

# Configuration Manual

MSc Internship  
Cyber Security

**Rohit Jain**  
Student ID: x18164455

School of Computing  
National College of Ireland

Supervisor: Christos Grecos

**National College of Ireland**  
**MSc Project Submission Sheet**  
**School of Computing**



**Student Name:** Rohit Jain  
**Student ID:** x18164455  
**Programme:** MSc Cyber Security **Year:** 2019-20  
**Module:** Internship  
**Lecturer:** Christos Grecos  
**Submission Due Date:** 13/12/2019  
**Project Title:** Enhancing the security of message in the QR Code using a Combination of Steganography and Cryptography

**Page Count: 7**

**Word Count:**1083

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

**Signature:** .....

**Date:** .....

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission,</b> to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project,</b> both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual

Rohit Jain

Student ID: x18164455

## 1 Introduction

In this document, we will provide a proper walkthrough of the proposed method. How the technique implemented and used step by step. The message is securely transmitting using cryptography, and steganography in a QR Code has proved the most effective way of message transmission. The suggested research concentrates on more security with a different method and a secure ECC algorithm. The Key size of the 160-bit ECC has an equal security level with 1024-bit RSA/ DSA. The advantage of ECC over RSA is distinct since, with a smaller length for the key, it can provide the same level of security [1]. This design uses the ECC algorithm, ECDH, AEAD algorithm for encryption, and decryption of the message. With the ECC algorithm, we are saving our data in a brainpoolP256r1 curve. ECDH algorithm we use for generating a secret key, and we are using that secret key and Nonce as an input to ChaCha20Poly1305 to encrypt our plaintext. ChaCha20 is a cipher, whereas Poly1305 is an authenticator, both as a separate algorithm and as a combined mode or Authenticated Encryption with Associated Data (AEAD) algorithm. For steganography, we used the LSB technique for watermarking. To increase security in steganography, we used the One Time Pad algorithm.

## 2 System Configuration - 1

- Operating system: Windows 10
- Processor: Two CPU
- System: 32 bits or 64 bits
- Hard Disk: 10 GB
- Memory: 2 GB

### 2.1 System Configuration – 2

- Operating system: Parrot security (Linux distribution)
- Processor: Two CPU
- System: 64 bits
- Hard Disk: 10 GB
- Memory: 4 GB

## 3 Working

### 3.1 Message encryption process

#### Libraries to Install

- In this part we will discuss how to install python libraries to run the message encryption and decryption code. first install the pip library<sup>1</sup>. Pip library used to install the software packages which is written in the python. After installing the pip library we can install tinyec library as shown in the figure 1. Tinyec library used for arithmetic operation on the elliptic curves in the python.

```
C:\Users\ROHIT>pip install tinyec
Collecting tinyec
  Using cached https://files.pythonhosted.org/packages/c2/00/977e7339ae19b42ae10e1219e5b13c0f54ef703e019be5d3e0b6bf5b90fe/tinyec-0.3.1.tar.gz
Installing collected packages: tinyec
  Running setup.py install for tinyec ... done
Successfully installed tinyec-0.3.1

C:\Users\ROHIT>
```

*Fig: 1 Install tinyec lib*

- The third library we need to install is chacha20poly1305. This library uses an authentication cipher with associated data (AEAD). This works with a secret key and a random nonce that never reused over the encryption. Figure 2 shows the command.



```
C:\Users\ROHIT>pip install chacha20poly1305
Collecting chacha20poly1305
  Using cached https://files.pythonhosted.org/packages/0e/0a/4a263a94aed4b32e723132ae2bbe8c6de4e3a260bd2ce4512f3fde8efcd/chacha20poly1305-0.0.2-py2.py3-none-any.whl
Installing collected packages: chacha20poly1305
Successfully installed chacha20poly1305-0.0.2

C:\Users\ROHIT>
```

*Fig: 2 Install chacha20poly1305 library*

#### ➤ Extraction

Extract the file “ecc code.zip” file. In this folder there is two files sender.py and receiver.py. As shown in the figure.

