Strengthening McEliece Cryptosystem

Pierre Loidreau

Project CODES, INRIA Rocquencourt Research Unit - B.P. 105 - 78153 Le Chesnay Cedex France
Pierre.Loidreau@inria.fr

Abstract. McEliece cryptosystem is a public-key cryptosystem based on error-correcting codes. It constitutes one of the few alternatives to cryptosystems relying on number theory. We present a modification of the McEliece cryptosystem which strengthens its security without increasing the size of the public key. We show that it is possible to use some properties of the automorphism groups of the codes to build decodable patterns of large weight errors. This greatly strengthens the system against the decoding attacks.

1 Introduction

Since public-key cryptography was introduced in 1977 in the fundamental paper of Diffie and Hellman, it has taken an increasing importance in research as well as application fields. Many public-key ciphers have been proposed during the last twenty years; they rely on various difficult problems such as factoring numbers, computing discrete logarithms, solving knapsack problems... However, the conjugate development of computing power and efficient algorithms have made many of them insecure. A common point between the non-yet broken systems is that they remain dangerously linked with only two problems of number theory – the difficulty of factoring an integer and the difficulty of computing a discrete logarithm – and we are not protected from a theoretical breakthrough.

McEliece proposed an alternative to such systems in 1978 [McE78]. It consists in a public-key cryptosystem based on error-correcting codes. Together with its Niederreiter [Nie86] version - of equivalent security [LDW94] - the original system based on the family of Goppa codes still resists cryptanalysis. The general security of the scheme relies on the inherent intractability of decoding a random code up to its error-correcting capability. The great advantage of systems based on error-correcting codes is the extremely low cost of their encryption and decryption procedures. It approaches the complexity of secret key encryption schemes. Furthermore, if by chance major breakthroughs were made in number theory problems, such systems would constitute one of the few possible alternatives; therefore the study of their security is essential. The cost of a general decoding attack on these systems depends on the size of the chosen code and its error-correcting capability. The best known algorithm based on this method points out [CS98] that the size of the public key of the original system is becoming short regarding the increasing power of the computers. A safer step should be

to take a larger key. Yet, in that case the huge size of key (more than 880kbytes) would be a major disadvantage for implementation on limited resource systems. In this paper we present a modification of McEliece cryptosystem which makes all decoding attacks infeasible without increasing the size of the public-key.

The underlying idea results from a trade-off between the strong security of the system against structural attack and its much weaker security regarding decoding attacks. We allow ourselves to reduce the size of the space of public-keys weakening the system against structural attacks to increase the security of the system regarding the decoding attacks. This can be done by using to some property of the automorphism group of Goppa codes. Namely, whenever the Frobenius automorphism lies in the automorphism group of the code we can generate large sets of decodable error-words of a larger weight than the constructed error-correcting capability of the code. We show that whenever such sets are used in the system, the cost decoding attacks is significantly increased.

2 McEliece Public-Key Cryptosystem

In the family of public-key cryptosystems based on coding theory the one proposed by McEliece is the most widely considered. Namely, it is not only the first encryption scheme using coding theory ever proposed but it has also ever resisted to the attacks attempting to recover the secret key.

Other McEliece-like systems using different families of codes have been structurally cryptanalysed [SS92]. The credit for its resistance can thus be given to the family of Goppa codes taken as the secret key space. Their poor structure prevents an attacker to find a way to reduce significantly the size of the key-space. However, the size of the public key has to be significantly large to avoid general decoding attacks. Even with such a constraint both encryption and decryption procedures for the system remain much faster than for RSA.

2.1 Description of the Cryptosystem

A linear binary code of length n and dimension k is a linear subspace of $\mathbf{F}_2^n$. It can be represented by a $k \times n$ binary matrix called generating matrix. Two codes C_1 and C_2 of length n are said to be equivalent if there exists a permutation of the n coordinate places changing C_1 into C_2.

The permutations of coordinate places sending a code C into itself form the automorphism group of the code C.

Irreducible Goppa Codes The secret key space is a family of irreducible Goppa codes [MS77] pp. 338. The receiver must thus consider some notions of finite fields algebra. Namely, in the construction of a Goppa code $\Gamma(L, g)$ [Gop70], we use

1. a finite field $\mathbf{F}_{2^m}$ with 2^m elements. $\mathbf{F}_{2^m}$ is the support field of the code,
2. a labeling L of $\mathbf{F}_{2^m}$. L is called generating vector of the code,
3. an irreducible polynomial g over $\mathbf{F}_{2^m}$ of degree t. g is called generating polynomial of the code.

