AN OVERVIEW OF SYMMETRIC ALTERNANT CODES AND THE STRUCTURAL CRYPTANALYSIS OF THEIR CORRESPONDING MCELIECE SCHEMES

FILIP STOJANOVIC

Abstract. The McEliece public-key cryptosystem (PKC) based on binary Goppa codes has proven to be very resilient in the face of both classical and quantum attacks. It, however, is not used in practice because its public key sizes are far larger than those of currently used cryptosystems for an equivalent level of security. As a remedy, it was first proposed in [G] to use symmetric codes as the secret codes of McEliece schemes so as to permit the reduction in their public key sizes. However, this adoption of symmetric codes makes the corresponding McEliece schemes vulnerable to the structural attacks devised by Faugère et al. in [F1], [F2], and [F3]. In this report, we will summarize basic results concerning the classes of symmetric Alternant codes suggested for use in the McEliece PKC, culminating in the observation that a shorter code of lower dimension can be constructed from any of these symmetric Alternant codes and that this reduced code gives away as much information about the private key as does the full code. We will then summarize the structural attack introduced in [F1] and how this observation by Faugère et al. greatly decreases the computational effort needed to recover the private key for a McEliece scheme based on these kinds of symmetric Alternant codes.

Acknowledgements

Once again, I sincerely thank Dr. Monica Nevins for offering me yet another opportunity to work with her on the subject of the McEliece cryptosystem. This project has continued broadening my mathematical horizons and has been a pleasure to work on, in no small part due to Monica's consistent helpfulness, enthusiasm, and expertise. I appreciate immensely her willingness to share her knowledge on all things from coding theory to the p-adic numbers and even further research opportunities and I'm incredibly lucky to have had her as a supervisor.

This work was supported by the grant "Quantum Security via Algebras and Representation Theory (Quasar)" from the Tri-Council New Frontiers in Research Fund of Canada.

1. Introduction

The security of the McEliece PKC based on binary Goppa codes first introduced in [M] has endured to this day: all known algorithms, both classical or quantum, that can decrypt a McEliece cipher without having access to the private key are of exponential complexity. Our motivation for studying symmetric codes follows from the main drawback of the McEliece PKC. Despite its longstanding security,

Date: January 19, 2021.

the PKC is not used in practice. The main reason for this is its large public key, which is a large generator matrix for the secret code. Even in systematic form, the binary representation of the public key is several orders of magnitude larger than the public keys of currently used cryptosystems like RSA for an equivalent level of security. However, the key sizes are dependent on the secret code used to define an instance of a McEliece scheme; for example, in the original proposal [M] as well as in the current National Institute of Standards and Technology proposal [Be], the secret code is a binary Goppa code.

Symmetric codes have been proposed to get around this drawback as they introduce redundancies in the public key that allows it to be described using less information than would be required for the public key of a code of equal length and dimension bearing no symmetries. The main classes of symmetric codes proposed for practical use were *quasi-cyclic* and *quasi-monoidal*, but these were shown to introduce structural weaknesses in their corresponding McEliece schemes, as revealed by the work done by Faugère et al. in [F1], [F2], and [F3].

The aim of this report is to be a guide to the work of Faugère et al. concerned with the structural cryptanalysis of McEliece schemes based on symmetric codes. We present the basic theory regarding symmetric codes that are either GRS codes or constructed from GRS codes by taking the subfield subcode (i.e. Alternant codes). We consider the symmetries of *quasi-cyclic* and *quasi-monoidal* GRS and Alternant codes, revealing that they are generated either by a single or a set of permutations that can be constructed from injective affine transformations. Given a symmetric code, we present the constructions of two notable related codes defined by it and its group of symmetries, $\mathbb{G}$, the $\mathbb{G}$-*subcode* and the *folded code*: the first, transforms its symmetries into redundancies; and the second, defined from the first, removes these redundancies. Lastly, we present the FOPT attack introduced in [F1] and how the folded code becomes instrumental in reducing the computational effort of this attack.

We invite the unfamiliar reader to refer to Section 5.1 in [St] for a recap of the McEliece PKC. Basic properties of GRS and Goppa codes are documented in sections 3 and 4 of the same report and although they will be recalled, it is useful for the reader to be familiar with certain results concerning the equivalence of GRS codes. We adopt the notational conventions of [St] for this report.

2. Symmetric Alternant Codes and the FOPT Attack

2.1. **Introduction to Symmetric GRS-Derived Codes.** We begin by describing code automorphisms: given a linear code C, a *code automorphism* is an isomorphism from C to itself that is isometric with respect to the Hamming distance. This definition implies that the automorphism group of a linear code will consist of permutations from C to itself, transformations that scale codewords of C by a non-zero value, and their compositions. We describe our notion of code symmetry from the set of such permutations, which we call the permutation group of the code and define formally as follows.

Definition 2.1.1. Let C be a (n,k) linear code. We define the *permutation group* of C by

$$Perm(C) := \{\sigma \in S_n : c^\sigma \in C \ \ \forall c \in C\},$$

where we denote the action of $\sigma \in S_n$ on $c \in C$ by $c^\sigma := (c_{\sigma(1)}, \ldots, c_{\sigma(n)})$.

