# A CONSIDERATION OF ATTACKS AND THEORY IN CODE-BASED CRYPTOGRAPHY

FILIP STOJANOVIC

Abstract. The McEliece public-key cryptosystem (PKC) based on binary, irreducible Goppa codes (known as Classic McEliece in its submission to the National Institute of Standards and Technology's post-quantum cryptography standardization project) is one of the most promising cryptosystems in post-quantum cryptography with it admitting no considerable speed-up from quantum attacks, as catalogued in [B]. We consider several structural attacks against McEliece schemes based on both Goppa codes and Generalized Reed-Solomon (GRS) codes, identifying when these attacks may threaten Classic McEliece. We notably extend the Sidelnikov-Shestakov attack to show that it is also successful on any McEliece scheme based on full-rank Goppa codes. We also collect a number of results about $\mathbb{F}_p$-linear subcodes of $\mathbb{F}_{p^m}$-linear codes and more specific results on GRS and Goppa codes. Ultimately, we use them to motivate a new approach for a key-recovery attack on the McEliece scheme based on Goppa codes.

## Acknowledgements

I want to offer my sincerest thanks to Dr. Monica Nevins both for providing me many opportunities to deepen my knowledge and interest for math over many years and for helping me find those early feelings of triumph in the face of challenging math problems that all mathematicians can invariably cite as a principal motivation for their decision to study what they love. Furthermore, her generous donation of her time, insight, and direction enriched the contents of this project greatly and made working on it an incredibly rewarding experience. I also wish to thank NSERC for providing me with the occasion to study this material through their support via an Undergraduate Student Research Award.

## 1. Introduction

In opposition to the threat of quantum computers on PKCs, the National Institute of Standards and Technology launched its initiative to find classically-secure and quantum-resistant PKCs. Classic McEliece has been identified as one of the most promising post-quantum cryptosystems by this initiative. In order to study its security, we present the requisite coding theory needed to describe the McEliece PKC in Part 1 of this article, and we describe attacks against it in Part 2.

In Part 1, we study the properties of subfield subcodes and trace codes, $\mathbb{F}_p$-linear subcodes of $\mathbb{F}_{p^m}$-linear codes, notably deriving a precise construction for each using subspaces of $\mathbb{F}_{p^m}^n$ that are invariant under the Galois group $\mathrm{Gal}(\mathbb{F}_{p^m}, \mathbb{F}_p)$. More generally, we also relate properties of $\mathbb{F}_p$-linear codes to those of their $\mathbb{F}_{p^m}$-linear

*Date*: January 22, 2021.

extensions. We define Goppa and GRS codes and we precisely describe the relationship of Goppa codes as subfield subcodes of GRS codes, consequently using it to characterize Goppa codes using the polynomial-evaluation definition of GRS codes.

In Part 2, we describe the McEliece PKC as well as the two main classes of attacks against it. We rely on results from [TS] to confirm the intractability of message attacks on the McEliece PKC and use it as motivation to consider the efficiency of structural attacks instead. We present several structural attacks against the McEliece PKC based on GRS codes and subcodes thereof proposed by Sidelnikov and Shestakov in [SS] and Wieschebrink in [W]. We note when it is possible to extend these attacks to efficient attacks against the McEliece PKC based on Goppa codes, notably identifying that the attack proposed by Sidelnikov and Shestakov applies in exactly the same way to a McEliece PKC based on full-rank Goppa codes as it does to one based on GRS codes. We illustrate why this attack does not extend to lower-dimensional Goppa codes and identify what information is lost due to this drop in dimension in the binary case. We ultimately propose a new approach for a key-recovery attack on the McEliece PKC based on Goppa codes motivated by results collected in Part 1. Its difficulty lies in the problem of identifying a linear code given its Galois closure and we derive results about the relationship between the parameters of GRS codes with the same Galois closure to ascertain the difficulty of this problem. We also consider a modification to this approach using punctured codes, which leads us to deriving a sufficient condition in the parameters of a GRS for when its subfield subcode is of full rank. In considering this new approach, we mention several open problems for future work.

1.1. **Prerequisites.** This material should be accessible to anyone who has taken a first course in coding theory. The particular topics one should be comfortable with are linear error-correcting codes, finite fields, and asymmetric cryptography. Sections 1, 2, 3.3, and 4.2 in [N] would be the best places to consult for familiarizing oneself with this material. Alternatives would be Sections 1, 2, and 3 in [Ro] or Sections 1, 3, 4 in [MS] for background on linear error-correcting codes and finite fields. The requisite familiarity with asymmetric cryptography can be acquired through a quick skim of the Wikipedia page on public-key cryptography.

1.2. **Statement on Notation.** With apologies to the computer scientists, we will write vectors as column vectors. We summarize certain other notable notation choices below.

- $\mathbb{F}_{p^m}$ denotes the finite field of size $p^m$ where $p$ is prime and $m \in \mathbb{N}_+$.
- A $(n, k)$ $\mathbb{F}$-linear code $C$ denotes a $k$-dimensional subspace of $\mathbb{F}^n$.
- $\omega : \mathbb{F}_{p^m}^n \to \mathbb{N}$ is used to denote the *Hamming weight*.
- $\mathbb{P}_k(\mathbb{F}_{p^m})$ denotes the vector space of polynomials of degree at most $k$ with coefficients in $\mathbb{F}_{p^m}$.
- If $\mathbf{M}$ is a matrix in $\mathcal{M}_{n,k}(\mathbb{F})$, then we denote the $i^{th}$ column of $\mathbf{M}$ by $\mathbf{M}_i$.

## Part 1. Properties of GRS, Goppa, and Alternant Codes

### 2. Subfield Subcodes and Trace Codes

For a linear code defined over an extension of a finite field, we will give the construction of two important classes of linear codes that we can derive from such a code and that will be vector spaces linear over the base field. We will prove certain properties of subfield subcodes and trace codes, in particular giving bounds on their dimensions. We will also give equivalent characterizations of subfield subcodes and trace codes in terms of invariant vector spaces under a particular Galois group.

2.1. **Preliminary Theory.** We begin first by presenting the material from Galois theory needed to understand the construction of subfield subcodes and trace codes as well as develop a particularly interesting equivalent characterization for each.

We begin by outlining a few important properties of the primary tool we will use to study $\mathbb{F}_p$-linear subcodes of $\mathbb{F}_{p^m}$-linear codes, the *Frobenius map*.

**Definition 2.1.1.** The *Frobenius map* is $\phi : \mathbb{F}_{p^m} \to \mathbb{F}_{p^m}$ given by $x \mapsto x^p$.

We may extend this map to $n$-tuples over $\mathbb{F}_{p^m}$. In this case, we label it by $\phi_n : \mathbb{F}_{p^m}^n \to \mathbb{F}_{p^m}^n$, and, naturally, it maps $(x_i)_{i=1}^n \mapsto (x_i^p)_{i=1}^n$.

**Lemma 2.1.2.** *Let $x \in \mathbb{F}_{p^m}$. We have $\phi(x) = x$ if and only if $x \in \mathbb{F}_p$.*

*Proof.* Notice that

$$\phi(x) = x^p = x \iff x(x^{p-1} - 1) = 0.$$

This latter condition holds if and only if $x = 0$ or $x \in \mu_{p-1}$. Immediately, we recognize since the set of the $p-1$ roots of unity is $\mu_{p-1} = \mathbb{F}_p^\times$, we have $\phi(x) = x$ if and only if $x \in \mathbb{F}_p^\times \cup \{0\} = \mathbb{F}_p$, as required. □

**Lemma 2.1.3.** *$\phi_n$ is $\mathbb{F}_p$-linear.*

*Proof.* Let $x, y \in \mathbb{F}_{p^m}^n$ and let $\alpha, \beta \in \mathbb{F}_p$.
We will show $\phi_n(\alpha x + \beta y) = \alpha\phi_n(x) + \beta\phi_n(y)$. Suppose $\alpha, \beta \neq 0$ because the result is trivial in this case. For any choice of $i \in \{1, \ldots, n\}$, we have

$$\begin{aligned}
(\phi_n(\alpha x + \beta y))_i &= \phi(\alpha x_i + \beta y_i) \\
&= \sum_{k=0}^{p} \binom{p}{k} \alpha^k x_i^k \beta^{p-k} y^{p-k} \\
&= \alpha^p x_i^p + \beta^p y_i^p \quad \text{since } \operatorname{char}(\mathbb{F}_{p^m}) = p \\
&= \phi(\alpha)\phi(x_i) + \phi(\beta)\phi(y_i) \\
&= \alpha\phi(x_i) + \beta\phi(y_i) \quad \text{by Lemma 2.1.2.}
\end{aligned}$$

Since this holds for each coordinate of $\phi_n(\alpha x + \beta y)$, we get the result. □

**Remark 2.1.4.** Since $\phi_n$ is the coordinate-wise application of $\phi$ to a $n$-tuple over $\mathbb{F}_{p^m}$, it follows immediately from this last lemma that for all $x \in \mathbb{F}_{p^m}^n$, we have $\phi_n(x) = x$ if and only if $x \in \mathbb{F}_p^n$.

**Definition 2.1.5.** Let $\mathbb{G}$ be a field and let $\mathbb{F}$ be a subfield of $\mathbb{G}$. The *Galois group* $\operatorname{Gal}(\mathbb{G}, \mathbb{F})$ is the group of all field automorphism of $\mathbb{G}$ such that for any map $\tau \in \operatorname{Gal}(\mathbb{G}, \mathbb{F})$ and for any element $x \in \mathbb{F}$, we have $\tau(x) = x$.

For notational brevity, we will denote $\mathrm{Gal}(\mathbb{F}_{p^m}, \mathbb{F}_p)$ by $\mathrm{Gal}(p^m, p)$. Furthermore, $\mathrm{Gal}(p^m, p)$ is a cyclic group of order $m$ generated by the Frobenius map $\phi$. A proof of this comes from Theorem 4.12 in [R].

**Definition 2.1.6.** Let $C$ be a subspace of $\mathbb{F}_{p^m}^n$. $C$ is said *Gal$(p^m, p)$-invariant* if for all vectors $c \in C$ and for any map $\tau \in \mathrm{Gal}(p^m, p)$, we have $\tau_n(c) \in C$ where $\tau_n : \mathbb{F}_{p^m}^n \to \mathbb{F}_{p^m}^n$ is the map whose coordinate-wise action on $c$ is $\tau$. Since $\tau$ is an automorphism, we will have $\tau_n(C) = C$.

We will characterize when a linear code is $\mathrm{Gal}(p^m, p)$-invariant.

**Lemma 2.1.7.** *Let $C$ be a subspace of $\mathbb{F}_{p^m}^n$. $C$ is Gal$(p^m, p)$-invariant if and only if $\phi_n(c) \in C$ for all $c \in C$.*

*Proof.* The forwards implication follows immediately from noticing $\phi \in \mathrm{Gal}(p^m, p)$. For the converse direction, let $c \in C$ be given.

$$\phi_n(c) \in C \implies \phi_n \circ \phi_n(c) \in C \implies \phi_n \circ \cdots \circ \phi_n(c) \in C$$

For any $r \in \{1, \ldots, m\}$, we may compose $\phi_n$ with itself $r$ times and get $\phi_n^r(c) \in C$. Since $\mathrm{Gal}(p^m, p)$ is the cyclic group of order $m$ generated by $\phi$, for all $\tau \in \mathrm{Gal}(p^m, p)$,

$$\exists r \in \{1, \ldots, m\} \text{ such that } \tau = \phi^r.$$

Hence, $\tau_n(c) = \phi_n^r(c) \in C$, as required. □

In fact, this last characterization remains true even when it's just a basis of $C$ that's contained in $C$ after the application of $\phi_n$. This characterization comes from Lemma 5.2.15 in [P].

**Lemma 2.1.8.** *Let $C$ be a $(n, k)$ $\mathbb{F}_{p^m}$-linear code and let $B$ be a basis for $C$. $C$ is Gal$(p^m, p)$-invariant if and only if $\phi_n(b) \in C$ for all $b \in B$.*

*Proof.* See [P]. □

**Definition 2.1.9.** Let $C$ be a subspace of $\mathbb{F}_{p^m}^n$. We will define the *Galois interior* and *Galois closure* of $C$, as

$$C^0 := \cap_{i=1}^m \phi_n^i(C) \text{ and}$$
$$C^* := \sum_{i=1}^m \phi_n^i(C), \text{ respectively.}$$

**Proposition 2.1.10.** *Let $C$ be a subspace of $\mathbb{F}_{p^m}^n$ and consider its Galois interior.*

(a) *$C^0$ is Gal$(p^m, p)$-invariant.*
(b) *$C$ is Gal$(p^m, p)$-invariant if and only if $C = C^0$.*
(c) *For any subspace $D \subseteq C$ such that $D$ is Gal$(p^m, p)$-invariant, we have $D \subseteq C^0$. In other words, $D$ is the largest Gal$(p^m, p)$-invariant subspace of $C$.*

*Proof.* We'll prove each item separately.

(a) Let $c \in C^0$, so $c \in \phi_n^s(C)$ for all $s = 0, \ldots, m-1$. Thus, for any $s \in \{0, \ldots, m-1\}$, there exists a vector $c_s' \in C$ such that $c = \phi_n^s(c_s')$. By Lemma 2.1.7, we just need to show $\phi_n(c) \in C^0$. We have $\phi_n(c) = \phi_n^{s+1}(c_s')$. But since $\mathrm{Gal}(p^m, m)$ is

a cyclic group of order $m$ generated by $\phi$, for any integer $r \in \mathbb{N}$, we see that $\phi_n^r = \phi_n^{r_m}$ such that $r_m \equiv r (\text{mod } m)$. Thus, it's clear to see that

$$\phi_n(c) = \begin{cases} \phi_n^s(c'_{s-1}) & s = 1, \ldots, m-1 \\ \phi_n^0(c'_{m-1}) & s = 0 \end{cases}$$

Thus, $\phi_n(c) \in \cap_{i=1}^m \phi_n^i(C) = C^0$, as required.

(b) The converse direction follows immediately from $C^0$ being $\text{Gal}(p^m, p)$-invariant. Let $C$ be $\text{Gal}(p^m, p)$-invariant and let $c \in C$ be given. By Lemma 2.1.7, we have that $\phi_n(c) \in C$. By successively applying Lemma 2.1.7, we get

$$\phi_n(c) \in C \implies \phi_n^2(c) \in C \implies \cdots \implies \phi_n^m(c) \in C.$$

Hence, the above implications give $\phi_n^i(C) \subseteq C$ for all $i = 1, \ldots, m$. Now, for all $i \in \{1, \ldots, n\}$ set $c_i := \phi_n^{m-i}(c)$. The above implications guarantees that $c_i \in C$. We therefore have

$$c = \phi_n^i(c_i) \implies c \in \phi_n^i(C) \ \forall i = 1, \ldots, m.$$

This means $C \subseteq \phi_n^i(C)$ for all $i = 1, \ldots, m$, which allows us to conclude using our previous inclusion that $\phi_n^i(C) = C$ for all $i = 1, \ldots, m$. Finally, we can conclude $C = \cap_{i=1}^m \phi_n^i(C) = C^0$, as required.

(c) Let $D \subseteq C$ be a subspace such that $D$ is $\text{Gal}(p^m, p)$-invariant. By part (b), this means $D = D^0$. However, because $D \subseteq C$, we get

$$D = \cap_{i=1}^m \phi_n^i(D) \subseteq \cap_{i=1}^m \phi_n^i(C) = C^0, \text{ as required.}$$

□

**Proposition 2.1.11.** *Let $C$ be a subspace of $\mathbb{F}_{p^m}^n$ and consider its Galois closure.*
(a) *$C^*$ is $Gal(p^m, p)$-invariant.*
(b) *$C^* \supseteq C$ and for all vector spaces $D \supseteq C$ such that $D$ is $Gal(p^m, p)$-invariant, we have $D \supseteq C^*$. In other words, $C^*$ is the smallest $Gal(p^m, p)$-invariant vector space containing $C$.*

*Proof.* Again, we'll prove both parts separately.

(a) By Lemma 2.1.7, we need only prove $\phi_n(C^*) \subseteq C^*$.
Since $C^* = \{c_1 + \cdots + c_m : c_i \in \phi_n^i(C) \text{ for all } i = 1, \ldots, m\}$, if we take $c \in C^*$, then there exist $c_1, \ldots, c_m \in C$ such that $c = \sum_{i=1}^m \phi_n^i(c_i)$. Let $c$ be as just described.

$$\phi_n(c) = \phi_n\left(\sum_{i=1}^m \phi_n^i(c_i)\right) = \sum_{i=1}^m \phi_n^{i+1}(c_i) \quad \text{by the } \mathbb{F}_p\text{-linearity of } \phi_n$$

Notice that for all $i = 1, \ldots, m-1$, we have $\phi_n^{i+1}(c_i) \in \phi_n^{i+1}(C)$ since $c_i \in C$. Since $\langle \phi \rangle$ is a cyclic group of order $m$, we also have $\phi_n^{m+1}(c_m) = \phi_n(c_m) \in \phi_n(C)$. Hence, $\phi_n(c) \in C^*$, as required.

(b) Since $\phi_n^m = id$, $C^* = C + \phi_n(C) + \cdots + \phi_n^{m-1}(C)$ clearly contains $C$. Now, suppose $D$ is a $\text{Gal}(p^m, p)$-invariant vector space such that $D \supseteq C$. This means for all vectors $d \in D$, we have $\phi_n^i(d) \in D$ for all $i = 1, \ldots, m$.

Let $b_1, \ldots, b_m \in C \subseteq D$ be given. We have $\phi_n^i(b_i) \in D$ for all $i = 1, \ldots, m$. But since $D$ is a vector space, we have $\sum_{i=1}^m \phi_n^i(b_i) \in D$. This vector is just an arbitrary element of $C^*$; thus, we conclude $C^* \subseteq D$.

□

**Definition 2.1.12.** Let $C$ be a $\mathbb{F}_p$-linear code. The *extension by scalars* of $C$ is the $\mathbb{F}_{p^m}$-linear code $\text{span}_{\mathbb{F}_{p^m}}(C)$. We denote this code by $C \otimes \mathbb{F}_{p^m}$.

We will now establish a few results that relate a $\mathbb{F}_p$-linear code to its extension by scalars over $\mathbb{F}_{p^m}$.

**Lemma 2.1.13.** *Let $C$ be a $(n, k)$ $\mathbb{F}_p$-linear code and let $\{b_1, \ldots, b_k\}$ be a basis for $C$. The set $\{b_1, \ldots, b_k\}$ will also be a basis for the code $C \otimes \mathbb{F}_{p^m}$.*

*Proof.* We begin first by showing $\{b_1, \ldots, b_k\}$ spans $C \otimes \mathbb{F}_{p^m}$.

$$\begin{aligned}
C \otimes \mathbb{F}_{p^m} &= \left\{ \sum_{c \in C} \lambda_c c : \lambda_c \in \mathbb{F}_{p^m} \right\} \\
&= \left\{ \sum_{c \in C} \lambda_c \left( \sum_{i=1}^k \gamma_i b_i \right) : \gamma_i \in \mathbb{F}_p, \;\; \lambda_c \in \mathbb{F}_{p^m} \right\} \\
&= \left\{ \sum_{i=1}^k \left( \sum_{c \in C} \gamma_i \lambda_c \right) b_i : \gamma_i \in \mathbb{F}_p, \;\; \lambda_c \in \mathbb{F}_{p^m} \right\} \\
&= \text{span}_{\mathbb{F}_{p^m}} \{b_1, \ldots, b_k\}
\end{aligned}$$

To prove the set's linear independence over $\mathbb{F}_{p^m}$, we first let $\{\alpha_1, \ldots, \alpha_m\}$ be a basis for $\mathbb{F}_{p^m}$ over $\mathbb{F}_p$. Next, let $\lambda_1, \ldots, \lambda_k \in \mathbb{F}_{p^m}$ such that $\lambda_i = \sum_{j=1}^m a_{i,j} \alpha_j$. Consider the equation $\sum_{i=1}^k \lambda_i b_i = 0$ and notice that it lets us develop the following.

$$\begin{aligned}
&\iff \sum_{i=1}^k \left( \sum_{j=1}^m a_{i.j} \alpha_j \right) b_i = 0 \\
&\iff \sum_{i=1}^k a_{i,1} \alpha_1 b_i \;\; + \cdots + \;\; \sum_{i=1}^k a_{i,m} \alpha_m b_i = 0 \\
&\iff \begin{bmatrix} \sum_{i=1}^k a_{i,1} (b_i)_1 \alpha_1 & + \cdots + & \sum_{i=1}^k a_{i,m} (b_i)_1 \alpha_m \\ & \vdots & \\ \sum_{i=1}^k a_{i,1} (b_i)_n \alpha_1 & + \cdots + & \sum_{i=1}^k a_{i,m} (b_i)_n \alpha_m \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \\
&\implies \sum_{i=1}^k a_{i,j} (b_i)_r = 0 \;\; \forall j = 1, \ldots, m, \;\; \forall r = 1, \ldots, n \\
&\iff \sum_{i=1}^k a_{i,j} b_i = 0 \;\; \forall j = 1, \ldots, m \\
&\implies a_{i,j} = 0 \;\; \forall i, j \;\; \text{by the linear independence over } \mathbb{F}_p \text{ of } \{b_1, \ldots, b_k\}
\end{aligned}$$

Hence, $\lambda_1, \ldots, \lambda_k = 0$, so $\{b_1, \ldots, b_k\}$ is linearly independent over $\mathbb{F}_{p^m}$. Thus, it is a basis for $C \otimes \mathbb{F}_{p^m}$. □

**Corollary 2.1.14.** *If $C$ is a $(n,k)$ $\mathbb{F}_p$-linear code, then*

$$\dim_{\mathbb{F}_p}(C) = \dim_{\mathbb{F}_{p^m}}(C \otimes \mathbb{F}_{p^m}).$$

*Proof.* This is a direct consequence of the previous lemma. □

The operations of extending a $\mathbb{F}_p$-linear code by scalars and of intersecting a $\mathbb{F}_{p^m}$-linear code with $\mathbb{F}_p^n$ allow us to describe another equivalent characterization for when a code is $\mathrm{Gal}(p^m, p)$-invariant.

**Lemma 2.1.15.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. $C \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m} = C$ if and only if $\dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) = k$.*

*Proof.* It is clear that $C \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m} \subseteq C$. Thus,

$$\begin{aligned} C \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m} = C &\iff \dim_{\mathbb{F}_{p^m}}(C \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m}) = \dim_{\mathbb{F}_{p^m}}(C) \\ &\iff \dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) = k \quad \text{by Corollary 2.1.14.} \end{aligned}$$

□

The following theorem given in [GP] provides us with an equivalent characterization for the $\mathrm{Gal}(\mathbb{F}_{p^m}, \mathbb{F}_p)$-invariance of a code.

**Theorem 2.1.16.** *Let $C$ be a $\mathbb{F}_{p^m}$-linear code. $C$ is $Gal(p^m, p)$-invariant if and only if $C = C \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m}$. This is equivalent to having $C$ admit a basis in $\mathbb{F}_p^n$.*

*Proof.* See [GP]. □

**Corollary 2.1.17.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. $C$ is $Gal(p^m, p)$-invariant if and only if $C \cap \mathbb{F}_p^n$ is of maximal rank.*

*Proof.* Successively applying Theorem 2.1.16 and Lemma 2.1.15 gives

$$C \text{ is } \mathrm{Gal}(p^m, p)\text{-invariant} \iff \dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) = k,$$

which by Proposition 2.3.1 is the maximal dimension of $C \cap \mathbb{F}_p^n$. □

Lastly before proceeding, we write a lemma to understand the interplay between the actions $\cap\mathbb{F}_p^n$ and $\otimes\mathbb{F}_{p^m}$ we can perform on linear codes.

**Lemma 2.1.18.** *If $C$ is a $(n,k)$ $\mathbb{F}_p$-linear code, then $C = C \otimes \mathbb{F}_{p^m} \cap \mathbb{F}_p^n$.*

*Proof.* Let $B = \{b_1, \ldots, b_k\}$ be a basis for $C$. By Lemma 2.1.13, $B$ is also a basis for $C \otimes \mathbb{F}_{p^m}$. Immediately, we will note that because $C = C \cap \mathbb{F}_p^n$, we have $C \subseteq C \otimes \mathbb{F}_{p^m} \cap \mathbb{F}_p^n$.

We will next show that $C \otimes \mathbb{F}_{p^m} \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m} = C \otimes \mathbb{F}_{p^m}$. By our previous observation, the $\supseteq$ inclusion is clear. For the inclusion the other way, let $s \in \mathbb{N}$ be given and let $d_1, \ldots, d_s$ be codewords of $C \otimes \mathbb{F}_{p^m} \cap \mathbb{F}_p^n$. This means that for all $j = 1, \ldots, s$, there exist scalars $\lambda_{j,1}, \ldots, \lambda_{j,k} \in \mathbb{F}_{p^m}$ such that $d_j = \sum_{i=1}^k \lambda_{j,i} b_i$ and $d_j \in \mathbb{F}_p^n$. Thus, for all $\gamma_1, \ldots, \gamma_s \in \mathbb{F}_{p^m}$, we have

$$\sum_{j=1}^{s} \gamma_j d_j = \sum_{j=1}^{s} \sum_{i=1}^{k} \gamma_j \lambda_{j,i} b_i \in \mathrm{span}_{\mathbb{F}_{p^m}}(B) = C \otimes \mathbb{F}_{p^m}.$$

Hence, $C \otimes \mathbb{F}_{p^m} = C \otimes \mathbb{F}_{p^m} \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m}$.

By Corollary 2.1.14, we get $\dim_{\mathbb{F}_{p^m}}(C \otimes \mathbb{F}_{p^m}) = \dim_{\mathbb{F}_p}(C \otimes \mathbb{F}_{p^m} \cap \mathbb{F}_p^n)$. But this means $\dim_{\mathbb{F}_p}(C) = \dim_{\mathbb{F}_p}(C \otimes \mathbb{F}_{p^m} \cap \mathbb{F}_p^n)$, which after recalling that we found that $C \subseteq C \otimes \mathbb{F}_{p^m} \cap \mathbb{F}_p^n$, means we can conclude $C = C \otimes \mathbb{F}_{p^m} \cap \mathbb{F}_p^n$. □

Note that $\cap\mathbb{F}_p^n$ and $\otimes\mathbb{F}_{p^m}$ are not each other's inverses. We've shown $C \otimes \mathbb{F}_{p^m} \cap \mathbb{F}_p^n = C$, but it is not generally true $C \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m} = C$. By Theorem 2.1.16, the latter is true if and only if C is $\mathrm{Gal}(p^m, p)$-invariant.

2.2. **Construction of Subfield Subcodes and Trace Codes.** We will now introduce *subfield subcodes* and *trace codes*, two important classes subcodes that lead to the construction of many families of linear codes, including the cryptographically-relevant family of Goppa codes.

**Definition 2.2.1.** The *trace map* is the map $Tr : \mathbb{F}_{p^m} \to \mathbb{F}_p$ given by

$$x \mapsto x + x^p + \cdots + x^{p^{m-1}} = \sum_{i=1}^{m} \phi^i(x),$$

where $\phi : \mathbb{F}_{p^m} \to \mathbb{F}_{p^m}$ is the Frobenius map.

We will first verify that $Tr(x) \in \mathbb{F}_p$ for all $x \in \mathbb{F}_{p^m}$. This will be done using Lemma 2.1.2, which states that $\phi(x) = x$ if and only if $x \in \mathbb{F}_p$. Let $x \in \mathbb{F}_{p^m}$ be given.

$$\begin{aligned}
\phi(Tr(x)) &= \phi\left(\sum_{i=1}^{m} \phi^i(x)\right) \\
&= \sum_{i=1}^{m} \phi^{i+1}(x) \quad \text{since } \phi \text{ is } \mathbb{F}_p\text{-linear} \\
&= \sum_{i=1}^{m} \phi^i(x) \quad \text{since } \langle\phi\rangle \text{ is a cyclic group of order } m
\end{aligned}$$

Hence, $Tr(x) \in \mathbb{F}_p$. Additionally, it's clear that $\phi$ being $\mathbb{F}_p$-linear implies that $Tr$ is also $\mathbb{F}_p$-linear.

