McEliece Cryptosystem Based on Punctured Convolutional Codes and the Pseudo-Random Generators

Hamza MOUFEK and Kenza GUENDA
University of Science and Technology (Algeria)

ken.guenda@gmail.com

Abstract

The purpose of this paper is to present a new version of the McEliece cryptosystem based on punctured convolutional codes and the pseudo-random generators. We use the modified self-shrinking generator to fill the punctured pattern. More precisely we propose to fill out the pattern punctured by the bits generated using a pseudo random generator LFSR.

Keywords
Punctured convolutional code, McEliece cryptosystem, self shrinking generator

1 Introduction

In 1978 Robert J. McEliece invented the first cryptosystem based on algebraic coding theory [10]. Since then different variants have been proposed [5].
Different attacks were made against these schemes. Among them, we mention the attack on the original McEliece system by Canteaut and Sendrier [6] and the attack on the cryptosystem based on convolutional codes by Landais and Tillich [7].
The purpose of this paper is to present a new version of the McEliece cryptosystem based on punctured convolutional codes and the pseudo-random generators. Instead of using time-varying convolutional codes as it was given in [8] and broken by [7], we use the modified self-shrinking generator to fill the punctured pattern. More precisely we propose to fill out the pattern punctured by the bits generated using a pseudo random generator LFSR.

2 Our New Variant

In this section we will give the description of our new variant McEliece Cryptosystem. To hide the structure of the convolutional code, we follow the method of puncturing described in [9]. Starting from a convolutional code of parameters (n, k, K), we construct an equivalent code of parameters (Mn, Mk, K_p), called grouped code. With the puncturing pattern T, whose the number of its coefficient corresponds to the number of columns of the grouped matrix, we obtain the generator matrix of punctured code with parameters $(N^{'}, Mk, K_p)$.
Since we use the modified self-shrinking generator to fill the punctured pattern, in the next paragraph we describe the modified self shrinking generator given by Kanso [6].

2.1 The Modified Self Shrinking Generator :

In [6] Kanso modified the method of Meier and Staffelbach [11] by using one LFSR of length s which operates as follows:
Let A be an LFSR that generates the sequence $a_t = a_0, a_1, a_2, \ldots$
At time i, we consider the triplet $(a_{3i}, a_{3i+1}, a_{3i+2})$. If the bit $a_{3i} \bigoplus a_{3i+1} = 1$, the output of the LFSR is a_{3i+2}. Else no output is produced.
The puncturing pattern is of size $n \times M$. So we proceed so that the modified self-shrinking generator product an output sequences of period greater or equal than $n \times M$.

2.2 Description of our Cryptosystem

Algorithm 1: Key Generation

1. Choose a generator matrix G for a convolutional code of parameters: its length n and its dimension k.
2. Write the polyphase decomposition of elements of G and then forming the polycyclic pseudocirculant matrix.
3. Replace the polynomials $g_{i,j}(D)$ of G by their Mth PCPC matrix and interlacing the lines and columns at depth M.
4. Generate a random sequence of bits in a modified self-shrinking generator, and fill the matrix T by the $n \times M$ output elements.
5. Apply the function ϕ to the matrix $G^{[M]}(D)$ to get the secret matrix G_p .
6. Choose randomly tow matrix: P the permutation matrix and S an invertible matrix.
7. Compute the public matrix $G^{'} = SG_pP$.

Algorithm 2: Encryption

For sending a message x to someone, we calculate: $c = xG^{'} + e$

Algorithm 3: Decryption

To decrypt the message, you must:

1. Compute $z = cP^{-1}$
2. Determining $z^{'}$ by correcting errors of u.
3. Compute $x = z^{'}S^{-1}$

To correct errors of z, the Viterbi decoding algorithm is used, because an algorithm which decodes the parent code, it also decodes the punctured code.

3 Security of our Scheme:

There are several ways to attack the McEliece encryption system. Among them we find algebraic methods and probabilistic methods.
In this section we give a proof that our scheme is secure against the structural and decoding attacks.

