ISSN 2347 - 3983

International Journal of Emerging Trends in Engineering Research (IJETER), Vol. 3 No.1, Pages : 01 – 04 (2015)
Special Issue of ICAET 2015 - Held on February 23, 2015, Cochin, India
http://warse.org/pdfs/2015/icaet2015sp01.pdf

Optimizing the Capacity of QR Code To Store Encrypted Image

Dipesh Rawat[1] , Ravindra Sahu[2] , Yashila Puthran[3]
[1]Rajiv Gandhi Institute Of Technology, India, rawat.dipesh@gmail.com
[2]Rajiv Gandhi Institute Of Technology, India, sahu.ravindra277@gmail.com
[3]Rajiv Gandhi Institute Of Technology, India, y.puthran01@gmail.com

Abstract : Requirements for securing data storage using encryption has increased over the decade. This has resulted increased use of various modes for storing data like Barcode, RFID, QR Code. From all the mentioned storage techniques QR code is derived to be more efficient. QR code (Quick Response code) is a machine-readable code consisting of an array black white pixels used for storage of data that can be read by scanners. Typically QR code is store small amount of data like website URL, phone number, product information but it can be optimized store more amount of information. We intend to use QR code to store high resolution image by implementing the process of image compression technique along with encryption. The pixel data of the image is converted to compressed data stream which is encoded and store in a sequence of QR codes. This sequence of QR codes can be decoded by applying decoding and decompression technique to obtain the original image. This technique uses the capacity of QR code more efficiently.

Key words : QR Code, Image Compression, Encryption, Base64 encoding

INTRODUCTION

QR code (abbreviated from Quick Response Code is a type of 2-dimensional bar code developed by Denso Wave[1].A barcode is a machine-readable optical label that contains information about the item to which it is attached. A QR ("Quick response") code a machine-readable code consisting of an array of black and white used for data for reading by the camera on a reader. Its 2-dimensional barcode. Data is accessed by capturing a photograph of the code using a camera and processing the image with a QR reader.

A QR code uses four standardized encoding modes (numeric, alphanumeric, byte / binary, and kanji) to efficiently store data. QR code can store. The maximum capacity of a QR Code varies according to the content encoded and the error recovery level. The higher the error correction level, the less storage capacity. The maximum capacity is 2,953 bytes, 4,296 alphanumeric characters, and 7,089 numeric digits [2].

This paper proposes a method in which data capacity of QR code can be optimized and used to store images by first compressing the image data and then encrypting it. To get back the image back first the encoded data from the QR code is decompressed and then the decompressed data is decrypted. The use of compression and encryption together provides both quality and security.

SYSTEM OVERVIEW

A. *Introduction to system design*

Typically QR code is store small amount of data like website URL,phone number ,product information but it can be optimized store more amount of information. In the proposed system to store high resolution images we use the Raw Pixel format which is useful for applications that need direct access to the pixel data without the burden of the complex computations required to determine the location of pixels within a compressed data stream. This simplifies reading the image for applications that are performing pixel-oriented image processing, such as filtering and edge detection. This format is useful to need write data back to the image.

As shown in Figure 1 in the proposed technique first the raw pixel data of the image is extracted then pixel data of the image is converted to compressed data stream which is encoded and store in a sequence of QR codes. This sequence of QR codes can be decoded by applying decoding and decompression technique to get back the pixel data to write data back to the image.

Java coding of QR code needs the use of libraries that can be used to generate QR codes and that are available with an open source software known as Zebra Crossing or ZXing [3]. This is used for a variety of barcodes and QR code too. The ZXing libraries need to be loaded into the java project, this helps develop programs for QR code with customized requirement in Java.

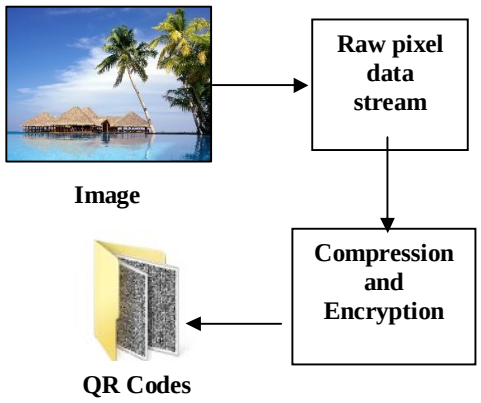

Fig 1: Steps for image to QR code conversion

B. Compression and Encryption

Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation [4]. The term Base64 originates from a specific MIME content transfer encoding. Base64 algorithm is designed to encode any binary data, and stream of bytes, into a stream of 64-printable characters.

