes to detect and correct the errors before the message is displayed [2]

A simple example can be seen in (*Figure*) below, which is a QR code holding a text-based URL [3]. Decoding a QR code starts from the bottom right corner, continuing up and around as per the orange arrows shown in a zigzag pattern. The first four bits, pictured as Enc below, gives the “encoding mode,” of which there are 10 different modes as of this publication.[[2]](#footnote-2) The second 8 bits, pictured as Len below, indicate how many characters should be read from the data stream.

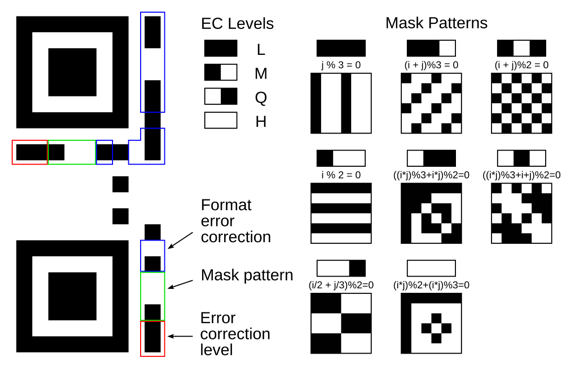
 

### Data Masking and Reversing

In analyzing the first 3 bytes (decoded as w, w, and w) from *Figure* or *Figure*, one may notice that these bytes seemingly do not match despite all bytes supposed to be representing the same data: w. From a cursory glance, the bytes are as follows:[[3]](#footnote-3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Original** | | **Segments** | | | |
| Decoding small QR codes by hand | | **Original Byte** | **Interpretation Order** | | **Interpretation** |
|  |  | | 10111011  (187) |
|  |  | | 10110100  (180) |
|  |  | | 01000100  (68) |
|  |  | | |  | |

This is because the *mask pattern* has not yet been applied. Masking occurs, “when an object (called the target) is affected by the presence of a second object (called the mask)” [4]. In this context, the *target* is the original byte of data while the mask is the predefined binary pattern applied to those bits (see *Figure*). This masking process alters the visual layout the QR code to prevent problematic patterns – such as large areas of uniform bits or sequences that may resemble finder patterns. In this QR code, the mask pattern in *Figure* is 001, and can be found from the format information in the QR code (*Figure*).



Thus, for every even-numbered row (starting from the top left corner of the QR code), the bit will be flipped. The even-numbered rows are highlighted below and the corresponding encodings are also given.

|  |  |  |  |
| --- | --- | --- | --- |
| **Original** | **Segments** | | |
|  | **Original Byte** | **Update** | **Interpretation** |
|  |  | 10001000  (136) |
|  |  | 10001000  (136) |
|  |  | 10001000  (136) |

Though while each byte now matches 136, the ASCII value for w is 119. The discrepancy lies in that, after a QR code has been interpreted from its *masked* version, the reader must thereby *unmask* it – flipping the outputs once more. This means that each byte of 10001000 becomes flipped, and the reader finally interprets the byte as 01110111, or 119 (w).

#### Reed-Solomon Codes

An issue arises when a QR code is not read perfectly: perhaps the screen holding the QR code has a smudge or water droplet, or perhaps the user has a finger over a portion of the QR code. When there is an issue with the QR code, *Reed-Solomon* *Codes* alleviate this by providing redundancy.

Formally, a Reed-Solomon code is a BCH[[4]](#footnote-4) code where is the total number of bytes in a QR code block – including the actual message and the error correction bytes; is how much real data (text, a URL, etc.) is being sent; is the minimum number of symbol differences that can be detected and corrected, and is all possible values. [2]

1. Choose a message 🡪 17 characters is 17 bytes
2. Choose QR code version and error correction level
3. Convert message into bytes
4. Build the message polynomial
   1. [www.wikipedia.org](http://www.wikipedia.org) 🡪 119 + 119x + 119x^2 + 46x^3
5. Multiply by x^(n-k) to shift the polynomial
6. Divide the generator polynomial
7. Combine data and error correction bytes
8. Scan and error correction in action

### Comparing Traditional Barcodes to QR Codes

The previous analysis underscores how much more sophisticated that QR codes are versus traditional barcodes (Figure). Some features merit even further discussion.

A black background with numbers

AI-generated content may be incorrect.

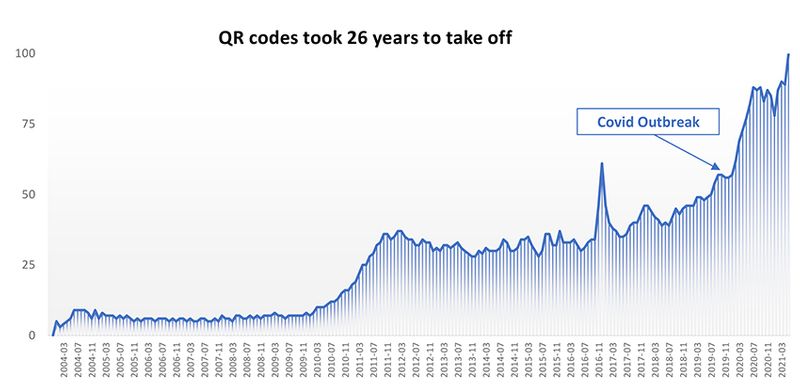
Traditional barcodes (e.g. a *universal product code* or *UPC*) can store only up to 20 digits. QR codes can store over 7,000 characters of data, including numeric data, alphanumeric data, binary data, and Kanji data. In addition, QR codes can encode URLs, e-mail addresses, and multimedia. As such, QR codes can encode 10 times more data than a barcode of the same size [5].