**Lemma 2.2.2.** *$Tr : \mathbb{F}_{p^m} \to \mathbb{F}_p$ is surjective.*

*Proof.* Suppose there exists $x \in \mathbb{F}_{p^m}$ such that $Tr(x) \neq 0$. For any $y \in \mathbb{F}_p$, we observe by the $\mathbb{F}_p$-linearity of $Tr$

$$Tr(yTr(x)^{-1}x) = yTr(x)^{-1}Tr(x) = y.$$

Hence, $Tr$ is surjective if and only of there exists $x \in \mathbb{F}_{p^m}$ such that $Tr(x) \neq 0$. To show the surjectivity of $Tr$, we will show it is not the zero map.

Suppose for a contradiction that for all $x \in \mathbb{F}_{p^m}$, $Tr(x) = 0$. This occurs if and only if

$$\begin{aligned}
\sum_{i=1}^{m} x^{p^i} &= 0 \\
\iff x + x^p + \cdots + x^{p^{m-1}} &= 0 \\
\iff x(1 + x^{p-1} + \cdots + x^{p^{m-1}-1}) &= 0.
\end{aligned}$$

Thus, for all $y \in \mathbb{F}_{p^m}^{\times}$, $y$ is a root of $f(x) := 1 + x^{p-1} + \cdots + x^{p^{m-1}-1}$. But $deg(f) = p^{m-1} - 1$, so it has at most $p^{m-1} - 1$ roots. However, $|\mathbb{F}_{p^m}^{\times}| = p^m - 1 > p^{m-1} - 1$, so it is not possible for all of $\mathbb{F}_{p^m}^{\times}$ to be roots of $f(x)$, a contradiction. □

**Proposition 2.2.3.** *For all $a \in \mathbb{F}_p$, we have $|Tr^{-1}(a)| = p^{m-1}$.*

*Proof.* By the previous lemma, $Im(Tr) = \mathbb{F}_p$. Furthermore, as a vector space over $\mathbb{F}_p$, $\mathbb{F}_{p^m} \cong \mathbb{F}_p^m$. Hence, by the Dimension Theorem,

$$m = \dim_{\mathbb{F}_p}(Im(Tr)) + \dim_{\mathbb{F}_p}(ker(Tr)) \iff m - 1 = \dim_{\mathbb{F}_p}(ker(Tr)).$$

Thus, $|ker(Tr)| = p^{m-1}$. Let $x_0 \in \mathbb{F}_{p^m}$ be a vector such that $Tr(x_0) = a$. But by the $\mathbb{F}_p$-linearity of $Tr$, we get $Tr^{-1}(a) = \{x_0\} + ker(Tr)$. Thus, we conclude $|Tr^{-1}(a)| = p^{m-1}$. □

We will now give an explicit formulation for the kernel of the trace map. It comes as the result of Exercise 4.23 in [R]

**Proposition 2.2.4.** *Let $Tr : \mathbb{F}_{p^m} \to \mathbb{F}_p$ be the trace map as defined previously. We have*

$$ker(Tr) = \{a \in \mathbb{F}_{p^m} : a = \phi(u) - u \text{ for some } u \in \mathbb{F}_{p^m}\}.$$

*Proof.* Define $\tau := \phi - id$ where $id : \mathbb{F}_{p^m} \to \mathbb{F}_{p^m}$ is the identity map.
We will show that $ker(Tr) = Im(\tau)$.

$$\begin{aligned} ker(\tau) &= \{x \in \mathbb{F}_{p^m} : \phi(x) - x = 0\} \\ &= \{x \in \mathbb{F}_{p^m} : \phi(x) = x\} \\ &= \mathbb{F}_p \quad \text{since } \phi(x) = x \iff x \in \mathbb{F}_p \end{aligned}$$

Furthermore, $\tau$ is $\mathbb{F}_p$-linear since $\phi$ is $\mathbb{F}_p$-linear. Since $\mathbb{F}_{p^m} \cong \mathbb{F}_p$ as a vector space over $\mathbb{F}_p$, so by the Dimension Theorem,

$$m = \dim_{\mathbb{F}_p}(Im(\tau)) + \dim_{\mathbb{F}_p}(ker(\tau)) \iff \dim_{\mathbb{F}_p}(Im(\tau)) = m - 1.$$

We recall that in the previous proposition, we found $\dim_{\mathbb{F}_p}(ker(Tr)) = m - 1$ as well. Hence, we need only show $Im(\tau) \subseteq ker(Tr)$. Let $u \in \mathbb{F}_{p^m}$ be given.

$$\begin{aligned} Tr(\tau(u)) &= Tr(\phi(u) - u) \\ &= \sum_{i=1}^{m} \phi^{i+1}(u) - \sum_{i=1}^{m} \phi^i(u) \quad \text{by the } \mathbb{F}_p\text{-linearity of } Tr \\ &= \sum_{i=1}^{m} \phi^i(u) - \sum_{i=1}^{m} \phi^i(u) \quad \text{since order}(\phi) = m \\ &= 0 \end{aligned}$$

Thus, $\tau(u) \in ker(Tr)$, giving us the result. □

The following result comes from Corollary 1.29 in [C1].

**Proposition 2.2.5.** *The map $L(\cdot,\cdot) : \mathbb{F}_{p^m} \times \mathbb{F}_{p^m} \to \mathbb{F}_p$ given by $(a, b) \mapsto Tr(ab)$ is a non-degenerate, bilinear form.*

*Proof.* The bilinearity of $L$ follows from the $\mathbb{F}_p$-linearity of $Tr$. Suppose for a contradiction that there exists $b \in \mathbb{F}_{p^m}^{\times}$ such that $Tr(ab) = 0$ for all $a \in \mathbb{F}_{p^m}$. Since $b \neq 0$, $b^{-1}$ exists. Let $x \in \mathbb{F}_{p^m}$ and take $a = xb^{-1}$.

$$Tr(ab) = Tr(xb^{-1}b) = Tr(x) = 0$$

Hence, $Tr$ is the zero map, but this is a contradiction since $Tr$ is surjective. $\square$

This map allows us to introduce the notion of the *dual basis* to a given basis of $\mathbb{F}_{p^m}$ over $\mathbb{F}_p$. The dual basis becomes an important ingredient in the description of a generating set for a *trace code*, which is about to be introduced. We will simply state the results concerning the dual basis and invite the interested reader to consult [C1] to get a better understanding.

We can also extend the trace map to $n$-tuples over $\mathbb{F}_{p^m}$ as $Tr : \mathbb{F}_{p^m}^n \to \mathbb{F}_p^n$ by $(x_i)_{i=1}^n \mapsto \left(\sum_{j=1}^m \phi^j(x_i)\right)_{i=1}^n$. We'll denote this map in the same way as $Tr : \mathbb{F}_{p^m} \to \mathbb{F}_p$, so we'll use the context to distinguish between these two maps.

We now define the first important class of linear subcodes, the trace code.

**Definition 2.2.6.** Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. The *trace code* of $C$ is defined to be $Tr(C) = \{Tr(c) : c \in C\} = \{\left(\sum_{i=1}^m \phi^i(c_1),\ \ldots,\ \sum_{i=1}^m \phi^i(c_n)\right) : c \in C\}$.

**Definition 2.2.7.** Let $\{\alpha_1, \ldots, \alpha_m\}$ be a basis for $\mathbb{F}_{p^m}$ over $\mathbb{F}_p$. The *dual basis* of $\{\alpha_1, \ldots, \alpha_m\}$ is another basis $\{\alpha_1^*, \ldots, \alpha_m^*\}$ such that for all $i, j \in \{1, \ldots, m\}$,

$$L(\alpha_i, \alpha_j^*) = Tr(\alpha_i \alpha_j^*) = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}.$$

For any given basis, the corresponding dual basis exists and is unique. This is verified in Proposition 1.32 in [C1].

The following result is presented in [C1] and gives us a generating set for a trace code.

**Proposition 2.2.8.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. Let $\{\alpha_1, \ldots, \alpha_m\}$ be a basis for $\mathbb{F}_{p^m}$ over $\mathbb{F}_p$ and let $\{\alpha_1^*, \ldots, \alpha_m^*\}$ be its corresponding dual basis. If $\{b_1, \ldots, b_k\}$ is a basis for $C$, then $\{Tr(b_i \alpha_j^*) : 1 \leq i \leq k, 1 \leq j \leq m\}$ generates $Tr(C)$.*

*Proof.* See [C1]. $\square$

We now define the other important class of linear subcodes, the subfield subcode.

**Definition 2.2.9.** Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. The *subfield subcode* of $C$ is $C \cap \mathbb{F}_p^n$.

The following theorem is due to Delsarte in [D] and it describes the relationship between subfield subcodes and trace codes.

**Theorem 2.2.10** (Delsarte Duality)**.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. We have $(C \cap \mathbb{F}_p^n)^\perp = Tr(C^\perp)$.*

*Proof.* See [D]. $\square$

For a $\mathbb{F}_{p^m}$-linear code $C$, we want to study the intersection of its corresponding trace code and subfield subcode. We first begin by considering when $C$ is $\mathrm{Gal}(p^m, p)$-invariant.

**Proposition 2.2.11.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. We have $C \cap \mathbb{F}_p^n = Tr(C)$ if and only if $C$ is $Gal(p^m, p)$-invariant.*

*Proof.* See [GP]. □

Next, we consider the more general case when $C$ need not be $\mathrm{Gal}(p^m, p)$-invariant.

**Proposition 2.2.12.** *Let $C$ be a $(n, k)$ $\mathbb{F}_{p^m}$-linear code. If $m \not\equiv 0 (mod\ p)$, then $C \cap \mathbb{F}_p^n \subseteq Tr(C)$.*

*Proof.* We first note that the map $Tr : \mathbb{F}_{p^m} \to \mathbb{F}_p$ behaves in an interesting way when applied to an element $x \in \mathbb{F}_{p^m}$. For all $x \in \mathbb{F}_p$, we have

$$Tr(x) = \sum_{i=1}^{m} \phi_i(x) = \sum_{i=1}^{m} x = mx.$$

Define $r$ to be the smallest element of the equivalence class $m \pmod p$, and $r$ is not zero by hypothesis. Let $c := (c_1, \ldots, c_n) \in C \cap \mathbb{F}_p^n$. We observe

$$Tr(c) = (Tr(c_i))_{i=1}^n = (rc_i)_{i=1}^n = rc.$$

By the $\mathbb{F}_p$-linearity of $Tr$, we have $Tr(r^{-1}c) = r^{-1}Tr(c) = r^{-1}rc = c$. Hence, for all codewords $c \in C \cap \mathbb{F}_p^n$, there exists $v := r^{-1}c \in C$ such that $Tr(v) = c$. This exactly means $C \cap \mathbb{F}_p^n \subseteq Tr(C)$. □

This proposition holds more generally. [GP] shows $C \cap \mathbb{F}_p^n \subseteq Tr(C)$ whenever $\mathbb{F}_{p^m}$ is a separable extension over $\mathbb{F}_p$. By Lemma 11.82 in [R], $\mathbb{F}_{p^m}$ is separable over $\mathbb{F}_p$ if and only if the trace form $L : \mathbb{F}_{p^m} \times \mathbb{F}_{p^m} \to \mathbb{F}_p$ is non-degenerate. By Proposition 2.2.5, $L$ is non-degenerate, so $\mathbb{F}_{p^m}$ is separable and $C \cap \mathbb{F}_p^n \subseteq Tr(C)$ always holds.

The following theorem gives an equivalent characterization for the trace code and subfield subcode of a $\mathbb{F}_{p^m}$-linear code in terms of its Galois closure and Galois interior.

**Theorem 2.2.13.** *Let $C$ be a $(n, k)$ $\mathbb{F}_{p^m}$-linear code. We have the following characterization for its subfield subcode and trace code.*

$$C \cap \mathbb{F}_p^n = C^0 \cap \mathbb{F}_p^n \ \ and$$
$$Tr(C) = C^* \cap \mathbb{F}_p^n$$

*Proof.* We'll first begin by showing the characterization for the subfield subcode. Since $C^0 \subseteq C$, it is clear that $C^0 \cap \mathbb{F}_p^n \subseteq C \cap \mathbb{F}_p^n$.

For the inclusion the other way, let $c \in C \cap \mathbb{F}_p^n$ be a codeword and let $r \in \{1, \ldots, m\}$. Since we know $\phi_n(x) = x$ if and only if $x \in \mathbb{F}_p^n$ by Remark 2.1.4, we have

$$\phi_n^r(c) = \phi_n \circ \cdots \circ \phi_n(c) = c.$$

Thus, we get $c = \phi_n^r(c)$ for all $r = 1, \ldots, m$, so we have $c \in \phi_n^r(C)$ for all $r$. This means exactly $c \in C^0$. But since $c \in \mathbb{F}_p^n$ as well, this gives $c \in C^0 \cap \mathbb{F}_p^n$. With the previous inclusion we derived, we get $C \cap \mathbb{F}_p^n = C^0 \cap \mathbb{F}_p^n$, as required.

Next, we will show the characterization for the trace code. We have

$$Tr(C) = \{\sum_{i=1}^{m} \phi_n^i(c) : c \in C\} \subseteq \sum_{i=1}^{m} \phi_n^i(C) = C^*.$$

Also, since $Tr : \mathbb{F}_{p^m}^n \to \mathbb{F}_p^n$, we have $Tr(C) \subseteq \mathbb{F}_p^n$. Hence, $Tr(C) \subseteq C^* \cap \mathbb{F}_p^n$.

For the other inclusion, we invoke Delsarte Duality to write $(C^\perp \cap \mathbb{F}_p^n)^\perp = Tr(C)$. We'll show the inclusion by showing $(C^\perp \cap \mathbb{F}_p^n)^\perp \supseteq C^* \cap \mathbb{F}_p^n$. Let $c = \sum_{i=1}^m \phi_n^i(c_i)$ be a codeword such that $c_1, \ldots, c_m \in C$. Suppose further that $c \in \mathbb{F}_p^n$ and let $b \in C^\perp \cap \mathbb{F}_p^n$. We will use $< \cdot, \cdot >$ to denote the canonical inner product.

$$\begin{aligned}
< c, b > &= < \sum_{i=1}^{m} \phi_n^i(c_i), b > \\
&= \sum_{i=1}^{m} < \phi_n^i(c_i), b > \\
&= \sum_{i=1}^{m} \sum_{t=1}^{n} \phi^i((c_i)_t) b_t \\
&= \sum_{i=1}^{m} \phi^i \left( \sum_{t=1}^{n} b_t (c_i)_t \right) \\
&= \sum_{i=1}^{m} \phi^i(< b, c_i >)
\end{aligned}$$

But since we have $c_i \in C = (C^\perp)^\perp$ for all $i = 1, \ldots, m$, we get

$$< b, c_i >= 0 \ \ \forall b \in C^\perp \cap \mathbb{F}_p^n.$$

Hence, $\sum_{i=1}^m \phi^i(< b, c_i >) = \sum_{i=1}^m \phi^i(0) = 0$. Because $c \in \mathbb{F}_p^n$ as well and we have $(C^\perp \cap \mathbb{F}_p^n)^\perp \subseteq \mathbb{F}_p^n$, we conclude $c \in (C^\perp \cap \mathbb{F}_p^n)^\perp$. Of course, this means $C^* \cap \mathbb{F}_p^n \subseteq Tr(C)$, which gives us the equality. □

Immediately following from this theorem, we can relate the dimension of the subfield subcode and the trace code of a linear code to the dimension of its Galois interior and Galois closure, respectively.

**Corollary 2.2.14.** *For a $(n, k)$ $\mathbb{F}_{p^m}$-linear code $C$, we have*

$$\dim_{\mathbb{F}_p}(Tr(C)) = \dim_{\mathbb{F}_{p^m}}(C^*) \quad \textit{and} \quad \dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) = \dim_{\mathbb{F}_{p^m}}(C^0).$$

*Proof.*

$$\dim_{\mathbb{F}_{p^m}}(C^0) = \dim_{\mathbb{F}_{p^m}}(C^0 \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m}) = \dim_{\mathbb{F}_p}(C^0 \cap \mathbb{F}_p^n) = \dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n)$$

and

$$\dim_{\mathbb{F}_{p^m}}(C^*) = \dim_{\mathbb{F}_{p^m}}(C^* \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m}) = \dim_{\mathbb{F}_p}(C^* \cap \mathbb{F}_p^n) = \dim_{\mathbb{F}_p}(Tr(C)).$$

□

2.3. **Bounds on the Dimension of Subfield Subcodes and Trace Codes.** We will begin by placing naïve bounds on the dimensions of the subfield subcodes and trace codes for a given linear code. We will also identify sufficient and necessary conditions for these subcodes to be of maximal dimension. Lastly, we will offer slight improvements on the naïve bounds for the dimensions of both subfield subcodes and trace codes.

We begin by outlining the trivial bounds for a subfield subcode.

**Proposition 2.3.1.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. The dimension of its subfield subcode will be bounded as follows:*

$$0 \leq \dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) \leq k.$$

*Proof.* The lower bound is trivial, so we'll just show that $k$ is an upper bound. We notice by Corollary 2.1.14 that $\dim_{\mathbb{F}_{p^m}}(C^0 \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m}) = \dim_{\mathbb{F}_p}(C^0 \cap \mathbb{F}_p^n)$. But because $C^0$ is $\mathrm{Gal}(p^m, p)$-invariant, by Theorem 2.1.16, we get $\dim_{\mathbb{F}_{p^m}}(C^0) = \dim_{\mathbb{F}_p}(C^0 \cap \mathbb{F}_p^n)$. Since $C^0 \subseteq C$ and recalling that by Theorem 2.2.13 we have $C \cap \mathbb{F}_p^n = C^0 \cap \mathbb{F}_p^n$, this gives the upper bound as

$$\dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) = \dim_{\mathbb{F}_p}(C^0 \cap \mathbb{F}_p^n) = \dim_{\mathbb{F}_{p^m}}(C^0) \leq \dim_{\mathbb{F}_{p^m}}(C).$$

□

Note that these are naïve bounds on the dimension of a subfield subcode and are therefore not necessarily any good. All subfield subcodes must satisfy these bounds, but they may not necessarily be attained.

Next, we will outline the naïve bounds for the dimension of the trace code.

**Proposition 2.3.2.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. $Tr(C)$ is a $\mathbb{F}_p$-linear code of dimension at most $mk$.*

*Proof.* Consider the map $Tr|_C : C \to \mathbb{F}_p^n$. We have $Tr(C) = Im(Tr|_C)$, so $Tr(C) \subseteq \mathbb{F}_p^n$. Since $C \cong \mathbb{F}_p^{mk}$ and $Tr|_C$ is $\mathbb{F}_p$-linear, by the Dimension Theorem,

$$\begin{aligned} mk &= \mathrm{rank}_{\mathbb{F}_p}(Tr|_C) + \dim_{\mathbb{F}_p}(ker(Tr|_C)) \\ \implies \mathrm{rank}_{\mathbb{F}_p}(Tr|_C) &= mk - \dim_{\mathbb{F}_p}(ker(Tr|_C)) \leq mk. \end{aligned}$$

□

The lower bound follows from Delsarte Duality and restates part of Lemma 2.1.13 in [St].

**Proposition 2.3.3.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. $Tr(C)$ has dimension at least $k$.*

*Proof.* By Delsarte Duality, $Tr(C) = (C^\perp \cap \mathbb{F}_p^n)^\perp$. Hence,

$$\begin{aligned} \dim_{\mathbb{F}_p}(Tr(C)) &= \dim_{\mathbb{F}_p}((C^\perp \cap \mathbb{F}_p^n)^\perp) \\ &= n - \dim_{\mathbb{F}_p}(C^\perp \cap \mathbb{F}_p^n) \\ &\geq n - \dim_{\mathbb{F}_{p^m}}(C^\perp) \\ &= n - (n - \dim_{\mathbb{F}_{p^m}}(C)) \\ &= k. \end{aligned}$$

□

The upper bounds from these previous propositions indicate the maximal possible dimensions of subfield subcodes and trace codes. We will give sufficient and necessary conditions for when the dimension of these codes attains their respective upper bounds. Again, we start by considering the subfield subcode.

**Proposition 2.3.4.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. The subfield subcode $C \cap \mathbb{F}_p^n$ is of maximal dimension if and only if $C^0 = C = C^*$.*

*Proof.* We start with the forwards implication.

If $C \cap \mathbb{F}_p^n$ is of maximal dimension, we have $\dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) = \dim_{\mathbb{F}_{p^m}}(C^0) = k$. But since $C^0 \subseteq C$, this means $C^0 = C$. Becuase $C^0$ is $\mathrm{Gal}(p^m, p)$-invariant, this means $C$ itself is $\mathrm{Gal}(p^m, p)$-invariant. By Proposition 2.1.10, $C^*$ is the smallest $\mathrm{Gal}(p^m, p)$-invariant vector space containing $C$, so because $C$ is $\mathrm{Gal}(p^m, p)$-invariant and it contains itself, we must have $C \supseteq C^*$ as well, giving us the last equality: $C = C^*$.

For the converse direction, if we have $C^0 = C$, then this means $\dim_{\mathbb{F}_{p^m}}(C^0) = k$. Because $\dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) = \dim_{\mathbb{F}_{p^m}}(C^0)$, it attains its upper bound. □

As a result, we may characterize when a subfield subcode is of maximal dimension in another way as well.

**Corollary 2.3.5.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. Let $B$ be a basis for $C$ and let $\mathbf{H}$ be a parity-check matrix. We observe that $C \cap \mathbb{F}_p^n$ is of maximal dimension if and only if*

$$\mathbf{H}\phi_n(b) = 0 \ \ \forall b \in B.$$

*Proof.* By the last proposition, $C \cap \mathbb{F}_p^n$ is of maximal dimension if and only if $C = C^0$, which, because $C^0$ is $\mathrm{Gal}(p^m, p)$-invariant, means $C$ is $\mathrm{Gal}(p^m, p)$-invariant. By Lemma 2.1.8, $C$ is $\mathrm{Gal}(p^m, p)$-invariant if and only if $\phi_n(b) \in C$ for all $b \in B$. Hence, this occurs if and only if $\mathbf{H}\phi_n(b) = 0$ for all $b \in B$. □

We next take a combinatorial perspective to describe the necessary relationship between the Galois interior and the Galois closure of a linear code when its trace code is of full rank.

**Proposition 2.3.6.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. If $\dim_{\mathbb{F}_p}(Tr(C)) = mk$, then $\dim_{\mathbb{F}_{p^m}}(C^0) = 0$.*

*Proof.* Begin by considering the Galois closure $C^*$. We have that $C^*$ is a vector space over $\mathbb{F}_{p^m}$, so if its dimension is $d$, it will be isomorphic to $\mathbb{F}_{p^m}^d$. Indeed, we know then that the cardinality of $C^*$ will be $(p^m)^d$, so we may in fact ascertain the dimension of $C^*$ from its cardinality. We will first approximate $|C^*|$ by assuming each distinct choice of $m$-tuple $v := (c_1, \ldots, c_m) \in C^m$ produces a different vector $s_v := \sum_{i=1}^m \phi_n^i(c_i) \in C^*$. In this case, we get

$$|C^*| = |C^m| = (p^m)^{mk}.$$

Define $T : C^m \to C^*$ by the mapping $v \mapsto s_v$. Note that we won't consider the linearity of $T$ in what follows. We will get the desired result by showing that $T$ is not injective.

However, we notice for a given $m$-tuple $(c_1, \ldots, c_m)$ if there exists $j \in \{1, \ldots, m\}$ such that $c_j \neq 0$ and $\phi_n^j(c_j) \in \phi_n^i(C)$ for some $i \neq j$ (WLOG $j > i$), then the choice of $m$-tuples $v = (c_1, \ldots, c_i, \ldots, c_j, \ldots, c_m)$ and $w := (c_1, \ldots, c_i + \phi_n^{j-i}(c_j), \ldots, 0, \ldots, c_m)$ will produce the same vector. This vector will therefore be double-counted in the first estimate of $|C^*|$. If $\cup_{j \neq i} \phi_n^i(C) \cap \phi_n^j(C) \neq \{0\}$ for all $i = 1, \ldots, m$, then there is a vector in $C^*$ that can be expressed as the image of two distinct $m$-tuples $v, w \in C^m$ in the above way. Hence, $T$ is not injective, meaning that $|C^m| = (p^m)^{mk} > |C^*|$, which then guarantees $\dim_{\mathbb{F}_{p^m}}(C^*) \neq mk$.

Notice that if $C^0 = \cap_{i=1}^m \phi_n^i(C) \neq \{0\}$, then for all $i = 1, \ldots, m$, we observe $\cup_{j \neq i} \phi_n^i(C) \cap \phi_n^j(C) \supsetneq \{0\}$, so $\dim_{\mathbb{F}_{p^m}}(C^*) \neq mk$ as per the above. But since $\dim_{\mathbb{F}_{p^m}}(C^*) = \dim_{\mathbb{F}_p}(Tr(C))$, we have $C^0 \neq \{0\}$ implies $\dim_{\mathbb{F}_p}(Tr(C)) \neq mk$. The contrapositve of this last implication is the result we wanted to show. □

Note that the converse doesn't hold. Having $\dim_{\mathbb{F}_{p^m}}(C^0) = 0$ isn't nearly a strong enough condition to guarantee that the dimension of the trace code attains its upper bound. For example, one can verify that if $\cap_{i=1}^n \phi_n^i(C) = \{0\}$ and there exists some $i \in \{1, \ldots, m\}$ such that $\phi_n^i(C) \cap \left(\sum_{j \neq i} \phi_n^j(C)\right) \supsetneq \{0\}$, the corresponding trace code $Tr(C)$ will not be of maximal dimension. We will soon prove this is the case as a part of our describing conditions on $C$ that will ensure $Tr(C)$ is of maximal rank. To do this, we will need to establish a result from linear algebra.

**Proposition 2.3.7.** *Let $V$ be finite-dimensional vector space over field $\mathbb{F}$. Let $U_1, \ldots, U_m$ be subspaces of $V$ such that $V = U_1 + \cdots + U_m$. We have that $V = \bigoplus_{i=1}^m U_i \iff \dim(V) = \sum_{i=1}^m \dim(U_i)$.*

*Proof.* To prove the forwards implication, we assume $V = \bigoplus_{i=1}^m U_i$. This means

$$V = \sum_{i=1}^m U_i \quad \text{and} \quad U_i \cap \left( \sum_{j \neq i} U_j \right) = \{0\}.$$

Notice that $V = \bigoplus_{i=1}^m U_i = (\bigoplus_{i=1}^{m-1} U_i) \oplus U_m$, which follows essentially from the associativity of addition and that $U_m \cap \left(\sum_{j \neq m} U_j\right) = \{0\}$. Let $W := \bigoplus_{i=1}^{m-1} U_i$ and let $B_1 := \{w_1, \ldots, w_k\}$ be a basis for $W$. Likewise, let $B_2 := \{u_1, \ldots, u_l\}$ be a basis for $U_m$. We will show that $B_1 \cup B_2$ is a basis for $W \oplus U_m$.

Let $v \in V$ be given. Clearly, there exists $w \in W$ and $u \in U_m$ such that $v = w + u$. Given that $B_1$ is a basis for $W$ and $B_2$ is a basis for $U_m$, there exist $\lambda_i, \gamma_j \in \mathbb{F}$ for all $i = 1, \ldots, k$ and for all $j = 1, \ldots, l$ such that $v = \sum_{i=1}^k \lambda_i w_i + \sum_{j=1}^l \gamma_j u_j$. Hence, $V \subseteq \text{span}(B_1 \cup B_2)$, giving us $V = \text{span}(B_1 \cup B_2)$.

Next, suppose for a contradiction that there exists $I \subseteq \{1, \ldots, k\}$ and $J \subseteq \{1, \ldots, l\}$ such that $\lambda_i \neq 0$ for all $i \in I$, $\gamma_j \neq 0$ for all $j \in J$, and $\sum_{i \in I} \lambda_i w_i + \sum_{j \in J} \gamma_j u_j = 0$. By the linear independence of $B_1$ and $B_2$, we can't have either $I = \emptyset$ or $J = \emptyset$ as if $I$ were empty, $\sum_{j \in J} \gamma_j w_j = 0$ implies $\gamma_j = 0$ for all $j \in J$, which is a contradiction, and likewise for if $J = \emptyset$. Thus, both $I \neq \emptyset$ and $J \neq \emptyset$.

