Data Validation Scheme Using Meaningless Reversible Degradation and NFC

Kevin S. Araujo and Max E. Vizcarra Melgar

***Abstract*— This paper presents a novel data validation system using meaningless reversible degradation and the Near-Field Communication (NFC) protocol. The NFC protocol is widely used as data exchange channel due to its technical properties, such as data transfer rate and transmission reliability. The proposed scheme exploits reversible degradation properties, using the systematic Berlekamp Reed-Solomon error correction algorithm and the NFC on three error correction levels: Low ($\cong 5\%$), Medium ($\cong 15\%$), and High ($\cong 25\%$). This new mechanism allows encoding data on a meaningless file, which can be posteriorly retrieved using an authorized NFC card. Thus, to retrieve and validate the file, the part that receives the corrupted file, and the authorizing part, must be physically near to allow decoding the file. Once the authorizing part removes the NFC card, the decoded file becomes meaningless again and is closed.**

***Keywords*— Near-Field Communication, Reversible Degradation, Reed-Solomon, Data Validation.**

I. Introduction

Near-Field Communication (NFC) is a standard-based radio frequency (RF) channel communication architecture that can be embedded into several devices, such as mobiles cellphones, laptops, tablets, and NFC card readers [1]. Its main function allows to exchange data between devices in a proximity up to 10cm [2] using the 13.56 MHz frequency [3], [4], [5] with maximum data transfer rate of 424 kbit/s and low bandwidth [4], [6]. NFC technology was proposed over a decade, but it became popular when it was incorporated into payment systems, Internet of Things (IoT), information security, and Industry 4.0 [1], [3], [7], [8], [9].

One of the working modes of the NFC technology is using proximity cards [2]. These card are used as anti-collision identification tokens on an NFC reader Proximity Coupling Device (PCD). The PCD is a reader hardware that emits an electromagnetic field. This field powers a tag/transmitter by inductivity. The cards can be used as authentication and item validation on information security architectures. Figure 1 shows an example of an NFC reader and cards.

Reversible degradation is a method for perceptive nature multimedia content files data exchange, such as audio, images, and video files [10]. The process consists on degrading a digital file in such a way that it becomes partially deteriorated, but it can still be distinguished by a third party. The sender becomes able to release the file for validation without the risk of deliver an unpaid valuable file. By making the degradation process reversible, the receiver gets a degraded version, validates it, and negotiates a key 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.

Kevin S. Araujo, e-mail: kevin_santana.araujo@hotmail.com; Max E. Vizcarra Melgar, e-mail: maxvizcarra@ieee.org, SGAS 613/614 Campus Edson Machado - IESB, Brasília - DF. This work was supported by the Centro Universitário IESB.

Fig. 1. NFC reader and cards.

Reversible degradation implementation is performed using an Error Correction Code (ECC). An ECC corrects errors in data transmission over unreliable or noisy channels. ECCs have been widely studied as base for cryptographic algorithms [11], [12].

We introduce the concept of Meaningless Reversible Degradation, which consists on generate redundancy data from an ECC and purposely corrupt the original data to turns it meaningless to the receiver part. The corrupted information becomes meaningful when corrected by the EEC using previous generated redundancy data. Thus, the redundancy data works as a recover key for 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 files. The ECC used in this work for meaningless reversible degradation is a systematic Berlekamp Reed-Solomon [13]. This ECC algorithm is very popular, being frequently used in 2D-Barcodes, satellite communications, and optical recording systems, like CD, DVD, and Bluy-Ray [14], [15], [16].

Nowadays, NFC security standards define data exchange

format and security protocols. It is expressly stipulated in the NFC security standard that a key agreement is required for communications between users [2]. NFC security parameters were deeply researched. Haselsteiner and Breitfu [17] explore different threats, for man-in-the-middle attack, data corruption, and data modification on NFC interactions. Mulliner [18] analyzes NFC vulnerabilities. These vulnerabilities can easily be exploited for spoofing of the tag content.