We formally say that a code is *symmetric* when it has a non-trivial permutation group. For example, the permutation group of a cyclic code (defined for instance in Definition 2.7.3 in [N]) will contain $\langle R \rangle$ where R is the permutation that rotates the coordinates of a vector by one step to the right.

It was first proposed in [G] to use a symmetric code (whose generator matrices exhibit certain redundancies) as the secret code of the McEliece scheme so that the public key sizes may be reduced. The rationale is that the public key is a permuted generator matrix for the secret code, so using a symmetric code whose generator matrix can be fully constructed knowing only a few of its columns will allow one to use those columns acted on by a random permutation in place of the full permuted generator matrix for the public key. This compression of information was achieved through the use of quasi-cyclic codes and a better form of compression was achieved through the use of quasi-monoidal codes as proposed in [B].

We will now outline the nature of these symmetries as well as how such symmetric GRS/Alternant/Goppa codes are constructed. Firstly, if $C \subseteq \mathbb{F}^n$ is a linear code where $char(\mathbb{F}) = p$, then its permutation group is isomorphic to

- $\mathbb{Z}/\lambda\mathbb{Z}$ for some $\lambda \leq n$ if C is *quasi-cyclic*, or
- $(\mathbb{Z}/p\mathbb{Z})^\lambda$ for some $\lambda \leq n$ if C is *quasi-monoidal*.

Quasi-dyadic codes often also pop up in literature discussing the use of symmetric codes to reduce the size of the public keys of McEliece schemes, but these are nothing more than quasi-monoidal codes defined over a field of characteristic $p = 2$. These can therefore be treated in the same way as quasi-monoidal codes. Before we begin outlining the constructions of GRS/Alternant/Goppa codes with the above symmetries, we will recall the definition of a GRS code and a result describing the equivalence of GRS codes given as a part of Theorem 3.2.1 in [St]

Definition 2.1.2. The (n,k) *GRS code* defined by the vectors $\alpha, \beta \in \mathbb{F}_{p^m}^n$ is

$$GRS_{n,k}(\alpha,\beta) := \{(\beta_1 f(\alpha_1), \ldots, \beta_n f(\alpha_n)) : f \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})\}$$

where $\alpha_i \neq \alpha_j$ for all $i \neq j$ and $\beta_i \neq 0$ for all i. The vectors α and β are typically referred to as the *locator* and *multiplier*, respectively.

Proposition 2.1.3. *If $\alpha, \beta \in \mathbb{F}_{p^m}^n$ are chosen such that $GRS_{n,k}(\alpha,\beta)$ is a (n,k) GRS code, then for any $\mu, \eta \in \mathbb{F}_{p^m}^\times$ and $\nu \in \mathbb{F}_{p^m}$, we may define $\alpha', \beta' \in \mathbb{F}_{p^m}^n$ by $\alpha'_i = \mu\alpha_i + \nu$ and $\beta'_i = \eta\beta_i$ for all $i = 1, \ldots, n$ such that*

$$GRS_{n,k}(\alpha,\beta) = GRS_{n,k}(\alpha',\beta').$$

Proof. See [St]. □

Since Alternant codes are subfield subcodes of GRS codes, this result extends to them too. Let $A_r(\alpha,\beta)$ denote the Alternant code $GRS_{n,n-k}(\alpha,\beta)^\perp \cap \mathbb{F}_p^n$ for some $r \leq n$. Take $\alpha, \beta, \alpha', \beta' \in \mathbb{F}_{p^m}^n$ as in the last proposition and consider the Alternant

codes $A_r(\alpha, \beta)$ and $A_r(\alpha', \beta')$. They must be equal as by the proposition, we have $GRS_{n,n-k}(\alpha, \beta) = GRS_{n,n-k}(\alpha', \beta')$; thus,

$$A_r(\alpha, \beta) = GRS_{n,n-k}(\alpha, \beta)^{\perp} \cap \mathbb{F}_{p^m}^n = GRS_{n,n-k}(\alpha', \beta')^{\perp} \cap \mathbb{F}_{p^m}^n = A_r(\alpha', \beta').$$

To convince oneself of this, it is a useful exercise to use Proposition 3.1.7 in [St] to show

$$GRS_{n,n-k}(\alpha, \beta)^{\perp} = GRS_{n,k}(\alpha, \gamma) \text{ and } GRS_{n,n-k}(\alpha', \beta')^{\perp} = GRS_{n,k}(\alpha', \eta^{-1}(\mu^{-1})^{n-1}\gamma)$$

for $\gamma \in (\mathbb{F}_{p^m}^{\times})^n$ as per the proposition.

Note that r typically called the *degree* of the Alternant code and it describes the number of rows in a full-rank parity-check matrix for the associated GRS code $GRS_{n,n-k}(\alpha, \beta)^{\perp}$. Equivalently, r is the codimension of this GRS code, meaning $r = n - k$. So as not to overload this section, we will recall some basic properties about Alternant codes that were not treated in [St] in the appendix.