Properties Every irreducible Goppa codes $\Gamma(L, g)$ has a fast polynomial time decoding algorithm [Pat75] up to its constructed error-correcting capability. The error-correcting capability of the codes is lower bounded by t, the degree of the generating polynomial, that is any error of weight less than t occurring on a codeword can be corrected.

Key Space To construct the scheme one takes the family $\mathcal{G}$ of irreducible Goppa codes of length $n = 2^m$, dimension k and error-correcting capability t. The cardinality of $\mathcal{G}$ is almost always equal to the number of irreducible polynomials of degree t over the finite field with 2^m elements that is approximately $2^m/t$. With the original parameters, $n = 1024,\ k = 524,\ t = 50$, the size of the space is around 2^{496}.

Cryptosystem It has the following form:

1. Private key: a Goppa code $\Gamma(L, g)$ randomly picked up in the family $\mathcal{G}$, a random $k \times k$ non-singular binary matrix S, and a random $n \times n$ permutation matrix P.
2. Public key: the product $G' = SGP$, where G is a generating matrix for $\Gamma(L, g)$.
3. Encryption: let x be the k-bit message to be encrypted, the sender computes $x' = xG' + e$ where e is a n-bit error-vector of weight t.
4. Decryption: the receiver computes $x'P^{-1} = xSG + eP^{-1}$, and then recovers xS by using the fast decoding algorithm of $\Gamma(L, g)$. Since S is non-singular the receiver recovers x.

The security of the system depends on the difficult problem of decoding a code up to its error-correcting capability.

Complexity of Encryption-Decryption This encryption scheme has an extremely low complexity compared to the RSA. Namely, [Can96]

– in the encryption procedure we can take for granted that the cost of generating a random word of length n and weight t is negligible compared to the cost of a matricial product. Hence the work factor for encryption is

$$W^C = nk/2$$

– by using the Euclidian algorithm –which is not the most efficient but whose complexity is the easiest to evaluate – to make the decoding the work factor for decryption is:

$$W^D \approx \underbrace{3mnt + 4m^2t^2}_{\text{decoding algorithm}} + k^2/2$$

Originally Goppa codes of length 2^{10} dimension 524, and degree of the generator 50 are taken. This gives:

– Number of binary operations per information bit for encryption:
$W^C/k = 512$, which is smaller than the 2402.7 binary operations per information bit required in the RSA-1024 encryption procedure.
– Number of binary operations per information bit for the decryption:
$W^D/k = 5101.7$, which is much smaller than the 738 112.5 binary operations per information bit required in the RSA-1024 decryption procedure.

With such parameters the system runs more than 100 times faster for decryption than the RSA-1024 [CS98].

However the system has three main drawbacks:

1. the transmission rate is low: k/n that is 51 percent in this case. Some attempts have been made to increase the transmission rate.
2. the size of the public key has to be huge: kn bits, approximately $500Kbits$. If keys are smaller the scheme does not resist to decoding attacks.
3. encrypting the same message twice is recognizable and the plaintext can be recovered straightforward.

Note that by using the Niederreiter variant [Nie86] of the system, we can completely eradicate the problem of encrypting the same message twice. Moreover it allows to increase the transmission rate and to halve the size of the public-key without reducing the security of the system [LDW94]

2.2 Attacks on the System

There are two main approaches to cryptanalyse the system. They rely on two separate difficult problems.

1. The first one consists reconstructing a decoder for the code generated by the public-key G' by studying its structure. A such approach is denoted structural attack.
   From the very construction of the system, the code C' generated by the public key G' is equivalent to $\Gamma(L, g)$. The attack consists in enumerating the codes in the family $\mathcal{G}$ to find a code $\Gamma \in \mathcal{G}$ which is equivalent to C'. Since equivalence classes of Goppa codes are constructible [Gib91] one can reduce the cost of the attack by examining a single element in each equivalence class. Yet the equivalence classes have a too small cardinality to decrease significantly the cost of the attack. For instance if we take the original parameters, – $t = 50$, and $n = 1024$ – there are $\approx 2^{496}$ irreducible polynomials of degree 50 over $\mathbf{F}_{2^{10}}$, and the equivalence classes have at most 2^{30} elements. Finding a code equivalent to C' implies thus to explore on average more than 2^{466} codes. This remains largely beyond the capabilities of the most powerful computers.
   Once Γ equivalent to C' has been found, one recovers the permutation between Γ and C' by applying for instance the Support Splitting Algorithm [Sen99].