The Base64 encoding process is to:

- Divide the input bytes stream into blocks of 3 bytes.
- Divide 24 bits of each 3-byte block into 4 groups of 6 bits.
- Map each group of 6 bits to 1 printable character, based on the 6-bit value using the Base64 character set map as shown in Table 1.
- If the last 3-byte block has only 1 byte of input data, pad 2 bytes of zero (\x0000). After encoding it as a normal block, override the last 2 characters with 2 equal signs (==), so the decoding process knows 2 bytes of zero were padded.
- If the last 3-byte block has only 2 bytes of input data, pad 1 byte of zero (\x00). After encoding it as a normal block, override the last 1 character with 1 equal signs (=), so the decoding process knows 1 byte of zero was padded.
- Carriage return (\r) and new line (\n) are inserted into the output character stream. They will be ignored by the decoding process [5].

Table 1: Character set map by Base64 encoding

VALUE	ENCODING
0-25	A-Z
26-51	a-z
52-61	0-9
62	+
63	/

C. Process flow Description

- Image to QR code

1. The image that needs to be stored in QR code is taken and its pixel data is extracted.
2. The extracted pixel data of the image is converted as data stream.
3. The data stream is compressed and encoded using Base64 technique.
4. Now the encrypted and compressed stream is divided into sub stream based on maximum capacity of QR code.
5. For each sub stream a QR code is generated and all these QR codes are stored together in a sequence.

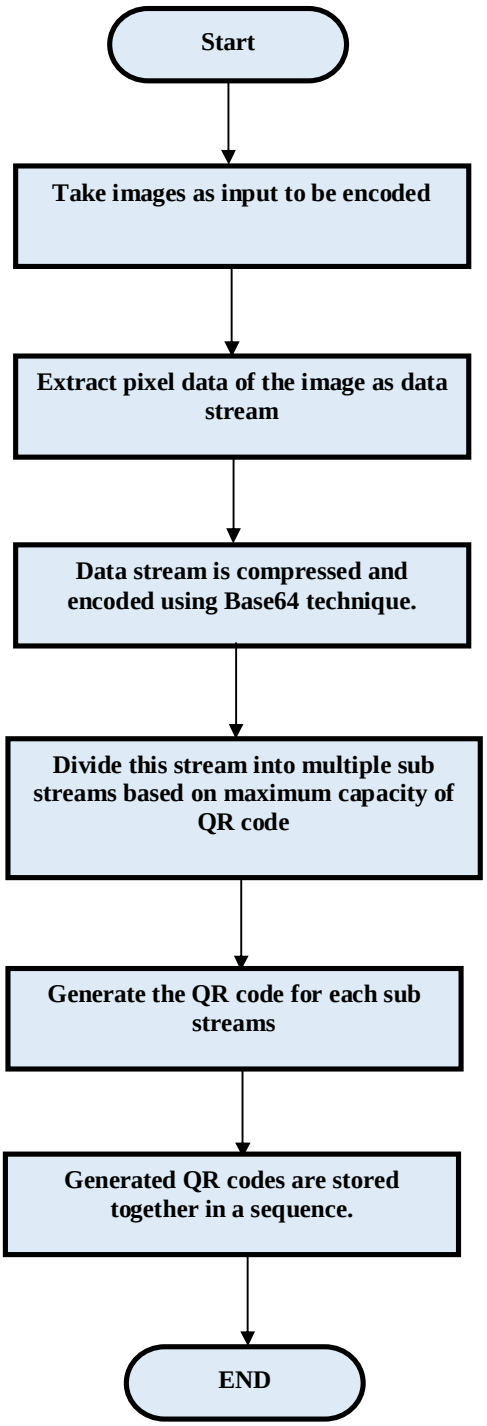

Fig 2: Conversion of Image to QR code

- QR Code to Image
  1. Take the sequence of QR code corresponding to a single image.
  2. Decode all these QR code to obtain a sequence of sub stream.
  3. Combine all the sub streams together to obtain the main data stream.
  4. Decompress and decode the data stream to get the pixel data.
  5. Construct the image from the pixel data.