In this paper we propose a two-channel data validation scheme using meaningless reversible degradation. The first channel uses a generic internet connection, which transmits a meaningless degraded file. The other channel validates an identification number to download the Reed-Solomon redundancy bytes. This identification number is stored into an NFC card.

This paper is divided as follows. In Section II, we describe the data validation scheme using NFC cards, an NFC reader, 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

Information security includes safety, two-way authentication of the sender and receiver. Confidentiality and integrity are the most important information security parameters for data exchange. Particularly, the authentication goal is to ensure the correct identity of entities operating on an NFC transaction.

The proposed data validation scheme is designed to encode any file to a meaningless file, which might be considered as a corrupted file due to degradation. The encoding process consists on purposely degrade some bytes from the original file. These corrupted bytes might be corrected, if and only if, an authorized NFC card is detected by the system.

The corrected file is degraded again if the authorized NFC card is removed from the NFC reader. Thus, the secret file is displayed only when an authorized part is present to allow its visualization.

The authorized NFC card must be physically present on the NFC reader to show the secret file. This feature makes the proposed system a hardware based data validation method.

A. Coding Process

To avoid NFC cards data collision, a SHA-256 hash public file is created from the NFC card Unique Identifier (UID) number [2]. This ID works as storage address for further redundancy retrieval. The Berlekamp Reed-Solomon (RS) error correction algorithm computes the meaningless reversible degradation coding process. A redundancy file is generated from the original file. The coding algorithm is performed every k bytes from the original file. A RS decoder can correct up to t symbols that contains errors in a codeword. Equation 1 shows how t is determined.

$$t = (n - k)/2, \tag{1}$$

where n is the maximum symbol quantity of the RS codeword, which is in a 2^8 Galois Field. We use a 8-bits length on each symbol for a suitable byte manipulation on any file. Thus, $n = 2^8 - 1 = 255$ in the proposed work.

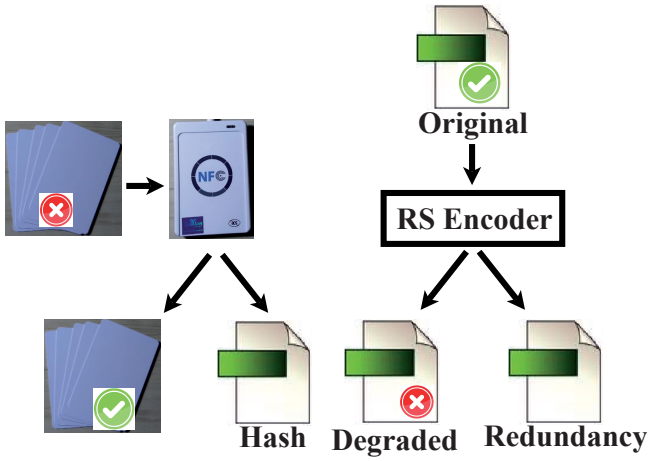


Fig. 2. Meaningless reversible degradation coding process.

The k bytes are the Reed-Solomon data input to generate the redundancy bytes [13]. Equation 2 shows the generated RS(n,k) symbols.

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

where I_k is the kth information symbol and RS_{n-k} is the last redundancy symbol.

The primitive polynomial used in the generation of redundancy bytes is shown in Equation 3.

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

After encoding the k bytes on the RS encoder, the redundancy file is generated and composed by the $n - k$ redundancy bytes. To complete the coding process, from every n bytes of the original file, $n - k$ pseudo-randomly bytes are changed to other pseudo-random byte value, which performs the corruption in the original file. This process is repeated every k bytes on the original file. The remainder bytes of the file are not considered. The encoding algorithm is shown below. Here, the NFC UID number is represented as UID, the original file as $File$, the Redundancy file as Red, the Reed-Solomon encoding algorithm as RS_{Enc}, and the Degraded file as Deg.