Name	Date modified	Type	Size
 receiver.py	12/11/2019 2:53 PM	PY File	3 KB
 sender.py	12/11/2019 6:17 PM	PY File	3 KB

*Fig: 3 Python message encryption and decryption files.*

---

<sup>1</sup> <https://pip.pypa.io/en/stable/installing/>

## ➤ Implementation

Now run the sender.py file for the encryption process. It will show random nonce and allows us to type any plaintext. After typing a plaintext, it will give an encrypted text. This encrypted text sender will send to the receiver for the decryption process along with the nonce value. As shown below fig 4.

```
Microsoft Windows [Version 10.0.18362.476]
(c) 2019 Microsoft Corporation. All rights reserved.

V:\steg image\Rohit\ecc code>python sender.py
nonce: d0dbe55da1198c9787c5922e
Enter text to encrypt: hello, how are you?
Encrypted Text: 838c4ca8c7abd95c8ffd6817d68f9d9a31aea167706ec42097170bdaa2df04ccd56920
```

*Fig: 4 Text encryption command*

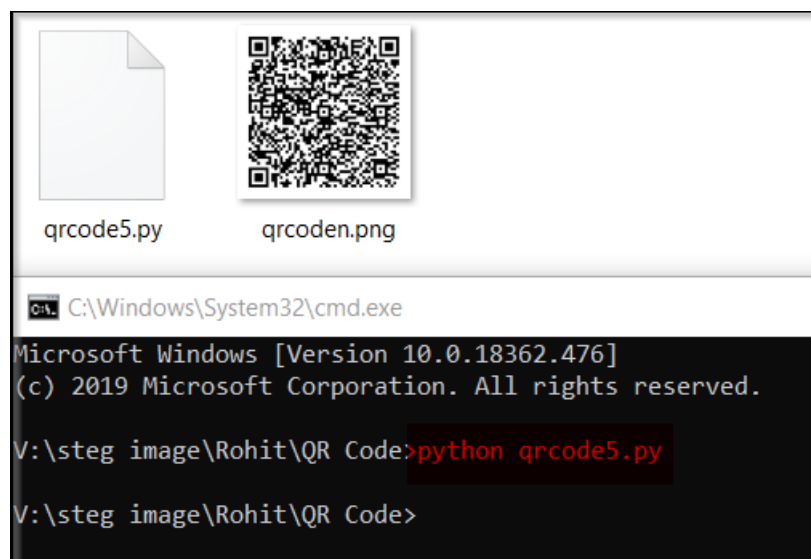
### 3.2 Hiding Ciphertext into QR code

In this method, we will hide ciphertext and random Nonce value, which we generated from the above process. This ciphertext and Nonce embedded into the QR Code using python code. We set a delimiter of “\$\$” after Nonce, and then we stored ciphertext. The system is shown in fig 5. In fig 6 run the QR Code python script, and it will create QR Code with the hidden texts

```
import qrcode
qr = qrcode.QRCode(
    version=5,
    error_correction=qrcode.constants.ERROR_CORRECT_H,
    box_size=10,
    border=4,
)
qr.add_data('d0dbe55da1198c9787c5922e $$ 838c4ca8c7abd95c8ffd6817d68f9d9a31aea167706ec42097170bdaa2df04ccd56920')
qr.make(fit=True)

img = qr.make_image(fill_color="black", back_color="white").save('qrcoden.png')
```

*Fig: 5 Nonce and ciphertext*



*Fig: 6 Create QR code*

Extract the QR Code from the python script zip file “QR code.zip”. In this folder there are two files as we can see in the figure 6. Qrcode5.py and qrcodeen.png.

### 3.3 Image embedding process

In this section we will discuss step by step installation and implementation of the proposed method.

#### ➤ Installation

First, we will install the MATLAB R2019b<sup>2</sup> in windows 10 to run the application in this software properly. MATLAB R2019b can be used to write the codes of the proposed method and run into it. This software is for 1-month trial version. MATLAB software will install properly without any errors.

#### ➤ Extraction

The next step is to extract the files from the zip folder (Stego code.zip). In the zip folder there are three files with “.m” extension. These files are used in the MATLAB. Four files are .bmp extension images for different pictures, three files of .png format used as an original cover image and one file is a QR Code which is our embedded image.