2. The second approach consists in decoding the intercepted ciphertexts m' relatively to the public code C' generated by the public-key. It is called decoding attack.
   Since $\Gamma(L, g)$ is equivalent to C' both codes have the same error-correcting capability t and the equation $x' = x + e,\ x \in C'$ has a unique solution (x, e) with e of weight less than t. The cost of the attack depends only on the parameters of C', its length, its dimension and its error-correcting capability. It implies that the parameters of the system have to be chosen very carefully and large enough. For this reason, the original parameters given by McEliece (length 1024, dimension 524, error-correcting capability $t = 50$) are becoming rather small for the state of art [CS98]: decoding one word takes on average 2^{64} binary operations. The next "safer" step would be to take $n = 2048$ for the code length. However the size of the key would become really prohibitive, for implementation on limited resource systems.

Whereas efficient decoding attacks were developed, the investigations concerning the reconstruction of a decoder remain rather scarce. In the general instance of the system there is no better way than exhaustive search on the key space - reduced modulo the equivalence relation -, testing the equivalence of each code with the code generated by the public-key.

One could replace the Goppa codes by any other family of codes with a fast polynomial-time decoding algorithm. Many codes are better than Goppa code regarding the decoding attacks. However the structure of theses families make the system insecure against structural attacks. For instance if one replaces the family of Goppa codes with the family of generalized Reed-Solomon codes or the family of concatenated codes, the recovering of a decoder can be done straightforward. [SS92, Sen98].

3 Tower Decodable Patterns

Taken randomly Goppa codes have a similar structure to random codes. In particular their automorphism group is usually trivial. Yet, Goppa codes with non trivial automorphism group are constructible: if the generating polynomial has coefficients in a subfield $\mathbf{F}_{2^s}$ of support field $\mathbf{F}_{2^m}$, then the automorphism group of the code is generated by the Frobenius automorphism. The attacker can detect this property by applying the Support Splitting Algorithm to the public key. This property was used to derive an almost realistic structural attack on the McEliece parameters, whenever the generating polynomial has binary coefficients [LS98].

Although such a property weakens the system against structural attack by reducing the size of the secret key space, we show that it can equally be used to strengthen the system against decoding attacks. By using properties of the automorphism group the conceiver can build sets of decodable patterns of large weight.

Moreover, from a cryptographic standpoint this set should satisfies some preliminary conditions: it must be large enough to avoid exhaustive search, the

error words must have a weight larger than the error-correcting capability of the code. If such sets are used in place of the error vectors added in the original system, the cost of decoding attacks is greatly increased without changing the size of the public key.

3.1 Automorphism Group of Goppa Codes

Suppose the support field is $\mathbf{F}_{2^m}$, and let $L = (\alpha_1, \ldots, \alpha_n)$ be a labeling of the support field. Let us consider the Goppa code $\Gamma(L, g)$ where the generating polynomial g has coefficients in a subfield $\mathbf{F}_{2^s}$ of $\mathbf{F}_{2^m}$. Then we have

Proposition 1. *The automorphism group of $\Gamma(L, g)$ contains the group generated by the Frobenius automorphism $\sigma : z \mapsto z^{2^s}$ of $\mathbf{F}_{2^m}/\mathbf{F}_{2^s}$.*

The proof can be derived from Moreno's theorem [MS77] pp 347.

This means that the code $\Gamma(L, g)$ is invariant under the action of the Frobenius automorphism. If any word c of length n is labeled by L, we have

$$\forall\ c = (c_{\alpha_1}, \ldots, c_{\alpha_n}) \in \Gamma(L, g), \quad \sigma(c) = (c_{\sigma(\alpha_1)}, \ldots, c_{\sigma(\alpha_n)}) \in \Gamma(L, g)$$

3.2 t-Tower Decodability

Definition 1. *Let $\mathcal{E}$ be a set of words of length $n = 2^m$, let $\mathbf{F}_{2^s}$ be a subfield of $\mathbf{F}_{2^m}$ and $\sigma : z \mapsto z^{2^s}$ the Frobenius automorphism of the extension field. We say that $\mathcal{E}$ is t-tower decodable if*