Furthermore, QR codes are “rapidly readable” from any direction, all made possible from the detection patterns in the 3 corners of the code. The structure of these codes also evades background interreference. UPCs do not have these features. [5]

### The Rising Popularity of QR Codes

While QR codes were created in 1994 as a means of tracking inventory, they become much more popular with the increasing use of smartphones.

According to a 2011 study [6], 20.1 million American smartphone owners used their device to scan a QR code in a 3-month average period. Among those tested, 44% scanned from a retail store and 59.4% scanned from home. This study hypothesized that the QR code was becoming the retailer’s “secret weapon.” For example, electronics retailer Best Buy adopted QR code integration on product tags to view and compare key features of the product.



But the popularity of QR codes in 2011 pales in comparison to what occurred during the COVID-19 outbreak (see *Figure*). Between March 2020 and December 2020, 8.74 million users began using QR codes for mobile payment, and QR code payments represented 85% of all mobile payments in 2020 [7]. In addition, there was a 110% increase in QR code usage in general from 2019 to 2020, and a 28% increase from 2020 to 2021. Overall, the pandemic was a catalyst for the widespread adoption of QR codes, transforming them from a technology used in specific contexts to one that was much more ubiquitous.

### QR Codes for Access Control

### Problem Statement

While QR codes offer an efficient way to validate access, current QR code detection methods have two critical shortcomings: poor detection when environmental conditions are challenging and weak protection against re-use.

### Objective

The primary objective of this project is to develop a secure and robust QR code authentication system that addresses challenges related to detection and security challenges.To achieve this, the system will integrate two key innovations:

First, this project implements adaptive thresholding, homography transformations, and deblurring techniques to improve QR code readability under the real-world conditions including poor lighting and motion blur. Second, this project refreshes QR codes, allowing each QR code to be used only once.

Together, these improvements deliver a more secure, user-friendly, and resilient QR code authentication solution suitable for use in schools and other environments needing access control.

## Literature Review

### Techniques in QR-Based Authentication

What approaches have been attempted? What were the results?

### Security Vulnerabilities in QR-Based Authentication

What approaches have been attempted? What were the results?

### Time-Based Authentication with QR Codes

What approaches have been attempted? What were the results?

## Methodology and Implementation

### System Overview

High level description of the approach with a flowchart.

### Technologies Used

What were all the technologies used?

### Enhanced QR Code Detection

#### Adaptive Thresholding

Text here.

#### Homography Transformations

Text here.

#### Deblurring Techniques

Text here.

### Time-Based QR Code Refresh System

Talk about the implementation of dynamic QR codes that prevent reuse or replay.

### Challenges and Solutions

What sorts of issues arose and how were they solved?

## Experimental Results and Evaluation

### Test Setup

How did I test my system, including what dataset and parameters?

### Performance Analysis

How well did the system hold up?

### Security Analysis

How well did the system hold up?

### User Experience

How well did the system hold up?

## Discussion

How should the results be interpreted?

## Conclusion

Summary of key findings and final remarks.

## Works Cited

|  |  |
| --- | --- |
| [1] | S. Tiwari, An Introduction to QR Code Technology, Bhubaneswar: 2016 International Conference on Information Technology (ICIT), 2016, pp. 39-44. |
| [2] | B. S. Bajaj, "Reliability on QR Codes and Reed-Solomon Codes," Canary Islands, 2025. |
| [3] | Wikipedia, "QR Code," 14 October 2011. [Online]. Available: https://commons.wikimedia.org/wiki/File:QR\_Character\_Placement.svg. [Accessed 27 March 2025]. |
| [4] | J. Zhang, D. Li, J. Jia, W. Sun and G. Zhai, "Protection and Hiding Algorithm of QR Code Based on Multi-channel Visual Masking," *2019 IEEE Visual Communications and Image Processing (VCIP),* pp. 1-4, 2019. |
| [5] | J. H. Chang, "An introduction to using QR codes in scholarly journals," *Sci Ed,* vol. 1, no. 2, pp. 113-117, 2014. |
| [6] | Retail TouchPoints, "QR Code Trends And Early Adopters," 14 February 2012. [Online]. Available: https://www.retailtouchpoints.com/topics/digital-marketing/mobile-marketing/qr-code-trends-and-early-adopters?utm\_source=chatgpt.com. [Accessed 8 April 2025]. |
| [7] | M. Tu, L. Wu, H. Wan, Z. Din, Z. Guo and J. Chen, "The Adoption of QR Code Mobile Payment During COVID-19: A Social Learning Perspective," *Frontiers in Psychology,* 2022. |

1. A *pixel* can be more appropriately referred to as a *module* [↑](#footnote-ref-1)
2. Modes include 0001 for numeric encoding, 0010 for alphanumeric encoding, 0100 for byte encoding, 1000 for Kanji encoding, etc. [3] [↑](#footnote-ref-2)
3. The three figures are recreations of the QR code for “www.wikipedia.com.” Each figure is a “w” in “www”. [↑](#footnote-ref-3)
4. Explain BCH. [↑](#footnote-ref-4)