ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 10, April 2020

An Efficient Implementation of the Niederreiter Cryptosystem based on Error Correcting Codes

Younes Bayane[1], Fatima Amounas[1] and Lahcen El Bermi[2]
[1] R.O.I Group, Computer Sciences Department, Moulay Ismaïl University, Faculty of Sciences and Technics, Errachidia, Morocco
[2] GL-ISI, Computer Sciences Department, Moulay Ismaïl University, Faculty of Sciences and Technics, Errachidia, Morocco

***Abstract*— With the development of information and communication technologies, securing data content is becoming more important. Different cryptographic systems have been put in place to fill this need. Up until now, number-based cryptography remains by far the most used. However the assumed advent of quantum computers makes it potentially threatened by quantum attacks. To get around this, more secure alternatives have emerged. Code-based cryptography is one of the most powerful candidates in this sense. In this paper we present a practical implementation of one of the most promising schemes in the field of cryptography based on error correcting codes, namely, the Niederreiter cryptosystem. In particular, we put forward its security proof; we describe the new construction as well than the results obtained during the tests.**

***Index Terms*—Public key cryptography, error-correcting codes, McEliece cryptosystem, Niederreiter cryptosystem, Syndrome decoding problem.**

I. INTRODUCTION

To break contemporary cryptosystems using any binary computer, one has to face a problem of at least sub-exponential complexity. However if quantum algorithms was effectively executed, the cryptographic primitives based on the number theory would not resist anymore [1]. Code-based cryptography on the other hand is not as threatened as number theoretic based cryptography as it relies on different problems. Thus, this kind of cryptography has attracted more attention last years, especially after showing that certain instances of problems based on the theory of error correcting codes remain intractable even employing quantum computer. So, some cryptography constructions based on error correcting codes belong to the post quantum cryptography. The first code-based construction was the cryptosystem devised by McEliece in the late seventies [2]. It makes use of binary irreducible Goppa codes and ensures a fast encryption and decryption in comparison, for example, with RSA. However it suffers from a major drawback, namely, a too much large private and public keys. In the eighties, Niederreiter proposed another variant of McEliece scheme [3]. He proposed to use Generalized Reed-Solomon (GRS) codes. A polynomial time algorithm reconstructing the code parameters from an arbitrary generator matrix was found afterwards by Sidelnikov and Shestakov [4].

Manuscript received: 21 March 2020
Manuscript received in revised form: 17 April 2020
Manuscript accepted: 03 May 2020
Manuscript Available online: 15 May 2020

Therefore the original Niederreiter scheme was completely broken. Next, a new variant of the system was proposed by Berger and Loidreau in [5], which intended to resist the Sidelnikov-Shestakov attack. The idea is to work with a sub code of a GRS code instead of a complete GRS code in order to hide the code structure.

Although the cryptosystem of Niederreiter seems to have a promising future, it remains little implemented in practical life [6]. In this work, we attempt to present a new practical implementation of the cryptosystem. We first give a brief review of the first code-based cryptosystem, namely the McEliece cryptosystem. Therefore, we discuss the basic idea of the implemented scheme, and then present the implementation in details.

II. BACKGROUND

A. The McEliece cryptosystem

The McEliece cryptosystem is the first cryptographic public key scheme to use randomization in the encryption process [7] – [8]. The algorithm has not yet found a good place in practical life, but it seems to have a flourishing future since it is immune to attacks using Shor's algorithm and so belongs to post-quantum cryptography. The scheme security is based on the hardness of decoding a general linear code. The principle can be summarized as follows: For each irreducible polynomial g over GF (2^m) of degree t, there exists a binary irreducible Goppa code C of length $n=2^m$, having an efficient decoding algorithm γ, able to correct any pattern of t errors. Since C is a linear code, it is a k-dimensional vector space over GF (2^m). So we can describe it by an $k \times n$ matrix G, called generated matrix. By using a $k \times k$ regular matrix S and an $n \times n$ permutation matrix P, a new generator matrix G’ is constructed to hide the structure of G, so that:

$$G' = SGP \qquad (1)$$

The public key consists then of G’, whereas the matrices S and P together with g forms the private key as shown in Fig. 1. The new matrix G’ is the generator matrix of another linear code that is assumed to be difficult to decode since the trapdoor information is not known.

