# Data Validation System Using QR Code and Meaningless Reversible Degradation

[1]Lucas F. Freitas, [2]Adalberto R. Nogueira, and [3]Max E. Vizcarra Melgar
[1,2,3]Department of Computer Engineering, Centro Universitário IESB, Brasília, DF - Brazil
[1] First Decision, Brasília, DF - Brazil [2] Autotrac, Brasília, DF - Brazil
E-mails: [1]lucasdf10@gmail.com, [2]betorolim1@gmail.com, [3]maxvizcarra@ieee.org

***Abstract*—QR Codes are used as information channel on several cryptographic architectures due to their technical properties, such as data capacity and retrieval reliability. This paper presents a novel string data validation system using QR Codes and meaningless reversible degradation. The proposed scheme exploits reversible degradation properties, using the systematic Berlekamp Reed-Solomon error correction algorithm and the QR Code. This new mechanism encodes up to 388 characters in two information channels: a dynamic version QR Code (channel 1) and a wireless network (channel 2). A byte mode QR Code stores partial corrupted and masked data input bits. Its version size varies between 1 and 11 according the stored data quantity. The wireless channel downloads a previous generated Reed-Solomon redundancy file to correct the QR Code retrieved information and decode the secret message. The QR Code information is meaningless when scanned by a standard QR Code reader. Compared to real-time retransmission data validation systems, the proposed scheme reduces the download data (channel 2) up to 50%.**

***Keywords*-Data validation; QR Code; Reversible Degradation; Reed-Solomon.**

## I. Introduction

The Quick-Response Code (QR Code) is a monochromatic two-dimensional (2D) barcode. It was originally proposed in 1994 by the Japanese company Denso Wave Incorporated. The QR code [1] has been widely used to store and retrieve real time data on e-commerce, entertainment industry, social network, and information security [2], [3], [4], [5]. Most applications link the stored data to additional on-line server content after being scanned, thus, the 2D barcode can be used as a tool to obtain higher stored data [6], [7]. Figure 1 shows an example of a QR Code.

The QR Code standard has 40 versions. The first version has 21 $\times$ 21 modules. For each version the size of the modules increases by 4 $\times$ 4, reaching version 40, which is composed by 171 $\times$ 171 modules. QR Codes also count with four Reed-Solomon (RS) error correction levels: L - 7%, M - 15%, Q - 25%, and H - 30% [1].

In general, printed QR Codes store data in a back-end database, when scanned, the 2D barcode shows an understandable plaintext string. As the QR code standard is a public patent, the QR code message can be read directly by any standard QR code reader. Such QR Code feature results in privacy concerns when an encoder tries to deliver private messages to a third part using QR codes.

Fig. 1. Example of QR Code version 5, L error-correction level on alphanumeric mode.

TABLE I
QR Code storage capacity on ECC level L [1].

| Version | Data Capacity Mode | | |
|---|---|---|---|
| | Numeric | Alphanumeric | Byte |
| 1 | 41 | 25 | 17 |
| 3 | 127 | 77 | 53 |
| 5 | 255 | 154 | 106 |
| 7 | 370 | 224 | 154 |
| 9 | 552 | 335 | 230 |
| 11 | 772 | 468 | 321 |

Since a smartphone uses its camera and an image processing software to retrieve data in a QR Code, a QR Code print-scan channel does not consume online bandwidth, which turns this technology into an off-line data transfer system [5]. The decoding software can be found on several applications for iOS or Android operational systems, most of them only read numeric and alphanumeric QR Codes.

The QR Code coding mode has to be considered in order to take full advantage of its size and do not waste modules on non-used bits. For example, if an application needs to code 8-bits symbols, it should use the byte mode to generate the QR Code. In this case, alphanumeric mode is not suitable due to its storage capacity of 6 bits per character [1] and the application must use 2 alphanumeric characters for each byte, wasting 4 remaining bits. Table I shows numeric, alphanumeric, and byte mode store capacity for ECC level L and versions 1, 3, 5, 7, 9, and 11.

Reversible degradation is a method for digital data exchange of perceptive nature multimedia content files (such as audio, images, and video) [8]. The process consists on degrading the digital file in such a way that it becomes partially deteriorated, but it can still be distinguished by the receiving party. The sender becomes able to release it for validation without the risk of deliver an unpaid valuable file. By making the degradation process reversible, the receiver gets a degraded copy, validates it, and then negotiates a key

ISBN 978-80-261-0812-2, © University of West Bohemia, 2019

for recovering the original full-quality item. On the original definition, reversible degradation is applied on meaningful digital files, which are recognized by the user.