FILE-ENCODING$(UID, File)$
1 $Stored_{Hash} = SHA_{256}(UID)$
2 $Red = \emptyset$
3 $Deg = \emptyset$
4 **for** $i = 1 : i = i + k$ **to** $File.length$
5 $\quad Red = Red \;||\; RS_{Enc}(File[i : i + k])$
6 **for** $i = 1 : i = i + k$ **to** $File.length$
7 $\quad Deg = Deg \;||\; Corrupt(File[i : k])$
8 **return** $(Stored_{Hash}, Red, Deg)$

After performing the coding process, the hash, degraded, and redundancy files are public elements. On the other hand, the registered NFC card becomes the private key to allow the original file retrieval. Figure 2 shows the coding process.

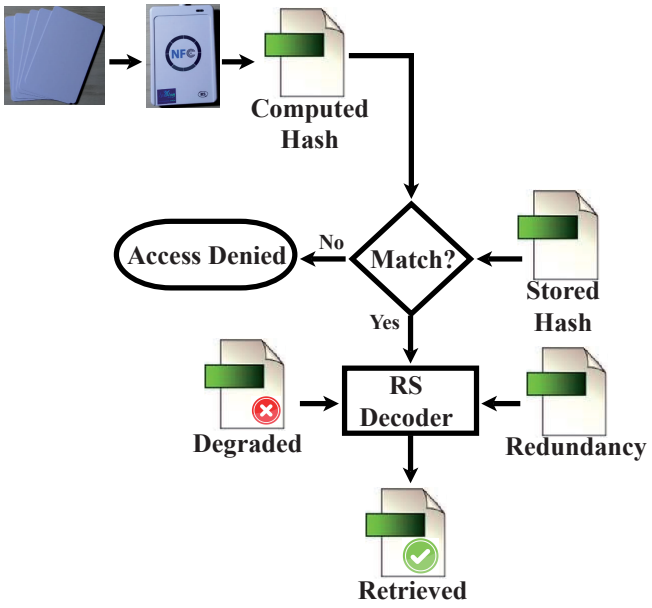


Fig. 3. Meaningless reversible degradation decoding process.

B. Decoding Process

Once the coding process is completed, the retrieving process is started when a receiver part gets the public files and the recover software. The authorized part holds a valid NFC card, which is essential to recover the original file. An NFC card is superimposed on the NFC reader and the SHA-256 hash algorithm is computed over the NFC card UID. An authorized part allows to show a secret file if the hash data matches any stored hash file. Then, the retrieving process begins.

The retrieving process consists on the following steps:

1) Get the degraded and redundancy files as input.
2) For each k bytes on the degraded file, get $n-k$ bytes from the redundancy file.
3) For each iteration on the previous step, compute the Berlekamp-Massey RS(n,k) error correction and correct the k bytes.
4) For each iteration on the previous step, store and concatenate the k bytes on the Retrieved file.
5) Repeat steps 2), 3), and 4) until the end of the file.
6) Show the secret file. Figure 3 illustrates the decoding process.

Once the decoding process finishes, a retrieved file is generated. The decoding algorithm is shown below. Here, the Retrieved file is represented as Ret, the Reed-Solomon decoding algorithm as RS_{Dec}, and a counter as $Cont$.

FILE-DECODING$(UID, Stored_{Hash}, Deg, Red)$
1 $Hash_{File} = SHA_{256}(UID)$
2 $Ret = \emptyset$
3 $Cont = 1$
4 **if** $Hash_{File}$ == $Stored_{Hash}$
5 **for** $i = 0 : i = i + k$ **to** $Deg.length$
6 $Ret = Ret \mid\mid RS_{Dec}(Deg[i : i + k], Red[Cont : Cont + n - k])$
7 $Cont = Cont + n - k$
8 **return** (Ret)

The decoding process depends on the presence of the authorized NFC card. This factor turns the proposed system on a data validation method that requires a hardware component (NFC card) to allow data visualization.

The retrieved file is identical to the original file. The retrieved file is available to the third part, if and only if, the authorized NFC card is superimposed on the NFC reader. If the authorized part removes the NFC card from the NFC reader, the retrieved file is encoded and becomes a meaningless file again.

TABLE I
PROPOSED SYSTEM RS PARAMETERS.