These results about the equivalence of codes will give us a recipe for constructing permutations under which GRS and Alternant codes will be invariant, which are the permutations induced by an injective affine transformation.

Definition 2.1.4. Let $\mu \in \mathbb{F}_{p^m}^{\times}$, $\nu \in \mathbb{F}_{p^m}$, and $\alpha \in \mathbb{F}_{p^m}^n$ such that $\alpha_i \neq \alpha_j$ for all $i \neq j$. If the set $\{\alpha_1, \ldots, \alpha_n\}$ is invariant under the affine transformation defined by $x \mapsto \mu x + \nu$, then we define the *permutation induced by this affine transformation* to be $\sigma \in S_n$ such that for all $i = 1, \ldots, n$, $\sigma(i)$ is the unique integer in $\{1, \ldots, n\}$ for which $\alpha_{\sigma(i)} = \mu\alpha_i + \nu$.

That is to say σ is the permutation whose action on the coordinates of α is the injective affine transformation $x \mapsto \mu x + \nu$. Under a minor condition, such a permutation will belong to the permutation group of a GRS code.

Proposition 2.1.5. *Let $\mu \in \mathbb{F}_{p^m}^{\times}$, $\nu \in \mathbb{F}_{p^m}$, and $\alpha \in \mathbb{F}_{p^m}^n$ such that $\alpha_i \neq \alpha_j$ for all $i \neq j$ and the set $\{\alpha_1, \ldots, \alpha_n\}$ is invariant under the affine transformation defined by $x \mapsto \mu x + \nu$. Let $\sigma \in S_n$ be the permutation induced by this affine transformation and let $l = order(\sigma)$. Let $\beta \in (\mathbb{F}_{p^m}^{\times})^n$ and suppose that there exists a l^{th} root of unity $\eta \in \mathbb{F}_{p^m}$ such that $\beta^{\sigma} = \eta\beta$. We therefore have $\sigma \in Perm(GRS_{n,k}(\alpha, \beta))$ for any $k \in \{1, \ldots, n\}$.*

Proof. Exercise.
Hint: The condition of η being a l^{th} root of unity is needed so that $\beta^{\sigma^l} = (\eta^l)\beta = \beta$ since $order(\sigma) = l$. It would be more helpful to simply consider η as some element in $\mathbb{F}_{p^m}^{\times}$. □

By the equivalence of Alternant codes we established earlier, any permutation induced by an affine transformation for which there exists $\eta \in \mathbb{F}_{p^m}^{\times}$ as in the last proposition will belong to the permutation group of Alternant code $A_r(\alpha, \beta) = GRS_{n,n-k}(\alpha, \beta)^{\perp} \cap \mathbb{F}_p^n$ for any r such that $A_r(\alpha, \beta)$ is an Alternant code.

An analogous result exists for Goppa codes. We recall that for $\Gamma(\alpha, g)$ to be a Goppa code, we must have that $\alpha \in \mathbb{F}_{p^m}^n$ such that its entries are distinct and $g \in \mathbb{F}_{p^m}[x]$ such that $g(\alpha_i) \neq 0$ for all $i = 1, \ldots, n$.

Proposition 2.1.6. *If we let $\mu \in \mathbb{F}_{p^m}^{\times}$ and $\nu \in \mathbb{F}_{p^m}$, and let $\{\alpha_1, \ldots, \alpha_n\}$ be invariant under the affine transformation defined by $x \mapsto \mu x + \nu$, then if we choose $g \in \mathbb{F}_{p^m}[x]$ as outlined in Proposition 4 of* [F2]*, the permutation induced by this affine transformation belongs to $Perm(\Gamma(\alpha, g))$. Furthermore, if we denote this permutation by σ and let $l = order(\sigma)$, then $Perm(\Gamma(\alpha, g)) = \langle\sigma\rangle$ and it is isomorphic to $\mathbb{Z}/l\mathbb{Z}$.*

Proof. See [F2] □

Such a Goppa code is therefore quasi-cyclic and we can see that its permutation group is generated by a *single* permutation induced by an injective affine transformation. Quasi-monoidic Alternant and Goppa codes can be constructed similarly (the precise details for which are given in Proposition 5 in [F2]), but the difference is that their permutation groups are generated by *a set* of permutations, each induced by an affine transformation.

Quasi-cyclic and quasi-monoidal codes exhibit redundancies that allow for their generator matrices to be constructed given only a subset of their columns (see [G] and [B] for precise details). In fact, all codes with non-trivial permutation groups possess redundancies. The nature of the redundancies may not be immediately obvious, but it can be illuminated by their $\mathbb{G}$-*subcode*, which we will now define.

Definition 2.1.7. Let C be a linear code and let $\mathbb{G} \subseteq Perm(C)$ be a subgroup of its permutation group. We define the $\mathbb{G}$-*subcode* of C as

$$\widetilde{C}^{\mathbb{G}} := \left\{ \sum_{\sigma \in \mathbb{G}} c^{\sigma} : c \in C \right\}.$$

We leave it as an exercise to verify that this really is a subcode of C. We next present how this subcode reveals the redundancies of C.