Reversible degradation implementation is performed using an Error Correction Code (ECC). An ECC deals with controlling errors in data transmission over unreliable/noisy channels. ECCs have been widely studied as the base for cryptographic algorithms [9], [10].

We introduce the concept of Meaningless Reversible Degradation, which consists on generate redundancy data from an ECC and purposely corrupt original data to make it meaningless to a third party. The corrupted information becomes meaningful when corrected by the EEC using previous generated redundancy data. Thus, the redundancy data is the key to recover the original file.

By introducing data validation into the critical path of data exchange, protocol designers can deploy authorized exchange-based information applications for generic bit streams. The ECC used in this work for meaningless reversible degradation is a systematic Berlekamp Reed-Solomon [11], suitable for digital content.

In this paper we propose a two-channel data validation scheme using meaningless reversible degradation. The first channel uses a printed QR Code, which contains masked and corrupted data. The other channel retrieves Reed-Solomon redundancy bytes through a wireless network. This paper is divided as follows. In Section II, we describe the data validation scheme using a QR Code and Reed-Solomon error correction algorithm. In Section III, we show the results of the proposed system. Finally, in Section IV we present our conclusions.

## II. Meaningless Reversible Degradation Data Validation Scheme

The proposed data validation scheme is designed to encode $c \leq 388$ characters, which might be used to describe properties of an item to be certified. We believe that 388 characters are enough information amount to describe any object property, such as personal identifications, vehicle titles, and commerce products.

The $c$ characters input are mapped on their correspondent bits using the 7-bit American Standard Code for Information Interchange (ASCII) table. The product description could have several information fields. Our propose use the character "&" to concatenate the filled fields and avoid non-instantaneous decode between them.

Our propose use the byte mode QR Code generation to display the lowest QR Code version and size. The characters ASCII table bits are concatenated generating $y$ bytes, which can use padding bits when $y$ is divisible by 8. This value is shown in Equation 1.

$$y = \begin{cases} \frac{7 \cdot c}{8}, & \text{if} \quad 7 \cdot c \bmod 8 = 0; \\ \frac{7 \cdot c + 8 - 7 \cdot c \bmod 8}{8}, & \text{if} \quad 7 \cdot c \bmod 8 \neq 0. \end{cases} \tag{1}$$

Meaningless reversible degradation is used on the first 2 bits of each $y$ bytes, which are concatenated to generate $k$ bytes and use padding when there are no enough bits to form the last byte. Figure 2 and Equation 2 show the selection process to generate $k$ bytes.

$$k = \begin{cases} \frac{7 \cdot y}{8}, & \text{if} \quad 7 \cdot y \bmod 8 = 0; \\ \frac{7 \cdot y + 8 - 7 \cdot y \bmod 8}{8}, & \text{if} \quad 7 \cdot y \bmod 8 \neq 0. \end{cases} \tag{2}$$

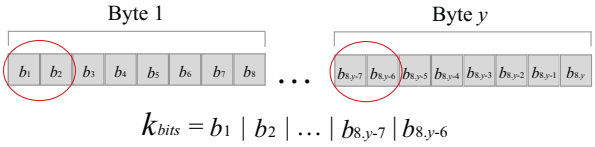


Fig. 2. Input $k$ bits selection.

An identification (ID) sequence is generated using the SHA-256 hash function over the $k$ bytes and generates a 64 hexadecimal string [12]. This ID works as storage address for further redundancy retrieval. A RS decoder can correct up to $t$ symbols that contain errors in a codeword. Equation 3 shows how $t$ is determined.

$$t = (n - k)/2. \tag{3}$$

The $k$ bytes are the Reed-Solomon data input to generate the redundancy bytes, which are equal to $2 \times k$ in order to correct the $k$ information symbols. Equation 4 shows the generated RS($n$,$k$) symbols.

$$RS(n, k) = [I_1 \cdots I_k \; RS_1 \cdots RS_{2 \times k}], \tag{4}$$

where $I_k$ is the $kth$ information symbol and $RS_{2k}$ is the $2 \times kth$ parity symbol.