ECC Level	Berlekamp RS Parameters		
	k	$n-k$	t
Low ($\cong 5\%$)	229	26	13
Medium ($\cong 15\%$)	177	78	39
High ($\cong 25\%$)	125	130	65

(a) (b)

Fig. 4. Example of (a) NFC reader with an NFC card and (b) NFC reader without an NFC card.

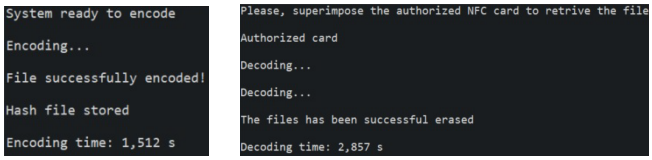
III. RESULTS

The proposed data validation system was implemented on Java language program and works on 3 different meaningless reversible degradation levels: low, medium, and high. The low, medium, and high levels degrades the original file on 5%, 15%, and 25%, respectively. Table I shows the RS parameters: information bytes, redundancy bytes, and error correction capacity bytes for each level. For each ECC level, all t possible bytes are ramdonly corrupted as shown in Section II.

We tested the prototype on several popular file formats, such as .pdf, .docx, .pptx, .xlxs, .exe, .jpg, .mp3, .mp4, .txt, etc. All degraded files were identified as corrupted and could not be opened by the Windows operational system due to the corrupted bytes.

Figure 4 shows the NFC reader with an (a) authorized NFC card and (b) without and NFC card. The NFC card must be superimposed on the NFC reader for the encoding and retrieving process. On the other hand, when the NFC reader detects that the NFC card was removed, the software encodes the secret file and turns it meaningless again.

We also measured the encoding and decoding time for 3 different file sizes. Table II shows this result. The short elapsed time for encoding and decoding shows that this prototype can be implemented on commercial applications for confidential data validation and share, such as Automatic Teller Machine (ATM) banking, intellectual property data temporal sharing, etc . Figures 5 (a) and (b) show the software encoding and decoding steps.



(a) (b)

Fig. 5. Software (a) encoding and (b) decoding report.

TABLE II
ENCODING AND DECODING ELAPSED TIME.

File Size	ECC Level	Coding Time	Decoding Time
3 MB	Low	2.75s	1.19s
	Medium	3.09s	4.09s
	High	3.86s	9.29s
4 MB	Low	3.40s	1.75s
	Medium	3.88s	5.44s
	High	4.60s	12.03s
5 MB	Low	4.31s	1.88s
	Medium	4.55s	6.40s
	High	5.50s	14.72s

The encoding process is performed one time for each file. The decoding process runs while the NFC card is superimposed on the NFC reader, allowing to retrieve and show the secret file. When the NFC card is removed from the NFC reader on the decoding process, the system encodes the file and erase the secret file.

IV. CONCLUSION

In this paper, we proposed a novel meaningless reversible degradation method for file validation. This method explores the Reed-Solomon reversible degradation capacity and the NFC architecture. A SHA-256 hash function is computed from a secret file and stored to be the data validation key. Then, the RS redundancy data is generated and stored and the secret file is partially corrupted until the RS error correction capacity, on three different levels: Low ($\cong 5\%$), Medium ($\cong 15\%$), and High ($\cong 25\%$). The output of the coding process is the valid NFC card, the hash file, the degraded file, and the redundancy file.

A valid NFC card is superimposed on the NFC reader to retrieve the file, this mechanism allows to begin the decoding process for data retrieving. Once a valid NFC card is read, the degraded file and the redundancy file are taken as input for the RS decoder. A retrieved file is generated, this file is exactly the same than the original file. The retrieved file is available, if and only if, the NFC card is superimposed on the NFC reader. If the NFC card is removed, the secret file is encoded again and the file becomes meaningless.

The advantage of this propose is that the secret file is decoded, if and only if, an authorized NFC card is superimposed on the NFC reader. Thus, an authorized part must be physically close to the NFC reader to superimpose the card on the NFC reader. The proposed system was implemented on the Java language program and computes the encoding and decoding process in few seconds, which turns this method on a feasible commercial system.