1. *for all $e \in \mathcal{E}$, there exists a linear combination*

$$E = \sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(e), \quad \epsilon_i \in \mathbf{F}_2$$

   having a Hamming weight less than t, where $\sigma(e)$ denotes the action of the Frobenius on the word e,
2. *the knowledge of E enables the receiver to recover e in $\mathcal{E}$ in a unique way.*

In other words $\mathcal{E}$ is a t-tower decodable set if there exists a linear combination of the powers of the Frobenius automorphism σ that is a one-to-one mapping from $\mathcal{E}$ into the vectors of length n and weight less than the correcting capability of the Goppa code.

The second condition in the definition is fundamental. It ensures that given a pattern we can invert all the operations to recover the original vector e.

The first condition is simple to achieve: Let us take $\mathcal{E}$, the set of all the binary words e of length n satisfying

$$\sum_{i=0}^{m/s-1} \sigma^i(e) = 0$$

However it does not satisfy the second condition. Namely every word in $\mathcal{E}$ is mapped onto the null word.

t-tower decodability is intimately linked with classical decodability up to t in the family of Goppa codes with a non-trivial automorphism group :

Proposition 2. *Let $\Gamma(L, g)$ be a Goppa code with generating vector of degree t over a subfield $\mathbf{F}_{2^s}$ of the support field $\mathbf{F}_{2^m}$, then any error vector of a t-tower decodable set $\mathcal{E}$ is correctable in $\Gamma(L, g)$.*

Proof. Let $x' = x + e$ where x is a codeword in $\Gamma(L, g)$ and $e \in \mathcal{E}$. By definition of $\mathcal{E}$ there exist a linear combination of the power of the Frobenius $E = \sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(e)$ having weight less than t.

From Sect. 3.1 the automorphism group of $\Gamma(L, g)$ contains σ. Thus, the linear combination $x' = \sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(x)$ is also in the code $\Gamma(L, g)$.

Since $\sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(x) = x' + E$, by applying the decoding algorithm of $\Gamma(L, g)$ one recovers E. From Definition 1, the error-vector e can be recovered in a unique manner. □

3.3 Modified Cryptosystem

Space of Secret Keys Let g_1 be an irreducible polynomial of degree t_1 over $\mathbf{F}_{2^m}$. g_1 is called hiding polynomial. Let $\mathcal{G}$ be the family of the Goppa codes $\Gamma(L, g_1 g)$ where g describes the family of irreducible polynomials of degree t over a subfield $\mathbf{F}_{2^s}$ of $\mathbf{F}_{2^m}$.

Private Key Not changing from the original scheme, it is made of 3 parts:

- a $k \times n$-generating matrix G of a code $\Gamma(L, g_1 g)$ randomly chosen in $\mathcal{G}$
- a $n \times n$ permutation matrix P,
- a $k \times k$ non-singular matrix S.

Public Key To the difference of the original scheme it consists in two parts

- the product $G' = SGP$,
- the way to generate a t-tower decodable set $\mathcal{E}$.

Encryption Let x be the k-bit message that has to be transmitted. The sender chooses randomly a word e in $\mathcal{E}$, then sends $x' = xG' + e$.

Decryption The receiver first computes $x'P^{-1} = xSG + eP^{-1}$.

Since e is in the t-tower decodable set $\mathcal{E}$, from Definition 1 there is a linear combination $\sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(e)$ of weight less than the error correcting capability t of $\Gamma(L, g)$.

The receiver computes

$$\sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(x'P^{-1}) = \sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(xSG) + \sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(eP^{-1})$$

Note that xSG is a word in the code $\Gamma(L, g_1 g)$. However, by construction, $\Gamma(L, g_1 g)$ is a subcode of $\Gamma(L, g)$. Therefore we can consider that xSG is a word in $\Gamma(L, g)$. Moreover, since σ is in the automorphism group $\Gamma(L, g)$ by construction, $\sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(mSG)$ is also a codeword of $\Gamma(L, g)$. Since P^{-1} is a permutation we have

$$\sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(eP^{-1}) = \left(\sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(e) \right) \cdot P^{-1}$$

which is a decodable pattern in $\Gamma(L, g)$. The receiver gets thus the vector$E = \left(\sum_{i=0}^{m/s-1} \epsilon_i \sigma^i(e) \right)$ of weight less than t. E can thus be recovered by applying the decoding algorithm of $\Gamma(L, g)$. The knowledge of E provides a unique way to find e.