The primitive polynomial used in the generation of redundancy bytes is shown in Equation 5.

$$p(D) = D^8 + D^4 + D^3 + D^2 + 1. \tag{5}$$

After encoding the $k$ bytes on the RS encoder, the $k$ bits are deleted to execute the meaningless reversible degradation. Four out of the 6 remaining bits of each $y$ byte are masked with the $8 \times (n - k)$ redundancy bits. The masking is computed using the One-Time Pad algorithm and matches the amount of bits due to $4y \leq 2k \times 8$ and $k \cong 2y/8$. The 2 less significant bits of each byte are plaintext. Figure 3 shows the masking process.

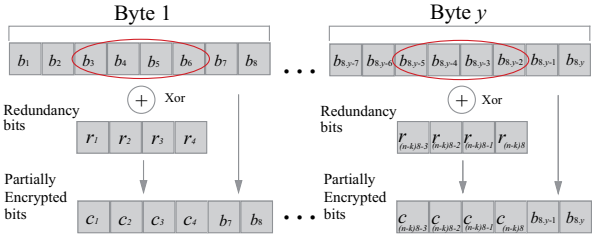


Fig. 3. One-Time Pad masking process.

The amount of encoded characters are represented in 2 bytes, which are concatenated with the 32 SHA-256 bytes and the resulting bytes from Figure 3. The entire

encoded message is stored in a QR Code on byte mode. The total $b$ bytes are stored in the QR Code, which is shown in Equation 6.

$$b = \begin{cases} \frac{6 \cdot y}{8} + 32 + 2 & , \text{if} \quad 6 \cdot y \bmod 8 = 0; \\ \frac{6 \cdot y + 8 - 6 \cdot y \bmod 8}{8} + 32 + 2, & \text{if} \quad 6 \cdot y \bmod 8 \neq 0. \end{cases} \tag{6}$$

The QR Code is printed and sticked on the desired product after the encoding process. Redundancy bytes are stored in a remote server. It is worth to note that the efficiency of retransmission is up to 50% of the original message. Equation 7 shows the efficiency of redundancy file download against an entire message retransmission.

$$e = 1 - \frac{2k}{y}, \tag{7}$$

where $k \cong y/4$, resulting on $e \cong 50\%$.

The decoding software reads the QR Code and identifies the item to be validated using its ID, and retrieves the corresponding redundancy bytes. The proposed algorithm use the $n - k$ redundancy bytes to decode the One-Time Pad ciphered bits and retrieves the $k$ bytes using the RS decoding process, which is able to retrieve the $k$ bytes from the $n - k = 2 \times k$ redundancy bytes. Finally, the original message is shown and data is validated.

## III. Results

The proposed architecture encodes any product description up to 388 characters. Only an authorized agent can read the encoded message through the download of the redundancy data. Any other regular QR Code reader is no able to decode the QR Code message due to the QR Code Byte encoding mode, the reversible degradation erasing mode, and the masking process present in the proposed solution.

We created an application called Easy Tracking to test the proposed system. We tested a set of 50 QR Codes on versions between 1 and 11, and ECC levels "L" and "H". The tested QR Codes were shown on printed on A4 paper and screen mirror. Due to the reading robustness of the QR Code decoder, all tested QR Codes were successfully read and their information was retrieved.

As a result example, a vehicle information was filled in 23 different fields. The input message has 272 characters and is represented by the following string: "12315616516&ABC1234&DF&131321313131313& Car&Flex&Renault&Fluence&2017&2017&4&89& Particular&Dourado&123.456.678-90&Lucas Feitosa de Freitas&teste123@gmail.com&(99) 99999-9999&12345678900&JOSE FRANCISCO CARLOS&MARIA DE LUZ DA SILVA&23/06/1995&Setor Hoteleiro Norte Quadra 8&".

The 272 encoded characters generates $y = 238$, $k = 60$, $n - k = 120$, and $b = 213$ bytes. The generated QR Code is on version 9 ($53 \times 53$ modules) and L error-correction level. This version size stores up to 230 bytes (See Table I). In this case, the QR Code stores 213 bytes. Figure 4 shows the QR Code capture image before the final message illustration and Figure 5 shows the retrieved message before its final illustration.