To encrypt a plaintext m $\in \{0, 1\}^k$: choose a random vector e $\in$ $\{0, 1\}^n$ of weight t and compute the cipher text as:

$$c = mG' + e \quad . \qquad (2)$$

To decrypt a cipher text c $\in$ $\{0, 1\}^n$, first calculate cP^{-1} as:

$$cP^{-1} = m(SG) + eP^{-1} \ . \qquad (3)$$

Subsequently, by using the decoding algorithmγ, the mS value can be easily computed from the following equation:

$$cP^{-1} = (mS)G + eP^{-1} \ . \qquad (4)$$

The message m is finally restituted by a multiplication with S^{-1}. Fig. 1 below summarizes the process.

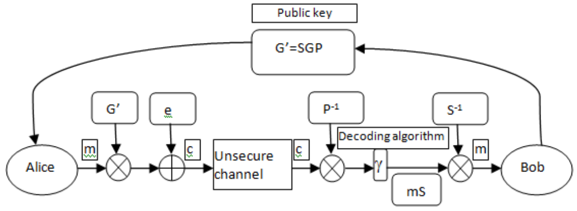


Fig. 1: McEliece cryptosystem

B. *The Niederreiter Cryptosystem*

The Niederreiter scheme is a dual variant of the McEliece cryptosystem that uses a parity check matrix for encryption, instead of a generator matrix [9]. In term of performance, the Niederreiter encryption is about ten times faster than the encryption of McEliece. Moreover, the Niederreiter can be used to construct a secure digital signature scheme [10].

Whereas the first version of the cryptosystem has been completely broken, it has been shown to be equivalent to the McEliece cryptosystem in term of security if appropriate parameters are chosen. Its major drawback is that it works on large keys, e.g., for 128-bit post quantum security (corresponding roughly to 256-bit classical security), it needs a large key of up to 1MB, using parameters proposed in [11].

From structure point of view, the Niederreiter cryptosystem implementation goes through three steps, namely the generation of both private and public keys, the encryption, and then the decryption processes. Here is the detail of each step:

1) *Key generation*

1. Alice selects a binary (n, k)-irreducible Goppa code C, able to correct up to t errors, that has an efficient decoding algorithmγ.
2. Alice generates an (n - k) × n parity check matrix H, for the code C.
3. Alice selects a random (n - k) × (n - k) non singular matrix S.
4. Alice selects a random n × n permutation matrix P.
5. Alice computes the (n - k) × n matrix:
$$H'=SHP \qquad (5)$$
6. Alice's public key is pk = (H', t). Her private key is sk = (S, H, P,γ).

2) *Message encryption*

To send a message *m* to Alice, whose public key is (H', t), Bob proceeds as follows:

1. Bob encodes the message *m* as a binary string of length n and weight at most t.
2. Bob computes the cipher text as:

$$c = H'm^T \ . \qquad (6)$$

3) *Message decryption*

Upon receipt of c from Bob, Alice does the following to retrieve the message *m*.

1. Alice computes
$$S^{-1}c = HPm^T \ . \qquad (7)$$
2. Alice applies the syndrome decoding algorithm γ for the code *C* to recover Pm^T.
3. Alice computes the message *m*, via :
$$m^T=P^{-1}Pm^T \ . \qquad (8)$$

The figure below illustrates the steps mentioned above:

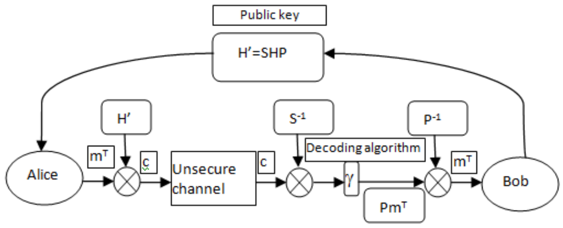


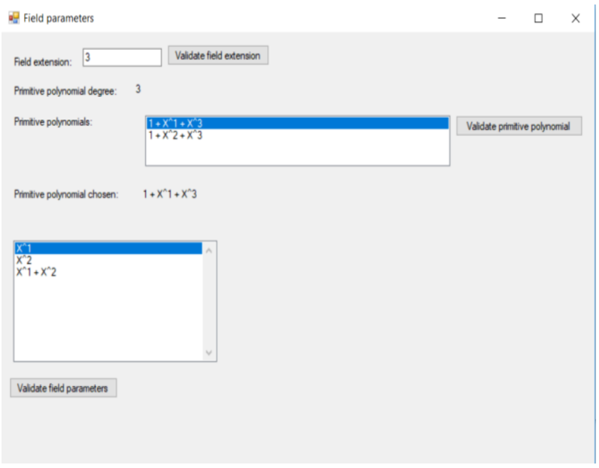
Fig. 2: Niederreeiter cryptosystem

III. IMPLEMENTATION AND RESULT

In this section, we present the results obtained from practical implementation of this algorithm using C # language. The designed system is subdivided into 4 steps: Choosing the Galois field, generation of keys, encryption, and decryption process.