Proposition 2.1.8. *Let C be a (n, k) linear code with $\mathbb{G} \subseteq Perm(C)$ a subgroup of its permutation group. Let $i \in \{1, \ldots, n\}$ and let $\mathbb{G}(i) := \{\sigma(i) : \sigma \in \mathbb{G}\}$ denote the orbit of i under $\mathbb{G}$. We have that for any $\tilde{c} \in \widetilde{C}^{\mathbb{G}}$, all coordinates of $\tilde{c}$ indexed by an element of $\mathbb{G}(i)$ are the same.*

Proof. Let $\tilde{c} \in \widetilde{C}^{\mathbb{G}}$ be given, so there exists $c \in C$ such that $\tilde{c} = \sum_{\sigma \in \mathbb{G}} c^{\sigma}$. Let $i \in \{1, \ldots, n\}$ and $\tau \in \mathbb{G}$ be given. We must show $\tilde{c}_i = \tilde{c}_{\tau(i)}$, which is straightforward.

$$\tilde{c}_{\tau(i)} = \sum_{\sigma \in \mathbb{G}} c_{\sigma\circ\tau(i)} = \sum_{\sigma' \in \mathbb{G}\tau} c_{\sigma'(i)} = \sum_{\sigma' \in \mathbb{G}} c_{\sigma'(i)} = \tilde{c}_i$$

The result hinges on the following equality of cosets: $\mathbb{G} = \mathbb{G}\tau$. This is clear since these cosets are not disjoint, seeing that $\tau \in \mathbb{G} \cap \mathbb{G}\tau$. □

This subcode effectively transforms the symmetries of C into redundancies. It turns out that removing the redundancies in $\widetilde{C}^{\mathbb{G}}$, meaning *puncturing* the codewords (see Section 9.1 in [St] for a definition) so that we keep only one of the coordinates indexed by an element of a distinct orbit of $\{1, \ldots, n\}$ under $\mathbb{G}$, defines the *folded code*, which becomes instrumental in attacking a McEliece scheme based on symmetric codes. Before moving our discussion to folded codes, we will outline one last result concerning the dimension of the $\mathbb{G}$-subcode.

Proposition 2.1.9. *Let C be a (n,k) linear code and let $\mathbb{G} = Perm(C)$ be its permutation group such that $l = |\mathbb{G}|$. Let $\{b_1, \dots, b_k\}$ be a basis of C and suppose that this basis is invariant under $\mathbb{G}$ where the action of $\mathbb{G}$ on the basis is such that $\sigma \in \mathbb{G}$ acts on $b \in \{b_1, \dots, b_k\}$ by $b \mapsto b^\sigma$. If we assume that each orbit is of size l (meaning that there are $\frac{k}{l}$ orbits in total), then the dimension of $\widetilde{C}^{\mathbb{G}}$ is $\frac{k}{l}$.*

Proof. Exercise. □

The condition on a basis of C in the proposition is satisfied for quasi-cyclic and quasi-monoidic codes.

2.2. **Folded Codes.** Given a symmetric code, we may define its $\mathbb{G}$-subcode, which transforms its symmetries into redundancies, and then we may remove them by *folding* the subcode. We will first explicitly define the folded code and give a preliminary result about folded subcodes of symmetric Alternant codes.

Definition 2.2.1. Let C be a (n,k) linear code and let $\mathbb{G} \subseteq Perm(C)$ be a subgroup of its permutation group. For each orbit $\mathbb{G}(i) := \{\sigma(i) : \sigma \in \mathbb{G}\}$ where $i = 1, \dots, n$, choose a representative (the smallest element, for instance) and list out the representatives as $i_1, \dots, i_s$. The *folded code* of C with respect to $\mathbb{G}$ is a code of length s defined as

$$\overline{C}^{\mathbb{G}} := \left\{ \left(\sum_{\sigma \in \mathbb{G}} c_{\sigma(i_j)} \right)_{j=1}^{s} : c \in C \right\}.$$

If C were a symmetric Alternant code, then the folded code $\overline{C}^{\mathbb{G}}$ will be a subcode of an Alternant code.

Proposition 2.2.2. *Let $A_r(\alpha, \beta)$ be an Alternant code of length n and let $\mathbb{G} \subseteq Perm(A_r(\alpha,\beta))$ be a subgroup of its permutation group. We have that the folded code $\overline{(A_r(\alpha,\beta))}^{\mathbb{G}}$ is a subcode of an Alternant code.*