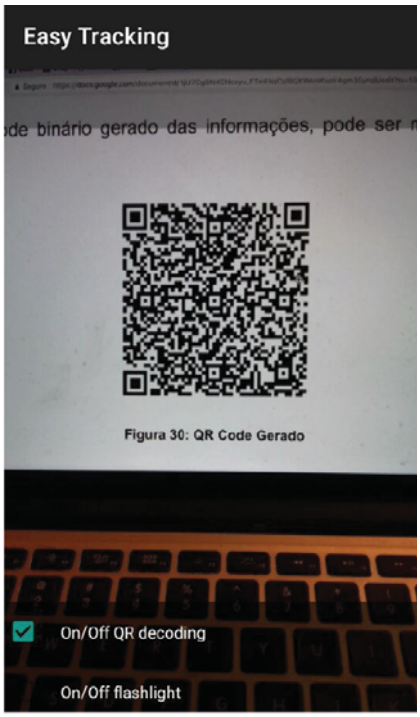


Fig. 4. QR Code retrieving information process.

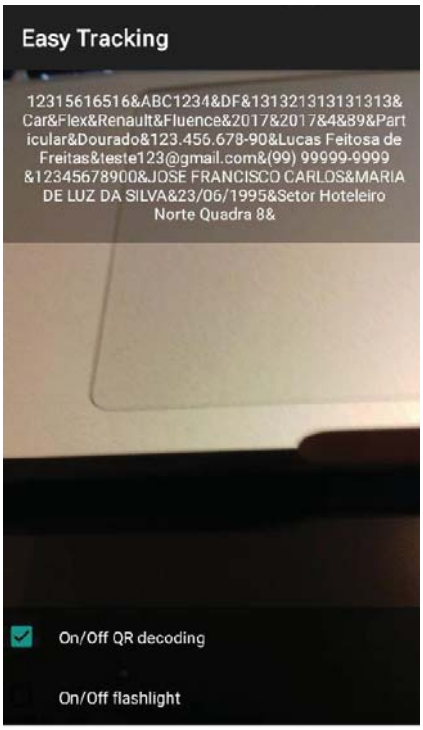


Fig. 5. QR Code retrieving information process.

Figure 6 shows a vehicle information for verification after decoding the $k$ RS information bytes and retrieve the final secret message. The prototype follows the reverse encoding process. In other words, it computes the following steps:

TABLE II
EFFICIENCY RESULTS.

| $c$ | $y$ | $k$ | $n$ | $b$ | $e$ |
|---|---|---|---|---|---|
| 78 | 69 | 18 | 54 | 86 | 47.83% |
| 111 | 98 | 25 | 75 | 108 | 48.98% |
| 155 | 136 | 34 | 102 | 135 | 50.00% |
| 199 | 175 | 44 | 132 | 166 | 49.71% |
| 232 | 203 | 51 | 153 | 187 | 49.75% |
| 276 | 242 | 61 | 183 | 216 | 49.59% |
| 320 | 280 | 70 | 210 | 244 | 50.00% |
| 388 | 340 | 85 | 255 | 289 | 50.00% |

1) Capture the QR Code using the software prototype.
2) Get the Hash-ID and the stored byte length.
3) Download from the server the redundancy associated with the Hash-ID.
4) Use the redundancy bits to decrypt the One-Time Pad message according Figure 3.
5) Use the redundancy bits to generate the $k$ bytes using the RS decoding process.
6) Show the secret message $c$.

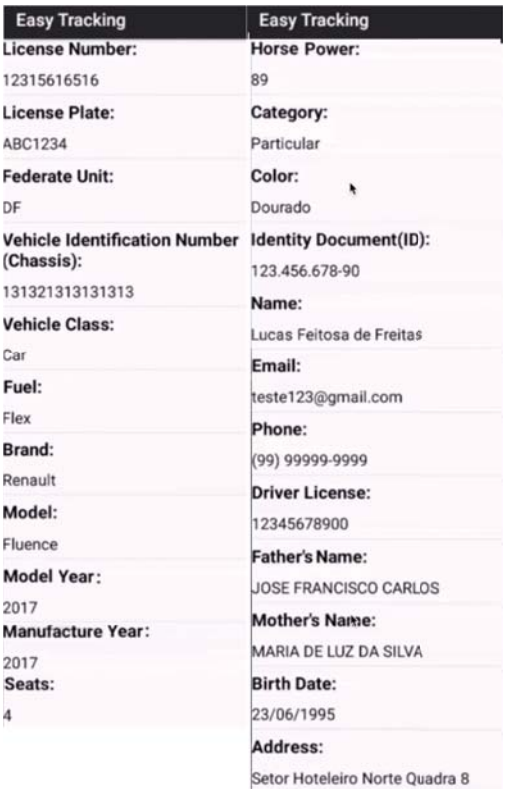


Fig. 6. Retrieved secret message.