Complexity of the Scheme The complexity of the encryption is exactly the same as in the original system, since consisting in matricial products and picking up a random vector.

The decryption requires additional operations. However, the cost strongly depends on the structure of the t-tower decodable set $\mathcal{E}$.

Conditions on $\mathcal{E}$ From a cryptological point of view, the t-tower decodable set must satisfy the following conditions:

1. $\mathcal{E}$ has to be a set of words of weight larger than the error-correcting capability of the code. This conditions strengthens the system against decoding attacks,
2. $\mathcal{E}$ has to be large enough to avoid enumeration. Namely, if an exhaustive search on the possible error-words were feasible the initial message x would be easily recovered,
3. the way to generate $\mathcal{E}$ must be public, and must not reveal information that could help an attacker.

Importance of the Hiding Polynomial g_1 We introduced the concept of hiding polynomial g_1 to satisfy the third condition on $\mathcal{E}$. If we used for $\mathcal{G}$ the family of irreducible Goppa codes with generating polynomial over $\mathbf{F}_{2^s}$, by applying the support splitting algorithm to the public key G' any attacker would be able to recover σ. Then one could apply linear transformations of the Frobenius automorphism and reduce the problem of finding the error vector e to the problem of finding the vector E of lower weight.

The codes $\Gamma(L, g_1 g)$ are subcodes of the codes $\Gamma(L, g)$ with a large structure. The introduction of the hiding polynomial scrambles the structure of the code rendering the automorphism group of $\Gamma(L, g_1 g)$ trivial. Moreover, the hiding polynomial g_1 can be published since its knowledge does not give any exploitable information.

4 Extension of Degree 5

In the previous section we introduced the theoretical concept of tower decodability and how to use it in cryptography. In practice however, it is uneasy to build t-tower decodable sets satisfying the cryptological requirements. Therefore we focus on the example of extensions of degree 5. They not only turned out to be suitable from a cryptological viewpoint but they also intervene in the original parameters of the system. When using such t-tower decodable sets we show that, without increasing the size of the public-key, the security of the modified system is increased.

4.1 Construction of a t-Tower Decodable Set

We consider the field extension $\mathbf{F}_{2^{5s}}$ of $\mathbf{F}_{2^s}$, and the corresponding Frobenius automorphism $\sigma : z \mapsto z^{2^s}$. Since 5 is prime, the orbits of the elements of $\mathbf{F}_{2^{5s}}$ have size 5 except the orbits of the elements of $\mathbf{F}_{2^s}$ of size 1. Hence there are exactly $N_5 = (2^{5s} - 2^s)/5$ orbits of size 5. Let $L = (\alpha_1, \ldots, \alpha_n)$ be a labeling of the field $\mathbf{F}_{2^{5s}}$. From now on, we suppose that any word of length n is labeled by L.

The action of the Frobenius automorphism σ on e corresponds exactly to the action of the automorphism on the coordinates of e: if $e = (e_{\alpha_1}, \ldots, e_{\alpha_n})$ then $\sigma(e) = \big(e_{\sigma(\alpha_1)}, \ldots, e_{\sigma(\alpha_n)}\big)$.

We define a t-tower decodable set with respect to the Frobenius automorphism as follows,

Definition 2. *Let $\mathcal{E}$ be the set of all the possible words of length $n = 2^{5s}$ constructed this way:*

1. *one chooses randomly p orbits out of the N_5 orbits of size 5 in the generating vector L, where p satisfies $p = \lfloor t/2 \rfloor$*
2. *puts randomly 3 bits on every chosen orbit.*
3. *puts the coordinates to zero on the remaining positions.*

The set $\mathcal{E}$ contains words of weight $3p = 3\lfloor t/2 \rfloor$. The construction of $\mathcal{E}$ relies on the knowledge of the position of the orbits in the labeling L of the field.

Proposition 3. *Let $\mathcal{E}$ be the set of words previously defined, we have*

1. *the cardinality of $\mathcal{E}$ is $10^p \cdot \binom{N_5}{p}$,*
2. *$\mathcal{E}$ is t-tower decodable.*

Proof. There are $\binom{N_5}{p}$ possibilities in choosing p orbits of size 5 out of N_5. Once these orbits have been chosen, there are $\binom{5}{3} = 10$ possibilities in choosing three bits out of 5, proving the first assertion.