Choosing a Galois Field

In this step, we choose a primitive polynomial as well as one of its roots among a set of polynomials generated in accordance with a given integer. A Galois field is then generated with an extension equal to the previously given number.



Field extension: 3

Primitive polynomial degree: 3

Primitive polynomials:
1+X^1+X^3
1+X^2+X^3

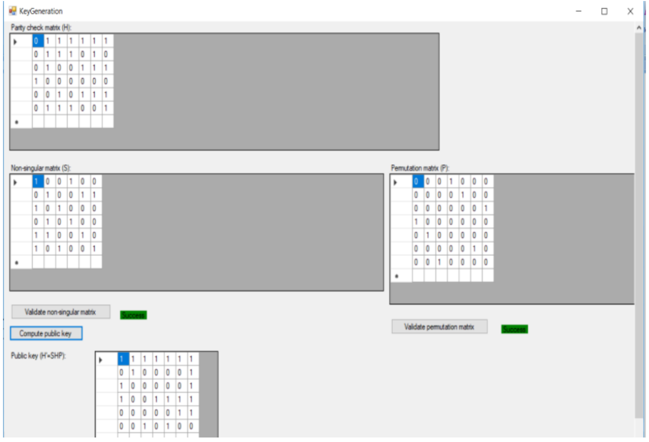
Primitive polynomial chosen: 1+X^1+X^3

X^1
X^2
X^1+X^2

Fig. 3: Choosing domain parameters

Keys generation

This is the most important step of the scheme. It starts by selecting a support for the code as well as an irreducible generator polynomial. A parity check matrix (the secret key) is then generated. Afterwards, a non singular and a permutation matrix are chosen. The public key is generated from the three matrices by a multiplication operation as shown in the Fig. 3.



Party check matrix (H):

0	1	1	1	1	1	1
0	1	1	1	0	1	0
0	1	0	0	1	1	1
1	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	1	1	0	0	1

Non-singular matrix (S):

1	0	0	1	0	0
0	1	0	0	1	1
1	0	1	0	0	0
0	1	0	1	0	0
1	1	0	0	1	0
1	0	1	0	0	1

Permutation matrix (P):

0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	0	1
1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	1	0
0	0	1	0	0	0	0

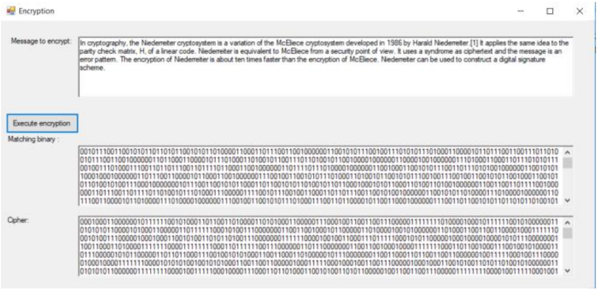
Public key (H'=SHP):

1	1	1	1	1	1	1
0	1	0	0	0	0	1
1	0	0	0	0	0	1
1	0	0	1	1	1	1
0	0	0	0	0	1	1
0	0	1	0	1	0	0

Fig. 4: Key Generation Interface

Encryption process

In this step, we first apply the ASCII encoding standard to convert the plain text message into a binary string. The public key is then used to encrypt the binary string into a cipher. Fig. 5 below shows a practical example of an encoding-encryption process.



Message to encrypt: In cryptography, the Niederreiter cryptosystem is a variation of the McEliece cryptosystem developed in 1986 by Harald Niederreiter [1] It applies the same idea to the parity check matrix, H, of a linear code. Niederreiter is equivalent to McEliece from a security point of view. It uses a syndrome as ciphertext and the message is an error pattern. The encryption of Niederreiter is about ten times faster than the encryption of McEliece. Niederreiter can be used to construct a digital signature scheme.