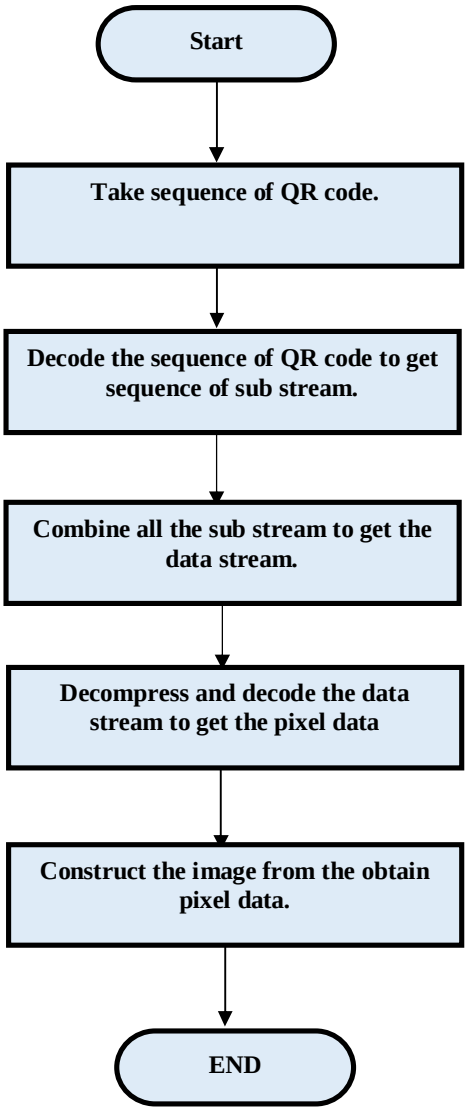

Fig 3: Conversion of QR Code back to image

EXPERIMENTAL RESULTS

The performance of the system was analyzed with several executions with the test images. The main objective of the tests was to evaluate the performance of the proposed technique.

A. Results

This section presents the performance analysis results while applying the proposed technique on various test images. Table 2 shows the number of QR codes generated for a particular resolution of an image.

Table 2: Experiment Result

RESOLUTION OF IMAGE	NO: OF QR CODE GENERATED
100x100	1
400x400	20
1024x768	82
1440x900	131
1600x1200	185

From the results portrayed, it can be seen that the proposed technique produces optimized acceptable results for all the test images.

B. Sample Case

Fig 4: 100x100 Original Image

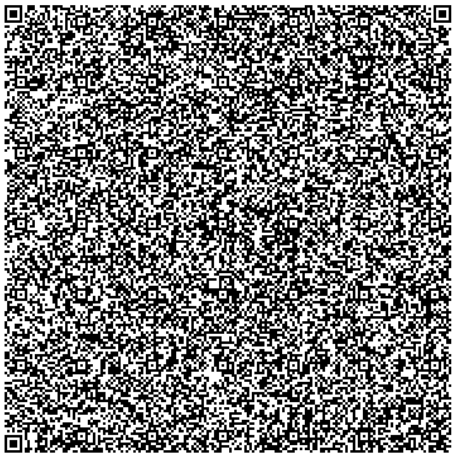

Fig 5: QR Code of above 100x100 image

APPLICATIONS

1) QR code can be used in banks for cheque truncation system to store and transfer images of cheque instead of manually transferring cheques.
2) It can be used for transfer of very important and sensitive images of documents etc.
3) QR codes of images can be used to print digital image on paper or anywhere which can be converted to original digital image.
4) The proposed technique could be further developed to store encrypted videos to prevent piracy.

ACKNOWLEDGMENT

We sincerely thank our guide Mr. Swapnil Gharat for his guidance and encouragement in successful completion of our paper work.

.

REFERENCES

[1] Denso wave incorporated,"http://www.densowave. com/qrcode/index-e.html"

[2] Peter Kieseberg, Manuel Leithner, Martin Mulazzani, , Schrittwieser, Mayank Sinha, Edgar Weippl " QR Code Security" , IEEE paper.

[3] ZXING- QR Code Library ,"http://code.google.com/ p/zxing"

[4] The Base16, Base32, and Base64 Data Encodings. IETF. October 2006. RFC 4648. Retrieved March 18, 2010.

[5] Congfu Xu; Yafang Chen; Chiew, K., "An Approach to Image Spam Filtering Based on Base64 Encoding and N-Gram Feature Extraction," Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on , vol.1, no., pp.171,177, 27-29 Oct. 2010.