We therefore have

$$\sum_{i\in I}\lambda_i w_i + \sum_{j\in J}\gamma_j u_j = 0$$
$$\iff \sum_{i\in I}\lambda_i w_i = \sum_{j\in J}(-\gamma_j)u_j.$$

But since $\gamma_j \neq 0$ for all $j \in J$ and $B_2$ is linearly independent, $\sum_{j\in J}(-\gamma_j)u_j \neq 0$. Therefore, there exists $u \in U_m$ such that $u \neq 0$ and $u \in W$. But this contradicts $W \oplus U_m = V$, so

$$\sum_{i=1}^{m}\lambda_i w_i + \sum_{j\in J}^{l}\gamma_j u_j = 0 \implies \lambda_i, \gamma_j = 0 \ \forall i, j.$$

Hence, $B_1 \cup B_2$ is linearly independent, so it's a basis for $V$.

This gives $\dim(V) = \dim(W) + \dim(U_m) = \dim(\bigoplus_{i=1}^{m-1} U_i) + \dim(U_m)$. By taking $m = m - j$ and iterating the above argument for all $j = 1, \ldots, m-2$, what we get is $\dim(V) = \sum_{i=1}^{m}\dim(U_i)$. This proves the forwards implication.

To prove the converse direction, suppose that $\dim(V) = \dim(\sum_{i=1}^{m} U_i) = \sum_{i=1}^{m}\dim(U_i)$. We must show this implies $V = \bigoplus_{i=1}^{m} U_i$, which amounts to showing $U_i \cap \Big(\sum_{j\neq i} U_j\Big) = \{0\}$ for all $i = 1, \ldots, m$. Let $i \in \{1, \ldots, m\}$ be given. By the associativity and commutativity of addition, we have $V = U_i + \sum_{j\neq i} U_j$. We also have $\dim(V) = \dim(U_i) + \sum_{j\neq i}\dim(U_j)$.

For any finite-dimensional vector spaces $U, W$ over $\mathbb{F}$, we have $\dim(U + W) = \dim(U) + \dim(W) - \dim(U \cap W)$. Hence,

$$\dim(V) = \dim(U_i) + \dim(\sum_{j\neq i} U_j) - \dim(U_i \cap \sum_{j\neq i} U_j).$$

But since $\dim(V) = \dim(U_i) + \sum_{j\neq i}\dim(U_j)$, we get

$$\dim(\sum_{j\neq i} U_j) - \dim(U_i \cap \sum_{j\neq i} U_j) = \sum_{j\neq i}\dim(U_j). \tag{2.1}$$

Let $B_j$ be a basis for $U_j$ for all $j \neq i$. As we've shown in proving the previous direction, $\cup_{j\neq i} B_j$ spans $\sum_{j\neq i} U_j$, so we get

$$\dim(\sum_{j\neq i} U_j) \leq |\cup_{j\neq i} B_j| \leq \sum_{j\neq i}\dim(U_j).$$

But because $\dim(U_i \cap \sum_{j\neq i} U_i) \geq 0$, the only way for (2.1) to hold is if $\dim(\sum_{j\neq i} U_j) = \sum_{j\neq i}\dim(U_j)$. Hence, $\dim(U_i \cap \sum j \neq i U_j) = 0$, as required. □

This proposition lets us place the sufficient and necessary conditions for a trace code of a linear code to be of maximal rank.

**Corollary 2.3.8.** *Let $C$ be a $(n, k)$ $\mathbb{F}_{p^m}$-linear code. The trace code $Tr(C)$ is of maximal dimension if and only if $C^* = \bigoplus_{i=1}^{m}\phi_n^i(C)$.*

*Proof.* By Corollary 2.2.14, we have $\dim_{\mathbb{F}_p}(Tr(C)) = \dim_{\mathbb{F}_{p^m}}(C^*)$. The maximal dimension of $Tr(C)$ is $mk = \sum_{i=1}^{m} \phi_n^i(C)$. By the previous proposition, we conclude $\dim_{\mathbb{F}_{p^m}}(C^*) = \sum_{i=1}^{m} \phi_n^i(C)$ if and only if $C^* = \bigoplus_{i=1}^{m} \phi_n^i(C)$. □

Next, we will describe a relationship between the dimensions of the Galois closure and Galois interior of a linear code.

**Proposition 2.3.9.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. Let $B^0$ be a basis for its Galois interior $C^0$ and let $B_i \supseteq B^0$ be a basis for $\phi_n^i(C)$ for all $i = 1, \ldots, m$. If we define $P := \{1, \ldots, m\}$, we observe*

$$\dim_{\mathbb{F}_{p^m}}(C^*) + (-1)^m \dim_{\mathbb{F}_{p^m}}(C^0) \leq \sum_{i=1}^{m-1} (-1)^{r+1} \sum_{\substack{I \subseteq P \\ |I|=r}} |\cap_{i \in I} B_i|.$$

*Proof.* Recall that $C^0 = \cap_{i=1}^{m} \phi_n^i(C)$. With this, for any $I \subseteq P$, we observe that $\cap_{i \in I} \phi_n^i(C)$ will admit $C^0$ as a subspace. Because of this, we can extend $B^0$ to be a basis for $\phi_n^i(C)$ for all $i = 1, \ldots, m$. The bases we get through this construction are $B_i := \{b_{i,1}, \ldots, b_{i,k}\}$ for all $i = 1, \ldots, m$.

We will first show that $\cup_{i=1}^{m} B_i$ spans $C^*$, and, thus, $\dim_{\mathbb{F}_{p^m}}(C^*) \leq |\cup_{i=1}^{m} B_i|$. Let $v \in C^*$ be given, so there exists $c_i \in \phi_n^i(C)$ for all $i = 1, \ldots m$ such that $v = \sum_{i=1}^{m} c_i$. Because for all $i = 1, \ldots, m$, we know $B_i$ is a basis for $\phi_n^i(C)$, there exists $\lambda_{i,1}, \ldots, \lambda_{i,k} \in \mathbb{F}_{p^m}$ such that $c_i = \sum_{j=1}^{k} \lambda_{i,j} b_{i,j}$. Hence,

$$v = \sum_{i=1}^{m} c_i = \sum_{i=1}^{m} \sum_{j=1}^{k} \lambda_{i,j} b_{i,k}.$$

This lets us conclude $C^* \subseteq \text{span}_{\mathbb{F}_{p^m}}(\cup_{i=1}^{m} B_i)$, which gives us the equality as the inclusion in the other direction follows immediately from the definition of $C^*$.

Since the size of any generating set of a vector space is greater or equal to the size of any linearly independent subset of the same vector space, $\dim_{\mathbb{F}_{p^m}}(C^*) \leq |\cup_{i=1}^{m} B_i|$. By inclusion-exclusion, the right-hand side simplifies to

$$|\cup_{i=1}^{m} B_i| = \sum_{r=1}^{m} (-1)^{r+1} \sum_{\substack{I \subseteq P \\ |I|=r}} |\cap_{i \in I} B_i|.$$

For each $i = 1, \ldots, m$, we extended $B^0$ to $B_i$ so that it may be a basis for $\phi_n^i(C)$. This means $B^0 \subseteq B_i$ for all $i$, and, hence, $\cap_{i=1}^{m} B_i \supseteq B^0$. To address the other inclusion, let $b \in B_i$ for all $i$ and suppose for a contradiction that $b \notin B^0$. This must mean that either $B^0 \cup \{b\}$ is not linearly independent or $b \notin C^0$. Taking $i \in \{1, \ldots, m\}$, since $B^0 \cup \{b\} \subseteq B_i$ and because $B_i$ is linearly independent, any subset thereof must also be linearly independent. Since $b \in B_i \subseteq \phi_n^i(C)$ for all $i = 1, \ldots, m$, we have $b \in C^0$, which is a contradiction. Since no such $b$ exists, we get $\cap_{i=1}^{m} B_i \subseteq B^0$, meaning $\cap_{i=1}^{m} B_i = B^0$. Thus,

$$|\cup_{i=1}^{m} B_i| = \sum_{i=1}^{m-1} (-1)^{r+1} \sum_{\substack{I \subseteq P \\ |I|=r}} |\cap_{i \in I} B_i| + (-1)^{m+1} |B^0|.$$

Using the fact that $\dim_{\mathbb{F}_{p^m}}(C^*) \leq |\cup_{i=1}^m B_i|$ and recognizing that $|B^0| = \dim_{\mathbb{F}_{p^m}}(C^0)$ gives us the result. □

The relationship between the dimensions of $C^*$ and $C^0$ established in the previous proposition is maintained if we replace $\dim_{\mathbb{F}_{p^m}}(C^*)$ by $\dim_{\mathbb{F}_p}(Tr(C))$ and $\dim_{\mathbb{F}_{p^m}}(C^0)$ by $\dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n)$ as a result of Corollary 2.2.14.

Finally, we will offer some improved bounds on the dimensions of the trace code and subfield subcode of a linear code. The first improvement is a new upper bound on the dimension of the trace code given by Proposition 9.1.4 in [St].

**Proposition 2.3.10.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code and let $U$ be a $\mathbb{F}_{p^m}$-subspace of $C$ such that $\phi_n(U) \subseteq C$. We have*

$$\dim_{\mathbb{F}_p}(Tr(C)) \leq m(k - \dim_{\mathbb{F}_{p^m}}(U)) + \dim_{\mathbb{F}_p}(U \cap \mathbb{F}_p^n).$$

*Proof.* See [St]. □

A new lower bound on the dimension of the subfield subcode is given by Exercise 9.3 in [St].

**Proposition 2.3.11.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. Suppose there is a matrix $\mathbf{M} \in \mathcal{M}_{r\times n}(\mathbb{F}_p)$ such that $\mathbf{M}$ is of rank $s$ and $\mathbf{M}c = 0$ for all $c \in C$. We have*

$$\dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) \geq k - (m-1)(n-s-k).$$

*Proof.* Let $\mathbf{M}_x$ be $\mathbf{M}$ viewed as a matrix in $\mathcal{M}_{r\times n}(\mathbb{F}_{p^m})$. Since we have that for all $c \in C$, $\mathbf{M}c = \mathbf{M}_x c = 0$, we observe that $C \cap \mathbb{F}_p^n \subseteq ker(\mathbf{M})$ and $C \subseteq ker(\mathbf{M}_x)$.

Define $V$ to be $ker(\mathbf{M}_x)$ viewed as a vector space over $\mathbb{F}_p$ and likewise let $W$ be $C$ viewed as a vector space over $\mathbb{F}_p$, so we have $W \subseteq V$. Consider the quotient map $T : V \to V/W$, whose kernel is necessarily $W$. Given that $ker(\mathbf{M}) = ker(\mathbf{M}_x) \cap \mathbb{F}_p^n$, we also have $ker(\mathbf{M}) \subseteq V$. Now, since $C \cap \mathbb{F}_p^n \subseteq ker(\mathbf{M})$, we have that the kernel of $T|_{ker(\mathbf{M})}$ will be $C \cap \mathbb{F}_p^n$. But since $Im(T) \supseteq Im(T|_{ker(\mathbf{M})})$, we get that $rank_{\mathbb{F}_p}(T) \geq rank_{\mathbb{F}_p}(T|_{ker(\mathbf{M})})$. But then by the Dimension Theorem, we have

$$\dim_{\mathbb{F}_p}(ker(\mathbf{M}_x)) - \dim_{\mathbb{F}_p}(C) \geq \dim_{\mathbb{F}_p}(ker(\mathbf{M})) - \dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n).$$

Note that since $rank_{\mathbb{F}_p}(\mathbf{M}) = s$, we also have $rank_{\mathbb{F}_{p^m}}(\mathbf{M}_x) = s$. Thus, we get by the Dimension Theorem that $\dim_{\mathbb{F}_p}(ker(\mathbf{M})) = n - s = \dim_{\mathbb{F}_{p^m}}(ker(\mathbf{M}_x))$. With this, we can rewrite the above equality to get the desired result.

$$\begin{aligned} m(\dim_{\mathbb{F}_{p^m}}(ker(\mathbf{M}_x)) - \dim_{\mathbb{F}_{p^m}}(C)) &\geq \dim_{\mathbb{F}_p}(ker(\mathbf{M})) - \dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) \\ &\iff \dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) \geq n - s - m(n-s-k) \\ &\iff \dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) \geq k - (m-1)(n-s-k) \end{aligned}$$

□

This last proposition then gives us an improved upper bound for the dimension of the corresponding trace code.

**Corollary 2.3.12.** *Let $C$ be a $(n,k)$ $\mathbb{F}_{p^m}$-linear code. We have*

$$\dim_{\mathbb{F}_p}(Tr(C)) \leq k + (m-1)(k - \dim_{\mathbb{F}_{p^m}}(C^0)).$$

*Proof.* Let $B := \{b_1, \ldots, b_s\}$ be a basis for $C \cap \mathbb{F}_p^n$ and let $\mathbf{M} \in \mathcal{M}_{s\times n}(\mathbb{F}_p)$ whose rows are the elements of $B$. Given that $(C^\perp)^\perp = C$, we will have for all $c \in C^\perp$, $\mathbf{M}c = 0$. Hence, by the previous proposition,

$$\dim_{\mathbb{F}_p}(C^\perp \cap \mathbb{F}_p^n) \geq \dim_{\mathbb{F}_{p^m}}(C^\perp) - (m-1)(n - s - \dim_{\mathbb{F}_{p^m}}(C^\perp)).$$

By Delsarte Duality, we have

$$\begin{aligned} &\dim_{\mathbb{F}_p}(Tr(C)^\perp) \geq \dim_{\mathbb{F}_{p^m}}(C^\perp) - (m-1)(n - s - \dim_{\mathbb{F}_{p^m}}(C^\perp)) \\ \iff\ & n - \dim_{\mathbb{F}_p}(Tr(C)) \geq n - k - (m-1)(k-s) \end{aligned}$$

But since $s = |B| = \dim_{\mathbb{F}_p}(C \cap \mathbb{F}_p^n) = \dim_{\mathbb{F}_{p^m}}(C^0)$, we conclude

$$\dim_{\mathbb{F}_p}(Tr(C)) \leq k + (m-1)(k - \dim_{\mathbb{F}_{p^m}}(C^0)).$$

□

## 3. GRS Codes

Generalized Reed-Solomon (GRS) codes are an important family of linear codes, being perhaps the most extensively-used error-correcting codes in practice. They saw direct cryptographic application in a McEliece scheme wherein the cryptographic primitive was based on a GRS code, but this scheme was proven to be insecure in [SS], as we outline in Section 6.1. Currently, their use in cryptography is more indirect as it is Goppa codes, subfield subcodes of GRS codes, that form the cryptographic primitive of current proposals of McEliece. We will study GRS codes in this section, outlining key properties of GRS codes and the ways in which the same GRS code may be defined by different pairs of parameters.

### 3.1. Properties of GRS Codes.

We will define GRS codes and establish a few of their basic properties.

**Definition 3.1.1.** A *GRS code* is defined by a pair of vectors $\alpha, \beta \in \mathbb{F}_{p^m}^n$ such that $\alpha_i \neq \alpha_j$ for all $i \neq j$ and $\beta_i \neq 0$ for all $i \in \{1, \ldots, n\}$. The $(n, k)$ GRS code defined by the pair $(\alpha, \beta)$ is

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

The vectors $\alpha$ and $\beta$ are typically called the *locator* and *multiplier*, respectively.

We next verify that GRS codes truly are linear codes.

**Proposition 3.1.2.** *$GRS_{n,k}(\alpha, \beta)$ is a vector space.*

*Proof.* Given that $GRS_{n,k}(\alpha, \beta) \subseteq \mathbb{F}_{p^m}^n$, we need only show it is a subspace of $\mathbb{F}_{p^m}^n$.

- $0 \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m}) \implies (0, \ldots, 0) \in GRS_{n,k}(\alpha, \beta)$
- Let $f, g \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ be given. Define $a := (\beta_1 f(\alpha_1), \ldots, \beta_n f(\alpha_n))$ and $b := (\beta_1 g(\alpha_1), \ldots, \beta_n g(\alpha_n))$. We observe

$$\begin{aligned} a + b &= (\beta_1 f(\alpha_1), \ldots, \beta_n f(\alpha_n)) + (\beta_1 g(\alpha_1), \ldots, \beta_n g(\alpha_n)) \\ &= (\beta_1 (f+g)(\alpha_1), \ldots, \beta_n (f+g)(\alpha_n)) \\ &\in GRS_{n,k}(\alpha, \beta). \end{aligned}$$

- Let $\lambda \in \mathbb{F}_{p^m}$. We observe

$$\lambda a = (\beta_1(\lambda f)(\alpha_1), \ldots, \beta_n(\lambda f)(\alpha_n)) \in GRS_{n,k}(\alpha, \beta).$$

□

We next verify that the dimension of $GRS_{n,k}(\alpha, \beta)$ is as purported.

**Proposition 3.1.3.** $\dim_{\mathbb{F}_{p^m}}(GRS_{n,k}(\alpha, \beta)) = k$.

*Proof.* Suppose for a contradiction $f, g \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ such that $f \neq g$ and we have $(\beta_1 f(\alpha_1), \ldots, \beta_n f(\alpha_n)) = (\beta_1 g(\alpha_1), \ldots, \beta_n g(\alpha_n))$. Since $\beta_i \neq 0 \forall i$, this implies as follows.

$$\begin{aligned} & f(\alpha_i) = g(\alpha_i) \quad \forall i \\ \iff & (f-g)(\alpha_i) = 0 \quad \forall i \\ \iff & (x - \alpha_i) \mid f - g \quad \forall i \end{aligned}$$

Therefore, we get $(f-g)(x) = q(x)\prod_{i=1}^{n}(x-\alpha_i)$ for some $q \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$. However, since $deg(f-g) \leq k-1$ and $deg\left(\prod_{i=1}^{n}(x-\alpha_i)\right) \geq n > k-1$, the above is a contradiction. Hence, codewords defined by different polynomials in $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ are different.

Thus, $|GRS_{n,k}(\alpha, \beta)| = |\mathbb{P}_{k-1}(\mathbb{F}_{p^m})| = p^{mk}$. But since $GRS_{n,k}(\alpha, \beta)$ is a vector space over $\mathbb{F}_{p^m}$, this means $GRS_{n,k}(\alpha, \beta) \cong \mathbb{F}_{p^m}^k$, so $\dim_{\mathbb{F}_{p^m}}(GRS_{n,k}(\alpha, \beta)) = k$. □

An important property about GRS codes that makes them particularly useful for error correction is that they are MDS, that is to say that they attain the Singleton bound. We will first remind ourselves of the Singleton bound before proving that GRS codes are MDS.

**Theorem 3.1.4** (Singleton bound)**.** *Let $C$ be a $(n, k)$ linear code over $\mathbb{F}_{p^m}$ and let $d$ be its minimum distance. We have $d \leq n - k + 1$.*

There are many references that offer a proof for this theorem. See Theorem 1.7.1 in [N] for a proof.

**Proposition 3.1.5.** *Let $d$ denote the minimum distance of $GRS_{n,k}(\alpha, \beta)$. $GRS_{n,k}(\alpha, \beta)$ is MDS, meaning it attains the Singleton bound:*

$$d = n - k + 1.$$

*Proof.* We present the proof as in [MS].
Let $c \in GRS_{n,k}(\alpha, \beta) \backslash \{0\}$, so there exists $f \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ such that $c = (\beta_1 f(\alpha_1), \ldots, \beta_n f(\alpha_n))$. But since $deg(f) \leq k-1$, there are at most $k-1$ non-zero entries in $c$.

Thus, $\omega(c) \geq n - (k-1) = n - k + 1$. Since $GRS_{n,k}(\alpha, \beta)$ is linear, $d \geq n - k + 1$. But then by the Singleton bound, $n - k + 1 \leq d \leq n - k + 1$, so $d$ attains the Singleton bound. □

As there are several choices for a generator matrix for a linear code, we understand that there is typically a "preferred choice" of generator matrix for each code. We will next establish the canonical form of the generator matrix for a GRS code, which is the "preferred" generator matrix.

To develop a generator matrix for $GRS_{n,k}(\alpha, \beta)$, we first need to find a basis for the GRS code. In fact, any basis for $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ can be used to define a basis for $GRS_{n,k}(\alpha, \beta)$.

**Proposition 3.1.6.** *Let $F := \{f_1, \ldots, f_k\}$ be a basis for $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$. The set $B := \{b_1, \ldots, b_k\}$ is a basis for $GRS_{n,k}(\alpha, \beta)$ where for all $i = 1, \ldots, k$, $b_i$ is defined by $b_i := (\beta_1 f_i(\alpha_1), \ldots, \beta_n f_i(\alpha_n))$.*

*Proof.* Suppose $c \in GRS_{n,k}(\alpha, \beta)$ such that $c = (\beta_1 f(\alpha_1), \ldots, \beta_n f(\alpha_n))$ for some $f \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$. Given that $F$ is a basis for $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$, there exists $\lambda_1, \ldots, \lambda_k \in \mathbb{F}_{p^m}$ such that $f(x) = \sum_{i=1}^{k} \lambda_i f_i(x)$. But then

$$c = (\beta_1 f(\alpha_1), \ldots, \beta_n f(\alpha_n)) = \sum_{i=1}^{k} (\beta_1 f_i(\alpha_i), \ldots, \beta_n f_i(\alpha_i)).$$

Hence, $GRS_{n,k}(\alpha, \beta) \subseteq \text{span}_{\mathbb{F}_{p^m}}(B)$. The inclusion in the other direction is evident, so this is an equality. Since $\dim_{\mathbb{F}_{p^m}}(GRS_{n,k}(\alpha, \beta)) = k = |B|$, this means $B$ is a basis for $GRS_{n,k}(\alpha, \beta)$. □

If the choice of basis for $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ used in the last proposition is the standard monomial basis, then the basis for $GRS_{n,k}(\alpha, \beta)$ defined from this basis is the canonical basis of the GRS code. This will be

$$\left\{(\beta_1, \ldots, \beta_n),\ (\beta_1\alpha_1, \ldots, \beta_n\alpha_n),\ \ldots,\ (\beta_1\alpha_1^{k-1}, \ldots, \beta_n\alpha_n^{k-1})\right\}.$$

The canonical generator matrix for $GRS_{n,k}(\alpha, \beta)$ is defined from this basis to be as follows.

$$\mathbf{G} := \begin{bmatrix} \beta_1 & \beta_1\alpha_1 & \ldots & \beta_1\alpha_1^{k-1} \\ \beta_2 & \beta_2\alpha_2 & \ldots & \beta_2\alpha_2^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_n & \beta_n\alpha_n & \ldots & \beta_n\alpha_n^{k-1} \end{bmatrix}$$

For the last of these preliminary properties, we will note that the dual code of a GRS code is another GRS code.

**Proposition 3.1.7.** *The dual code of $GRS_{n,k}(\alpha, \beta)$ is $GRS_{n,k}(\alpha, \beta)^{\perp} = GRS_{n,n-k}(\alpha, \gamma)$ such that $\gamma \in \mathbb{F}_{p^m}^n$ is defined by $\gamma_i = \beta_i^{-1} \prod_{j \neq i}(\alpha_i - \alpha_j)^{-1}$ for all $i = 1, \ldots, n$.*

*Proof.* See Theorem 4 in Chapter 10 of [MS]. □

**Remark 3.1.8.** Consequently, a parity-check matrix for $GRS_{n,k}(\alpha, \beta)$ will be the transpose of a generator matrix for $GRS_{n,n-k}(\alpha, \gamma)$.

Using the canonical generator matrix of $GRS_{n,n-k}(\alpha,\gamma)$, we can write a parity-check matrix for $GRS_{n,k}(\alpha,\beta)$ to be $\mathbf{H}$ as follows.

$$\begin{aligned}\mathbf{H} &= \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_n \\ \gamma_1\alpha_1 & \gamma_2\alpha_2 & \cdots & \gamma_n\alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_1\alpha_1^{n-k-1} & \gamma_2\alpha_2^{n-k-1} & \cdots & \gamma_n\alpha_n^{n-k-1} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \cdots & \alpha_n^{n-k-1} \end{bmatrix} \begin{bmatrix} \gamma_1 & & & \\ & \gamma_2 & & \\ & & \ddots & \\ & & & \gamma_n \end{bmatrix} \\ &= \mathbf{KC}\end{aligned}$$

We see from this that $c \in ker(\mathbf{H})$ if and only if $\mathbf{C}c \in ker(\mathbf{K})$. Thus, $\mathbf{K}$ is a parity-check matrix for $\mathbf{C}(GRS_{n,k}(\alpha,\beta)) = \{\mathbf{C}c : c \in GRS_{n,k}(\alpha,\beta)\}$.

3.2. **Equivalence of GRS Codes.** The same GRS code may be defined by multiple different pairs of parameters. We will describe conditions on the pairs of vectors in $\mathbb{F}_{p^m}^n$ that may define a GRS code such all pairs that satisfy these conditions will define the same GRS code.

The following result is motivated by Problem 5.4 in [Ro].

**Theorem 3.2.1.** *Let $\alpha,\beta \in \mathbb{F}_{p^m}^n$ such that $\alpha_i \neq \alpha_j$ $\forall i \neq j$ and $\beta_i \neq 0$ for all $i = 1,\ldots,n$. If we define vectors $\alpha',\beta' \in \mathbb{F}_{p^m}^n$ in one of the three following ways, $GRS_{n,k}(\alpha,\beta) = GRS_{n,k}(\alpha',\beta')$.*

- *(Case-1 Equality) Let $\mu,\nu,\eta \in \mathbb{F}_{p^m}$ such that $\mu,\eta \neq 0$. Define $\alpha'$ and $\beta'$ such that $\alpha'_i = \mu\alpha_i + \nu$ and $\beta'_i = \eta\beta_i$ $\forall i = 1,\ldots,n$.*
- *(Case-2 Equality) Suppose further that $\alpha_i \neq 0$ $\forall i = 1,\ldots,n$. Define $\alpha'$ such that $\alpha'_i = \alpha_i^{-1}$ and $\beta'_i = \beta_i\alpha_i^{-(n-k-1)}\prod_{j\neq i}(-\alpha_i\alpha_j)$ $\forall = 1,\ldots,n$.*
- *(Case-3 Equality) Let $\mu,\nu,\sigma,\tau,\delta \in \mathbb{F}_{p^m}$ such that $\mu\tau \neq \sigma\nu$ and $\delta \neq 0$. Suppose further that $\sigma\alpha_i + \tau \neq 0$ $\forall i = 1,\ldots,n$. Define $\alpha'$ such that $\alpha'_i = \frac{\mu\alpha_i+\nu}{\sigma\alpha_i+\tau}$ and define $\beta'$ such that*

$$\beta'_i = \delta\beta_i(\sigma\alpha_i+\tau)^{-(n-k-1)}\prod_{j\neq i}[-(\sigma\alpha_i+\tau)(\sigma\alpha_j+\tau)] \quad \forall i = 1,\ldots,n.$$

*Proof.* We'll prove each case separately.
Case-1 Equality

Let $\alpha'$ and $\beta'$ be as described above. Define $\tau : \mathbb{F}_{p^m} \to \mathbb{F}_{p^m}$ by $x \mapsto \mu x + \nu$ and define $\sigma : \mathbb{P}_{k-1}(\mathbb{F}_{p^m}) \to \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ by $f \mapsto \eta f$. Notice that for all $f \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$, we observe $\sigma \circ f \circ \tau(x) \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ too.

But each codeword of $GRS_{n,k}(\alpha',\beta')$ can be expressed for some $f \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ as $(\eta\beta_1 f(\tau(\alpha_1)),\ldots,\eta\beta_n f(\tau(\alpha_n))) = (\beta_1\sigma\circ f\circ\tau(\alpha_1),\ldots,\beta_n\sigma\circ f\circ\tau(\alpha_n))$. But this right-hand vector is a codeword of $GRS_{n,k}(\alpha,\beta)$. Hence, $GRS_{n,k}(\alpha',\beta') \subseteq GRS_{n,k}(\alpha,\beta)$, and since both vector spaces are of the same size, this is an equality.