Let $e = (e_{\alpha_1}, \ldots, e_{\alpha_n})$ be a word in the set $\mathcal{E}$. We reorder the labeling L of the support in such a way that e is written

$$e = (e_1, e_2, \ldots, e_{N_5}, \underbrace{0, \ldots, 0}_{\mathbf{F}_{2^s}})$$

where the $e_i = (e_{\alpha_i}, e_{\sigma(\alpha_i)}, e_{\sigma^2(\alpha_i)}, e_{\sigma^3(\alpha_i)}, e_{\sigma^4(\alpha_i)})$ denote the subvectors of length 5 of e labeled by the orbit corresponding to the element α_i. By construction of $\mathcal{E}$, the e_i have either weight 0 or weight 3.

After the reordering, the action of the Frobenius automorphism σ on the word e becomes a combination of cyclic shifts on the vectors e_i. Therefore all 5-bit patterns of weight 3 can be divided into two classes f_1 and f_2 up to the Frobenius shifting equivalence:

Type 1	Type 2
$f_1 = (11100)$	$f_2 = (11010)$
$\sigma(f_1) = (01110)$	$\sigma(f_2) = (01101)$
$\sigma^2(f_1) = (00111)$	$\sigma^2(f_2) = (10110)$
$\sigma^3(f_1) = (10011)$	$\sigma^3(f_2) = (01011)$
$\sigma^4(f_1) = (11001)$	$\sigma^4(f_2) = (10101)$

The patterns f_1 and f_2 play dual roles. There exist linear combinations of f_1, f_2 and of their Frobenius images that enables one to reduce the weight of one pattern from 3 to 1 preserving the weight of the other. The average weight of the pattern is thus decreased. Namely we have

$$f_1 + \sigma(f_1) + \sigma^2(f_1) = (10101), \quad f_2 + \sigma(f_2) + \sigma^2(f_2) = (00001)$$

and

$$f_1 + \sigma^2(f_1) + \sigma^3(f_1) = (01000), \quad f_2 + \sigma^2(f_2) + \sigma^3(f_2) = (00111)$$

Whenever one has the image $f + \sigma(f) + \sigma^2(f)$ or $f + \sigma^2(f) + \sigma^3(f)$ of a pattern f of weight 3 , f is recoverable in a unique way:

1. from the weight of the obtained pattern one gets the type of the pattern, either type 1 or type 2,
2. from the positions of the bits on the obtained pattern, one gets the original one.

To prove that $\mathcal{E}$ is t-tower decodable, it is sufficient to prove that one of the linear combinations $e+\sigma(e)+\sigma^2(e)$ and $e+\sigma^2(e)+\sigma^3(e)$ has weight less than t. Suppose now that e is made of p_1 patterns of type 1 and p_2 patterns of 2. Then $e+\sigma(e)+\sigma^2(e)$ has weight $3p_1+p_2$, and $e+\sigma^2(e)+\sigma^3(e)$ has weight p_1+3p_2. Since by construction $p = p_1 + p_2 = \lfloor t/2 \rfloor$, we have

$$2(p_1 + p_2) \leq t$$

This implies in particular that at least either $3p_1+p_2$ or p_1+3p_2 is less than t. Hence at least one of the images of e by the previous combinations has weight less than t.

For instance if $e + \sigma^2(e) + \sigma^3(e)$ has weight less than t then by using the property of one-to-one correspondence between the patterns of weight 3 and their image by this transformation, the word e can be recovered entirely.

Thus $\mathcal{E}$ is t-tower decodable. □

Remark 1. The optimal parameters for $\mathcal{E}$ are $2p = t$. In this case each word in $\mathcal{E}$ has weight $3t/2$. With this method one can decode up to one half beyond the error-correcting capability t.

McEliece Parameters The set considered is the set of irreducible polynomials over $\mathbf{F}_{2^{10}}$ and error-correcting capability 50. Since $m = 10$, we have $s = 2$. The number of orbits of size 5 is 204. By taking the parameter $p = 25$, then the set $\mathcal{E}$ generated is composed of 2^{188} words of weight 75. Still it is negligible compared to the 2^{284} patterns of weight 50 but remains large enough to avoid enumeration.