3.1 Structural Attacks:

The objective of a structural attack is to find an equivalent code to the public code whose a polynomial decoding algorithm is known. For this, we used a puncturing pattern to hide the structure of the code, whose the attacker cannot imagine. Moreover, the equivalence of punctured codes is an NP-complete problem [12]. This makes impossible the cryptanalysis of the system. Our system is resistant to the structural attack cited in [7] because the shape of our generator matrix is different from that used in the attacked cryptosystem [8].

3.1.1 Exhaustive Search Attack:

In our scheme, the private key (G_p, S, P, T) is obtained randomly. In this Section we will show that our scheme is secure against the exhaustive search attack. This is equivalent to show that it is a difficult task to find the private key. For that we start by showing that the choice of the matrices S and P is very large. Namely, since the number of invertible matrices in $\mathbb{F}_q$ is $\prod_{i=1}^{k}(q^k - q^{i-1})$ and the number of permutation matrices of size n is equal to $n!$. Then in order to find the two selected matrices we have to try $n! \prod_{i=1}^{k}(q^k - q^{i-1})$ matrices.
Now, we will show that the complexity of finding the matrix G_p is very large. For that, let A be an r-sequence generated by a primitive LFSR of length s and let L be the set of positions of the columns removed during the puncturing step from the matrix G. To find a subset $L^{'} \subseteq L$, it is necessary to know a part of the sequence generated by the LFSR. For constructing r bit-triples i.e.; $s = 3r$, we get a total of $2^{0.862s}$ possible states for the LFSR. Further our method to select the matrix T has a complexity of the order 2^{80} for s=93 [6]. This makes impossible to find the private key by exhaustive research.
For example, if we puncture a (400,300)-convolutional code C, the grouped code associated to C will be of parameters (800,600) and the puncturing pattern T will be of size 400×2. Then

the Modified Self Shrinking Generator must generate at least 800 output bits. Therefore the initial state of the LFSR must be at least of length 2400. The complexity of the cryptanalysis of this LFSR by an exhaustive search attack is $\mathcal{O}(2^{2069})$.

3.2 Information Set Decoding:

For security level around 2^{80} measured by Canteaut-Chabaud's algorithm [4], we propose the following set of parameters.
Let C be an $(400, 343)$-convolutional code. After a puncturing of depth 7 in 56 positions, we obtain a punctured code of length 2744 and dimension 2401.
For a code of rate $R = 3/4$, we propose a puncturing of depth 3 for an $(570, 421)$- convolutional code in 26 positions. Thereafter we will have an $(1684, 1263)$-punctured code.

In the Table 1 we give examples of different (n, k) - convolutional code and their $(N^{'}, k^{'})$-punctured code associated with different security level.

Security level	n	k	M	Number of deleted coulumns	$k^{'} = kM$	$N^{'}$	Rate
80	305	150	4	20	600	1200	1/2
	284	71	5	30	355	1420	1/4
	570	421	3	26	1263	1684	3/4
	125	1050	2	100	250	2000	1/8
100	316	154	5	40	770	1540	1/2
	625	155	3	15	465	1860	1/4
	730	540	3	30	1620	2160	3/4
	68	550	5	30	340	2720	1/8

Table 1: Suggested parameters of our cryptosystem for different security levels

3.3 Message-Resent Attack:

Suppose now that, through some accident, or as a result of action in the part of the cryptanalyst, both $c_1 = mSGP + e_1$ and $c_2 = mSGP + e_2$ with $e_1 \neq e_2$ are sent. We call this a message-resent condition.