Case-2 Equality

Let $\alpha'$ and $\beta'$ be as described above. Recall the definitions of $\mathbf{K}$ and $\mathbf{C}$ used in the discussion immediately proceeding Remark 3.1.8.

$$\mathbf{K} := \begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \dots & \alpha_n^{n-k-1} \end{bmatrix}, \quad \mathbf{C} := \begin{bmatrix} \gamma_1 & & & \\ & \gamma_2 & & \\ & & \ddots & \\ & & & \gamma_n \end{bmatrix}$$

Define $\gamma' \in \mathbb{F}_{p^m}^n$ such that $\gamma'_i = (\beta'_i)^{-1} \prod_{j \neq i} (\alpha'_i - \alpha'_j)$ for all $i = 1, \dots, n$. Now, notice that using the definitions of $\alpha'$ and $\beta'$, we can simplify $\gamma'_i$ for all $i = 1, \dots, n$ as follows:

$$\begin{aligned}
\gamma'_i &= (\beta'_i)^{-1} \prod_{j \neq i} (\alpha_i^{-1} - \alpha_j^{-1})^{-1} \\
&= (\beta'_i)^{-1} \prod_{j \neq i} \left[ \left( -\frac{1}{\alpha_i \alpha_j} \right) (\alpha_i - \alpha_j) \right]^{-1} \\
&= (\beta'_i)^{-1} \prod_{j \neq i} (-\alpha_i \alpha_j) \prod_{j \neq i} (\alpha_i - \alpha_j)^{-1} \\
&= \left( \beta_i \alpha_i^{-(n-k-1)} \prod_{j \neq i} (-\alpha_i \alpha_j) \right)^{-1} \prod_{j \neq i} (-\alpha_i \alpha_j) \prod_{j \neq i} (\alpha_i - \alpha_j)^{-1} \\
&= \alpha_i^{n-k-1} \beta_i^{-1} \prod_{j \neq i} (\alpha_i - \alpha_j)^{-1} \\
&= \alpha_i^{n-k-1} \gamma_i.
\end{aligned}$$

Hence, by Proposition 3.1.7 and Remark 3.1.8, $\mathbf{H}'$ defined as follows is a parity-check matrix for $GRS_{n,k}(\alpha', \beta')$.

$$\mathbf{H}' = \mathbf{K}'\mathbf{D}\mathbf{C} \quad \text{where}$$

$$\mathbf{K}' := \begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha_1^{-1} & \alpha_2^{-1} & \dots & \alpha_n^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{-(n-k-1)} & \alpha_2^{-(n-k-1)} & \dots & \alpha_n^{-(n-k-1)} \end{bmatrix}, \mathbf{D} := \begin{bmatrix} \alpha_1^{n-k-1} & & & \\ & \alpha_2^{n-k-1} & & \\ & & \ddots & \\ & & & \alpha_n^{n-k-1} \end{bmatrix}.$$

Since $\alpha_i \neq \alpha_j$ for all $i \neq j$ and since $\beta_i \neq 0$ for all $i = 1, \dots, n$, it is clear that $det(\mathbf{C}) \neq 0$, meaning that $\mathbf{C}$ is invertible. We also notice that $\mathbf{K}'\mathbf{D}$ is a parity-check matrix for $\mathbf{C}(GRS_{n,k}(\alpha', \beta'))$. Simplifying this product, we notice

$$\mathbf{K}'\mathbf{D} = \begin{bmatrix} \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \dots & \alpha_n^{n-k-1} \\ \alpha_1^{n-k-2} & \alpha_2^{n-k-2} & \dots & \alpha_n^{n-k-2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}.$$

Let $c \in \mathbf{C}(GRS_{n,k}(\alpha,\beta)) = ker(\mathbf{K})$. We observe that the $j^{th}$ entry of $\mathbf{K'D}c$ will be

$$(\mathbf{K'D}c)_j = \sum_{i=1}^{n} \alpha_i^{n-k-j} c_i = 0$$

since $\sum_{i=1}^{n} \alpha_i^j c_i = 0$ for all $j = 0, \ldots, n-k-1$, as $c \in ker(\mathbf{K})$. Hence,

$$\mathbf{C}(GRS_{n,k}(\alpha',\beta')) = ker(\mathbf{K'D}) \supseteq ker(\mathbf{K}) = \mathbf{C}(GRS_{n,k}(\alpha,\beta)).$$

Recall that $\mathbf{C}$ is invertible, so taking the image of $\mathbf{C}(GRS_{n,k}(\alpha',\beta'))$ and $\mathbf{C}(GRS_{n,k}(\alpha,\beta))$ under the map corresponding to multiplication by $\mathbf{C}^{-1}$ gives us that $GRS_{n,k}(\alpha',\beta') \supseteq GRS_{n,k}(\alpha,\beta)$. Since these GRS codes are vector spaces of the same size, the inclusion is an equality.

Case-3 Equality

Let $\mu, \nu, \sigma, \tau \in \mathbb{F}_{p^m}$ such that $\mu\tau \neq \sigma\nu$ and define $\alpha' \in \mathbb{F}_{p^m}^n$ such that $\alpha'_i = \frac{\mu\alpha_i + \nu}{\sigma\alpha_i + \tau}$ for all $i = 1, \ldots, n$. Define the vectors $\lambda, \gamma \in \mathbb{F}_{p^m}^n$ such that $\lambda_i = \mu\alpha_i + \nu$ and $\gamma_i = \sigma\alpha_i + \tau$ for all $i = 1, \ldots, n$. Next, define $\gamma' \in \mathbb{F}_{p^m}^n$ by $\gamma'_i = \gamma_i^{-1}$. We've already shown through Case-1 and Case-2 Equalities that for $\lambda, \gamma,$ and $\gamma'$, there exist vectors $b, b', b'' \in \mathbb{F}_{p^m}^n$ such that

$$GRS_{n,k}(\alpha,\beta) = GRS_{n,k}(\lambda, b) \quad \text{and}$$
$$GRS_{n,k}(\alpha,\beta) = GRS_{n,k}(\gamma, b') = GRS_{n,k}(\gamma', b'')$$

Maintaining that $\mu\tau \neq \sigma\nu$, if we also impose $\mu, \sigma \neq 0$, then

$$\alpha'_i = \frac{\mu\alpha_i + \nu}{\sigma\alpha_i + \tau} = \frac{\mu\sigma^{-1}(\sigma\alpha_i + \tau) + \nu - \mu\sigma^{-1}\tau}{\sigma\alpha_i + \tau} = \mu\sigma^{-1} + \frac{\nu - \mu\sigma^{-1}\tau}{\sigma\alpha_i + \tau}.$$

Define $x := \mu\sigma^{-1}$ and $z := \nu - \mu\sigma^{-1}\tau$. By Case-1 Equality, there exists $\beta' \in (\mathbb{F}_{p^m}^{\times})^n$ such that

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

We composed Case-1, Case-2, and then Case-1 Equalities in order to get this result. By the relation of the equivalent parameters to the original parameters in those cases, $\beta'$ may be chosen such that

$$\beta'_i = \delta\beta_i(\sigma\alpha_i + \tau)^{-(n-k-1)} \prod_{j \neq i} [-(\sigma\alpha_i + \tau)(\sigma\alpha_j + \tau)]$$

for any $\delta \neq 0$ $\forall i = 1, \ldots, n$, and it satisfies the desired equality.

Now, if $\mu = 0$ and $\sigma \neq 0$, we see that $\alpha'_i = \frac{\nu}{\sigma\alpha_i + \tau} = \nu\gamma'_i$ for all $i = 1, \ldots, n$. By Case-1 Equality, there exists a vector $\beta' \in (\mathbb{F}_{p^m}^{\times})^n$ such that

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

We again composed Case-1, Case-2, and then Case-1 Equalities in order to get this result. Hence, by the relationships between equivalent and original parameters in these cases, any $\beta'$ related to the original parameters such that

$$\beta'_i = \delta\beta_i(\sigma\alpha_i + \tau)^{-(n-k-1)} \prod_{j \neq i} [-(\sigma\alpha_i + \tau)(\sigma\alpha_j + \tau)]$$

for any $\delta \neq 0$ $\forall i = 1, \ldots, n$ satisfies the equality.

Finally, if $\mu \neq 0$ and $\sigma = 0$, we see that $\alpha'_i = \frac{\mu\alpha_i + \nu}{\tau} = \tau^{-1}\lambda_i$ for all $i = 1, \ldots, n$. Note that $\tau \neq 0$ since $\mu\tau \neq \sigma\nu$. Again by Case-1 Equality, there exists a vector $\beta' \in (\mathbb{F}_{p^m}^{\times})^n$ such that

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

We applied Case-1 Equality twice in order to get this result, so any $\beta'$ related to the original parameters such that $\beta'_i = \delta\beta_i$ for any $\delta \neq 0$ and for all $i = 1, \ldots, n$ satisfies the equality. It's easy to see that $\beta'$ defined in this way also adheres to

$$\beta'_i = \delta\beta_i(\sigma\alpha_i + \tau)^{-(n-k-1)} \prod_{j \neq i} [-(\sigma\alpha_i + \tau)(\sigma\alpha_j + \tau)]$$

for any $\delta \neq 0$ and for all $i = 1, \ldots, n$ for $\sigma = 0$ since $\tau \neq 0$.

This covers all possible choices of the scalars, $\mu, \nu, \sigma, \tau$ such that $\mu\tau \neq \sigma\nu$, so this proves Case-3 Equality. □

It's easy to see (especially given how we proved these case-equalities) that Case-3 Equality encompasses both Case-1 and Case-2 Equalities. Hence, any pairs of parameters that defines a GRS code that meet the conditions for Case-3 Equality define the same GRS code.

## 4. Goppa Codes

Goppa codes are the linear error-correcting codes that form the cryptographic primitive of the longest-enduring variant of the McEliece public-key cryptosystem. We will define Goppa codes and relate them to GRS codes, in particular interpreting them using a similar definition to Definition 3.1.1 for GRS codes. We will also identify basic properties including those about their dimension and minimum distance.

### 4.1. **Properties of Goppa Codes.**

We will define Goppa codes and establish a few of their basic properties, in particular in how they relate to GRS codes.

We first introduce a function we will need to define a Goppa code.

**Definition 4.1.1.** Let $g \in \mathbb{F}_{p^m}[x]$ be a polynomial of degree $t$ and let $a \in \mathbb{F}_{p^m}^n$ such that $a_i \neq a_j$ for all $i \neq j$ and $g(a_i) \neq 0$ for all $i = 1, \ldots, n$. The *syndrome function* is a map $S : \mathbb{F}_p^n \to \mathbb{F}_{p^m}[x]/\langle g \rangle$ defined by

$$c \mapsto \sum_{i=1}^{n} \frac{c_i}{x - a_i} \pmod{g}.$$

We should be able to see that the choices for the entries of $a$ guarantee that $x - a_i$ is invertible in $\mathbb{F}_{p^m}[x]/\langle g \rangle$. Since $g(a_i) \neq 0$, we have that $x - a_i$ does not divide $g$, meaning then that $\gcd(x - a_i, g) = 1$. As a consequence of Bézout's identity, $x - a_i$ is invertible mod $g$. With this, we understand that $(x - a_i)^{-1}$ is a polynomial in the quotient ring $\mathbb{F}_{p^m}[x]/\langle g \rangle$ for all $i = 1, \ldots, n$, a fact we will make use of shortly.

Note that $g$ is called the *Goppa polynomial* and $a$ is often called the *locator*. Following this definition, we may now define a Goppa code.

**Definition 4.1.2.** Let $g \in \mathbb{F}_{p^m}[x]$ and $a \in \mathbb{F}_{p^m}^n$ be chosen as in Definition 4.1.1. The *Goppa code* defined by the pair $(a, g)$ is $\Gamma(a, g) = ker(S)$, where $S$ is the syndrome function based on the pair $(a, g)$. More explicitly,

$$\Gamma(a,g) = \left\{ c \in \mathbb{F}_p^n : \sum_{i=1}^{n} \frac{c_i}{x - a_i} \equiv 0 \pmod{g} \right\}.$$

We may deduce some basic facts about the dimension and minimum distances of Goppa codes from this definition.

**Proposition 4.1.3.** *Let $\Gamma(a, g)$ be a $k_\Gamma$-dimensional Goppa code defined by degree-$t$ Goppa polynomial $g \in \mathbb{F}_{p^m}[x]$ and locator $a \in \mathbb{F}_{p^m}^n$. The code's dimension satisfies $k_\Gamma \geq n - mt$.*

*Proof.* It is clear from its definition that the syndrome function is a $\mathbb{F}_p$-linear map. Note also that $\mathbb{F}_{p^m}[x]/\langle g \rangle \cong \mathbb{F}_{p^m}^t \cong \mathbb{F}_p^{mt}$, so $\dim_{\mathbb{F}_p}(\mathbb{F}_{p^m}[x]/\langle g \rangle) = mt$. Since $\Gamma(a, g) = ker(S)$, by the Dimension Theorem, we have

$$\dim_{\mathbb{F}_p}(\Gamma(a,g)) = \dim_{\mathbb{F}_p}(\mathbb{F}_p^n) - \dim_{\mathbb{F}_p}(Im(S)).$$

But since $Im(S) \subseteq \mathbb{F}_{p^m}[x]/\langle g \rangle$, $\dim_{\mathbb{F}_p}(\mathbb{F}_{p^m}[x]/\langle g \rangle) \leq mt$, so we conclude

$$k_\Gamma \geq n - mt.$$

□

The following proposition appears as part of Theorem 2.1 in [J].

**Proposition 4.1.4.** *Let $\Gamma(a, g)$ be a Goppa code defined by degree-$t$ Goppa polynomial $g \in \mathbb{F}_{p^m}[x]$ and locator $a \in \mathbb{F}_{p^m}^n$. The minimum distance of $\Gamma(a, g)$ is at least $t + 1$.*

*Proof.* See [J]. □

Given the relationship between the minimum distance of a code and the number of errors it can correct, we understand that the larger a code's minimum distance, the more errors it can correct. The next theorem stated in [J] will reveal that a certain kind of binary Goppa code will possess a greater lower bound on its minimum distance, thereby making these codes more desirable in situations that call for their error-correcting capabilities.

**Definition 4.1.5.** A polynomial is called *separable* if it has no roots of multiplicity greater than one.

**Theorem 4.1.6.** *Let $\Gamma(a, g)$ be a binary Goppa code defined by a separable degree-$t$ Goppa polynomial $g \in \mathbb{F}_{2^m}[x]$ and locator $a \in \mathbb{F}_{2^m}^n$. The minimum distance of $\Gamma(a, g)$ is at least $2t + 1$.*

*Proof.* See Theorem 2.2 in [J]. □

Because a linear code $C$ with minimum distance $d$ can correct up to $\frac{d-1}{2}$ errors, binary Goppa codes defined by degree-$t$ separable Goppa polynomials can correct up to $t$ errors. A typical way to construct a Goppa code with this greater minimum distance is to choose the Goppa polynomial to be an irreducible polynomial in

$\mathbb{F}_{p^m}[x]$. Since irreducible polynomials of degree greater than one have no roots, this ensures they are separable; hence, the last theorem applies. In fact, a Goppa code constructed from an irreducible Goppa polynomial carries a special name: these codes are called *irreducible* Goppa codes. Irreducible Goppa codes are very useful; for example, the McEliece cryptosystem outlined in [B] is based a binary irreducible Goppa code.

We will explore the link between Goppa codes and GRS codes starting with their parity-check matrices. Consider the Goppa code $\Gamma(a, g)$ with syndrome function $S$ mapping $c \mapsto \sum_{i=1}^{n} \frac{c_i}{x-a_i} \pmod g$. We will rewrite $\frac{1}{x-a_i}$ as a polynomial $p_i \in \mathbb{F}_{p^m}[x]/\langle g \rangle$ for all $i = 1, \ldots, n$.

$$\frac{1}{x - a_i} = \sum_{j=1}^{t} p_{j,i} x^{j-1} =: p_i(x)$$

Hence, for all codewords $c \in \Gamma(a, g)$,

$$S(c) = \sum_{i=1}^{n} c_i \sum_{j=1}^{t} p_{j,i} x^{j-1} \equiv 0 \pmod g.$$

Define the coordinate isomorphism on the standard basis of $\mathbb{F}_{p^m}[m]/\langle g \rangle$ as $\varphi : \mathbb{F}_{p^m}[m]/\langle g \rangle \to \mathbb{F}_{p^m}^t$ such that

$$\forall f(x) = \sum_{i=1}^{t} f_i x^i \in \mathbb{F}_{p^m}[m]/\langle g \rangle, \quad \varphi(f) = \begin{bmatrix} f_1 \\ \vdots \\ f_t \end{bmatrix}.$$

We can see that each codeword of $\Gamma(a, g)$ satisfies $c \in \mathbb{F}_p^n$ and $S(c) \equiv 0 \pmod g$. This latter condition is equivalent to the following.

$$\varphi^{-1} \circ \varphi(S(c)) = 0 \pmod g$$

$$\iff \varphi^{-1}\left( \sum_{i=1}^{n} c_i \begin{bmatrix} p_{1,i} \\ \vdots \\ p_{t,i} \end{bmatrix} \right) = 0 \pmod g \quad \text{by the linearity of } \varphi$$

$$\iff \sum_{i=1}^{n} c_i \begin{bmatrix} p_{1,i} \\ \vdots \\ p_{t,i} \end{bmatrix} = 0 \quad \text{since } \varphi^{-1} \text{ is injective}$$

$$\iff \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ p_{2,1} & \cdots & p_{2,n} \\ \vdots & \ddots & \vdots \\ p_{t,1} & \cdots & p_{t,n} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = 0$$

This means the vectors in the kernel of this last matrix that are also in $\mathbb{F}_p^n$ are the codewords of $\Gamma(a, g)$. Define

$$\mathbf{H} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ p_{2,1} & \cdots & p_{2,n} \\ \vdots & \ddots & \vdots \\ p_{t,1} & \cdots & p_{t,n} \end{bmatrix}.$$

Explicitly, $ker(\mathbf{H}) \cap \mathbb{F}_p^n = \Gamma(a, g)$. However, $\mathbf{H}$ has a familiar form, which will be revealed by following the development from Section 2.4 of [J].

We notice that

$$p_i(x) = \frac{1}{x - a_i} \equiv -\frac{g(x) - g(a_i)}{x - a_i} g(a_i)^{-1} \pmod{g} \quad \forall i = 1, \ldots, k.$$

Write out $g(x)$ as the following polynomial: $g(x) = \sum_{i=0}^{t} g_i x^i$. With this, we may rewrite $p_i(x)$.

$$\begin{aligned} p_i(x) &\equiv -\frac{\sum_{j=0}^{t} g_j (x^j - a_i^j)}{x - a_i} g(a_i)^{-1} \pmod{g} \\ &\equiv -\sum_{j=1}^{t} g_j \left( \sum_{l=0}^{j-1} x^{j-l-1} a_i^l \right) g(a_i)^{-1} \pmod{g} \end{aligned}$$

Comparing the coefficients of each power of $x$ in this expression to those of $p_i(x) = \sum_{j=1}^{t} p_{j,i} x^{j-1}$, we get the following set of equalities.

$$\begin{cases} p_{1,i} = -(g_1 + g_2 a_i + \cdots + g_t a_i^{t-1}) g(a_i)^{-1} \\ p_{2,i} = -(g_2 + g_3 a_i + \cdots + g_t a_i^{t-2}) g(a_i)^{-1} \\ \vdots \\ p_{t-1,i} = -(g_{t-1} + g_t a_i) g(a_i)^{-1} \\ p_{t,i} = -(g_t) g(a_i)^{-1} \end{cases}$$

But this then identifies the values of the parity-check matrix $\mathbf{H}$.

$$\mathbf{H} = \begin{bmatrix} -\sum_{j=1}^{t} g_j a_1^{j-1} g(a_1)^{-1} & \cdots & -\sum_{j=1}^{t} g_j a_n^{j-1} g(a_n)^{-1} \\ -\sum_{j=2}^{t} g_j a_1^{j-2} g(a_1)^{-1} & \cdots & -\sum_{j=1}^{t} g_j a_n^{j-2} g(a_n)^{-1} \\ \vdots & \ddots & \vdots \\ -g_t g(a_1)^t & \cdots & -g_t g(a_n)^{-1} \end{bmatrix} = \mathbf{UKC}$$

We define

$$\mathbf{U} := \begin{bmatrix} -g_t & -g_{t-1} & -g_{t-2} & \cdots & -g_1 \\ 0 & -g_t & -g_{t-1} & \cdots & -g_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -g_t \end{bmatrix}, \mathbf{K} := \begin{bmatrix} a_1^{t-1} & a_2^{t-1} & \cdots & a_n^{t-1} \\ a_1^{t-2} & a_2^{t-2} & \cdots & a_n^{t-2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix},$$

$$\mathbf{C} := \begin{bmatrix} g(a_1)^{-1} & & & \\ & g(a_2)^{-1} & & \\ & & \ddots & \\ & & & g(a_n)^{-1} \end{bmatrix}.$$

Note that $deg(g) = t$ implies that $g_t \neq 0$. Because $\mathbf{U}$ is upper-triangular, its determinant, which is the product of its diagonal entries, therefore cannot be 0, meaning $\mathbf{U}$ is invertible. Hence, we will have $\Gamma(a, g) = ker(\mathbf{KC}) \cap \mathbb{F}_p^n$ as well since for any vector $x \in \mathbb{F}_{p^m}^n$,

$$\mathbf{UKC}x = 0 \iff \mathbf{KC}x = 0.$$

For the same reason, if we multiply $\mathbf{KC}$ by any permutation matrix from the left, because permutation matrices are invertible, the kernel of this product intersected with $\mathbb{F}_{p^m}^n$ will also be $\Gamma(a, g)$. One such product will be $\mathbf{K'C}$ such that

$$\mathbf{K'} := \begin{bmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{t-1} & a_2^{t-1} & \dots & a_n^{t-1} \end{bmatrix}.$$

However, notice that by the discussion following Remark 3.1.8, $\mathbf{K'C}$ is a parity-check matrix for a GRS. The particular GRS code of which it is a parity-check matrix is $GRS_{n,k}(\alpha, \beta)$ such that $t = n - k$, $\alpha = a$, and

$$\beta_i^{-1} \prod_{j \neq i} (\alpha_i - \alpha_j)^{-1} = g(a_i)^{-1} \iff \beta_i = \frac{g(a_i)}{\prod_{j \neq i} \alpha_i - \alpha_j} \quad \forall i = 1, \dots, n$$

by Proposition 3.1.7. Choosing $\alpha$ and $\beta$ as just described, we see $ker(\mathbf{K'C}) = GRS_{n,k}(\alpha, \beta)$, meaning then that $\Gamma(a, g) = GRS_{n,k}(\alpha, \beta) \cap \mathbb{F}_p^n$. This is the key relationship between Goppa codes and GRS codes: Goppa codes are subfield subcodes of GRS codes. We'll state this formally as a proposition.

**Proposition 4.1.7.** *Consider the Goppa code $\Gamma(a, g)$ defined by degree-$t$ Goppa polynomial $g \in \mathbb{F}_{p^m}[x]$ and locator $a \in \mathbb{F}_{p^m}^n$. This code is the subfield subcode of $GRS_{n,k}(\alpha, \beta)$ such that $t = n - k$, $\alpha = a$, and $\beta_i = \frac{g(a_i)}{\prod_{j \neq i} \alpha_i - \alpha_j}$ for all $i = 1, \dots, n$.*

*Proof.* We have proven this with the above discussion. □

We should note that the family of codes wherein each member is a subfield subcode of a GRS code carries a special name.

**Definition 4.1.8.** An *Alternant* code is a subfield subcode of a GRS code.

It's important to note that by what we've shown, a Goppa code is necessarily an Alternant code, but this does not imply that an Alternant code is a Goppa code. A particular relationship given in Proposition 4.1.7 between the parameters of a Goppa code and its corresponding GRS code must be satisfied in order for the subfield subcode of the GRS code to be a Goppa code. Returning to this relationship from Proposition 4.1.7, we see that the $n$ entries of $\beta$ are determined by the $t+1 \leq n$ coefficients of $g$ (along with the entries of $a$), so there's less freedom of choice in choosing $\beta$ for a GRS code whose subfield subcode is a Goppa code than there is in choosing $\beta$ for an arbitrary GRS code. We won't prove this rigorously here, but one can be made to believe that as a result, we can define a GRS code whose subfield subcode cannot be a Goppa code.

4.2. **Reinterpreting Goppa Codes.** With the knowledge that a Goppa code is a subfield subcode of some GRS code, we will use the polynomial-evaluation definition of GRS codes to develop an analogous characterization for Goppa codes.

The main tool we will use to develop this characterization is the following result given as a part of Theorem 1 from [SB].

**Theorem 4.2.1.** *Let $GRS_{n,k}(\alpha, \beta)$ be a $(n, k)$ GRS code whose subfield subcode is a $k_\Gamma$-dimensional Goppa code $\Gamma(a, g)$. If $\mathbf{G}$ is a generator matrix for $GRS_{n,k}(\alpha, \beta)$,*

*then there exists a matrix* $\mathbf{\Gamma} \in \mathcal{M}_{k\times k_\Gamma}(\mathbb{F}_{p^m})$ *such that a generator matrix* $\mathbf{G_\Gamma}$ *for* $\Gamma(a,g)$ *can be expressed as* $\mathbf{G_\Gamma} = \mathbf{G\Gamma}$*. Furthermore,* $\mathbf{\Gamma}$ *can be found from* $\alpha$ *and* $\beta$*.*

*Proof.* See [SB]. □

We will now give the polynomial-evaluation characterization for a Goppa code.

**Proposition 4.2.2.** *Let* $\Gamma(a,g)$ *be a* $(n,k_\Gamma)$ *Goppa code that is the subfield subcode of* $GRS_{n,k}(\alpha,\beta)$ *where these codes satisfy the relationship outlined in Proposition 4.1.7. We have that*

$$\Gamma(a,g) = \{(\beta_1 q(a_1),\dots,\beta_n q(a_n)) : q \in \mathcal{P}\}$$

*where* $\mathcal{P}$ *is a* $\mathbb{F}_p$*-linear subspace of* $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ *that is of dimension* $k_\Gamma$ *and* $\mathcal{P}$ *can be determined from* $a$ *and* $\beta$*.*

*Proof.* Let $\mathbf{G}$ be a generator matrix for $GRS_{n,k}(\alpha,\beta)$ and let $C_i$ denote its $i^{th}$ column. By Theorem 4.2.1, there exists a matrix $\mathbf{\Gamma} \in \mathcal{M}_{k\times k_\Gamma}(\mathbb{F}_{p^m})$ such that $\mathbf{G\Gamma}$ is a generator matrix for $\Gamma(a,g)$. Let $\gamma_{i,j}$ denote the $(i,j)^{th}$ entry of $\mathbf{\Gamma}$. Explicitly, we write out the generator matrix $\mathbf{G\Gamma}$.

$$\begin{aligned}\mathbf{G\Gamma} &= \begin{bmatrix} C_1 & C_2 & \dots & C_n \end{bmatrix} \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \dots & \gamma 1,k_\Gamma \\ \gamma_{2,1} & \gamma_{2,2} & \dots & \gamma 2,k_\Gamma \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{k,1} & \gamma_{k,2} & \dots & \gamma k,k_\Gamma \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^k \gamma_{i,1} C_i & \sum_{i=1}^k \gamma_{i,2} C_i & \dots & \sum_{i=1}^k \gamma_{i,k_\Gamma} C_i \end{bmatrix}\end{aligned}$$

But since $GRS_{n,k}(\alpha,\beta) = \{(\beta_1 p(a_1),\dots,\beta_n p(a_n)) : p \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})\}$, each column $C_i$ is a basis vector of this code, meaning that we can write it as

$$C_i = \begin{bmatrix} \beta_1 p_i(a_1) \\ \beta_2 p_i(a_2) \\ \vdots \\ \beta_n p_i(a_n) \end{bmatrix}$$

for some $p_i \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ such that $\{p_i(x) : i = 1,\dots,k\}$ is a basis for $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$.

With this, we write the $j^{th}$ column of $\mathbf{G\Gamma}$ is

$$(\mathbf{G\Gamma})_j = \sum_{i=1}^k \gamma_{i,j} C_i = \begin{bmatrix} \beta_1 \sum_{i=1}^k \gamma_{i,j} p_i(a_1) \\ \beta_2 \sum_{i=1}^k \gamma_{i,j} p_i(a_2) \\ \vdots \\ \beta_n \sum_{i=1}^k \gamma_{i,j} p_i(a_n) \end{bmatrix} = \begin{bmatrix} \beta_1 q_j(a_1) \\ \beta_2 q_j(a_2) \\ \vdots \\ \beta_n q_j(a_n) \end{bmatrix}$$

such that $q_j(x) := \left(\sum_{i=1}^k \gamma_{i,j} p_i\right)(x) \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$. Thus, we can associate the $j^{th}$ column of $\mathbf{G\Gamma}$ to a polynomial $q_j \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$.