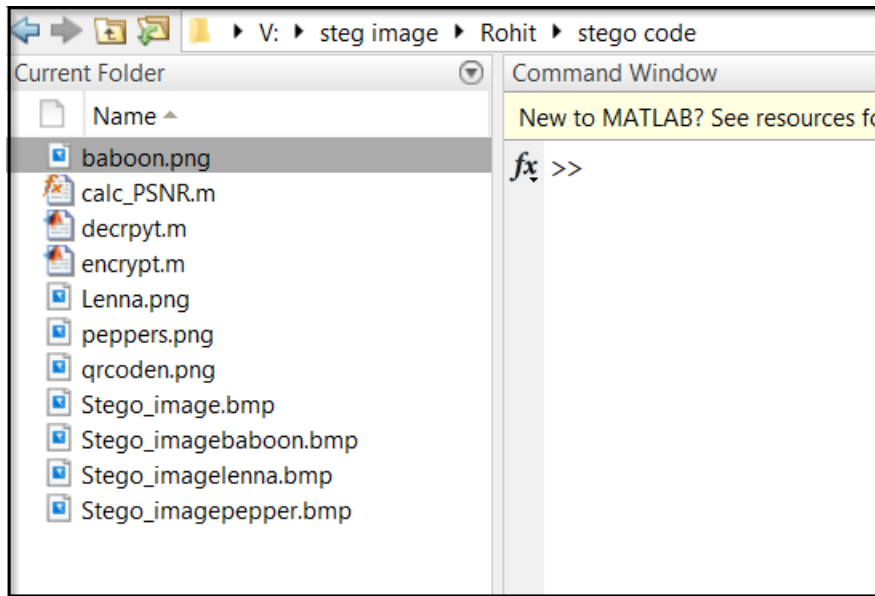
	Stego_image.bmp	Type: BMP File Dimensions: 512 x 512	Size: 768 KB
	Stego_imagebaboon.bmp	Type: BMP File Dimensions: 512 x 512	Size: 768 KB
	Stego_imagelenna.bmp	Type: BMP File Dimensions: 512 x 512	Size: 768 KB
	Stego_imagepepper.bmp	Type: BMP File Dimensions: 512 x 512	Size: 768 KB
	calc_PSNR.m Type: M File		Date modified: 11/24/2019 4:21 PM Size: 198 bytes
	decrypt.m Type: M File		Date modified: 12/12/2019 12:44 AM Size: 1.00 KB
	encrypt.m Type: M File		Date modified: 12/12/2019 1:25 AM Size: 2.30 KB
	baboon.png	Type: PNG File Dimensions: 512 x 512	Size: 622 KB
	Lenna.png	Type: PNG File Dimensions: 512 x 512	Size: 462 KB
	peppers.png	Type: PNG File Dimensions: 512 x 512	Size: 526 KB
	qrcoden.png	Type: PNG File Dimensions: 690 x 690	Size: 1.95 KB

Fig: 7 Stego code zip folder after extraction

<sup>2</sup><https://uk.mathworks.com/campaigns/products/trials.html>

➤ **Implementation**

Now run the MATLAB program as an administrator and click on the “Open folder” in the MATLAB upper left corner. Open all the extracted files from the stego image.zip folder. After doing this step it looks like this as you can see in the figure 8.