4.2 Application to the Cryptosystem

Section 3.3 was dedicated to the modification of McEliece system by using the general properties of t-tower decodability to strengthen the system against decoding attacks. In this section we apply this modification with the t-tower decodable sets previously defined over extensions of degree 5. In particular we show that it is possible to publish how to generate $\mathcal{E}$ without giving the possibility for an attacker to reduce the complexity of the attacks on the system.

Parameters of the System

Family of Goppa Codes As a hiding polynomial we take an irreducible polynomial g_1 of degree 2 over $\mathbf{F}_{2^{5s}}$. Let L be a labeling of the field $\mathbf{F}_{2^{5s}}$, we consider the family $\mathcal{G}$ of Goppa codes $\Gamma(L, g_1 g)$ where g has degree t and coefficients over $\mathbf{F}_{2^s}$.

Private Key It consists in 3 parts:

- a $k \times n$-generating matrix G of a code picked up randomly in $\mathcal{G}$,
- a $n \times n$-permutation matrix P,
- a non-singular $k \times k$-matrix S.

Public Key

1. the matrix $G' = SGP$,
2. the positions of the N_5 orbits of cardinality 5 in the generating vector L.

Note that if the positions of the orbits are in some way canonical the size of the public-key can be made as low as the size of G'.

Encryption-Decryption Since the positions of the orbits are public, the sender can generate the set $\mathcal{E}$ of t-tower decodable words described in the previous section.

Encryption Let x be the k-bit plaintext one has to transmit, the sender chooses randomly a word e in $\mathcal{E}$: he first picks up $\lfloor t/2 \rfloor$ orbits out of the N_5 possible and puts randomly 3 bits on each orbit. The corresponding ciphertext is $x' = xG' + e$.

Decryption The receiver computes $x'P^{-1} = xSG + eP^{-1}$. Since permuting the coordinates does not change the structure of the automorphism group, we can consider that eP^{-1} is still in $\mathcal{E}$. It was shown in 3 that E is t-tower decodable, therefore by applying the right linear combinations of the powers of the Frobenius automorphism, the receiver first recovers eP^{-1}, then recovers x.

To evaluate the relative cost of the procedure compared to the original scheme, we have to separate it into different steps.

1. First one has to compute the two linear combinations $x'_1 = x' + \sigma(x') + \sigma^2(x')$ and $x'_2 = x' + \sigma^2(x') + \sigma^3(x')$. Let s_n be the cost of computing the action of σ on a vector of length n, and let a_n be the cost of xoring two words of length n. Overall the cost is $3(s_n + a_n)$. The action of σ is the product of cyclic shifts thus we neglect the cost of this step compared to the complexity of the decoding part.
2. Decoding part: let $A_t = 3mnt + 4m^2t^2$ be the cost (given in 2.1) of decoding one word corrupted by a t-bit error-vector. In the original system the decoding part costs exactly A_t operations. In this modification the cost is at most $2A_t$ and can be greatly reduced. Namely we first try to decode x'_1 and only if the decoding fails then we decode x'_2. Thus the additive cost of the procedure to recover E from x'_1 or x'_2 is on average at most $1/2A_t$.
3. The cost to recover e from E is a few times the cost of running over the n positions of the word, so it can be neglected compared to the cost of the decoding procedure.

Thus if D is the cost of the decryption in the original scheme, and D_1 is the cost in time of the decryption in the modified scheme we have

$$D_1 \leq D + 1/2A_t$$

By taking the original parameters – $n = 1024,\ t = 50$ – the number of binary operations per information bit for decryption becomes: $W^D/k = 7521.5$, which remains much smaller than the 738 112.5 operations per information bit required for the RSA-1024 decryption.

The memory cost is identical in both schemes.

Security of the System In the conception of the scheme the positions of the orbits of size 5 in the support of the code are public stuff. It does not jeopardize the scheme since it does not provide a potential attacker with exploitable information. Given indeed this information it seems difficult to recover additional properties enabling to recover the Frobenius automorphism. This would imply that given the public code one could build a larger code from which we only know the non-ordered orbits through its automorphism group.

By considering the McEliece parameters we show that this system provides a better security against decoding attacks than the original scheme.