Since each codeword of $\Gamma(a,g)$ can be expressed as a linear combination over $\mathbb{F}_p$ of the columns of $\mathbf{G\Gamma}$, we have for all codewords $c \in \Gamma(a,g)$, there exists

$\lambda_1, \ldots, \lambda_{k_\Gamma} \in \mathbb{F}_p$ such that

$$c = \sum_{j=1}^{k_\Gamma} \lambda_j \begin{bmatrix} \beta_1 q_j(a_1) \\ \beta_2 q_j(a_2) \\ \vdots \\ \beta_n q_j(a_n) \end{bmatrix} = \begin{bmatrix} \beta_1 \left( \sum_{j=1}^{k_\Gamma} \lambda_j q_j \right)(a_1) \\ \beta_2 \left( \sum_{j=1}^{k_\Gamma} \lambda_j q_j \right)(a_2) \\ \vdots \\ \beta_n \left( \sum_{j=1}^{k_\Gamma} \lambda_j q_j \right)(a_n) \end{bmatrix}.$$

We see that $q(x) := \left( \sum_{j=1}^{k_\Gamma} \lambda_j q_j \right)(x) \in \text{span}_{\mathbb{F}_p}\{q_1, \ldots, q_{k_\Gamma}\}$. Thus, we've shown for each codeword $c \in \Gamma(a, g)$, there exists a polynomial $q \in \text{span}_{\mathbb{F}_p}\{q_1, \ldots, q_{k_\Gamma}\}$ such that $c = (\beta_1 q(a_1), \ldots, \beta_n q(a_n))$. Hence,

$$\Gamma(a, g) \subseteq \{(\beta_1 q(a_1), \ldots, \beta_n q(a_n)) : q \in \text{span}_{\mathbb{F}_p}\{q_1, \ldots, q_{k_\Gamma}\}.$$

The inclusion in the other direction is immediately apparent since $\text{span}_{\mathbb{F}_p}\{q_1, \ldots, q_{k_\Gamma}\} \subseteq \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$. Defining $\mathcal{P} := \text{span}_{\mathbb{F}_p}\{q_1, \ldots, q_{k_\Gamma}\}$, we finally get

$$\Gamma(a, g) = \{(\beta_1 q(a_1), \ldots, \beta_n q(a_n)) : q \in \mathcal{P}\}.$$

□

Note that to explicitly determine $\mathcal{P}$, we need to know the basis used to form the generator matrix $\mathbf{G}$ and then we need to use this along with the GRS code parameters $\alpha = a$ and $\beta$ to identify $\mathbf{\Gamma}$ by Theorem 4.2.1. By the discussion following Proposition 3.1.6, we can take $\mathbf{G}$ to be the canonical generator matrix for $GRS_{n,k}(\alpha, \beta)$, meaning we can explicitly construct it given $a$ and $\beta$. Hence, we can determine $\mathcal{P}$ given $a$ and $\beta$.

## Part 2. Attacks on the McEliece PKC

### 5. The McEliece PKC and the Two Classes of Attacks Against It

We briefly outline the scheme for a McEliece PKC and we describe the two main classes of attacks against it, message attacks and structural attacks. The complexity of the best known message attacks is considered, ultimately motivating us to explore in more detail structural attacks as a means of cryptanalyzing the McEliece PKC.

5.1. **The McEliece PKC.** We reiterate the general form of the cryptosystem McEliece introduced in [M].

**Algorithm 1:** The McEliece PKC

- Private Key
  - $\mathbf{G}$, a $n \times k$ generator matrix for a $(n, k)$ linear code $C$
  - $\mathbf{S} \in \mathrm{GL}_k(\mathbb{F}_{p^m})$, the scrambler matrix
  - $\mathbf{P}$, a $n \times n$ permutation matrix
  - $D_G$, an efficient error-correction algorithm specific to the code $C$ that can only be used if one has knowledge of $\mathbf{G}$
- Public Key
  - $\mathbf{M} = \mathbf{PGS}$, which is the generator $\mathbf{G}$ that has been masked by $\mathbf{P}$ and $\mathbf{S}$. In fact, $\mathbf{M}$ is a generator matrix for $\mathbf{P}(C)$ given that the invertibility of $\mathbf{S}$ implies $Im(\mathbf{GS}) = Im(\mathbf{G}) = C$.
  - $t$, the number of errors $C$ can correct
- Encryption
  - We generate ciphers from plaintexts, which are $k$-tuples over $\mathbb{F}_{p^m}$.
  - To generate a cipher $c$ from a plaintext $m \in \mathbb{F}_{p^m}^k$, we first choose a vector $z \in \mathbb{F}_{p^m}^n$ such that $\omega(z) = t$ and then we encode $m$ as a codeword of $\mathbf{P}(C)$ that has been damaged in the non-zero entries of $z$ as follows:
$$c = \mathbf{M}m + z.$$
- Decryption
  - To decrypt a cipher $c$, we first calculate
$$c' = \mathbf{P}^{-1}c = \mathbf{GS}m + \mathbf{P}^{-1}z.$$
We notice that because $Im(\mathbf{GS}) = C$ and $\omega(\mathbf{P}^{-1}z) = t$, $c'$ is a codeword of $C$ that has been subjected to $t$ errors.
  - Because $C$ can correct $t$ errors, we apply the error correction algorithm to $c'$, recovering $D_G(c') = \mathbf{GS}m$ given that $d(c', \mathbf{GS}m) = t$.
  - Let $\mathbf{G}_{LI}$ denote a left inverse of $\mathbf{G}$. We recover the plaintext with a final calculation:
$$\mathbf{S}^{-1}\mathbf{G}_{LI}\mathbf{GS}m = m.$$

At the heart of the McEliece cryptosystem is a trapdoor one-way function. The one-way function is the application of $t$ errors to a codeword of a linear code $C$. Indeed, the one-wayness of the function comes from the fact that identifying if there is a codeword in a binary code $C$ of distance at most $t$ to a given vector $y \in \mathbb{F}_2^n$ corresponds to a $NP$-Complete problem. This problem is equivalent to identifying

for a given $(n, k)$ binary code $C$ and vector $y \in \mathbb{F}_2^n$ with syndrome $s \in \mathbb{F}_2^{n-k}$ if there is an error vector $z \in \mathbb{F}_2^n$ with the same syndrome $s$ such that $\omega(z) \leq t$ and $y = c + z$ for some codeword $c \in C$. This latter problem was proven to be $NP$-Complete in [BMT]. The trapdoor is the code-specific, efficient, error-correction algorithm $D_G$.

5.2. **Classes of Attacks Against the McEliece PKC.** Attacking the McEliece PKC means trying to decrypt a McEliece cipher without access to the private key. The two primary strategies to accomplish this are encompassed in the two following classes of attacks:

- *message attacks*, which replace the trapdoor with an efficient, generic error-correction algorithm;
- *structural attacks*, which consist of ad hoc methods to reconstruct the generator $\mathbf{G}$ given the information gleaned from the public key, thereby granting an attacker access to the trapdoor.

We will now briefly study message attacks by examining the complexity of the best known generic error-correction algorithm, which is Information Set Decoding (ISD). ISD solves problems from the family of Computational Syndrome Decoding (CSD) problems, which we define using the following definition from [TS].

**Definition 5.2.1.** The *Computational Syndrome Decoding* problem considering the input $(n, k, t)$ is denoted $CSD_{n,k,t}$ and it consists of correcting $t$ errors applied to a codeword of a binary $(n, k)$ code.

When the code $C$ in the definition of the McEliece PKC is a binary code, its cryptographic primitive is exactly a computational syndrome decoding problem. Consider a McEliece scheme based on a $(n, k)$ binary code $C$ that can correct $t$ errors where $t = \mathbf{o}(n)$. Hence, by the result of [TS], for any variant of ISD, the expected number of binary operations needed to decrypt a cipher from such a McEliece scheme is $\mathcal{O}(2^{ct(1+\mathbf{o}(1))})$ where $c = \log_2 \frac{1}{1-R}$ such that $R = lim_{n\to\infty} \frac{k}{n}$. In [TS], it is noted that this result extends to the McEliece scheme based on binary Goppa codes as the error-correcting capacity for such codes is $t = \mathcal{O}(n/\log(n))$. The McEliece scheme based on binary Goppa codes is the longest-enduring variant and the one currently proposed for NIST's PQC standardization project, so this result shows message attacks to be intractable against it given that the best instance of such an attack is of exponential complexity.

Structural attacks, however, are in a position to better exploit the algebraic structure of the particular codes used to form a McEliece scheme. As a result, there are instances of structural attacks of polynomial complexity against McEliece schemes based on codes other than Goppa codes (notably GRS codes and random subcodes thereof) that can give an attacker access to the McEliece trapdoor. It is for this reason and because of the intractability of message attacks against the McEliece scheme based on binary Goppa codes that we will focus on structural attacks in our analysis going forward.

## 6. The Sidelnikov-Shestakov Attack

The Sidelnikov-Shestakov (S-S) attack is a structural attack of polynomial complexity that grants an attacker access to the trapdoor of a McEliece scheme based

on GRS codes. We will present a slight modification to the S-S attack Wieschebrink delivered in [W] that reduces its complexity. We will also identify a subset of the family of Goppa codes for which a McEliece scheme based on any code from this subset will also be vulnerable to the attack. Lastly, we will consider codes outside of this subset and outline the difficulties of applying the S-S attack to a McEliece scheme based on Goppa codes in general.

6.1. **The Sidelnikov-Shestakov Attack on GRS Codes.** We will begin by adapting the McEliece scheme presented in Algorithm 1 so that the secret code is a GRS code. For such a scheme, the private and public keys are as follows.

- Private key
  - $\mathbf{G}$, a $n \times k$ generator matrix for $GRS_{n,k}(\alpha, \beta)$
  - $\mathbf{S} \in \mathrm{GL}_k(\mathbb{F}_{p^m})$, the scrambler matrix
  - $(\alpha, \beta)$, the code parameters
- Public key
  - $\mathbf{M} = \mathbf{GS}$, the public matrix
  - $t$, the number of errors $GRS_{n,k}(\alpha, \beta)$ can correct

We note the primary differences between the above and its analogue in Algorithm 1 is the removal of the permutation matrix $\mathbf{P}$ and the replacement of the error-correction algorithm $D_G$ with the GRS code parameters $(\alpha, \beta)$. The former change follows from a shift in our perspective as attackers. For the S-S attack, we are satisfied with recovering a given message up to permutation from a cipher since the permutation can be undone by methods based on the Support Splitting Algorithm introduced by Sendrier in [S]. As such, the permutation matrix from Algorithm 1 disappears into the generator matrix $\mathbf{G}$ as we may now think of a permutation of the GRS code being the secret code. The latter change results from the efficient, code-specific error-correction algorithms for GRS codes (as well as for Goppa codes) being accessible immediately from the code parameters. Hence, the goal of the S-S attack becomes to recover the code parameters from public key.

In fact, the goal of the S-S attack can be changed to the recovery of equivalent parameters defining the same GRS code, as per the case-equalities outlined in Theorem 3.2.1. Any error-correction algorithm for a $(n, k)$ GRS code defined by equivalent parameters $(\alpha', \beta')$ will apply to $GRS_{n,k}(\alpha, \beta)$ because they are the same codes. It is for this reason that we may start the S-S attack with the knowledge of a part of the code parameters, as Wieschebrink suggests in [W].

**Lemma 6.1.1.** *Without loss of generality,* $\alpha_1 = 0$*,* $\alpha_2 = 1$*, and* $\beta_1 = 1$*.*

*Proof.* Take $\mu = (\alpha_2 - \alpha_1)^{-1}$, $\nu = (-\alpha_1)(\alpha_2 - \alpha_1)^{-1}$, and $\eta = \beta_1^{-1}$. These satisfy $\mu, \eta \neq 0$. Define $\alpha', \beta' \in \mathbb{F}_{p^m}^n$ such that $\alpha'_i = \mu\alpha_i + \nu$ and $\beta'_i = \eta\beta_i$ for all $i = 1, \ldots, n$. It is clear that $\alpha'_1 = 0$, $\alpha'_2 = 1$, and $\beta'_1 = 1$. Because it suffices to solve for any pair of equivalent parameters to gain access to the McEliece trapdoor, we will instead choose to solve for $(\alpha', \beta')$, which we already know in part. □

We will now present our slight modification of Wieschebrink's presentation of the Sidelnikov-Shestakov attack in [W]. Without loss of generality, by possibly permuting the rows of $\mathbf{G}$, we assume that the first $k$ rows $\mathbf{G}$ are linearly independent. We also assume that $2 \leq k \leq n - 2$ so that we may proceed with the attack. We

start by transposing $\mathbf{M}$ and bringing $\mathbf{M}^\intercal$ to its RREF form, denoted $E(\mathbf{M}^\intercal)$.

$$\mathbf{M}^\intercal \sim E(\mathbf{M}^\intercal) = [\mathbf{I}_k|A] = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_k \end{bmatrix}$$

By the first assumption, the first $k \times k$ submatrix of $E(\mathbf{M}^\intercal)$ is the identity. Notice that we write $E(\mathbf{M}^\intercal)$ as a matrix of row vectors where $R_i$ denotes its $i^{th}$ row. Because $\text{Row}(E(\mathbf{M}^\intercal)) = Im(\mathbf{GS}) = GRS_{n,k}(\alpha, \beta)$, the rows of $E(\mathbf{M}^\intercal)$ are codewords of $GRS_{n,k}(\alpha, \beta)$. Hence, we write for all $i = 1, \ldots, k$,

$$R_i = (\beta_1 p_{R_i}(\alpha_1), \ldots, \beta_n p_{R_i}(\alpha_n)) \text{ for some } p_{R_i} \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m}).$$

Because the first $k \times k$ submatrix of $E(\mathbf{M}^\intercal)$ is $\mathbf{I}_k$, we observe

$$(R_i)_j = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases} \quad \forall i = 1, \ldots, k, \implies p_{R_i}(\alpha_j) = 0 \ \forall j \in \{1, \ldots, k\} \backslash \{i\} \text{ since } \beta_j \neq 0.$$

This immediately implies $(x - \alpha_j) \mid p_{R_i}(x)$ for all $j \in \{1, \ldots, k\} \backslash \{i\}$, which, because degree-1 factors are irreducible, implies $\prod_{j \in \{1,\ldots,k\} \backslash \{i\}} (x - \alpha_j) \mid p_{R_i}(x)$. Given that $deg(p_{R_i}) \leq k - 1$ and the degree of this product of degree-1 factors is $k - 1$, $p_{R_i}$ must be a non-zero scalar multiple of this product. Written explicitly,

$$p_{R_i}(x) = c_i \cdot \prod_{j \in \{1,\ldots,k\} \backslash \{i\}} (x - \alpha_j) \quad \text{for some } c_i \in \mathbb{F}_{p^m}^\times.$$

The strategy in the S-S attack to recover the code parameters is to divide non-zero entries of the rows of $E(\mathbf{M}^\intercal)$ to get degree-1 rational functions in the entires of the parameters and to solve for code parameters entry by entry. We begin by recovering $\alpha$.

Pick $j \in \{k + 1, \ldots, n\}$. The quotient $\frac{(R_1)_j}{(R_2)_J}$ simplifies as follows:

$$\frac{(R_1)_j}{(R_2)_j} = \frac{\beta_j p_{R_1}(\alpha_j)}{\beta_j p_{R_2}(\alpha_j)} = \frac{c_1 \prod_{r \in \{1,\ldots,k\} \backslash \{1\}} (\alpha_j - \alpha_r)}{c_2 \prod_{r \in \{1,\ldots,k\} \backslash \{2\}} (\alpha_j - \alpha_r)} = \frac{c_1(\alpha_j - \alpha_2)}{c_2(\alpha_j - \alpha_1)}.$$

We recall that we assumed $\alpha_1 = 0$ and $\alpha_2 = 1$, so $\frac{(R_1)_j}{(R_2)_j} = \frac{c_1(\alpha_j - 1)}{c_2(\alpha_j)}$. The original attack presented by Wieschebrink had us make a guess for the value of $\frac{c_1}{c_2}$, but under the slightly stronger hypothesis that $k \geq 3$, we can compute it without any guesswork in a fixed number of operations by Lemma 6.1.2. Thus, the only unknown in this last expression is $\alpha_j$. Rearranging this equation to reflect this, we get $\frac{c_2}{c_1} \frac{(R_1)_j}{(R_2)_j} = \frac{\alpha_j - 1}{\alpha_j}$, which relates $\alpha_j$ to known values by a fractional linear transformation. Fractional linear transformations are bijective, so we may uniquely recover $\alpha_j$ for all $j = k + 1, \ldots, n$ from these equations.

To recover $\alpha_i$ for all $i \in \{3, \ldots, k\}$, we pick distinct values $j_1, j_2 \in \{k+1, \ldots, n\}$ and calculate the quotients $\frac{(R_1)_{j_1}}{(R_i)_{j_1}}$ and $\frac{(R_1)_{j_2}}{(R_i)_{j_2}}$. We get

$$\frac{(R_1)_{j_1}}{(R_i)_{j_1}} = \frac{c_1}{c_i} \frac{\alpha_{j_1} - \alpha_i}{\alpha_{j_1}} \quad \text{and} \quad \frac{(R_1)_{j_2}}{(R_i)_{j_2}} = \frac{c_1}{c_i} \frac{\alpha_{j_2} - \alpha_i}{\alpha_{j_2}}.$$

Rearranging these expressions for $\frac{c_1}{c_i}$ and then equating them, we get a an expression for $\alpha_i$.

$$\frac{\alpha_{j_1}}{\alpha_{j_1}-\alpha_i}\frac{(R_1)_{j_1}}{(R_i)_{j_1}} = \frac{c_1}{c_i} = \frac{\alpha_{j_2}}{\alpha_{j_2}-\alpha_i}\frac{(R_1)_{j_2}}{(R_i)_{j_2}} \implies \frac{(R_1)_{j_1}(R_i)_{j_2}}{(R_1)_{j_2}(R_i)_{j_1}}\frac{\alpha_{j_1}}{\alpha_{j_2}} = \frac{\alpha_{j_1}-\alpha_i}{\alpha_{j_2}-\alpha_i}$$

The rightmost equation relates an expression consisting of known values to the unknown $\alpha_i$ by a fractional linear transformation since $\alpha_{j_1} \neq \alpha_{j_2}$, so we can use this equation to uniquely solve for $\alpha_i$ for all $i = 3, \ldots, k$.

We will recover $\beta$ in an approach that differs from Wieschebrink's in [W]. First, we divide the diagonal entries of $E(\mathbf{M}^\intercal)$, giving us

$$\frac{(R_i)_i}{(R_j)_j} = \frac{\beta_i c_i \prod_{r\in\{1,\ldots,k\}\setminus\{i\}}(\alpha_i-\alpha_r)}{\beta_j c_j \prod_{r\in\{1,\ldots,k\}\setminus\{j\}}(\alpha_j-\alpha_r)} \text{ for some } i,j \in \{1,\ldots,k\} \text{ such that } i \neq j.$$

Noticing that each element of the diagonal of $E(\mathbf{M}^\intercal)$ is 1 and taking $i = 1$, this last equation may be rearranged into the following expression:

$$\beta_j = \frac{c_1}{c_j}\frac{\prod_{r\in\{2,\ldots,k\}}(-\alpha_r)}{\prod_{r\in\{1,\ldots,k\}\setminus\{2\}}(\alpha_j-\alpha_r)}.$$

Since we solved for $\alpha$ already and by Lemma 6.1.2, the right-hand side is fully known and can be calculated in $\mathcal{O}(k)$ operations. We use this last equation to uniquely solve for $\beta_j$ for all $j = 2, \ldots, k$.

Next, we divide different non-zero entries of $R_1$. Pick $j \in \{k+1, \ldots, n\}$ and compute

$$\frac{(R_1)_1}{(R_1)_j} = \frac{1\cdot c_1 \prod_{r\in\{2,\ldots,k\}}(0-\alpha_r)}{\beta_j c_1 \prod_{r\in\{2,\ldots,k\}}(\alpha_j-\alpha_r)} \iff \beta_j = (R_1)_j \prod_{r\in\{2,\ldots,k\}}\frac{-\alpha_r}{\alpha_j-\alpha_r}.$$

Again, everything is known on the right-hand side of this last equation, which lets us uniquely determine $\beta_j$ for all $j = k+1, \ldots, n$. With this, we've recovered the code parameters entirely. ♢

This improves the complexity of the variant of the S-S attack Wieschebrink outlines in [W]. Wieschebrink's suggestion to guess $\frac{c_1}{c_2}$ and approach for solving for $\beta$ resultin his attack having both an expected and worst-case complexity of $\mathcal{O}((n+p^m k)k^2)$. The approach outlined above reduces the complexity of the attack to $\mathcal{O}(nk^2)$, the complexity of row-reducing $\mathbf{M}^\intercal$. We should however note the difference in hypotheses: Wieschebrink's original proposal only required $k \geq 2$ whereas the above approach making use of Lemma 6.1.2 requires $k \geq 3$.

The following lemma is what let us avoid the need to guess in the above attack.

**Lemma 6.1.2.** *If $GRS_{n,k}(\alpha,\beta)$ is a GRS code such that. $3 \leq k$, then $\frac{c_1}{c_i}$ can be computed in $\mathcal{O}(1)$ operations for all $i = 1, \ldots, k$.*

*Proof.* We start with the $(n,k)$ GRS code over $\mathbb{F}_{p^m}$ $GRS_{n,k}(\alpha,\beta)$ and assume that $k \geq 3$ so that any generator matrix of $GRS_{n,k}(\alpha,\beta)$ has 3 columns. Thus, $E(\mathbf{M}^\intercal)$ has at least 3 rows, so we may divide the non-zero entries of row 1 of $E(\mathbf{M}^\intercal)$ by

those of row 2 and likewise divide the non-zero entries of row 1 by those of row 3. This gives us the following equations.

$$\frac{(R_1)_j}{(R_2)_j} = \frac{c_1}{c_2}\frac{\alpha_j - 1}{\alpha_j} \qquad \forall j \in \{k+1, \ldots, n\} \quad \text{and}$$
$$\frac{(R_1)_j}{(R_3)_j} = \frac{c_1}{c_3}\frac{\alpha_j - \alpha_3}{\alpha_j} \qquad \forall j \in \{k+1, \ldots, n\}$$

We recall that we assumed $\alpha_1 = 0$ and $\alpha_2 = 1$. We can then rewrite these equations as follows.

$$\left(\frac{(R_1)_j}{(R_2)_j} - \frac{c_1}{c_2}\right)\left(\frac{c_1}{c_2}\right)^{-1} = -\alpha_j \qquad \forall j \in \{k+1, \ldots, n\}$$
$$\left(\frac{(R_1)_j}{(R_3)_j} - \frac{c_1}{c_3}\right)\left(\frac{c_1}{c_3}\right)^{-1}\alpha_3^{-1} = -\alpha_j \qquad \forall j \in \{k+1, \ldots, n\}$$

Equating the top and bottom expressions for $-\alpha_j$, we get

$$\frac{(R_1)_j}{(R_2)_j}\left(\frac{c_1}{c_2}\right)^{-1} - 1 = \left(\frac{(R_1)_j}{(R_3)_j}\left(\frac{c_1}{c_3}\right)^{-1} - 1\right)\alpha_3^{-1} \qquad \forall j \in \{k+1, \ldots, n\}.$$

Define $x := \left(\frac{c_1}{c_2}\right)^{-1}$, $y := \left(\frac{c_1}{c_3}\right)^{-1}$, $z := \alpha_3^{-1}$, $a_j := \frac{(R_1)_{j+k}}{(R_2)_{j+k}}$, and $b_j := \frac{(R_1)_{j+k}}{(R_3)_{j+k}}$. Finding $\frac{c_1}{c_2}$ amounts to solving for $x$ in the following equations where $a_j$ and $b_j$ are the known values.

$$a_j x - 1 = (b_j y - 1)z \qquad \forall j \in \{1, \ldots, n-k\}$$

By rearranging for $z$ and equating the $z$s, we get

$$\frac{a_1 x - 1}{b_1 y - 1} = \frac{a_2 x - 1}{b_2 y - 1} = \cdots = \frac{a_{n-k} x - 1}{b_{n-k} y - 1} \neq 0.$$

Of course, we are assuming $b_j y - 1 \neq 0$ and $a_j x - 1 \neq 0$ for all $j$. However, if there exists some $j \in \{1, \ldots, n-k\}$ such that one of $b_j y - 1 = 0$ or $a_j x - 1 = 0$, then the other would also necessarily hold since $z \neq 0$, so we have a solution for $x$ as desired.

From the leftmost equality, we get

$$\begin{aligned} & (a_1 x - 1)(b_2 y - 1) = (a_2 x - 1)(b_1 y - 1) \\ \iff & (a_1 b_2 x - b_2 + b_1 - a_2 b_1 x)y = (a_1 - a_2)x \\ \iff & y = \frac{(a_1 - a_2)x}{(a_1 b_2 - a_2 b_1)x + b_1 - b_2}. \end{aligned}$$

The last if and only if holds because $a_1 b_2 x - b_2 + b_1 - a_2 b_1 x \neq 0$. Suppose for a contradiction that it were 0. This implies $(0)y = 0 = (a_1 - a_2)x$ by the second line above. Since $\frac{(R_1)_j}{(R_2)_j} \neq \frac{(R_1)_i}{(R_2)_i}$ for all $i \neq j \in \{k+1, \ldots, n\}$, we have $a_i \neq a_j$ for all $i \neq j$. Thus, we must have $x = 0$. But since $x = \left(\frac{c_1}{c_2}\right)^{-1}$ is invertible, $x \in \mathbb{F}_{p^m}^{\times}$, which means $x = 0$ is a contradiction.

Using distinct $i, j \in \{2, \dots, n-k\}$ and the above equation for $y$, we can recover $x$. We start with the equality

$$\frac{a_i x - 1}{b_i y - 1} = \frac{a_j x - 1}{b_j y - 1}.$$

Proceeding as before, we write

$$\begin{aligned} & (a_i x - 1)(b_j y - 1) = (a_j x - 1)(b_i x - 1) \\ \iff & (a_i b_j y - a_j b_i y + a_j - a_i)x = (b_j - b_i)y \\ & \iff x = \frac{b_j - b_i}{a_i b_j - a_j b_i + (a_j - a_i)y^{-1}}. \end{aligned}$$

This last if and only if holds because $a_i b_j y - a_j b_i y + a_j - a_i \neq 0$. If it were 0, then, again, we have $0 = (b_j - b_i)y$. For all $l \in \{3, \dots, k\}$, $\frac{(R_1)_j}{(R_l)_j} \neq \frac{(R_1)_i}{(R_l)_i} \quad \forall i \neq j \in \{k+1, \dots, n\}$, so $b_j \neq b_i$ for our choice of distinct $i, j$. This implies $y = \left(\frac{c_1}{c_3}\right)^{-1} = 0$, which of course is a contradiction since $y \in \mathbb{F}_{p^m}^{\times}$.

Substituting our previous expression for $y$ in the place of $y^{-1}$ in this last expression gives

$$x = \frac{b_j - b_i}{a_i b_j - a_j b_i + (a_j - a_i)\left(\frac{(a_1 b_2 - a_2 b_1)x + b_1 - b_2}{(a_1 - a_2)x}\right)}.$$

From this expression for $x$, we find

$$x\left[a_i b_j - a_j b_i + (a_j - a_i)\left(\frac{a_1 b_2 - a_2 b_1 + (b_1 - b_2)x^{-1}}{a_1 - a_2}\right)\right] = b_j - b_i$$

$$(*) \quad \iff \left[(a_i b_j - a_j b_i) + \frac{(a_j - a_i)(a_1 b_2 - a_2 b_1)}{a_1 - a_2}\right] x = b_j - b_i + \frac{(a_j - a_i)(b_2 - b_1)}{a_1 - a_2}.$$

Finally, we write

$$x = \frac{(b_j - b_i) + \frac{(a_j - a_i)(b_2 - b_1)}{a_1 - a_2}}{(a_i b_j - a_j b_i) + \frac{(a_j - a_i)(a_1 b_2 - a_2 b_1)}{a_1 - a_2}}.$$

We conclude this because $(a_i b_j - a_j b_i) + \frac{(a_j - a_i)(a_1 b_2 - a_2 b_1)}{a_1 - a_2} \neq 0$. Recalling that we derived all preceding equations from the equality

$$\frac{a_1 x - 1}{b_1 y - 1} = \frac{a_2 x - 1}{b_2 y - 1} = \frac{a_i x - 1}{b_i y - 1} = \frac{a_j x - 1}{b_j y - 1},$$

which we may rewrite as

$$\frac{a_1 x - 1}{a_2 x - 1} = \frac{b_1 y - 1}{b_2 y - 1} = \frac{b_j y - 1}{b_i y - 1} = \frac{a_j x - 1}{a_i x - 1},$$

we can view these as fractional linear transformations in $x$.