The transmission of the redundancy file is up to 50% the size of an entire message retransmission. The efficiency shown in Equation 7 was computed for different $c$ characters. Table II shows some cases.

## IV. CONCLUSION

In this paper, we proposed a novel meaningless reversible degradation method for product description validation. This method explores the Reed-Solomon reversible degradation capacity and the QR Code byte mode structure.

The message uses the Reed-Solomon redundancy bytes to partially encrypt the message with the One-Time Pad algorithm. The encrypted data is concatenated with a Hash-ID and a 2 byte header with the amount of bytes on the QR Code on byte mode. Finally, the encoded message is printed on the QR Code. A smartphone application was developed as prototype. This prototype reads the QR Code, get the Hash-ID and the stored byte length. Then, it downloads from the server a redundancy file, which is associated with the Hash-ID, and use the redundancy bits to decrypt the One-Time Pad message. Then, the software use the redundancy bits to generate a part of the encoded message using the RS decoding process. Finally, it shows the secret message.

The advantage of this propose is that the transmission of RS redundancy bytes requires up to 50% bandwidth compared to an entire message retransmission. In addition, it preserves on secret the stored information on the QR Code. In other words, the QR Code is meaningless to any standard QR Code reader. Only an authorized part, which has the software prototype and is able to download the redundancy file, can read the message and validate the product information.

## ACKNOWLEDGMENT

This work was supported by the Centro Universitário IESB.

## REFERENCES

[1] "ISO/IEC 18004:2005 - Information technology - Automatic identification and data capture techniques - QR Code 2005 bar code symbology specification," August 2005.
[2] Max E. Vizcarra Melgar and Luz M. Santander, "An alternative proposal of tracking products using digital signatures and QR codes," in *Proceedings of the 2014 IEEE Colombian Conference on Communications and Computing*, June 2014.
[3] H. Bagherinia and R. Manduchi, "A Theory of Color Barcodes," in *Proceedings of the IEEE Color and Photometry in Computer Vision Workshop*, 2011.
[4] M. S. B. Akila, B. Hema, "Secured Data Encoding Technique in High Capacity Color Barcodes for M-Ticket Application," in *International Journal of Electronics and Computer Science Engineering*, 2008.
[5] Max E. Vizcarra Melgar and Mylène C. Q. Farias, "High Density Two-Dimensional Color Code," in *Multimedia Tools and Applications*, vol. 78, July 2018, pp. 1949–1970.
[6] Max E. Vizcarra Melgar and Luz M. Santander, "Channel Capacity Analysis of 2D Barcodes: QR Code and CQR Code-5," in *Proceedings of the 2016 IEEE Colombian Conference on Communications and Computing*, April 2016.
[7] Max E. Vizcarra Melgar, Mylène C. Q. Farias. Flávio de Barros Vidal, and Alexandre Zaghetto, "A High Density Colored 2D-Barcode: CQR Code-9," in *29th SIBGRAPI Conference on Graphics, Patterns and Images*, October 2016, pp. 329–334.
[8] Fabio Piva and Ricardo Dahab, "E-Commerce and Fair Exchange The Problem of Item Validation," in *Proceedings of the International Conference on Security and Cryptography*, July 2011, pp. 317–324.
[9] Yasuo Sugiyama, Masao Kasahara, Shigeichi Hirasawa, and Toshihiko Namekawa, "An erasures-and-errors decoding algorithm for Goppa codes," in *IEEE Transactions on Information Theory*, March 1976, pp. 238–241.
[10] R.J. McEliece, "A Public-Key Cryptosystem Based On Algebraic Coding Theory," in *Deep Space Network Progress Report*, vol. 44, 1978, pp. 114–116.
[11] Berlekamp E., "Nonbinary BCH decoding," in *Proceedings of the International Symposium on Information Theory (ISIT)*, vol. 14 n.2, 1967.
[12] Pei-Yu Lin, "Distributed Secret Sharing Approach with Cheater Prevention based on QR Code," in *Proceedings of the IEEE Transactions on Industrial Informatics*, 2016.