ACKNOWLEDGMENT

This work was supported by the Centro Universitário IESB.

REFERENCES

[1] Ali Alshehri, Johann A. Briffa, Steve Schneider, and Stephan Wesemeyer, *Formal Security Analysis of NFC M-coupon Protocols using Casper/FDR*. 5th International Workshop on Near Field Communication (NFC), February, 2013.

[2] ISO/IEC 14443-3, *Identification cards - Contactless integrated circuit(s) cards - Proximity cards - Part 3*, February, 2001.

[3] Marti Boada, Antonio Lazaro, Ramon Villarino, and David Girbau, *Battery-Less NFC Sensor for pH Monitoring*, IEEE Access, vol 6, March, 2019.

[4] Siti Ummi Masruroh, Andrew Fiade, Imelda Ristanti Julia, *NFC Based Mobile Attendence System with Facial Authorization on Raspberry Pi and Cloud Server*, The 6th International Conference on Cyber and IT Service Management (CITSM 2018), August, 2018.

[5] Fan Dang, Ennan Zhai, Zhenhua Li, and Kaigui Bian, *Pricing Data Tampering in Automated Fare Collection with NFC-equipped Smartphones*, IEEE Transactions on Mobile Computing, vol 18, pg 1159-1173, May, 2019.

[6] Josep. I. Cairó, Jordi Bonache, Ferran Paredes, and Ferran Martín, *Reconfigurable System for Wireless Power Transfer (WPT) and Near Field Communications (NFC)*, IEEE Journal of Radio Frequency Identification, vol 1, pg 253-259, December, 2017.

[7] K. Finkenzeller and D. Müller, *RFID Handbook Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, Hoboken, NJ, USA: Wiley, 2010.

[8] D. Paret, *Design Constraints for NFC Devices*, Hoboken, NJ, USA Wiley, 2016.

[9] Ali Al-Haj and Mayyadah Adnan Al-Tameemi, *Providing Security for NFC-Based Payment Systems Using a Management Authentication Server*, 4th IEEE International Conference on Information Management, May, 2018.

[10] Fabio Piva and Ricardo Dahab, *E-Commerce and Fair Exchange The Problem of Item Validation*, Proceedings of the International Conference on Security and Cryptography, pg. 317-324, July, 2011.

[11] Yasuo Sugiyama, Masao Kasahara, Shigeichi Hirasawa, and Toshihiko Namekawa, *An erasures-and-errors decoding algorithm for Goppa codes*, IEEE Transactions on Information Theory, pg 238-241, March, 1976.

[12] R. J. McEliece, *A Public-Key Cryptosystem Based On Algebraic Coding Theory*, Deep Space Network Progress Report, vol. 44, pg 114-116, 1978.

[13] Berlekamp E., *Nonbinary BCH decoding*, Proceedings of the International Symposium on Information Theory (ISIT), vol. 14, 1967.

[14] Max E. Vizcarra Melgar and Mylène C. Q. Farias, *High density two-dimensional color code*, Multimedia Tools and Applications, vol. 78, pg 1949-1970, January, 2019.

[15] Max E. Vizcarra Melgar, Mylène C. Q. Farias. Flávio de Barros Vidal, and Alexandre Zaghetto, *A High Density Colored 2D-Barcode: CQR Code-9*, 29th SIBGRAPI Conference on Graphics, Patterns and Images, pg 329-334, October, 2016.

[16] J. Lee and K. A. S. Immink, *An efficient decoding strategy of 2D-ECC for optical recording systems*, IEEE Transactions on Consumer Electronics, vol. 55, pg 1360-1363, 2009.

[17] Haselsteiner E and K. Breitfu, *Security in near field communication (NFC)*, In Proceedings of the International workshop on RFIDSecurity, RFIDSec, June, 2006.

[18] Collin Mulliner, *Vulnerability analysis and attacks on NFC-enabled mobile phones*, In ARES, pg 695-700, 2009.