Consider the maps $x \mapsto \frac{a_1 x - 1}{a_2 x - 1}$ and $x \mapsto \frac{a_j x - 1}{a_i x - 1}$. These are fractional linear transformations since $a_r \neq a_s$ if and only if $(-1)a_r \neq (-1)a_s$ for all $r \neq s$. This means they are bijective, so there exists a unique value of $x$ satisfying this equality and any equations derived directly from it. In particular, this means there is a unique value

of $x$ satisfying $(*)$. Suppose for a contradiction $(a_ib_j - a_jb_i) + \frac{(a_j-a_i)(a_1b_2-a_2b_1)}{a_1-a_2} = 0$. The right-hand side of $(*)$ may be zero or non-zero. If it is non-zero, then $(*)$ has no solutions for $x$, which is a contradiction. Likewise, if it is zero, then each element in $\mathbb{F}_{p^m}$ is a solution, which is also a contradiction. Thus, we conclude the expression for $x$, which gives us an explicit expression for $\frac{c_1}{c_2}$ by reciprocating.

We now move on to finding an explicit expression for $\frac{c_1}{c_i}$ for all i $\in \{3, \ldots, k\}$. Let $i \in \{3, \ldots, k\}$ be given. Dividing the non-zero entries of $R_1$ by those of $R_2$ and dividing the non-zero entries of $R_1$ by those of $R_i$, we get

$$\frac{(R_1)_j}{(R_2)_j} = \frac{c_1}{c_2}\frac{\alpha_j - 1}{\alpha_j} \qquad \forall j \in \{k+1, \ldots, n\}$$
$$\frac{(R_1)_j}{(R_i)_j} = \frac{c_1}{c_i}\frac{\alpha_j - \alpha_i}{\alpha_j} \qquad \forall j \in \{k+1, \ldots, n\}.$$

By the same development as in the case where $i = 3$, we get

$$\frac{(R_1)_j}{(R_2)_j}\left(\frac{c_1}{c_2}\right)^{-1} - 1 = \left(\frac{(R_1)_j}{(R_i)_j}\left(\frac{c_1}{c_i}\right)^{-1} - 1\right)\alpha_i^{-1} \qquad \forall j \in \{k+1, \ldots, n\}.$$

Define $x := \left(\frac{c_1}{c_2}\right)^{-1}$, $y := \left(\frac{c_1}{c_i}\right)^{-1}$, $z := \alpha_i^{-1}$, $d_j := \frac{(R_1)_{j+k}}{(R_2)_{j+k}}$, and $f_j := \frac{(R_1)_{j+k}}{(R_i)_{j+k}}$. Using these definitions, we rewrite the preceding equations as follows.

$$\frac{d_1x - 1}{f_1y - 1} = \frac{d_2x - 1}{f_2y - 1} = \cdots = \frac{d_{n-k}x - 1}{f_{n-k}y - 1} \neq 0.$$

Of course, we assume $d_jx - 1 \neq 0$ and $f_jy - 1 \neq 0$ for all $j$, but if this weren't true, we'd already have our desired expression for $y$. From the leftmost equality, we get an expression for $y$:

$$y = \frac{(d_1 - d_2)x}{(d_1f_2 - d_2f_1)x + f_1 - f_2}.$$

Given that we have an expression for $x$, we know everything on the right-hand side of this equality. Reciprocating, we get an explicit expression for $\frac{c_1}{c_i}$ in terms of known values:

$$\frac{c_1}{c_i} = \frac{(f_1 - f_2)\frac{c_1}{c_2} + d_1f_2 - d_2f_1}{d_1 - d_2}.$$

□

6.2. **The Sidelnikov-Shestakov Attack on Full-Rank Goppa Codes.** When the secret code of a McEliece scheme is a Goppa code of maximal dimension, the S-S attack can be applied to such a scheme to recover the code parameters.

**Proposition 6.2.1.** *Consider a McEliece scheme based on a full-rank Goppa code. The S-S attack can be applied to this scheme to recover the Goppa polynomial and locator with the exact same complexity with which the S-S attack can recover the parameters of the corresponding GRS code.*

*Proof.* Consider a McEliece scheme based on a $(n, k)$ full-rank Goppa code $\Gamma(a, g)$ such that $\Gamma(a, g) = GRS_{n,k}(a, \beta) \cap \mathbb{F}_p^n$. We note that the public matrix $\mathbf{M}$ and the private generator matrix $\mathbf{G}$ both generate the secret Goppa code. By Lemma 2.1.15, $\Gamma(a, g)$ being of full-rank implies $\Gamma(a, g) \otimes \mathbb{F}_{p^m} = GRS_{n,k}(a, \beta)$. By Lemma

2.1.13, any basis for $\Gamma(a, g)$ will also be a basis for $GRS_{n,k}(a, \beta)$, so both $\mathbf{G}$ and $\mathbf{M}$ will be generator matrices for $GRS_{n,k}(a, \beta)$ (if viewed as matrices over $\mathbb{F}_{p^m}$). Hence, the private-key-public-key pair of this McEliece scheme corresponds exactly to one for a McEliece scheme based on a GRS code, the latter of which we've shown is vulnerable to the S-S attack. Thus, the S-S attack applies perfectly well to a McEliece scheme based on full-rank Goppa codes to recover the parameters $(a, \beta)$, the code parameters of $GRS_{n,k}(a, \beta)$.

However, we recall that the Goppa polynomial $g$ is related to $(a, \beta)$ by $g(a_j) = \beta_j \prod_{l \neq j}(a_j - a_l)$. Having recovered $(a, \beta)$ we may reconstruct $g$ by Lagrangian interpolation as follows.

$$\begin{aligned} g(x) &= \sum_{i=1}^{n} \beta_i \prod_{j \neq i}(a_i - a_j) \prod_{l \neq i} \left( \frac{x - a_l}{a_i - a_l} \right) \\ &= \sum_{i=1}^{n} \beta_i \prod_{l \neq i}(x - a_l) \end{aligned}$$

This, of course, follows from recognizing that $g$ passes through the $n$ points $D := \left\{ \left( a_i, \ \beta_i \prod_{j \neq i}(a_i - a_j)^{-1} \right) : i = 1, \ldots, n \right\}$. It is also possible to reconstruct $g$ through Lagrangian interpolation using a subset of $t + 1$ points from $D$ given that $deg(g) = t$. In any case, because Lagrangian interpolation over a set of size $n$ has complexity $\mathcal{O}(n\log(n))$, the addition of the recovery of $g$ to the S-S attack presented before means the complexity of recovering the code parameters for a full-rank Goppa code is also $\mathcal{O}(nk^2)$, the same as for a GRS code. □

In fact, by Corollary 2.1.17, we can also characterize GRS codes with subfield subcodes that are full-rank Goppa codes as being $\mathrm{Gal}(p^m, p)$-invariant. Hence, the S-S attack is successful on any Goppa code defined as a subfield subcode of a $\mathrm{Gal}(p^m, p)$-invariant GRS code.

While, from an attacker's perspective, it is heartening to see that any McEliece scheme based on a full-rank Goppa code is vulnerable to the S-S attack, we will show that such codes lead to an insecure McEliece scheme for another, more concerning reason. Consider a Goppa code $\Gamma(a, g)$ defined by the degree-$t$ Goppa polynomial $g \in \mathbb{F}_{p^m}[x]$ and the locator $a \in \mathbb{F}_{p^m}^n$. As a consequence of the GRS-Goppa code relationship detailed in Proposition 4.1.7, $\Gamma(a, g)$ is a subfield subcode of a GRS code $GRS_{n,k}(\alpha, \beta)$, where notably $k = n - t$. Let $d$ denote the minimum distance of $\Gamma(a, g)$. By Proposition 3.1.5, $GRS_{n,k}(\alpha, \beta)$ is MDS, so because $\Gamma(a, g)$ is a subcode of $GRS_{n,k}(\alpha, \beta)$, we may place a lower bound on its minimum distance: $d \geq n - k + 1 = t + 1$. Thus, we conclude the maximum number of zero positions in any codeword of $\Gamma(a, g)$ is

$$n - d \leq n - (t + 1) = k - 1.$$

This isn't surprising. It's the exact same bound we got by considering the polynomial interpretation of the Goppa code from Proposition 4.2.2,

$$\Gamma(a, g) = \{(\beta_1 q(a_1), \ldots, \beta_n q(a_n)) : q \in \mathcal{P}\},$$

with $\mathcal{P}$ a $\mathbb{F}_p$-linear subspace of $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ as defined before. For each codeword in $\Gamma(a, g)$, the polynomial $q$ associated to it has at most $k-1$ roots, translating to the codeword having at most $k-1$ zeros.

Goppa codes are linear, so by the Singleton bound,

$$n - k + 1 \leq d \leq n - k_\Gamma + 1.$$

Clearly, if $k_\Gamma = k$, then $\Gamma(a, g)$ is MDS. This fact about a full-rank Goppa code is what we will use to show it is unsuitable for a McEliece scheme. The unsuitability follows from this next proposition.

**Proposition 6.2.2.** *Let $C$ be a $(n, k)$ code and suppose it is MDS. All columns of the systematic generator matrix for $C$ will therefore be of minimum Hamming weight.*

*Proof.* Let $\mathbf{G}$ be the systematic generator matrix for $C$ and let $i \in \{1, \ldots, k\}$ be given. We know that

$$(\mathbf{G}_i)_j = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases} \qquad \forall j \in \{1, \ldots, k\}.$$

Using $d$ to denote the minimum distance of $C$, we have $d = n - k + 1$, since $C$ is MDS. The linearity of $C$ implies that each column of $\mathbf{G}$ must have Hamming weight greater than the code's minimum distance: $\omega(\mathbf{G}_i) \geq n - k + 1$ for all $i = 1, \ldots, k$. But for each column $\mathbf{G}_i$, we have $|\{j \in \{1, \ldots, k\} : (\mathbf{G}_i)_j \neq 0\}| = 1$, so for the inequality to hold, we must also have $(\mathbf{G}_i)_j \neq 0$ for all $j \in \{k+1, \ldots, n\}$. Thus, we conclude $\omega(\mathbf{G}_i) = n - k + 1 = d$ for all $i = 1, \ldots, k$. □

**Corollary 6.2.3.** *Any binary, full-rank Goppa code can correct at most 1 error.*

*Proof.* Consider a binary, full-rank Goppa code $\Gamma(a, g)$ of dimension $k$ with a generator matrix $\mathbf{M}$. The RREF form of $\mathbf{M}^\intercal$ will be the transpose of the systematic generator matrix for $\Gamma(a, g)$, written

$$E(\mathbf{M}^\intercal) = \left[\mathbf{I}_k | \mathbf{A}\right]$$

such that $\mathbf{A} \in \mathcal{M}_{k_\times (n-k)}(\mathbb{F}_2)$. By the last proposition, since $\Gamma(a, g)$ being of full rank means it is MDS, every entry of $\mathbf{A}$ is 1. For any two rows $R_i$ and $R_j$ such that $i \neq j$, we have $R_i + R_j \in \Gamma(a, g)$, but since the last $n - k$ positions of $R_i + R_j$ are all 0, $\omega(R_i + R_j) = 2$. Because the minimum distance is $d = t + 1$, we must have $2 \geq t + 1$, which implies that $t = 1$. As a consequence of Proposition 4.1.4 and Theorem 4.1.6, a Goppa code can correct at most $\frac{t}{2}$ errors and a binary, separable Goppa code can correct at most $t$ errors. Thus, if $\Gamma(a, g)$ were separable, it could correct at most $t = 1$ error; else, it could correct no errors. □

Note that this result does not extend to non-binary, full-rank Goppa codes. The sum of any pair of rows $R_i$ and $R_j$ of the RREF form a transposed generator matrix for such a code will not necessarily be a codeword of weight 2 precisely because we can't ensure $(R_j)_l = (R_i)_l^{-1}$ for all $l = k+1, \ldots, n$ as we could in the binary case, where $(R_i)_l = (R_j)_l = 1$.

Since all practical implementations of a McEliece scheme based on Goppa codes call for the code to be binary, the last corollary ensures that a full-rank Goppa

code used in practice can correct at most 1 error. Availing oneself of the S-S attack to correct the errors added in generating a cipher possesses little advantage over a message attack on such a McEliece scheme. Indeed, since at most 1 error was added in creating a cipher, an attacker only needs to check at worst $n$ different vectors in $\mathbb{F}_2^n$ to correct the error without needing the code-specific error-correction algorithm, whereas the recovery of the code parameters for a full-rank Goppa code alone needs $\mathcal{O}(nk^2)$ operations. Both approaches will identify the error vector $z$, after which point the message (up to permutation) can be recovered by identifying a left inverse of $\mathbf{GS}$. We illustrate this by letting $c \in \mathbb{F}_2^n$ denote a cipher generated by a plaintext $m \in \mathbb{F}_2^k$ and by noticing the following:

$$c = \mathbf{GS}m + z \implies m = \mathbf{GS}_{LI}(c - z).$$

Finding a left inverse of $\mathbf{GS}$ amounts to row-reducing a $n \times k$ matrix, which can be done in $\mathcal{O}(n^2k)$ operations. If we follow this model and neglect the complexity of applying the code-specific error-correction algorithm (which is valid given that the complexity of the decoding algorithm is $\mathbf{o}(n^2)$ as can be gleaned from the its complexity breakdown in Chapter 3 of [Bi]), the complexity of the structural attack required to recover $m$ is $\mathcal{O}(n^2k)$ whereas the complexity of the message attack required to accomplish the same is $\mathcal{O}(n^3k)$. Their complexities are comparable, so a McEliece scheme based on binary, full-rank Goppa codes is susceptible to both structural and message attacks. However, in practice no McEliece scheme would use a Goppa code capable of correcting only one error; indeed, $t$ is chosen to be far larger in the McEliece parameter sets proposed in [B], so practical implementations of McEliece aren't vulnerable in this way.

6.3. **The Sidelnikov-Shestakov Attack on (k-1)-Dimensional Goppa Codes.** We consider a McEliece scheme based on a Goppa code $\Gamma(a, g) = GRS_{n,k}(a, \beta)$ of dimension $k_\Gamma = k - 1$. We will elaborate on the progress we can make in adapting the S-S attack to this scheme and, ultimately, why it is difficult to do so for any McEliece scheme based on Goppa codes that are not of maximal dimension.

Recall by Proposition 4.2.2, $\Gamma(a, g) = \{(\beta_1 q(a_1), \ldots, \beta_n q(a_n)) : q \in \mathcal{P}\}$ where $\mathcal{P} \subset \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ such that $\mathcal{P}$ is $\mathbb{F}_p$-linear and $\dim_{\mathbb{F}_p}(\mathcal{P}) = k_\Gamma = k-1$. For a McEliece scheme based on $\Gamma(a, g)$, the public matrix $\mathbf{M}$ and the private generator matrix $\mathbf{G}$ both generate $\Gamma(a, g)$. The goal of the S-S attack is the recover the code parameters $(a, g)$, which by the discussion in Section 6.2 is equivalent to recovering the code parameters of the corresponding GRS code $GRS_{n,k}(a, \beta)$. We can see that Lemma 6.1.1 applies similarly here and that we need only recover equivalent parameters for $GRS_{n,k}(a, \beta)$ in order to access the efficient decoding algorithm for $\Gamma(a, g)$.

**Lemma 6.3.1.** *Without loss of generality,* $a_1 = 0$, $a_2 = 1$, *and* $\beta_1 = 1$.

*Proof.* By Lemma 6.1.1, there is a GRS code defined by equivalent parameters $(a', \beta')$ such that $GRS_{n,k}(a, \beta) = GRS_{n,k}(a', \beta')$ where $a'_1 = 0$, $a'_2 = 1$, and $\beta'_1 = 1$.

Let $\Gamma(a', g') = GRS_{n,k}(a', \beta') \cap \mathbb{F}_p^n$ where $g'$ satisfies

$$g'(a'_j) = \beta'_j \prod_{l \neq j} (a'_j - a'_l) \ \ \forall j = 1, \ldots, n.$$

But then $\Gamma(a,g) = GRS_{n,k}(a,\beta) \cap \mathbb{F}_p^n = GRS_{n,k}(a',\beta') \cap \mathbb{F}_p^n = \Gamma(a',g')$, so the decoding algorithm for $\Gamma(a,g)$ is accessible for either pair of parameters $(a,\beta)$ and $(a',\beta')$. We will choose to find $(a',\beta')$. □

To get a sense of where the difficulties for the S-S attack lie, we will try to apply it to the McEliece scheme we're considering. Again, WLOG, by possibly permuting rows of $\mathbf{G}$, we assume that the first $k_\Gamma$ rows of $\mathbf{G}$ are linearly independent. We must again find the RREF form of $\mathbf{M}^\intercal$, which will also have a systematic form by our assumption.

$$\mathbf{M}^\intercal \sim E(\mathbf{M}^\intercal) = [\mathbf{I}_{k_\Gamma}|A] = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_{k_\Gamma} \end{bmatrix}$$

Since the rows are codewords of $\Gamma(a,g)$, we interpret them as such: for all $i = 1,\dots,k_\Gamma$,

$$R_i = (\beta_1 q_{R_i}(a_1),\dots,\beta_n q_{R_i}(a_n)) \text{ for some } q_{R_i} \in \mathcal{P}.$$

As before, the zero entries of each row imply for all $i = 1,\dots,k_\Gamma$,

$$\prod_{j\in\{1,\dots,k_\Gamma\}\setminus\{i\}} (x - a_j) \mid q_{R_i}(x).$$

The difference now is that because the product of degree-1 factors is a polynomial of degree $k_\Gamma - 1$ and $deg(q_{R_i}) \leq k-1$, we don't know if this identifies $q_{R_i}$ up to scalar multiple as before. Indeed, our next proposition will show that for any $i = 1,\dots,k_\Gamma$, $q_{R_i}$ will be a scalar multiple of this product with low probability. However, to complete the factorization of $q_{R_i}$ using this product, we must introduce another polynomial $\rho_i \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ such that

$$q_{R_i}(x) = \rho_i(x) \prod_{j\in\{1,\dots,k_\Gamma\}\setminus\{i\}} (x - a_j) \quad \text{where } \rho_i(x) = \sum_{j=0}^{k-k_\Gamma} \lambda_{i,j} x^j.$$

By considering the RREF form of $\mathbf{M}^\intercal$, we get for all $i = 1,\dots,k_\Gamma$,

$$\beta_j q_{R_i}(a_j) = \begin{cases} 0, & j \in \{1,\dots,k_\Gamma\}\setminus\{i\} \\ 1, & j = i \\ (R_i)_j, & j \in \{k_\Gamma+1,\dots,n\} \end{cases}$$

$$\iff \rho_i(a_j) \prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}} (a_j - a_l) = \begin{cases} 0, & j \in \{1,\dots,k_\Gamma\}\setminus\{i\} \\ \beta_i^{-1}, & j = i \\ \beta_j^{-1}(R_i)_j, & j \in \{k_\Gamma+1,\dots,n\} \end{cases}.$$

From these, we are able to glean some useful equations about $\rho_i$.

$$\rho_i(a_j) = \begin{cases} \frac{\beta_i^{-1}}{\prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}}(a_i - a_l)}, & ,j = i \\ \frac{\beta_j^{-1}(R_i)_j}{\prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}}(a_j - a_l)}, & ,j \in \{k_\Gamma+1,\dots,n\} \end{cases}$$

It would appear we have no information on the degrees of the row polynomials $q_{R_i}$, but this changes in the binary case, as is summarized in the following proposition. Indeed, the information we know is particularly relevant when $k_\Gamma = k-1$, but

it more generally shows that the product $\prod_{j\in\{1,\dots,k_\Gamma\}\setminus\{i\}}(x-\alpha_j)$ will very rarely identify $q_{R_i}$ up to scalar multiple.

**Proposition 6.3.2.** *Let $\Gamma(a,g)$ be a $(n,k_\Gamma)$ Goppa code over $\mathbb{F}_2$ that is a subfield subcode of a GRS code of dimension $k > k_\Gamma$. There exists at most one $i \in \{1,\dots,k_\Gamma\}$ such that $deg(q_{R_i}) = k_\Gamma - 1$.*

*Proof.* Let $i \in \{1,\dots,k_\Gamma\}$ be given. We have $q_{R_i}(x) = \left(\prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}}(x-a_l)\right)\rho_i(x)$. We also developed the following equations.

$$\rho_i(a_j) = \begin{cases} \frac{\beta_i^{-1}}{\prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}}(a_i-a_l)} & ,j=i \\ \frac{\beta_j^{-1}(R_i)_j}{\prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}}(a_j-a_l)} & ,j\in\{k_\Gamma+1,\dots,n\} \end{cases}$$

Notice that $deg(q_{R_i}) = k_\Gamma - 1$ if and only if $\rho_i$ is constant and non-zero. But we know $\rho_i$ is constant if and only if

$$\frac{\beta_i^{-1}}{\prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}}(a_i-a_l)} = \frac{\beta_j^{-1}(R_i)_j}{\prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}}(a_j-a_l)} \qquad \forall j\in\{k_\Gamma+1,\dots,n\}.$$

Since these equations give the equality of $n-k_\Gamma+1$ $y$-values, each for different points $\rho_i$ passes through. The maximum degree of $\rho_i$ is $k-k_\Gamma < n-k_\Gamma+1$, so the converse direction holds as this guarantees $\rho_i$ is constant.

The above holds if and only if

$$\frac{\beta_j}{\beta_i}\prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}}\left(\frac{a_j-a_l}{a_i-a_l}\right) = (R_i)_j \qquad \forall j\in\{k_\Gamma+1,\dots,n\}.$$

Recalling that $j \geq k_\Gamma+1$, we notice we can't have $(R_i)_j = 0$ or else it would mean there exists some $l \neq j$ such that $a_l = a_j$, a contradiction. Also, this would then force $\rho_i$ to be the zero polynomial, which we said it wasn't. Since $\Gamma(a,g)$ is binary, $(R_i)_j \neq 0$ implies $(R_i)_j = 1$. We get the following rearrangement.

$$\begin{aligned} &\frac{\beta_j}{\beta_i}\prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}}\left(\frac{a_j-a_l}{a_i-a_l}\right) = 1 \\ \iff &\beta_j\prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}}(a_j-a_l) = \beta_i\prod_{l\in\{1,\dots,k_\Gamma\}\setminus\{i\}}(a_i-a_l) \end{aligned}$$

Recall that $g(a_j) = \beta_j\prod_{l\neq j}(a_j-a_l)$. With this, we rewrite the last equation as follows.

$$\begin{aligned} &\frac{g(a_j)}{\left[\prod_{l\in\{k_\Gamma+1,\dots,n\}\setminus\{j\}}(a_j-a_l)\right](a_j-a_i)} = \frac{g(a_i)}{\prod_{l\in\{k_\Gamma+1,\dots,n\}}(a_i-a_l)} \\ \iff &\frac{g(a_j)}{\prod_{l\in\{k_\Gamma+1,\dots,n\}\setminus\{j\}}(a_j-a_l)} = \frac{-g(a_i)}{\prod_{l\in\{k_\Gamma+1,\dots,n\}\setminus\{j\}}(a_i-a_l)} \end{aligned}$$

$$\iff g(a_j) = -\left(\prod_{l\in\{k_\Gamma+1,\ldots,n\}\setminus\{j\}} \frac{(a_j - a_l)}{(a_i - a_l)}\right) g(a_i)$$

This holds for all $j \in \{k_\Gamma + 1, \ldots, n\}$. If there exists some $i' \in \{2, \ldots, k_\Gamma\}\setminus\{i\}$ such that $\rho_{i'}(x)$ is constant, then we must have

$$g(a_j) = -\left(\prod_{l\in\{k_\Gamma+1,\ldots,n\}\setminus\{j\}} \frac{(a_j - a_l)}{(a_{i'} - a_l)}\right) g(a_{i'}) \qquad \forall j \in \{k_\Gamma + 1, \ldots, n\}.$$

Therefore,

$$\frac{g(a_i)}{\prod_{l\in\{k_\Gamma+1,\ldots,n\}\setminus\{j\}}(a_i - a_l)} = \frac{g(a_{i'})}{\prod_{l\in\{k_\Gamma+1,\ldots,n\}\setminus\{j\}}(a_{i'} - a_l)} \qquad \forall j \in \{k_\Gamma+1, \ldots, n\},$$

and thus,

$$g(a_{i'}) = \left(\prod_{l\in\{k_\Gamma+1,\ldots,n\}\setminus\{j\}} \frac{(a_{i'} - a_l)}{(a_i - a_l)}\right) g(a_i) \qquad \forall j \in \{k_\Gamma + 1, \ldots, n\}.$$

Since $g(a_i)$ and $g(a_{i'})$ are just values in $\mathbb{F}_{2^m}$ independent of the choice of $j$, this equation holding for all $j \in \{k_\Gamma + 1, \ldots, n\}$ implies

$$\prod_{l\in\{k_\Gamma+1,\ldots,n\}\setminus\{j\}} \frac{(a_{i'} - a_l)}{(a_i - a_l)} = \prod_{l\in\{k_\Gamma+1,\ldots,n\}\setminus\{h\}} \frac{(a_{i'} - a_l)}{(a_i - a_l)} \;\; \forall j, h \in \{k_\Gamma+1, \ldots, n\} \text{ such that } j \neq h.$$

This implies

$$\frac{(a_{i'} - a_h)}{(a_i - a_h)} = \frac{(a_{i'} - a_j)}{(a_i - a_j)} \qquad \forall j, h \in \{k_\Gamma + 1, \ldots, n\} \text{ such that } j \neq h.$$

Since $a_i \neq a_{i'}$ because $i \neq i'$, both the left-hand and right-hand expressions can be viewed as fractional linear transformations in $a_h$ and $a_j$, respectively. Define the map $T$ by $x \mapsto \frac{-x+a_{i'}}{-x+a_i}$. The value of $a_h$ is some fixed number in $\mathbb{F}_{2^m}$ and let $j \in \{k_\Gamma + 1, \ldots, n\}\setminus\{h\}$ be given. We want to solve $T(a_j) = \frac{a_{i'}-a_h}{a_i-a_h}$. Since $T$ is bijective, there is only one solution to this equation. We see that by taking $a_j = a_h$, we solve the equation, so $a_j = a_h$ is the unique solution. However, this is a contradiction since $a_i \neq a_j$ for all $i \neq j$. Therefore, we can't have two distinct values of $i, i' \in \{2, \ldots, k_\Gamma\}$ such that $\rho_i(x)$ and $\rho_{i'}(x)$ are both constant. □

The direct application of this proposition to Goppa codes of dimension $k_\Gamma = k-1$ is that there is at most one row polynomial of degree $k-2$. All others must be of degree $k-1$ and since all irreducible polynomials over $\mathbb{F}_{2^m}$ of degree one have roots in $\mathbb{F}_{2^m}$, this means these other row polynomials factor over $\mathbb{F}_{2^m}$. More precisely, we know the degrees of all row polynomials for a McEliece scheme based on such Goppa codes and we know that at most one row polynomial $q_{R_i}$ won't admit an additional, degree-1 factor $\rho_i$.

Notice that as a consequence of this proposition, the quotient $\frac{(R_1)_j}{(R_2)_j}$ for some $j \in \{k_\Gamma + 1, \ldots, n\}$ (or the quotient of the same entries in any two different rows, for that matter) will never simplify to a fractional linear transformation in $a_j$, as

it did in the S-S attack on a McEliece scheme based on GRS codes. Instead, it simplifies as follows:

$$\frac{(R_1)_j}{(R_2)_j} = \frac{\beta_j q_{R_1}(a_j)}{\beta_j q_{R_2}(a_j)} = \frac{\rho_1(a_j)\prod_{r\in\{1,\dots,k_\Gamma\}\setminus\{1\}}(a_j - a_r)}{\rho_2(a_j)\prod_{r\in\{1,\dots,k_\Gamma\}\setminus\{2\}}(a_j - a_r)} = \frac{\rho_1(a_j)(a_j-1)}{\rho_2(a_j)(a_j)}. \tag{6.1}$$

For large enough $k$, it stands to reason that with high probability neither $\rho_1$ nor $\rho_2$ are constant, so we will be forced to assume this. The rightmost expression in (6.1) is a rational function in $a_j$ of degree at most $k - k_\Gamma + 1$ and at least 2. The injectivity of this rational function is only guaranteed when it is of degree 1 since it is a fractional linear transformation in this case, so we will not necessarily be able to invert this function as we outlined in the S-S attack before in order to relate $a_j$ to known values. Even then, inverting a high-degree rational function is difficult to do. This blocks direct application of the S-S attack.