Using an $(1684, 1263)$-punctured code with a free distance $d_{free} = 131$. We compare the effect of this attack on our system and the McEliece cryptosystem, we get the following graph:

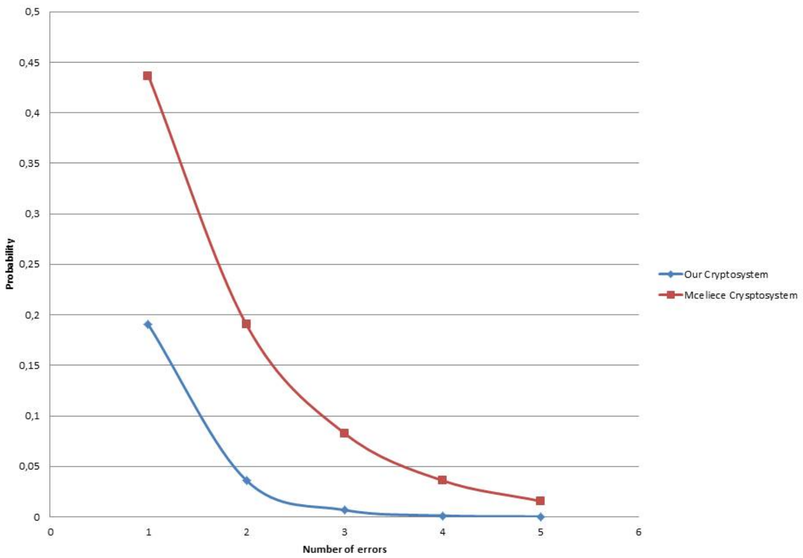

Figure 1: Probability of guessing k ungarbled columns from those indexed by L_0 depending on the number of errors

These results are better than the result obtained by attacking Mceliece crptosystem. We remark that whenever we increase the number of errors, the probability of avoiding this attack increases.

References

[1] M. Barbier and P.S.L.M. Baretto, Key reduction of McEliece's cryptosystem using list decoding, Intern.Sym. Inform. Theory ISIT2011, Saint-Pettersburg, 2011.

[2] T.P. Berger, P.L. Cayrel, P. Gaborit and A. Otmani, Reducing key length of the McEliece cryptosystem, In Prog. Crypt. AfricaCrypt 2009.

[3] T.A. Berson, Failure of the McEliece public-key cryptosystem under message-resend and related-message attack. In Burton S. Kaliski Jr., editor, Advances in Cryptology-CRYPTO '97, volume 1294 of Lecture Notes in Computer Science, pages 213-220. Springer-Verlag, 17-21 August 1997.

[4] A. Canteaut and F. Chabaud, A new algorithm for finding minimum-weight words in a linear code: Application to McElieces cryptosystem and to narrow-sense BCH codes of length 511. IEEE Transactions on Information Theory, 44(1):367378, 1998.

[5] R. Johannesson and K. SZigangirov, Fundamentals of convolutional coding. IEEE Press, USA, 1999.

[6] A. Kanso. Modified self-shrinking generator. Comp. Elec. Engineering 36(5): 993-100, 2010.

[7] G. Landais and J.P. Tillich, An Efficient Attack of a McEliece Cryptosystem Variant Based on Convolutional Codes. Lecture Notes in Computer Science vol. 7932, pp 102-117, 2013.

[8] C. Londahl and T. Johansson, A new version of McEliece PKC based on convolutional codes. Int. Conf. Inform. Com Security ICICS 2012, Hong-Kong, Oct. 2012.

[9] M. Marazin, R. Gautier and G. Burel, Algebraic method for blind recovery of punctured convolutional encoders from an erroneous bitstream, IET Signal Proces. 2011.

[10] R.J. McEliece, A public-key cryptosystem based on algebraic coding theory. DSN Progress Report 42-44, 114-116, 1978.

[11] W. Meier and O. Statfelbach, The Self-Shrinking Generator, adv.Crypt., Eurocrypt 94, Lecture note in computer science vol.950, pages 205-214, 1995.

[12] C. Wieschebrink, Two NP-complete problems in coding theory with an application in code based cryptography, Intern. Sym. Inform. Theory ISIT06 Seattle, Jul. 2006.