Proof. Let C denote $A_r(\alpha,\beta)$ and let $I = \{i_1, \dots, i_s\}$ be the set of representatives of the orbits of $\{1, \dots, n\}$ under $\mathbb{G}$. Recall that the $\mathbb{G}$-subcode of C is $\widetilde{C}^{\mathbb{G}} = \left\{ \sum_{\sigma \in \mathbb{G}} c^\sigma : c \in C \right\}$. Let us define $P_{\overline{I}}(\widetilde{C}^{\mathbb{G}})$ by puncturing $\widetilde{C}^{\mathbb{G}}$ so that for each codeword, we keep only the coordinates indexed by I. Explicitly, we have

$$P_{\overline{I}}(\widetilde{C}^{\mathbb{G}}) := \{(\tilde{c}_i)_{i \notin I} : \tilde{c} \in \widetilde{C}^{\mathbb{G}}\} = \left\{ \left(\sum_{\sigma \in \mathbb{G}} c_{\sigma(i_j)} \right)_{j=1}^{s} : c \in C \right\} = \overline{C}^{\mathbb{G}}.$$

If we puncture C similarly so that $P_{\overline{I}}(C) := \{(c_i)_{i \notin I} : c \in C\}$, we find that $\overline{C}^{\mathbb{G}} = P_{\overline{I}}(\widetilde{C}^{\mathbb{G}}) \subseteq P_{\overline{I}}(C)$ since $\widetilde{C}^{\mathbb{G}}$ is a subcode of C. By Remark 9.1.6 in [St], a punctured Alternant code will be the subcode of an Alternant code, so since puncturing is a morphism of vector spaces, $\overline{C}^{\mathbb{G}}$ is a subcode of an Alternant code. □

It is a fact that the dimension of the folded code $\overline{C}^{\mathbb{G}}$ is equal to the dimension of $\widetilde{C}^{\mathbb{G}}$. Thus, if C is a quasi-monoidic code, then the folded code $\overline{C}^{\mathbb{G}}$ is a shorter code of lower dimension of than C. Narrowing our scope to Goppa codes, if we consider a Goppa code with a permutation group generated either by a single permutation induced by an affine transformation or by a set of permutations each induced by an affine transformation, it is also true that the Goppa code folded with respect to

a subgroup $\mathbb{G}$ of its permutation group will be a subcode of another shorter Goppa code of lower dimension.

While it may not generally be true that a folded Goppa code with an affine-induced permutation group is itself also a Goppa code, a similar characterization can be made about the dual of such a symmetric Goppa code. The analogous characterization exists for Alternant codes with permutation groups generated by a single or a set of permutations induced by affine transformations. That is to say, the dual of folded codes with such affine-induced permutation groups will again be the dual of a code from the same family (be it Goppa or Alternant). We will give an overview of the result here and suggest the reader consults [F2] for more details.

Theorem 2.2.3. *Let $A_r(\alpha,\beta) = GRS_{n,n-k}(\alpha,\beta)^\perp \cap \mathbb{F}_p^n$ be an Alternant code defined by $\alpha,\beta \in \mathbb{F}_{p^m}^n$ with the usual conditions on α,β.*

- *If $\mathbb{G} = Perm(A_r(\alpha,\beta)) = \langle\sigma\rangle$ where $\sigma \in S_n$ is a permutation of order l induced by an injective affine transformation for which there exists an l^{th} root of unity $\eta \in \mathbb{F}_{p^m}$ such that $\beta^\sigma = \eta\beta$, then there exist $x,y \in \mathbb{F}_{p^m}^{n/l}$ and integer r' such that $\overline{(A_r(\alpha,\beta)^\perp)}^{\mathbb{G}} = A_{r'}(x,y)^\perp$. The parameters x,y,r' are related to α,β,r as described by Theorem 1 in* [F2].
- *If $\mathbb{G} = Perm(A_r(\alpha,\beta)) \cong (\mathbb{Z}/p\mathbb{Z})^\lambda$ is of the form given by Proposition 5 in* [F2]*, then $\overline{(A_r(\alpha,\beta)^\perp)}^{\mathbb{G}}$ is the dual of a shorter Alternant code. The relationship between the parameters of the shorter code and of the full code are described by Theorem 1 in* [F2].

Proof. See [F2]. □

The perfectly analogous result for Goppa codes is demonstrated in [F2]. It suggests that if $\Gamma(\alpha,g)$ is a quasi-cyclic or quasi-monoidal Goppa code with permutation group $\mathbb{G}$, then the folded code with respect to $\mathbb{G}$ is a shorter Goppa code of lower dimension. Furthermore, the parameters of the folded Goppa code are related to the parameters of the original code in such a way that given knowledge of the original code's permutation group and the folded code's parameters, we can recover the original parameters.

In all of the cases considered in the above theorem, the folded code defined from a symmetric Alternant code (or Goppa code) ends up being a shorter code of lower dimension and from the same family as the original symmetric code. What's especially important to note is that in all cases, the parameters of the folded code are related to the parameters of the original code in such a way that *recovering the parameters of the folded code will allow for the recovery of the parameters of the original code*. This becomes a critical flaw for a McEliece scheme based on such symmetric codes since we can mount an attack against the folded code instead of the original code, an approach that is less computationally expensive since the folded code is smaller than the original.