Matching binary :

```
0010111001100101011011010110010101101000011000110111001100100000011001010111001001110101011101000110000101101110011001110111011010
0101110011001000000110110001100001011101000110100101100111011010010110010000100000011000010010000001110100011000110111010101110111
0010011101000111001101101110011011110110001100100000011011110111010000100000011001000110010101110011011101101001000000110010101101
1000100010000000110111001100001011000110010000000111001001100101011101000110100101100101011100100111001001100101011001000110010
0110100101001110001000000010111001100101011000110110010101101001011011000100010101100010100110101001101001000000110011001101111001000
0001101110011011110110100101110100011100000011110010111001001100011011011100110010100100000011001010110100001110100001000000110
1110011000010110100001110100001000000011100100110010101110100011100110110000101100110001000000011100110110010101101101011010100101
```

Cipher:

```
0001000110000000101111110010100011011001101000011010100011000001110001001100110011100000111111101000010001011111100101000000011
0101010110000101000110000011011111100010100111000000011001100100010110000011010000100101000000110100011001100110000100011111110
0010100111000001000100011001010011010110100111000000011111110000100100110001110111110001010110000010001000010000101011100000001
1001100011010000111111100001111111100011011111110011100000011011100000001100110010001000011111110001101100100011100100101000011
0111000001010110000011011011000111001001010100011001100011010000010111000000011001100011011001100110000001001111100010011110000
0100010000111111100001010101001001010100011001100110000010001111100010001001100111000001000100011001010011010110100101000000011
0101010110000001111111100001001111100010000111000110110100011001010011010110000010011001100111000001111111100001001111110001001
```

Fig. 5: Encoding-Encryption process

Decryption process

The decryption process consists of recovering sent binary string by applying a Syndrome Decoding Algorithm. The plain text message is then restored by an ASCII decoding. To perform decoding process in order to find the error vector, we have opted for the Patterson's algorithm [12]. A brief outline of that algorithm is presented below:

- For received codeword y find syndrome:

$$s(x) = \sum_{i=1}^{n} y_i / x - \alpha_i \bmod g(x)_{.} \qquad (9)$$

- Find h(x), inverse of s(x) in the algebra of polynomials mod g(x).
- Find d(x) such that:

$$d^2(x) \equiv h(x) + x \bmod g(x)_{.} \qquad (10)$$

- By Applying the extended Euclidean algorithm, find a(x) and b(x) of least degree such that:

$$d(x)b(x) \equiv a(x) \bmod g(x)_{.} \qquad (11)$$

- Compute:

$$\sigma(x) = a^2(x) + xb^2(x). \qquad (12)$$

- $\sigma(x)$ is used to determine the set of error Locations

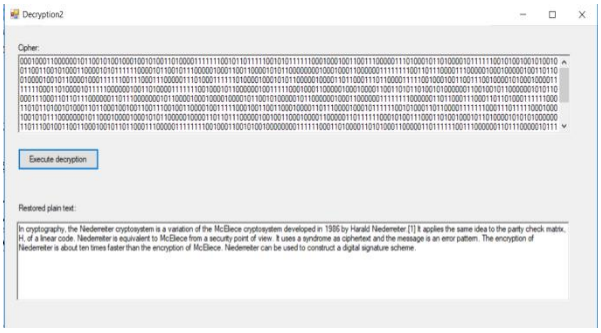
$$E = \{i, \sigma(\alpha_i) = 0\}. \qquad (13)$$

- Error positions are *i* such that

$$i \in E. \qquad (14)$$

- Codeword *e* is found as *y* – *c*.

Fig. 6 shows the decryption-decoding process of the encrypted message.



Cipher:

```
0001000110000000101100101001000100101001101000011111110010110111110010101111110001000100110011100000111010001011010000101111110010100100101010010
0110011001010001100001010111111100001011001011100000100011001100001010110000000010001000110000001111111001101110000111000001000100000100110110
0100001001011000010001111110011100011100000111010001111110100001000101011000001000011011000111011000011111001000100110011100100001010001000011
1111100011010000101111100000010011010000111111100100010110000001001111100010001100000100010000110011010110100101000000110010010110000001010110
0001110001101101110000001101110000000101100001000100001000010110010100000101100000010001100000011111111000000110110001110001101101000111111000
1101011010010100011011000100100110011100100110000100111110001001100110001000011011100001000101111110010100011011000011111110001110111110001000
1001010111000000010110001000010001010110000010000110110111000001001001100010000110000010111111000101001110001101001000101101000010101010000000
1101110010011001100010010110110001110000011111111001000110010100100000000111111000110100001101010001100000110111111001110000000110111000001011
```

Restored plain text:

In cryptography, the Niederreiter cryptosystem is a variation of the McEliece cryptosystem developed in 1986 by Harald Niederreiter [1] It applies the same idea to the parity check matrix, H, of a linear code. Niederreiter is equivalent to McEliece from a security point of view. It uses a syndrome as ciphertext and the message is an error pattern. The encryption of Niederreiter is about ten times faster than the encryption of McEliece. Niederreiter can be used to construct a digital signature scheme.