*Fig: 8 Files in MATLAB*

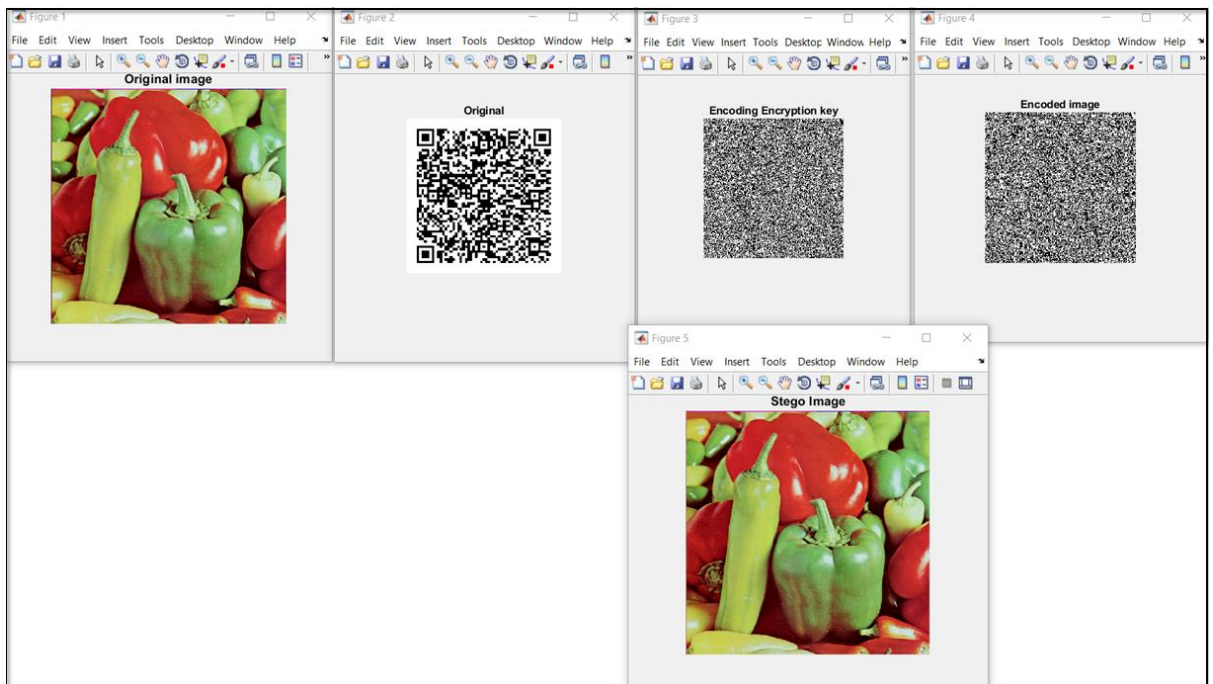
- Next step is to double click on the encrypt.m file and choose any cover image to run the test. As shown in the figure 9. If you want to use Lenna or baboon image remove % (comment) sign tag.

```
% Reading the images and converting the QR image to binary
tic;
QR = imresize(imread('qrcoden.png'), [195 195]);
QR = mat2gray(QR(:, :, 1)) > 0.5;
%Cover = imread('Lenna.png');
%Cover = imread('baboon.png');
Cover = imread('peppers.png');
```

*Fig: 9 Choose any Cover Image*

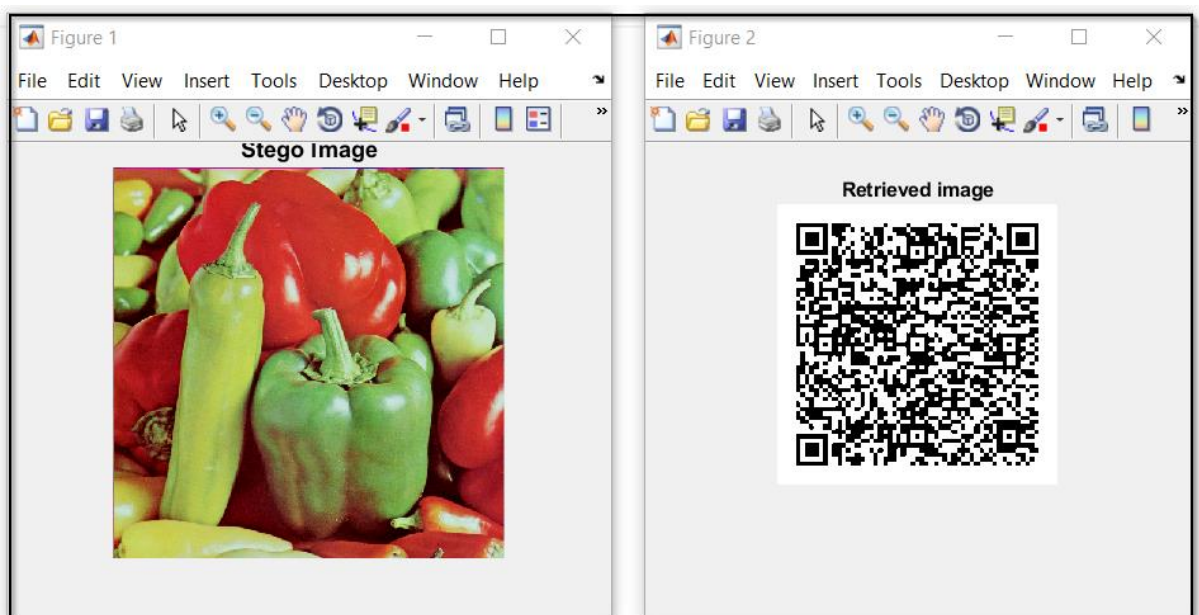
- Now run the encrypt.m file and you will get 5 images. Image 1 shows the original cover image, image 2 shows the Original QR Code, image 3 shows the encoding encrypted key, image 4 shows the encoded image, and image 5<sup>th</sup> shows the Stego Image in which QR Code is embedded into the original cover image. With the help of