2.3. **FOPT Attack on Symmetric Alternant Codes.** Given a McEliece scheme based on a quasi-cyclic or quasi-monoidal Alternant code, the approach of recovering the parameters of the folded Alternant code to then recover the parameters of the original code is realized through the attack introduced by Faugère et al.

in [F1]. We will call this attack the FOPT attack and at its heart, it relies on solving a homogeneous system involving multivariate polynomials. We will describe how to perform the FOPT attack on an arbitrary Alternant code, but the intended target of the attack for a McEliece scheme based on a symmetric Alternant code is its folded code (which is in fact an Alternant code by Theorem 2.2.3). Thus, in order to attack such a McEliece scheme, one needs to first construct the folded code given the public key. This process is considered in [F3] and the interested reader is encouraged to consult this article.

To outline the FOPT attack, let $A_r(\alpha, \beta)$ be a (n, k) Alternant code. By the development in the appendix, a parity-check matrix $\mathbf{H}$ over $\mathbb{F}_{p^m}$ for $A_r(\alpha, \beta)$ can be written as follows.

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{r-1} & \alpha_2^{r-1} & \dots & \alpha_n^{r-1} \end{bmatrix} \begin{bmatrix} \beta_1 & & & \\ & \beta_2 & & \\ & & \ddots & \\ & & & \beta_n \end{bmatrix}$$

It is clear that if we are given a parity-check matrix in this form (known as the Alternant form), we can immediately recover the Alternant code's parameters from the first two rows of $\mathbf{H}$.

If $A_r(\alpha, \beta)$ is the secret code of a McEliece scheme, then the public code is generated by matrix $\mathbf{M} = \mathbf{PGS}$ such that $\mathbf{G}$ is a generator matrix for $A_r(\alpha, \beta)$, $\mathbf{P}$ is a random $n \times n$ permutation matrix, and $\mathbf{S}$ is a $k \times k$ invertible matrix. We can eliminate the permutation matrix from the scheme as if we can recover the parameters for the permutation-equivalent Alternant code $\mathbf{P}(A_r(\alpha, \beta))$, we can efficiently identify $\mathbf{P}$ by the Support-Splitting Algorithm introduced in [S]. Thus, with $\mathbf{M} = \mathbf{GS}$ taken to be the public matrix, an attacker's "win condition" becomes to find the Alternant form of a parity-check matrix for $A_r(\alpha, \beta)$ as this will give the attacker access to the code's parameters, and, hence, an efficient error-correction algorithm for it.

For $x, y \in \mathbb{F}_{p^m}^n$, we define $V_r(x, y)$ to be the matrix in Alternant form as follows.

$$V_r(x, y) = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{r-1} & x_2^{r-1} & \dots & x_n^{r-1} \end{bmatrix} \begin{bmatrix} y_1 & & & \\ & y_2 & & \\ & & \ddots & \\ & & & y_n \end{bmatrix}$$

An attacker has access to McEliece's trapdoor once equivalent parameters for the code $x, y \in \mathbb{F}_{p^m}^n$ have been identified such that $ker(V_r(x, y)) \cap \mathbb{F}_p^n = A_r(\alpha, \beta)$. We notice that this requires for all $c \in A_r(\alpha, \beta)$, we must have $V_r(x, y)c = 0$, which is equivalent to $V_r(x, y)\mathbf{M} = 0$. We will develop this into a system of equations in the coordinates of x and y.

For consistency with the notation in [F1], let $M_{ij} = g_{i,j}$ for all $i = 1, \ldots, n$, $j = 1, \ldots, k$. We notice that the $(i,j)^{th}$ entry of the product $V_r(x,y)\mathbf{M}$ is

$$\begin{bmatrix} y_1 x_1^{i-1} & y_2 x_2^{i-1} & \ldots & y_n x_n^{i-1} \end{bmatrix} \begin{bmatrix} g_{1,j} \\ g_{2,j} \\ \vdots \\ g_{n,j} \end{bmatrix} = 0.$$

But this is just $\sum_{l=1}^{n} g_{l,j} x_l^{i-1} y_l = 0$, so we represent the equation $V_r(x,y)\mathbf{M} = 0$ as the following system of rk equations in $2n$ unknowns.

$$\left\{ \sum_{l=1}^{n} g_{l,j} x_l^{i-1} y_l = 0 : i \in \{1, \ldots, r\}, j \in \{1, \ldots, k\} \right\} \tag{2.1}$$

We can see the equivalence of $V_r(x,y)\mathbf{M} = 0$ and $\mathbf{M}^\intercal V_r(x,y)^\intercal = 0$ by taking the transpose of both sides. This latter equation gives us the following system.

$$\begin{cases} \mathbf{M}^\intercal \begin{bmatrix} y_1 & y_2 & \ldots & y_n \end{bmatrix}^\intercal & = 0 \\ \mathbf{M}^\intercal \begin{bmatrix} y_1 x_1 & y_2 x_2 & \ldots & y_n x_n \end{bmatrix}^\intercal & = 0 \\ & \vdots \\ \mathbf{M}^\intercal \begin{bmatrix} y_1 x_1^{r-1} & y_2 x_2^{r-1} & \ldots & y_n x_n^{r-1} \end{bmatrix}^\intercal & = 0 \end{cases}$$