Fig. 6: Decryption-Decoding process

IV. CONCLUSION

In this paper, we have developed an implementation of the Niederreiter cryptosystem using Visual studio and the C # language as tool. The implementation covers all steps of the cryptosystem, namely the definition of Goppa code, the keys generation, the encryption and the decryption algorithms. To support various levels of security and take into account the performance of the platforms on which the system is supposed to run, we adopted an approach that allows ensuring a wide choice concerning the parameters. Thus, one can choose the extension of Galois field, the generator polynomial, the non singular matrix as well as the permutation matrix. We run the cryptosystem for different instances of Galois fields, secret and public keys, and plain text. The results obtained shows that the system has a high degree of flexibility, a good capacity in term of error correction and a high level of security. From this, it turns out that the designed system can be implemented in practical life, especially to secure data over the Internet.

REFERENCES

[1] V. Mavroeidis, K. Vishi, M. D. Zych, A. Jøsang," The Impact of Quantum Computing on Present Cryptography, International Journal of Advanced Computer Science and Applications (IJACSA), 9(3), 405-414, 2018.

[2] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory". DSN Progress Report, Jet Prop. Lab., California Inst. Tech. 42–44, pp.114–116, 1978.

[3] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory". Problems of Control and Information Theory 15, 159–166, 1986.

[4] V. M. SIDELNIKOV, S. O. SHESTAKOV, "On insecurity of cryptosystems based on generalized Reed-Solomon codes". Discrete Math. Appl.2 (4), pp. 439-444, 1992.

[5] T. P. Berger, P. Loidreau, “How to mask the structure of codes for a cryptographic use”. Designs, Codes and Cryptography 35 (1) pp. 63–79, 2005.

[6] M. Kratochvíl, “Implementation of cryptosystem based on error-correcting codes”, Phd thesis, 2013.

[7] G. Hofmann, “Implementation of McEliece using quasi-dyadic Goppa codes”. Phd thesis, 2011.

[8] P. L. Cayrel, G. Hoffmann, and E. Persichetti, “Efficient Implementation of a CCA2-Secure Variant of McEliece Using Generalized Srivastava Codes”, International Association for Crypto logic Research, pp. 138–155, 2012.

[9] H. Imai, M. Hagiwara, “Error-correcting codes and cryptography”, Applicable Algebra in Engineering, Communication and Computing, Volume 19, Issue 3, pp 213–228, 2008.

[10] A K. Morozov, P. S. Roy, R. Steinwandt, R. Xu, “On the security of the Courtois-Finiasz-Sendrier signature”, Open Mathematics, Volume 16, Issue 1, Pages 161–167, ISSN (Online) 239-455, DOI: https://doi.org/10.1515/math-2018-0011, 2018.

[11] D. Augot, L. Batina, D. J. Bernstein, J. Bos, J. Buchmann, W. Castryck, O. Dunkelman, T. G¨uneysu, S. Gueron, A. H¨ulsing, T. Lange, M. S. E. Mohamed, C. Rechberger, P. Schwabe, N. Sendrier, F. Vercauteren, B. Y. Yang, “Initial recommendations of long-term secure post-quantum systems”. Tech. rep., PQCRYPTO ICT-645622, 2015.

[12] N. J. Patterson, “The Algebraic Decoding of Goppa Codes”. IEEE Trans. Information Theory, 21:203{207, 1975.], 1975.

AUTHOR BIOGRAPHY

Lahcen Elbermi

is full professor at department of computer sciences in Moulay Ismaïl University, Faculty of Sciences and Technics, Errachidia, Morocco. He is interested in cryptography.

Fatima Amounas

is a professor at department computer sciences in Faculty of Sciences and Technics, Errachidia, Morocco. Her research interests include elliptic curve and cryptography.

Younes Bayane

is a software engineer graduated from the Mohammadia School of Engineering in Morocco. He is currently PhD student in Faculty of Sciences and Technics Errachidia in Morocco. His research area is cryptography.