LSB technique. Also, we used One time pad algorithm in image 3. As you can see in the fig 10.



*Fig: 10 Stego Image*

- Now we will recover our QR Code image from the Stego Image. For this process we have to run the decrypt.m file for the decryption process. In the fig 11.



*Fig: 11 Recovered QR Code Image*



### 3.4 Message decryption process

- To decrypt the message we have to read the QR Code with any QR Code reading application from the play store<sup>3</sup>. We read this QR Code with random scanning application as shown in the figure 12.

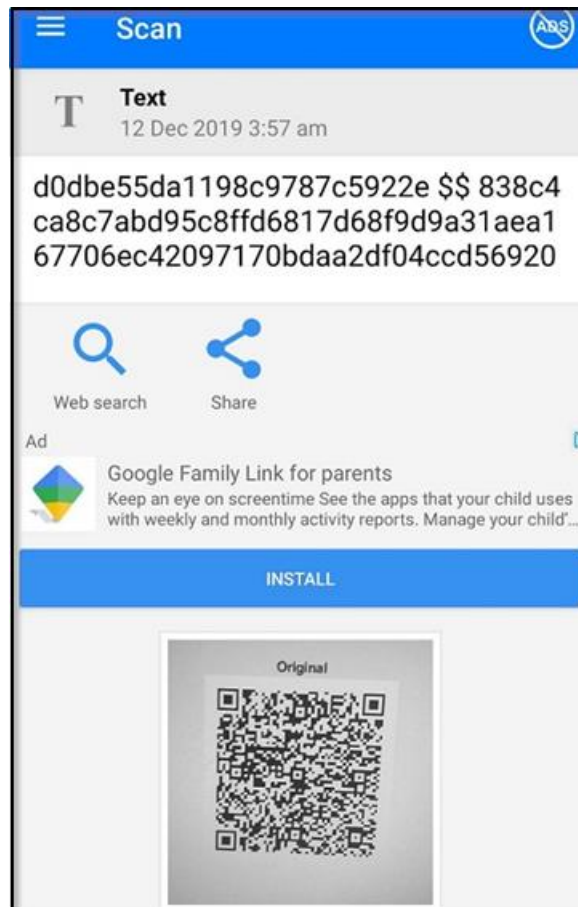


Fig: 13 Reading QR Code

- Now enter this Nonce value and ciphertext which we recovered from the Stego Image.
- We share this encrypted text to the sender through any online channel or we can create ftp server to share the information.
- After sharing the information type. The receiver will type Nonce and ciphertext to decrypt the ciphertext and read the plaintext.
- Run the receiver.py file as shown in the fig 14.

```
[rohit@parrot]~/Downloads/Rohit
└─$ python3 receiver.py
Enter nonce:d0dbe55da1198c9787c5922e
nonce: d0dbe55da1198c9787c5922e
Enter Encrypted text:838c4ca8c7abd95c8ffd6817d68f9d9a31aea167706ec42097170bdaa2df04ccd56920
Decrypted Text: hello, how are you?
[rohit@parrot]~/Downloads/Rohit
└─$
```

Fig: 14 Decrypting ciphertext

<sup>3</sup> <https://play.google.com/store/apps/details?id=com.gamma.scan>

## 4 References

- [1] M. Bafandehkar, R. Mahmood, S. M. Yasin and Z. M. Hanapi, "Comparison of ECC and RSA Algorithm in Resource Constrained Devices," in *2013 International Conference on IT Convergence and Security (ICITCS)*, Macao, China, 2013.