From the first block $\mathbf{M}^\intercal y = 0$, we have k linear equations in n unknowns. Given that $rank_{\mathbb{F}_{p^m}}(\mathbf{M}) \leq k$ (since the columns of $\mathbf{M}$ are only guaranteed to be $\mathbb{F}_p$-linearly independent), we have that $d = \dim_{\mathbb{F}_{p^m}}(ker(\mathbf{M}^\intercal)) \geq n - k$. Often, d will much significantly less than n. Since y belongs to the kernel of $\mathbf{M}^\intercal$, we can describe y with respect to $d \leq n$ variables that represent its coordinates with respect to a basis for the kernel. These coordinates define the projection of y onto the kernel and we will denote the vector with these coordinates by $y' \in \mathbb{F}_{p^m}^d$. As soon as we identify y' we can simplify system (2.1) and obtain the following system.

$$\left\{ \sum_{l=1}^{n} g'_{l,j} x_l^{i-1} = 0 : i \in \{1, \ldots, r\}, j \in \{1, \ldots, k\} \right\} \tag{2.2}$$

Here, $g'_{l,j}$ denotes the product $g_{l,j} y_l$ and this is known since identifying y' reveals y. System (2.2) is a consistent homogeneous system involving polynomials whose monomial terms are all univariate and it is far easier to solve than system (2.1). What we want in solving system (2.2) is the set of common zeros of the set of polynomials appearing in this system, which we denote by $F = \{\sum_{l=1}^{n} g'_{l,j} X_l^{i-1} : i = 1, \ldots, r, j = 1, \ldots, k\}$. The set of common zeros is also called the *variety* of F and it is defined as

$$\mathcal{V}(F) := \{(z_1, \ldots, z_n) \in \mathbb{F}_{p^m}^n : f(z_1, \ldots, z_n) = 0 \ \forall f \in F\}.$$

Identifying $\mathcal{V}(F)$ can be done using Gröbner basis techniques, which the interest reader can begin exploring in [Stu].

The hardest part of the attack, computationally speaking, is the identification of y', which can again be done with Gröbner bases by a method described in [F1].

With this task effectively being the bottleneck for attack, we can consider accomplishing this as our de facto goal. We will give a sense of the weakness of schemes based on symmetric codes by showing how much more easily this goal is achieved when considering the folded code.

Consider now the attack on the folded code $\overline{A_r(\alpha,\beta)}^{\mathbb{G}}$. The length and dimension of the folded code are $\frac{n}{l}$ and $\frac{k}{l}$, respectively, where l is the size of the permutation group of $A_r(\alpha,\beta)$, denoted $\mathbb{G}$. To show how much attacking the folded code helps in expediting the process of recovering equivalent code parameters, suppose $A_r(\alpha,\beta)$ is quasi-monoidic, so $\mathbb{G} \cong (\mathbb{Z}/p\mathbb{Z})^{\lambda}$ for some $\lambda \leq n$. We therefore have that $l = |\mathbb{G}| = p^{\lambda}$, so the system used to identify y' will have both its number of equations and unknowns reduced exponentially if we choose to attack $\overline{A_r(\alpha,\beta)}^{\mathbb{G}}$ instead of $A_r(\alpha,\beta)$, which greatly reduces the computational effort needed to recover equivalent parameters.

For reference, a $(8192, 4096)$ binary, quasi-monoidic Goppa code whose permutation group is isomorphic to $(\mathbb{Z}/2\mathbb{Z})^7$ can be folded into a $(64, 32)$ binary Goppa code. While a Goppa code possessing the length and dimension of this full code should offer 256-bit security for a McEliece scheme based on it (as reported in [B]), if we considered trying to identify y' in the context of this folded code, we are trying to find a binary vector of length between $64 - 32 = 32$ and 64. Even in the worst case where we are guessing its coordinates, it will require no more than $2^{64} << 2^{256}$ guesses to identify it. Given that this is the hardest part of the attack, attacking the folded code (even by brute force) provides a substantial speed-up over attacking the original code traditionally.

References

[B] P.S.L.M. Barreto, R. Lindner, R. Misoczki, "Monoidic Codes in Cryptography," in Yang BY. (ed.) Post-Quantum Cryptography, PQCrypto 2011. Lecture Notes in Computer Science, **7071**: 179-199. Springer, Berlin, Heidelberg, 2011.

[Be] D. J. Bernstein, T. Chou, T. Lange, I. V. Mauri, R. Misoczki, R. Niederhagen, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, J. Szefer, W. Wang, "Classic McEliece: conservative code-based cryptography," NIST PQC Competition, 2019.

[F1] J. Faugère, A. Otmani, L. Perret , J. Tillich, "Algebraic Cryptanalysis of McEliece Variants with Compact Keys," Eurocrypt 2010 - 29th International Conference on Cryptology, 279-298. Monaco, Monaco, 2010.

[F2] J. Faugère, A. Otmani, L. Perret , F. de Portzamparc, J. Tillich, "Folding Alternant and Goppa Codes with Non-Trivial Automorphism Groups," IEEE Transactions on Information Theory, **62**(1): 184-198, 2016.