We will illustrate the difficulty of using (6.1) to partially recover $\alpha$. Let $r \in \mathbb{F}_{2^m}$ such that $\rho_2(r) = 0$. Define $S : \mathbb{F}_{2^m} \setminus \{0, r\} \to \mathbb{F}_{2^m}$ by the map $x \mapsto \frac{\rho_1(x)(x-1)}{\rho_2(x)x}$. Suppose for all $j \in \{k_\Gamma + 1, \dots, n\}$ we are always able to identify $S^{-1}\left(\frac{(R_1)_j}{(R_2)_j}\right)$ and that this preimage contains only 2 elements. However, both elements are equally likely to be $a_j$, so we have no better option but to arbitrarily take $a_j$ to be one of these two values. Repeating this for all of $j = k_\Gamma + 1, \dots, n$, we observe that the partial recovery of $a_{k_\Gamma+1}, \dots, a_n$ requires at worst $\mathcal{O}(2^{n-k_\Gamma})$ operations in $\mathbb{F}_{2^m}$, excluding the complexity needed to identify $S^{-1}\left(\frac{(R_1)_j}{(R_2)_j}\right)$. We immediately see that it is intractable to use (6.1) to solve for $a_j$; hence, the S-S attack cannot be applied directly to Goppa codes that are not of full rank.

## 7. Wieschebrink's Squaring Attack

Wieschebrink mounts a structural attack on a McEliece scheme based on random subcodes of GRS codes in [W] that takes advantage of the polynomial-evaluation interpretation of GRS codes and that the square (defined using the componentwise product) of random subcodes of GRS codes will be a GRS code itself. We will outline this attack for random subcodes of GRS and then show that this attack doesn't generalize to subfield subcodes of GRS codes (that is to say, to Alternant codes).

7.1. **The Squaring Attack.** Consider the McEliece scheme using a random subcode of the code $GRS_{n,k}(\alpha, \beta)$ of dimension $k - l$ over $\mathbb{F}_{p^m}$ where $l$ is chosen to be some value in $\{1, \dots, k-1\}$. Following the model presented in Section 6.1, the public matrix of this scheme will be $\mathbf{M} = \mathbf{GS}$ such that $\mathbf{G}$ is a $n \times k$ generator matrix for $GRS_{n,k}(\alpha, \beta)$ and $\mathbf{S}$ is now a $k \times (k-l)$ matrix of rank $k-l$ is a random matrix of rank $k - l$. As usual, $\mathbf{M}$ is a generator matrix for the public code, but the public code is now a $(k-l)$-dimensional subspace of $GRS_{n,k}(\alpha, \beta)$.

At the risk of possible confusion with the degree of the field extension $m = [\mathbb{F}_{p^m} : \mathbb{F}_p]$, in order to stay consistent with Wieschebrink's notation, we define $m := k - l$. Note that in the rest of this section, $m$ will always mean $k - l$ and it will only ever refer to $[\mathbb{F}_{p^m} : \mathbb{F}_p]$ when written as the exponent of $p^m$ in $\mathbb{F}_{p^m}$. $\mathbf{M}$ is therefore a $n \times m$ matrix whose transposed, RREF form $E(\mathbf{M}^\intercal)$ has rows $R_1, \dots, R_m$, to

which are associated the row polynomials $p_{R_1}, \ldots, p_{R_m} \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$. Next, we define the bilinear operation called the component-wise product (or Schur product) as $* : \mathbb{F}_{p^m}^n \times \mathbb{F}_{p^m}^n \to \mathbb{F}_{p^m}^n$ by $a * b \mapsto (a_1 b_1, a_2 b_2, \ldots, a_n b_n)$. With this, we introduce the square of a linear code $C$ as

$$C^{(*2)} := \text{span}_{\mathbb{F}_{p^m}}(\{c * d : \ c, d \in C\}).$$

For a GRS code $GRS_{n,k}(\alpha, \beta) = \{(\beta_1 p(\alpha_1), \ldots, \beta_n p(\alpha_n)) : p \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})\}$, its square is

$$\begin{aligned} GRS_{n,k}(\alpha, \beta)^{(*2)} &= \text{span}_{\mathbb{F}_{p^m}}(\{(\beta_1^2 pq(\alpha_1), \ldots, \beta_n^2 pq(\alpha_n)) : p, q \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})\}) \\ &\subseteq \{(\beta_1^2 p(\alpha_1), \ldots, \beta_n^2 p(\alpha_n)) : p \in \mathbb{P}_{2k-2}(\mathbb{F}_{p^m})\} \end{aligned}$$

since for $p, q \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$, we have $p(x)q(x) = pq(x) \in \mathbb{P}_{2k-2}(\mathbb{F}_{p^m})$ given the ring structure of $\mathbb{F}_{p^m}[x]$, to which both $p$ and $q$ belong. For $2k - 1 \leq n$, the square is contained in the GRS code $GRS_{n,2k-1}(\alpha, \beta')$ such that $\beta' := \beta * \beta$. In fact, the square is exactly this GRS code.

Before we show this, let us define the *evaluation map* and use it to prove an intermediary result.

**Definition 7.1.1.** The *evaluation map* for a vector $\alpha \in \mathbb{F}_{p^m}^n$ is the linear map $ev_\alpha : \mathbb{F}_{p^m}[x] \to \mathbb{F}_{p^m}^n$ given by $p \mapsto (p(\alpha_1), \ldots, p(\alpha_n))$.

**Lemma 7.1.2.** *Consider the $(n, k)$ GRS code $GRS_{n,k}(\alpha, \beta)$ and let a basis for $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ be $\{p_i(x) : i = 1, \ldots, k\}$. We observe that*

$$GRS_{n,k}(\alpha, \beta)^{(*2)} = \left\{(\beta_1^2 p(\alpha_1), \ldots, \beta_n^2 p(\alpha_n)) : p \in span_{\mathbb{F}_{p^m}}\{p_i(x)p_j(x) : i \leq j\}\right\}.$$

*Proof.* The $\supseteq$ inclusion is straightforward. We will just show the $\subseteq$ inclusion.
Let $p, q \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$, so the vectors $\beta * ev_\alpha(p)$ and $\beta * ev_\alpha(q)$ both belong to $GRS_{n,k}(\alpha, \beta)$. We notice we have

$$(\beta * ev_\alpha(p)) * (\beta * ev_\alpha(q)) = \left(\beta_1^2 p(\alpha_1)q(\alpha_1), \ldots, \beta_n^2 p(\alpha_n)q(\alpha_n)\right).$$

This will be rewritten as follows.

$$\begin{aligned} &\left(\beta_1^2 \left(\sum_{i=1}^k \lambda_i p_i(\alpha_1)\right)\left(\sum_{j=1}^k \gamma_j p_j(\alpha_1)\right), \ \ldots \ , \ \beta_n^2 \left(\sum_{i=1}^k \lambda_i p_i(\alpha_n)\right)\left(\sum_{j=1}^k \gamma_j p_j(\alpha_n)\right)\right) \\ &\qquad = \left(\beta_1^2 \sum_{i=1}^k \sum_{j=1}^k \lambda_i \gamma_j p_i \cdot p_j(\alpha_1), \ \ldots \ , \ \beta_n^2 \sum_{i=1}^k \sum_{j=1}^k \lambda_i \gamma_j p_i \cdot p_j(\alpha_n)\right) \end{aligned}$$

We can rewrite the polynomial appearing in each entry of $(\beta * ev_\alpha(p)) * (\beta * ev_\alpha(q))$. We see that

$$\sum_{i=1}^k \sum_{j=1}^k \lambda_i \gamma_j p_i \cdot p_j(x) = \sum_{j=1}^k \sum_{i=1}^j (\lambda_i \gamma_j + \lambda_j \gamma_i) p_i \cdot p_j(x) - \sum_{j=1}^k \lambda_j \gamma_j p_j^2(x)$$

and this is clearly in $\text{span}_{\mathbb{F}_{p^m}}\{p_i \cdot p_j : i \leq j\}$. Hence, we may conclude

$$GRS_{n,k}(\alpha, \beta)^{(*2)} \subseteq \left\{(\beta_1^2 p(\alpha_1), \ldots, \beta_n^2 p(\alpha_n)) : p \in \text{span}_{\mathbb{F}_{p^m}}\{p_i(x)p_j(x) : i \leq j\}\right\},$$

which gives us the result. □

**Proposition 7.1.3.** *Consider the $(n,k)$ GRS code $GRS_{n,k}(\alpha,\beta)$ such that $2k-1 \leq n$. Defining $\beta' := \beta * \beta$, we have*

$$GRS_{n,k}(\alpha,\beta)^{(*2)} = GRS_{n,2k-1}(\alpha,\beta').$$

*Proof.* Let the basis $\{p_i(x) : i = 1,\dots,k\}$ for $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ used in the last lemma be the standard monomial basis $\{1, x, \dots, x^{k-1}\}$. It's clear that the standard monomial basis of $\mathbb{P}_{2k-2}(\mathbb{F}_{p^m})$ will be contained in $\text{span}_{\mathbb{F}_{p^m}}\{p_i(x)p_j(x) : i \leq j\}$, so by the last lemma, we get

$$GRS_{n,k}(\alpha,\beta)^{(*2)} = \left\{(\beta_1^2 p(\alpha_1),\dots,\beta_n^2 p(\alpha_n)) : p \in \mathbb{P}_{2k-2}(\mathbb{F}_{p^m})\right\}.$$

Since $2k-1 \leq n$, this vector space is $GRS_{n,2k-1}(\alpha,\beta')$. □

Wieschebrink considers two cases for his squaring attack in which we can recover the code parameters to this random subcode of $GRS_{n,k}(\alpha,\beta)$ of dimension $m = k-l$. We'll present the recovery of the code parameters in each case as separate lemmas.

**Lemma 7.1.4.** *Consider the $(n,k)$ GRS code $GRS_{n,k}(\alpha,\beta)$ such that $2k \leq n-2$. If $\mathbf{M}$ is the public matrix for a McEliece scheme such that it generates a random subcode of $GRS_{n,k}(\alpha,\beta)$ of dimension $m = k-l$, then we can recover $(\alpha,\beta)$ with high probability[1] using $\mathbf{M}$.*

*Proof.* We row-reduce the transpose of the public matrix to get $E(\mathbf{M}^\intercal)$, the matrix whose rows we denote by $R_1,\dots,R_m$. The code generated by the rows of $E(\mathbf{M}^\intercal)$ is a subcode of $GRS_{n,k}(\alpha,\beta)$, so by the last proposition, the square of $\text{span}_{\mathbb{F}_{p^m}}\{R_1,\dots,R_m\}$ will be a subspace of $GRS_{n,k}(\alpha,\beta)^{(*2)} = GRS_{n,2k-1}(\alpha,\beta')$. It is Wieschebrink's claim in [W] that for $C = Im(\mathbf{M})$, the probability that $C^{(*2)} = GRS_{n,2k-1}(\alpha,\beta')$ is very high. Recall that we associated each row $R_i$ to a polynomial $p_{R_i}$ such that $R_i = ev_\alpha(p_{R_i}) * \beta$.

It's easy to see that the same development used to prove Lemma 7.1.2 applies to $C$ as well, giving us

$$C^{(*2)} = \left\{(\beta_1^2 p(\alpha_1),\dots,\beta_n^2 p(\alpha_n)) : p \in \text{span}_{\mathbb{F}_{p^m}}\{p_{R_i}(x)p_{R_j}(x) : 1 \leq i \leq j \leq m\}\right\}.$$

We can continue simplifying this.

$$\begin{aligned}
C^{(*2)} &= \left\{\beta * ev_\alpha(p) * \beta * ev_\alpha(q) : p \in \text{span}_{\mathbb{F}_{p^m}}\{p_{R_j}(x) : 1 \leq j \leq m\},\ \ q \in \text{span}_{\mathbb{F}_{p^m}}\{p_{R_i}(x) : 1 \leq i \leq j\}\right\} \\
&= \left\{\beta * ev_\alpha\left(\sum_{j=1}^{m}\lambda_j p_{R_j}\right) * \beta * ev_\alpha\left(\sum_{i=1}^{j}\gamma_i p_{R_i}\right) : \lambda_j,\gamma_i \in \mathbb{F}_{p^m}\right\} \\
&= \left\{\left(\sum_{j=1}^{m}\lambda_j(\beta * ev_\alpha(p_{R_j}))\right) * \left(\sum_{i=1}^{j}\gamma_i(\beta * ev_\alpha(p_{R_i}))\right) : \lambda_j,\gamma_i \in \mathbb{F}_{p^m}\right\} \\
&\qquad \text{by the associativity and bilinearity of } * \text{ and the linearity of } ev_\alpha \\
&= \left\{\left(\sum_{j=1}^{m}\lambda_j R_j\right) * \left(\sum_{i=1}^{j}\gamma_i R_i\right) : \lambda_j,\gamma_i \in \mathbb{F}_{p^m}\right\} \\
&= \text{span}_{\mathbb{F}_{p^m}}\{R_i * R_j : 1 \leq i \leq j \leq m\}
\end{aligned}$$

[1] See [W] for a discussion on what is meant by "high probability"

Thus, we can generate $C^{(*2)}$ by computing the component-wise product between pairs of rows $R_i$ and $R_j$ for all $1 \leq i \leq j \leq m$. $C^{(*2)}$ will be the GRS code $GRS_{n,2k-1}(\alpha, \beta')$ with very high probability, so since $\dim(C^{(*2)}) = 2k-1 \leq n-2$, we may apply the Sidelnikov-Shestakov attack to a generator matrix of $C^{(*2)}$ to recover $\alpha$ and $\beta'$, from which point it is easy to recover $\beta$. □

The other case where if $2k-1 > n-2$, requires a somewhat different approach. To tackle this case, we need the notion of a *shortened code*.

**Definition 7.1.5.** Let $C \subseteq \mathbb{F}_{p^m}^n$ be a $(n,k)$ code and let $d \in \{0,\ldots,k\}$. The *shortened code* $S_d(C)$ is the code consisting of all codewords $(s_1,\ldots,s_{n-d}) \in \mathbb{F}_{p^m}^{n-d}$ such that $(0,\ldots,0,s_1,\ldots,s_{n-d}) \in C$.

**Remark 7.1.6.** It's clear that if $\mathbf{G} = [\mathbf{I}_k|\mathbf{A}]^\intercal$ is the systematic generator matrix for $C$, then the last $k-d$ columns of $\mathbf{G}$ restricted to their last $n-d$ entries will form a basis of $S_d(C)$.

We can now address the other case addressed by Wieschebrink.

**Lemma 7.1.7.** *Consider the $(n,k)$ GRS code $GRS_{n,k}(\alpha,\beta)$ such that $2k > n-2$. Let $\mathbf{M}$ be the public matrix for a McEliece scheme such that it generates a random subcode of $GRS_{n,k}(\alpha,\beta)$ of dimension $m = k-l$. If there exists and integer $d$ such that $d \leq m-1$ and $2(k-d)-1 \leq n-d-2$, then we can recover $(\alpha,\beta)$ with high probability[2] using $\mathbf{M}$.*

*Proof Sketch.* Again, row-reduce the transpose of the public matrix to get $E(\mathbf{M}^\intercal)$, the matrix whose $i^{th}$ row is $R_i$ to which we associate the polynomial $p_{R_i}$. Again, we define $C := Im(\mathbf{M})$.

Let $d$ be an integer and define the set $I := \{d+1,\ldots,n\}$ to be used for puncturing the rows of $E(\mathbf{M}^\intercal)$ to generate a basis for the shortened code $S_d(C)$ for some value $d$ satisfying $d \leq m-1$ and $2(k-d)-1 \leq n-d-2$. Since $E(\mathbf{M}^\intercal)$ is the transposed form of the systematic generator matrix for $C$, by Remark 7.1.6, a basis for the shortened code $S_d(C)$ is $\{(R_{d+i})_I : i = 1,\ldots,m-d\}$. Note that because $d \leq m-1$, this really will be a valid shortened code. Also by Remark 7.1.6, a systematic generator matrix for $S_d(C)$ can be defined as follows:

$$\mathbf{M}_S := [(R_{d+1})_I{}^\intercal \ (R_{d+2})_I{}^\intercal \ \ldots \ (R_m)_I{}^\intercal].$$

For simplicity, we will let $S_i$ denote the $i^{th}$ column of $\mathbf{M}_S$. We recognize that

$$(0,\ldots,0,(S_i)_1,\ldots,(S_i)_{n-d}) \in C \subseteq GRS_{n,k}(\alpha,\beta).$$

Thus, we know the $i^{th}$ column, $S_i$, will have the form $S_i = (\beta_{d+1}p(\alpha_{d+1}),\ldots,\beta_n p(\alpha_n))$ for some $p \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$. However, the codeword from which $S_i$ is punctured is in $GRS_{n,k}(\alpha,\beta)$, so we have

$$(0,\ldots,0,(S_i)_1,\ldots,(S_i)_{n-d}) = (\beta_1 p(\alpha_1),\ldots,\beta_n p(\alpha_n)),$$

which means we must have $p(\alpha_j) = 0$ for all $j = 1,\ldots,d$ since $\beta_j \neq 0$ for all $j = 1,\ldots,n$. Therefore, we may write

$$p(x) = h(x)\prod_{j=1}^{d}(x-\alpha_j) \quad \text{for some } h(x) \in \mathbb{F}_{p^m}[x] \text{ such that } deg(h) \leq k-1-d.$$

[2]Again, see [W] for further details

Thus, for all $i = 1, \ldots, m-d$, we have

$$S_i = \left( \left( \beta_{d+1} \prod_{j=1}^{d} (\alpha_{d+1} - \alpha_j) \right) h(\alpha_{d+1}), \ \ldots \ , \ \left( \beta_n \prod_{j=1}^{d} (\alpha_n - \alpha_j) \right) h(\alpha_n) \right).$$

Defining $z := \left( \beta_{d+1} \prod_{j=1}^{d} (\alpha_{d+1} - \alpha_j), \ldots, \beta_n \prod_{j=1}^{d} (\alpha_n - \alpha_j) \right)$ and $\alpha' := (\alpha_{d+1}, \ldots, \alpha_n)$, we can see that for each column, $S_i \in GRS_{n-d,k-d}(\alpha', z)$ for all $i = 1, \ldots, m-d$. Hence, $S_d(C) \subseteq GRS_{n-d,k-d}(\alpha', z)$.

Picking $C$ to be a random, $m$-dimensional subcode of $GRS_{n,k}(\alpha, \beta)$ is equivalent to picking $Q := \{p_1, \ldots, p_m\}$ to be a random, linearly independent subset of $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ such that $C = \{\beta * ev_\alpha(p) : p \in \text{span}_{\mathbb{F}_{p^m}}(Q)\}$. $Q$ being randomly chosen means that the basis $\{p_{R_1}, \ldots, p_{R_m}\}$ of row polynomials, where these row polynomials are all linear combinations of the polynomials in $Q$, will also be a set of $m$ random, linearly independent vectors in $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$.

Define the polynomial $h_{R_i}(x)$ such that $p_{R_i}(x) = h_{R_i}(x) \prod_{j=1}^{d} (x - \alpha_j)$. Since $p_{R_i}(x)$ is randomly chosen and $\prod_{j=1}^{d} (x - \alpha_j)$ is a fixed polynomial defined entirely by $\alpha$, the randomness must come from $h_{R_i}(x)$. Hence, $\{h_{R_i} : i = d+1, \ldots, m\}$ is a set of randomly-chosen vectors in $\mathbb{P}_{k-d-1}(\mathbb{F}_{p^m})$ that is linearly independent since the corresponding columns of $\mathbf{M}_S$ $(S_{d+1}, \ldots, S_m)$ form a linearly independent set and a set of vectors from a GRS code are linearly independent if and only if the set of their corresponding polynomials is linearly independent. This is easy to verify.

Thus, $S_d(C) = \left\{ z * ev_{\alpha'}(h) : h \in \text{span}_{\mathbb{F}_{p^m}} \{h_{R_i} : i = d+1, \ldots, m\} \right\}$ is a random, $(m-d)$-dimensional subcode of $GRS_{n-d,k-d}(\alpha', z)$. By Lemma 7.1.4, with high probability the square of this code will be a GRS code, so we can find the square code $S_d(C)^{(*2)}$ and if $2(k-d) - 1 \leq n - d - 2$ holds, then we can and apply the Sidelnikov-Shestakov attack to recover the parameters $(\alpha', z)$. By the hypothesis, $d$ satisfies $2(k-d) - 1 \leq n - d - 2$, so we can apply the S-S attack. The remaining entries of $\alpha$ can be recovered by going through the same process as above with a permutation of $\mathbf{M}$ with the precise details of the recovery left in [W]. $\square$

For this attack to be successful, we must be able to pick $d$ such that $d \leq m-1$ and $2(k-d) - 1 \leq n - d - 2$. Both of these conditions together can be written as $2k - n + 1 \leq d \leq m - 1$. This condition for the success of the attack explains in part why McEliece using Goppa codes won't succumb to this attack. Consider the $(n, k_\Gamma)$ Goppa code $\Gamma(\alpha, g)$ that is the subfield subcode of $GRS_{n,k}(\alpha, \beta)$. Since $\Gamma(\alpha, g)$ is a $k_\Gamma$-dimensional subspace of $GRS_{n,k}(\alpha, \beta)$, we have $m = k_\Gamma$; further usage of $m$ will now refer to the degree of the field extension $m = [\mathbb{F}_{p^m} : \mathbb{F}_p]$. Reinterpreting the conditions on $d$ for the success of the attack, we get $2k - n + 1 \leq d \leq k_\Gamma - 1$. We recall the lower bound on the dimension of a Goppa code given in Proposition 4.1.3:

$$k_\Gamma \geq n - mt,$$

where $m$ is the degree of the field extension $[\mathbb{F}_{p^m} : \mathbb{F}_p]$ and $t$ is the degree of the Goppa polynomial $g$. Thus, because $t = n - k$, we may rewrite the above as

$$k_\Gamma \geq n - 2t - (m-2)t = 2k - n - (m-2)t.$$

Since $m$ is taken to greater than 1 or else the Goppa code is not a subfield subcode of a GRS code and the Sidelnikov-Shestakov attack applies, we also have $2k - n + 1 > 2k - n - (m-2)t$. If the Goppa code is taken to be of minimum dimension, then there is no way to pick $d$ to proceed with the squaring attack since

$$\nexists d \in \mathbb{N} \text{ such that } 2k - n + 1 \leq d \leq k_\Gamma - 1 = 2k - n - 1 - (m-2)t < 2k - n + 1.$$

Also, if the Goppa code is taken to be of dimension $k_\Gamma$ such that $k_\Gamma < 2k - n + 2 = n - 2t + 2$, then the attack fails because we can't pick a value of $d$ satisfying the required condition. Thus, for any parameters used to define a particular McEliece scheme using Goppa codes, a private key can be chosen such that the Goppa code it generates will resist applications of the squaring attack.

There is also another reason why Goppa codes and, more generally, Alternant codes will resist applications of the squaring attack. The key-recovery procedure in the attack is performed by the Sidelnikov-Shestakov attack, so the critical condition we need in order to use the Sidelnikov-Shestakov attack is for $C^{(*2)}$ or $S_d(C)^{(*2)}$ to be a GRS code, where $C$ is the code spanned by the columns of $\mathbf{M}$.

Suppose $C = GRS_{n,k}(\alpha, \beta) \cap \mathbb{F}_p^n$ is an Alternant code of dimension $k_\Gamma$. For the squaring attack to be successful, we need either $C^{(*2)} = GRS_{n,2k-1}(\alpha, \beta')$ or $S_d(C)^{(*2)} = GRS_{n-d,2(k-d)}(\alpha', z * z)$ for an appropriate choice of integer $d$ outlined in Lemma 7.1.7, where the notation used for the parameters $\alpha'$ and $\beta'$ in these GRS codes carries the same meaning as before. Since $C$ and $S_d(C) \subseteq \mathbb{F}_p^n$, both $C^{(*2)}$ and $S_d(C)^{(*2)} \subseteq \mathbb{F}_p^n$ as well. For these codes to be equal to their respective GRS codes, the GRS codes must also be subspaces of $\mathbb{F}_p^n$. This will never hold true in practice by the next lemmas.

**Lemma 7.1.8.** *If $n > p$, then $GRS_{n,2k-1}(\alpha, \beta') \not\subseteq \mathbb{F}_p^n$.*

*Proof.* Suppose for a contradiction that $GRS_{n,2k-1}(\alpha, \beta') \subseteq \mathbb{F}_p^n$.

$$GRS_{n,2k-1}(\alpha, \beta') = \{(\beta_1^2 p(\alpha_1), \ldots, \beta_n^2 p(\alpha_n)) : p \in \mathbb{P}_{2k-2}(\mathbb{F}_{p^m})\},$$

so since $\{1, x, \ldots, x^{2k-2}\} \subseteq \mathbb{P}_{2k-2}(\mathbb{F}_{p^m})$, we must have that the codeword entries defined by these polynomials are in $\mathbb{F}_p$:

$$\beta_i^2 \alpha_i^j \in \mathbb{F}_p \ \ \forall i = 1, \ldots, n, \ \ \forall j = 0, \ldots, 2k-2.$$

This means for all $i = 1, \ldots, n$ and for all $j = 0, \ldots, 2k-2$, there exists $\gamma \in \mathbb{F}_p$ such that $\beta_i^2 \alpha_i^j = \gamma \iff \alpha_i^j = \gamma(\beta_i^2)^{-1}$ since $\beta_i \neq 0$ for all $i$. Thus, we get $\alpha_i^j \in \mathbb{F}_p$ for all $i, j$.

In particular for $j = 1$, we have $\alpha_i \in \mathbb{F}_p$ for all $i = 1, \ldots, n$. Since $n > p$, by the Pigeonhole Principle, there exist $i, j \in \{1, \ldots, n\}$ such that $i \neq j$ and $a_i = a_j$. But this is a contradiction. □

**Lemma 7.1.9.** *Let $d(C)$ denote the minimum distance of $C$. If $d(C) > p$, then $GRS_{n-d,2(k-d)}(\alpha', z * z) \not\subseteq \mathbb{F}_p^n$.*

*Proof.* Suppose for a contradiction that $GRS_{n-d,2(k-d)}(\alpha', z * z) \subseteq \mathbb{F}_p^n$.

$$GRS_{n-d,2(k-d)}(\alpha', z * z) = \{(z_1^2 p(\alpha_1'), \ldots, z_n^2 p(\alpha_n')) : p \in \mathbb{P}_{2(k-d)-1}(\mathbb{F}_{p^m})\},$$

so, just as in the last lemma, we will have $\alpha_i' \in \mathbb{F}_p$ for all $i = 1, \ldots, n-d$. The bounds for $d$ in Lemma 7.1.7 give us that $d \leq k_\Gamma - 1$, so $n - d \geq n - k_\Gamma + 1 \geq d(C)$ by the Singleton bound. But since $d(C) > p$, by the Pigeonhole Principle, there exist $i, j \in \{1, \ldots, n-d\}$ such that $i \neq j$ and $a_i = a_j$. This is again a contradiction. $\square$

For a McEliece scheme using a $(n, k)$ subcode of $\mathbb{F}_p^n$ as the secret code, the ciphers will be vectors in $\mathbb{F}_p^n$ and the public key contains a matrix over $\mathbb{F}_p$. The expansion that results from encoding these objects in terms of their binary equivalents is exponential in $\lceil log(p) \rceil$. In order to minimize this expansion, we take $p$ to be as small as possible, optimally taking it to be 2. In all practical implementations of McEliece, this is exactly what is done. We will certainly have $n \geq d(C) > 2$ or else if $d(C) \leq 2$, the code can't correct any errors and it would not be appropriate for the McEliece PKC. Thus, by the two preceding lemmas, if $C$ and $S_d(C)$ are chosen with respect to the virtually gratuitous condition $d(C) > p$, the squares of these codes can't possibly be equal to the corresponding GRS codes they're contained within.