McEliece Parameters If $\mathcal{G}$ is the set $\Gamma(L, g_1 g)$ where L is a labeling of $\mathbf{F}_{2^{10}}$ and g runs over the polynomials of degree 50 over $\mathbf{F}_{2^2}$ then

- the size of $\mathcal{G}$ is approximately 2^{95},
- the size of the public key is of the same order as in the original system. $\Gamma(L, g_1 g)$ being a subcode of $\Gamma(L, g)$ the size a generating matrix for $\Gamma(L, g_1 g)$ will be slightly smaller than the size of a generating matrix for $\Gamma(L, g)$,
- $\mathcal{E}$ is a family of 50-tower decodable codewords of weight 75 and has cardinality 2^{188}. This is very few compared to the set of patterns of length 1024 and weight 50 having cardinality 2^{284} that are decodable, but still remains largely out of range for the computers.
  In that case applying the best algorithm for decoding attack [CS98] gives roughly 2^{91} binary operations compared to the 2^{64} involved for breaking the original system.

5 Conclusion

In the paper we showed how to use the automorphism group of Goppa codes to increase the security of the McEliece system against decoding attacks. This approach can be easily transferred to its Niederreiter type version, the security of which is the same. Of course the specific structure we require from the family of Goppa codes enables any attacker to greatly reduce the complexity of a structural attack compared to the cost of a structural attack on the original version. However, in the example developed above concerning the extensions of degree 5 the size of the family of codes to enumerate remains largely beyond the capabilities of the computers. The security is thus the result of a trade-off between the two kinds of attacks.

Such an approach can be generalized to any finite field extension with characteristic 2. Still, in that case the problem is to find t-tower decodable sets satisfying the simple cryptographical constraints such as being a large set of large-weight words. The ideal would be to find a decodable set whose words have weight larger than half of the code-length. Decoding attacks would be then completely obsolete, and as a consequence, the main problematic factor which is the large size of the public-key would vanish. Gabidulin, Paramonov and Tretjakov proposed such a cryptosystem based on error-correcting codes [GPT91] with very nice properties. This system is unbreakable with a decoding attack and has a very low key size (less than 10kbits). Unfortunately the codes in the key space have so much structure that in its first version it was efficiently broken by K. Gibson [Gib95].

References

[Can96] Anne Canteaut. *Attaques de cryptosystèmes à mots de poids faible et construction de fonctions t-résilientes.* PhD thesis, Université Paris-VI, 1996.

[CS98] A. Canteaut and N. Sendrier. Cryptanalysis of the original McEliece cryptosystem. In Kazuo Ohta and Dingyi Pei, editors, *Advances in Cryptology - ASIACRYPT'98*, number 1514 in LNCS, pages 187–199, 1998.

[Gib91] J. K. Gibson. Equivalent Goppa codes and trapdoors to McEliece's public key cryptosystem. In D. W. Davies, editor, *Advances in Cryptology - EUROCRYPT'91*, number 547 in LNCS, pages 517–521. Springer-Verlag, 1991.

[Gib95] J. K. Gibson. Severely Denting the Gabidulin Version of the McEliece Public Key Cryptosystem. *Designs, Codes and Cryptography*, 6:37–45, 1995.

[Gop70] V. D. Goppa. A new class of linear error-correcting codes. *Problemy Peredachi Informatsii*, 6(3):207–212, 1970.

[GPT91] E .M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. Ideals over a non-commutative ring and their application in cryptology. *LNCS*, 573:482 – 489, 1991.

[LDW94] Y. X. Li, R. H. Deng, and X. M. Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Transactions Information Theory*, 40(1):271–273, 1994.

[LS98] P. Loidreau and N. Sendrier. Some weak keys in McEliece public-key cryptosystem. In *IEEE International Symposium on Information Theory, ISIT'98, Boston*, page 382, 1998.

[McE78] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. Technical report, Jet Propulsion Lab. DSN Progress Report, 1978.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error–Correcting Codes*. North Holland, 1977.

[Nie86] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159 – 166, 1986.

[Pat75] N. J. Patterson. The algebraic decoding of GOPPA codes. *IEEE Transactions Information Theory*, 21:203–207, 1975.

[Sen98] N. Sendrier. On the concatenated structure of a linear code. *AAECC*, 9(3):221–242, 1998.

[Sen99] Nicolas Sendrier. The Support Splitting Algorithm. Technical Report 3637, INRIA, March 1999. http://www.inria.fr/RRRT/RR-3637.html.

[SS92] V. M. Sidel'nikov and S. O. Shestakov. On cryptosystems based on generalized REED-SOLOMON codes. *Discrete Mathematics*, 4(3):57–63, 1992. in russian.