[F3] —, "Structural cryptanalysis of McEliece schemes with compact keys", Designs, Codes, and Cryptography, **79**: 87–112, 2016.

[MS] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North Holland, 1978.

[M] R. J. McEliece, "A public key cryptosystem based on algebraic coding theory", DSN Progress Report 44, 1978.

[N] M. Nevins, *MAT 3743: Algèbre linéaire appliquée*, lecture notes, University of Ottawa, delivered 14 March 2020.

[G] P. Gaborit, "Shorter keys for code based cryptography," in Proceedings of the 2005 International Workshop on Coding and Cryptography, 81-91. ACM Press, Bergen, Norway, 2005.

[S] N. Sendrier, "The Support Splitting Algorithm," Research Report 3637, INRIA, 1999. Available at https://hal.inria.fr/inria-00073037.

[St] F. Stojanovic, "A Consideration of Attacks and Theory in Code-Based Cryptography," 2020.

[Stu] B. Sturmfels, "What is . . . a Gröbner basis?" Notices of the American Mathematical Society, **52**(10):1199-1200, 2005.

Appendix

Basic Properties of Alternant Codes. We will present some of the basic properties given in Ch.12 §2 of [MS]. Let's first recall that we define an Alternant code as the subfield subcode of a $\mathbb{F}_{p^m}$-linear GRS code. Thus, the parameters used to define an Alternant code are $\alpha, \beta \in \mathbb{F}_{p^m}^n$ such that the coordinates of α are distinct and the coordinates of β are non-zero. Given these parameters, we define the Alternant code $A_r(\alpha, \beta)$ as $GRS_{n,n-k}(\alpha, \beta)^\perp \cap \mathbb{F}_p^n$. This somewhat unintuitive definition is used so that the notation $A_r(\alpha, \beta)$ can convey information about the canonical parity-check matrix of $A_r(\alpha, \beta)$.

Since $A_r(\alpha, \beta)$ is contained in $GRS_{n,n-k}(\alpha, \beta)^\perp$, a parity-check matrix $\mathbf{H}$ for $GRS_{n,n-k}(\alpha, \beta)^\perp$ also has the property that $\mathbf{H}c = 0$ for all $c \in A_r(\alpha, \beta)$. One choice for $\mathbf{H}$ is the transpose of the canonical form of a generator matrix for $GRS_{n,n-k}(\alpha, \beta)$, which has the following form.

$$\mathbf{H} = \begin{bmatrix} \beta_1 & \beta_2 & \dots & \beta_n \\ \beta_1\alpha_1 & \beta_2\alpha_2 & \dots & \beta_n\alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \beta_1\alpha_1^{n-k-1} & \beta_2\alpha_2^{n-k-1} & \dots & \beta_n\alpha_n^{n-k-1} \end{bmatrix}$$

The *degree* of $A_r(\alpha, \beta)$ is the number of rows of $\mathbf{H}$, so we have $r = n - k$. This is also the codimension of $GRS_{n,n-k}(\alpha, \beta)^\perp$ given that

$$\dim_{\mathbb{F}_{p^m}}(GRS_{n,n-k}(\alpha, \beta)^\perp) = n - \dim_{\mathbb{F}_{p^m}}(GRS_{n,n-k}(\alpha, \beta)) = n - (n - k) = k.$$

Given that $\mathbf{H}$ is a parity-check matrix for $GRS_{n,n-k}(\alpha, \beta)^\perp$, we can see that $A_r(\alpha, \beta) = ker(\mathbf{H}) \cap \mathbb{F}_p^n$. For this reason, we will say that $\mathbf{H}$ is a parity-check matrix over $\mathbb{F}_{p^m}$ for $A_r(\alpha, \beta)$, but to get a "true" parity-check matrix for $A_r(\alpha, \beta)$, we need a matrix to represent a $\mathbb{F}_p$-linear map from $\mathbb{F}_p^n$ whose kernel is $A_r(\alpha, \beta)$. We precisely get such a matrix by writing the entries of $\mathbf{H}$ as column vectors of length m over $\mathbb{F}_p$, i.e. we apply to each entry of $\mathbf{H}$ the coordinate isomorphism from $\mathbb{F}_{p^m}$ to $\mathbb{F}_p^m$ with respect to a basis of $\mathbb{F}_{p^m}$ over $\mathbb{F}_p$. Denote this matrix by $\overline{\mathbf{H}}$.

We know the rank of $\overline{\mathbf{H}}$ is at most mr, so by the Dimension Theorem, we conclude $\dim_{\mathbb{F}_p}(ker(\overline{\mathbf{H}})) \geq n - rm$. But since $ker(\overline{\mathbf{H}}) = A_r(\alpha, \beta)$, if we denote the dimension of $A_r(\alpha, \beta)$ by k_A, we obtain

$$k_A \geq n - rm \iff r \geq \frac{n - k_A}{m}.$$

This covers the basic properties of Alternant codes mentioned in Section 2 and that one might see in related literature.