By this discussion, binary Goppa codes in particular are resistant to the squaring attack. As binary, irreducible Goppa codes are the Goppa codes that allow for the greatest error correction by Theorem 4.1.6, these are the most useful for constructing a McEliece scheme. The most popular form of the McEliece PKC as seen in [B] therefore isn't threatened directly by this attack.

## 8. Wieschebrink's Guessing Attack

There are no known structural attacks on the McEliece scheme using Goppa code that run in polynomial time. We would like to have a sense of the least amount of work we need to put into acquiring these equivalent parameters before a polynomial-time attack becomes available to finish the job. To that end, we describe an attack based on one of the attacks Wieschebrink presents in [W] that recovers the parameters $(a, g)$ of a Goppa code if a subset of the entries of $a$ were known in advance.

### 8.1. **The Guessing Attack.**

We detail an attack based on Section 4.2 in [W] that can be applied to a McEliece scheme based on Goppa codes to recover the parameters in polynomial time if a certain subset of the code parameters is known in advance and given certain conditions. We then interpret this as a lower bound on the amount of information needed on the code parameters of a Goppa code before we can efficiently acquire the trapdoor of a McEliece scheme based on said Goppa code.

**Proposition 8.1.1.** *Consider the McEliece scheme using a $(n, k_\Gamma)$ Goppa code $\Gamma(a, g)$ that is the subfield subcode of $GRS_{n,k}(a, \beta)$ defined by degree-$t$ Goppa polynomial $g(x)$. Define $l := k - k_\Gamma = n - t - k_\Gamma$. Suppose $n \geq k_\Gamma + 2l + 4$ and that the entries $a_{k_\Gamma+1}, \ldots, a_{k_\Gamma+2l+4}$ of the parameter $a$ were known and that none of the entries $a_{k_\Gamma}, \ldots, a_n$ are roots of any of the row polynomials $q_{R_i}$ for all $i = 1, \ldots, k_\Gamma$, where these polynomials possess the same meaning as in Section 6.3. There exists an attack of polynomial algebraic complexity that recovers the rest of $(a, g)$.*

*Proof.* We begin by row-reducing the transpose of the public matrix $\mathbf{M}$, bringing it to the form $E(\mathbf{M}^\intercal)$, whereupon we get the following interpretation of each row

of $E(\mathbf{M}^\intercal)$:

$$R_i = (\beta_1 q_{R_i}(a_1), \ldots, \beta_n q_{R_i}(a_n)) \;\; \forall i \in \{1, \ldots, k_\Gamma\}.$$

We recall that each row polynomial $q_{R_i}$ can be written as

$$q_{R_i} = \left( \prod_{j \in \{1,\ldots,k_\Gamma\} \setminus \{i\}} (x - a_j) \right) \rho_i(x) \quad \text{such that } \rho_i \in \mathbb{P}_{k-k_\Gamma}(\mathbb{F}_{p^m})$$

by the characterization immediately preceding Proposition 6.3.2. We divide by the non-zero entries of the rows of $E(\mathbf{M}^\intercal)$ and we get the equations

$$\frac{(R_i)_j}{(R_h)_r} = \frac{\beta_j q_{R_i}(a_j)}{\beta_r q_{R_h}(a_r)} \quad \forall i, h \in \{1, \ldots, k_\Gamma\}, \;\; j, r \in \{k_\Gamma{+}1, \ldots, n\} \text{ such that } (R_h)_r \neq 0.$$

For $j = r$, these equations become

$$\frac{(R_i)_j}{(R_h)_j} = \frac{(a_j - a_h)\rho_i(a_j)}{(a_j - a_i)\rho_h(a_j)} \quad \forall i, h\{1, \ldots, k_\Gamma\}, \;\; j \in \{k_\Gamma{+}1, \ldots, n\} \text{ such that } (R_h)_j \neq 0.$$

Take $h = k_\Gamma$. Define $\tilde{P}_i(x) := (x - a_{k_\Gamma})\rho_i(x)$ and $\tilde{Q}_i(x) := (x - a_i)\rho_{k_\Gamma}(x)$. With this, we rewrite the last equation as

(8.1)
$$\frac{(R_i)_j}{(R_{k_\Gamma})_j} = \frac{\tilde{P}_i(a_j)}{\tilde{Q}_i(a_j)} \quad \forall i \in \{1, \ldots k_\Gamma\}, \;\; j \in \{k_\Gamma + 1, \ldots, n\} \text{ such that } (R_{k_\Gamma})_j \neq 0.$$

Since the degrees of $deg(\tilde{P}_i)$ and $deg(\tilde{Q}_i)$ are both less than or equal to $k - k_\Gamma + 1 = l + 1$, we need to know at least $l + 2$ points that either one passes through to be able to interpolate it. Since we know $a_{k_\Gamma+1}, \ldots, a_{k_\Gamma+2l+4}$, we know $2l + 4$ points $\tilde{P}_i$ and $\tilde{Q}_i$ pass through, so we can interpolate them. From equation (8.1), we recognize that $\tilde{P}_i$ passes through the points $\left\{(a_j, \frac{(R_i)_j}{(R_{k_\Gamma})_j}\tilde{Q}_i(a_j)) : j = k_\Gamma + 1, \ldots, k_\Gamma + 2l + 4\right\}$ and $\tilde{Q}_i$ passes through the points $\left\{(a_j, \frac{(R_{k_\Gamma})_j}{(R_i)_j}\tilde{P}_i(a_j)) : j = k_\Gamma + 1, \ldots, k_\Gamma + 2l + 4\right\}$. Let $\tilde{P}_i(x) = \sum_{r=0}^{l+1} p_r x^r$ and $\tilde{Q}_i(x) = \sum_{r=0}^{l+1} q_r x^r$.

We will define two matrices as follows:

$$\mathbf{A} := \begin{bmatrix} 1 & a_{k_\Gamma+1} & \ldots & {a_{k_\Gamma+1}}^{l+1} \\ 1 & a_{k_\Gamma+2} & \ldots & {a_{k_\Gamma+2}}^{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & a_{k_\Gamma+l+2} & \ldots & {a_{k_\Gamma+l+2}}^{l+1} \end{bmatrix} \quad \text{and} \quad \mathbf{B} := \begin{bmatrix} 1 & a_{k_\Gamma+l+3} & \ldots & {a_{k_\Gamma+l+3}}^{l+1} \\ 1 & a_{k_\Gamma+l+4} & \ldots & {a_{k_\Gamma+l+4}}^{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & a_{k_\Gamma+2l+4} & \ldots & {a_{k_\Gamma+2l+4}}^{l+1} \end{bmatrix}.$$

We recognize immediately that $\mathbf{A}$ and $\mathbf{B}$ are both square Vandermonde matrices, so because $a_i \neq a_j$ for all $i \neq j$, $\mathbf{A}$ and $\mathbf{B}$ are both invertible.

We also define two vectors $\mathbf{p}, \mathbf{q} \in \mathbb{F}_{p^m}^{l+2}$ to be the vectors of the coefficients of $\tilde{P}_i$ and $\tilde{Q}_i$, respectively: $\mathbf{p} := [p_0, p_1, \ldots, p_{l+1}]^\intercal$ and $\mathbf{q} := [q_0, q_1, \ldots, q_{l+1}]^\intercal$. The

following system follows directly from matrix multiplication.

$$\mathbf{Ap} = \begin{bmatrix} \tilde{P}_i(a_{k_\Gamma+1}) \\ \tilde{P}_i(a_{k_\Gamma+2}) \\ \vdots \\ \tilde{P}_i(a_{k_\Gamma+l+2}) \end{bmatrix} = \begin{bmatrix} \tilde{Q}_i(a_{k_\Gamma+1})\frac{(R_i)_{k_\Gamma+1}}{(R_{k_\Gamma})_{k_\Gamma+1}} \\ \tilde{Q}_i(a_{k_\Gamma+2})\frac{(R_i)_{k_\Gamma+2}}{(R_{k_\Gamma})_{k_\Gamma+2}} \\ \vdots \\ \tilde{Q}_i(a_{k_\Gamma+l+2})\frac{(R_i)_{k_\Gamma+l+2}}{(R_{k_\Gamma})_{k_\Gamma+l+2}} \end{bmatrix}$$

Rewrite the above as follows.

$$\mathbf{p} = \mathbf{A}^{-1} \begin{bmatrix} \tilde{Q}_i(a_{k_\Gamma+1})\frac{(R_i)_{k_\Gamma+1}}{(R_{k_\Gamma})_{k_\Gamma+1}} \\ \tilde{Q}_i(a_{k_\Gamma+2})\frac{(R_i)_{k_\Gamma+2}}{(R_{k_\Gamma})_{k_\Gamma+2}} \\ \vdots \\ \tilde{Q}_i(a_{k_\Gamma+l+2})\frac{(R_i)_{k_\Gamma+l+2}}{(R_{k_\Gamma})_{k_\Gamma+l+2}} \end{bmatrix} \tag{8.2}$$

Likewise, we also have the following system involving the remaining $a_{k_\Gamma+l+3}, \ldots, a_{k_\Gamma+2l+4}$ and $\tilde{Q}_i$.

$$\mathbf{Bq} = \begin{bmatrix} \tilde{Q}_i(a_{k_\Gamma+l+3}) \\ \tilde{Q}_i(a_{k_\Gamma+l+4}) \\ \vdots \\ \tilde{Q}_i(a_{k_\Gamma+2l+4}) \end{bmatrix} = \begin{bmatrix} \tilde{P}_i(a_{k_\Gamma+l+3})\frac{(R_i)_{k_\Gamma+l+3}}{(R_{k_\Gamma})_{k_\Gamma+l+3}} \\ \tilde{P}_i(a_{k_\Gamma+l+4})\frac{(R_i)_{k_\Gamma+l+4}}{(R_{k_\Gamma})_{k_\Gamma+l+4}} \\ \vdots \\ \tilde{P}_i(a_{k_\Gamma+2l+4})\frac{(R_i)_{k_\Gamma+2l+4}}{(R_{k_\Gamma})_{k_\Gamma+2l+4}} \end{bmatrix}$$

We also rewrite this by inverting $\mathbf{B}$.

$$\mathbf{q} = \mathbf{B}^{-1} \begin{bmatrix} \tilde{P}_i(a_{k_\Gamma+l+3})\frac{(R_i)_{k_\Gamma+l+3}}{(R_{k_\Gamma})_{k_\Gamma+l+3}} \\ \tilde{P}_i(a_{k_\Gamma+l+4})\frac{(R_i)_{k_\Gamma+l+4}}{(R_{k_\Gamma})_{k_\Gamma+l+4}} \\ \vdots \\ \tilde{P}_i(a_{k_\Gamma+2l+4})\frac{(R_i)_{k_\Gamma+2l+4}}{(R_{k_\Gamma})_{k_\Gamma+2l+4}} \end{bmatrix} = \mathbf{B}^{-1} Diag(\tfrac{(R_i)_{k_\Gamma+l+3}}{(R_{k_\Gamma})_{k_\Gamma+l+3}}, \ldots, \tfrac{(R_i)_{k_\Gamma+2l+4}}{(R_{k_\Gamma})_{k_\Gamma+2l+4}}) \begin{bmatrix} \tilde{P}_i(a_{k_\Gamma+l+3}) \\ \tilde{P}_i(a_{k_\Gamma+l+4}) \\ \vdots \\ \tilde{P}_i(a_{k_\Gamma+2l+4}) \end{bmatrix}$$

But we see this can easily be rewritten as

$$\mathbf{q} = \mathbf{B}^{-1} Diag(\tfrac{(R_i)_{k_\Gamma+l+3}}{(R_{k_\Gamma})_{k_\Gamma+l+3}}, \ldots, \tfrac{(R_i)_{k_\Gamma+2l+4}}{(R_{k_\Gamma})_{k_\Gamma+2l+4}})\mathbf{Bp}.$$

Define $\mathbf{D}_1 := Diag(\frac{(R_i)_{k_\Gamma+l+3}}{(R_{k_\Gamma})_{k_\Gamma+l+3}}, \ldots, \frac{(R_i)_{k_\Gamma+2l+4}}{(R_{k_\Gamma})_{k_\Gamma+2l+4}})$.
Define $\mathbf{D}_2 := Diag(\frac{(R_i)_{k_\Gamma+1}}{(R_{k_\Gamma})_{k_\Gamma+1}}, \ldots, \frac{(R_i)_{k_\Gamma+l+2}}{(R_{k_\Gamma})_{k_\Gamma+l+2}})$.

By our expression for $\mathbf{p}$ in equation (8.2), this becomes

$$\mathbf{q} = \mathbf{B}^{-1}\mathbf{D}_1\mathbf{B}\mathbf{A}^{-1}\mathbf{D}_2\mathbf{A}\mathbf{q}.$$

Evidently, $\mathbf{q}$ is a 1-eigenvector of $\mathbf{V} := \mathbf{B}^{-1}\mathbf{D}_1\mathbf{B}\mathbf{A}^{-1}\mathbf{D}_2\mathbf{A}$, so we can find $\mathbf{q}$ from $\ker(\mathbf{V}-\mathbf{I})$. Suppose we have found $\mathbf{q}$, so we've identified $\tilde{Q}_i(x)$ (note that we won't question its uniqueness in this argument).

We can use $\tilde{Q}_i(x)$ to find $a_1, \ldots, a_{k_\Gamma - 1}$. We will factor $\tilde{Q}_i(x)$ into irreducibles over $\mathbb{F}_{p^m}$ for all $i = 1, \ldots, k_\Gamma - 1$. We get

$$\tilde{Q}_i(x) = (x - a_i)\rho_{k_\Gamma}(x) = (x - a_i)\prod_{j=1}^{u} r_j(x)$$

such that $\rho_{k_\Gamma}(x) = r_1(x) \cdots r_u(x)$ is the factorization into irreducibles of $\rho_{k_\Gamma}(x)$. The factors $r_j(x)$ for all $j = 1, \ldots, u$ will all factor $\tilde{Q}_i$ no matter the choice of $i$, but since $a_i \neq a_j$ for all $i \neq j$, the factor $x - a_i$ will be different for each different choice of $i$. For any particular choice of $i$, we may partition the factors of $\tilde{Q}_i$ into the multiset of factors shared by all of $\tilde{Q}_i$ for all $i = 1, \ldots, k_\Gamma - 1$, which will be $\{r_j(x) : j = 1, \ldots, u\}$, and the one linear factor $x - a_i$ not belonging to this multiset. Note that because this is a multiset, even if there were some $j \in \{1, \ldots, u\}$ such that $r_j(x) = x - a_i$, we won't have $x - a_i \in \{r_j(x) : j = 1, \ldots, u\}$ as well since this would require $|\{r_j(x) : j = 1, \ldots, u\}| > deg(\rho_{k_\Gamma})$, which can't happen. With this, we can identify $a_i$ for any $i \in \{1, \ldots, k_\Gamma - 1\}$ from the constant term in the linear factor of $\tilde{Q}_i$ not belonging to $\{r_j(x) : j = 1, \ldots, u\}$.

To recover the remaining entries $a_{k_\Gamma}, a_{k_\Gamma + 2l + 5}, a_{k_\Gamma + 2l + 6}, \ldots, a_n$, we will repeat the above process for an appropriately-chosen permutation of $E(\mathbf{M}^\intercal)$. We begin by finding a column $E(\mathbf{M}^\intercal)_s$ of $E(\mathbf{M}^\intercal)$ among $\{E(\mathbf{M}^\intercal)_i : i \in \{k_\Gamma\} \cup \{k_\Gamma + 2 + 5, \ldots, n\}\}$ such that the column's first entry, $(E(\mathbf{M}^\intercal)_s)_1$, is non-zero, thereby ensuring that $E(\mathbf{M}^\intercal)_s$ is a linear combination of the first column of $E(\mathbf{M}^\intercal)$ (and possibly some others).

Define the following permutation:

$$\pi_s : \mathbb{F}_{p^m}^n \to \mathbb{F}_{p^m}^n \text{ by } (v_1, \ldots, v_n) \mapsto (v_s, v_2, \ldots, v_{s-1}, v_1, v_{s+1}, \ldots, v_n).$$

Define $\mathbf{M}'$ to be the matrix $E(\mathbf{M}^\intercal)$ with $\pi_s$ applied to each of its rows. Row-reduce $\mathbf{M}'$, bringing it to its RREF form $E(\mathbf{M}')$. We'll say that the $i^{th}$ row of $E(\mathbf{M}')$ is $R'_i = (\pi_s(\beta)_1 q_{R'_i}(\pi_s(a)_1), \ \ldots \ , \pi_s(\beta)_n q_{R'_i}(\pi_s(a)_n))$, where we are again associating a polynomial $q_{R'_i} \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ to the $i^{th}$ row $R'_i$. By dividing $R'_i$ by $R'_{k_\Gamma}$ on their non-zero entries, we get

$$\frac{(R'_i)_j}{(R'_{k_\Gamma})_j} = \frac{(\pi_s(a)_j - \pi_s(a)_{k_\Gamma})\rho'_i(\pi_s(a)_j)}{(\pi_s(a)_j - \pi_s(a)_i)\rho'_{k_\Gamma}(\pi_s(a)_j)} \quad \text{such that } \rho'_i, \rho'_{k_\Gamma} \in \mathbb{P}_l(\mathbb{F}_{p^m}).$$

Define $\tilde{P}'_i(x) := (x - \pi_s(a)_{k_\Gamma})\rho'_i(x)$ and $\tilde{Q}'_i(x) := (x - \pi_s(a)_i)\rho'_{k_\Gamma}(x)$ for each $i = 1, \ldots, k_\Gamma - 1$. Proceed with the method from before to identify $\tilde{Q}'_i(x)$. Notice that for all $i = 2, \ldots, k_\Gamma - 1$, we have $\pi_s(a)_i = a_i$. Since we already know $a_2, \ldots, a_{k_\Gamma - 1}$, we can identify $\rho'_{k_\Gamma}(x)$ by dividing $\tilde{Q}'_i(x)$ by $x - a_i$ for all $i \in \{2, \ldots, k_\Gamma - 1\}$. Having identified $\rho'_{k_\Gamma}(x)$, we can find $\pi_s(a)_1 = a_s$ by dividing $\tilde{Q}_1(x)$ by $\rho'_{k_\Gamma}(x)$ and extracting the constant term of the quotient.

Repeat this for all $s \in \{k_\Gamma\} \cup \{k_\Gamma + 2l + 5, \ldots, n\} \backslash \{i : a_i \text{ is known}\}$ such that the $s^{th}$ column satisfies $(E(\mathbf{M}^\intercal)_s)_1 \neq 0$. If there remain unidentified entries of $a$, proceed by repeating this process for all $s \in \{k_\Gamma\} \cup \{k_\Gamma + 2l + 5, \ldots, n\} \backslash \{i : a_i \text{ is known}\}$ such that $(E(\mathbf{M}^\intercal)_s)_2 \neq 0$, adapting the process as needed for having chosen the second entry of $E(\mathbf{M}')_s$ instead of the first in the latter condition. For all choices of $j \in \{3, \ldots, k_\Gamma\}$, continue repeating the selection of $s \in \{k_\Gamma\} \cup \{k_\Gamma + 2l + 5, \ldots, n\} \backslash \{i :$

$a_i$ is known$\}$ such that $(E(\mathbf{M}^\intercal)_s)_j \neq 0$ and following through with the according process to find $a_s$ until there are no more values of $s$ to pick. As long as no column of $E(\mathbf{M}^\intercal)$ is the zero vector, this algorithm will finish by fully identifying $a$. Of course, because the rows of $E(\mathbf{M}^\intercal)$ generate $\Gamma(a,g)$, a column of zeros means that every codeword in $\Gamma(a,g)$ has a zero in the position indexed by that column, say for instance column $i$. This happens if and only if $a_i$ is a root to all polynomials in $\mathcal{P}$, which happens if and only if $x-a_i$ factors all polynomials in any basis of $\mathcal{P}$. This is unlikely to hold if $\mathcal{P}$ was chosen randomly, but even if it did occur, we could still find $a_i$ by the same approach as above with minor modifications. The modified approach to recover $a_i$ will be treated in the appendix.

Because all the work we needed to perform to identify $a$ involved solving linear systems, row-reducing matrices, and dividing polynomials, all of which can be performed with polynomial complexity, and since these were all performed a polynomial number of times in $k_\Gamma$, identifying all of $a$ requires a polynomial number of arithmetic operations. In particular, the more streamlined method proposed by Wieschebrink in [W] requires $\mathcal{O}(k_\Gamma^2 n + k_\Gamma l^3)$ operations to find $a$.

To find $\beta$, which then lets us find the Goppa polynomial $g(x)$, we recall the following fact about the dual of GRS codes given in Proposition 3.1.7: if $\alpha$ and $\beta$ are vectors in $\mathbb{F}_{p^m}^n$ such that $GRS_{n,k}(\alpha,\beta)$ is a $(n,k)$ GRS code, then for $\gamma \in \mathbb{F}_{p^m}^n$ whose $i^{th}$ entry is defined by $\gamma_i := \beta_i^{-1}\prod_{j\neq i}(\alpha_i-\alpha_j)^{-1}$, we have $GRS_{n,k}(\alpha,\beta)^\perp = GRS_{n,n-k}(\alpha,\gamma)$. Since $\Gamma(a,g) \subseteq GRS_{n,k}(a,\beta)$, the rows of $E(\mathbf{M}^\intercal)$ are codewords of $GRS_{n,k}(a,\beta)$, so we can see by this fact about the dual of GRS codes that

$$R_i \begin{bmatrix} \gamma_1 a_1^j \\ \gamma_2 a_2^j \\ \vdots \\ \gamma_n a_n^j \end{bmatrix} = 0 \quad \forall i = 1,\dots,k_\Gamma, \ \ \forall j = 0,\dots,n-k-1,$$

given that $(\gamma_1 a_1^j,\dots,\gamma_n a_n^j) \in \{(\gamma_1 p(a_1),\dots,\gamma_n p(a_n)) : p \in \mathbb{P}_{n-k-1}(\mathbb{F}_{p^m})\}$, where the latter set is $GRS_{n,n-k}(a,\gamma) = GRS_{n,k}(a,\beta)^\intercal$. But because $\gamma$ is related to $a$ and $\beta$ by the relation $\gamma_i = \beta_i^{-1}\prod_{j\neq i}(a_i-a_j)^{-1}$, we need only find $\gamma$ satisfying the above to find $\beta$ given that $a$ is already known. The last set of equations can be rewritten as the homogeneous system

$$[(R_i)_1 a_1^j \ \ \dots \ \ (R_i)_n a_n^j]_{\substack{1\leq i\leq k_\Gamma \\ 0\leq j\leq n-k-1}} \gamma = \mathbf{0}_{k_\Gamma(n-k)\times 1}.$$

This gives $k_\Gamma(n-k)$ equations in $n$ unknowns. Typically, $k_\Gamma(n-k) \geq n$, so this should be enough to uniquely determine $\gamma$ (which then fully determines $\beta$) as indicated in [W]. Indeed, the minimum dimension of a Goppa code is $k_\Gamma = n-mt$, so we'll have $(t-1)n \geq mt^2 \iff (n-mt)t \geq n \implies k_\Gamma(n-k) \geq n$. We have that

$$(t-1)n \geq mt^2 \tag{8.3}$$

holds for all proposed parameters by Bernstein et al. in Chapter 3 of [B], so we can uniquely solve for $\beta$ in practice.

TABLE 1. Verification that the parameters in [B] satisfy (8.3)

| Parameters | $(t-1)n$ | $mt^2$ |
|---|---|---|
| $n = 3488$, $m = 12$, $t = 64$ | 219744 | 49152 |
| $n = 4608$, $m = 13$, $t = 96$ | 437760 | 119808 |
| $n = 6688$, $m = 13$, $t = 128$ | 849376 | 212992 |
| $n = 6960$, $m = 13$, $t = 119$ | 821280 | 184093 |
| $n = 8192$, $m = 13$, $t = 128$ | 1040384 | 212992 |

Finding $\gamma$ can be done in $\mathcal{O}(n(k_\Gamma(n-k))^2)$ arithmetic operations and then finding $\beta$ will take another $\mathcal{O}(n^2)$ operations. From this, $g(x)$ can be reconstructed by Lagrangian interpolation in $\mathcal{O}(nlog(n))$ operations. Thus, the Goppa polynomial can be recovered in $\mathcal{O}(n(k_\Gamma(n-k))^2)$ operations, giving us a polynomial complexity for the whole attack. □

Instead of assuming $a_1 = 0$ and $a_2 = 1$, we may instead assume $a_{k_\Gamma+1} = 0$ and $a_{k_\Gamma+2} = 1$ and the above algorithm will find equivalent parameters to the secret Goppa code $\Gamma(a, g)$ as suggested in [W]. This reduces the amount of information we need to know before we can quickly solve for the private key from knowing $2l+4$ entries of $a$ to knowing just $2l + 2$ entries.

As a consequence, if we are attacking a McEliece scheme based on a $(n, k_\Gamma)$ Goppa code that is a subcode of $GRS_{n,k}(a, \beta)$, our condition for breaking this scheme is to efficiently identify $a_{k_\Gamma+1}, \ldots, a_{k_\Gamma+2l+2}$ (as long as $n \geq k_\Gamma + 2l + 2$ and none of the last $n - k_\Gamma + 1$ entries are roots of any row polynomial $q_{R_i}$) as we can apply the attack of Proposition 8.1.1 given these values to find the full code parameters with polynomial complexity.

We can also see that Proposition 8.1.1 applies if we instead knew any $2l + 2$ entries of the last $n - k_\Gamma$ entries of $a$ (assuming that none of these are 0 or 1 because we assume two other entries among the last $n - k_\Gamma$ are 0 and 1). The only modifications to the attack are that we change the matrices $\mathbf{A}$ and $\mathbf{B}$ to reflect the new known values for the identification of $a_1, \ldots a_{k_\Gamma-1}$ and, to identify the remaining entries of $a$, when we pick the $s^{th}$ column of $E(\mathbf{M}^\intercal)$, we choose $s$ from $\{1, \ldots, n\} \setminus \{i : a_i \text{ is known}\}$ at each step of the recovery.

## 9. Galois-Closure-Based Attack

We present further constructions of linear codes defined by operations we can perform on a parent code in the form of punctured and shortened codes. We then introduce a new key-recovery approach for McEliece schemes based on Goppa codes whose difficulty lies in the problem of identifying a linear code given its Galois closure. We consider a modification of this approach using punctured codes motivated by the Guessing Attack from the last section and several easy cases we may potentially run into. Ultimately, we reveal that these easy cases will not occur if we assume the McEliece scheme is based on full-rank Goppa codes, but the development results in a condition on the parameters of a GRS code that can be used to inform us of when its subfield subcode is of full rank. We also mention several open problems for future work.

9.1. **Results About Puncturing and Shortening.** We present the code operations of puncturing and shortening on an indexing set and study the properties of punctured GRS, Alternant, and Goppa codes.

**Definition 9.1.1.** Let $C$ be a $(n, k)$ linear code and let $I \subseteq \{1, \ldots, n\}$ be a set of indices. We define the *punctured code* of $C$ by $I$ as

$$P_I(C) := \{(c_i)_{i \notin I} : c \in C\}.$$

Perhaps counterintuitively, this definition means that the coordinates of codewords of $C$ indexed by $I$ are *removed* by the puncturing operation, not kept.

We reintroduce code shortening (Definition 7.1.5) with a bit more generality so as to describe the relationship between it and code puncturing.

**Definition 9.1.2.** Let $C$ be a $(n, k)$ linear code and let $I \subseteq \{1, \ldots, n\}$ be a set of indices. We define the *shortened code* of $C$ by $I$ as

$$S_I(C) := \{(c_i)_{i \notin I} : c \in C \text{ and } \forall i \in I,\, c_i = 0\}.$$

There exists a relationship between shortened and punctured codes that bears a strong resemblance to Delsarte duality.

**Proposition 9.1.3.** *Let $C$ be a $(n, k)$ linear code and let $I \subseteq \{1, \ldots, n\}$ be a set of indices. We have both*

$$S_I(C)^\perp = P_I(C^\perp) \text{ and } P_I(C)^\perp = S_I(C^\perp).$$

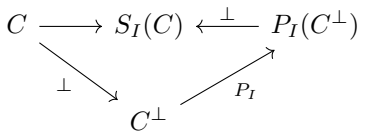
*Proof.* See Theorem 1.5.7 in [H]. □

Next, we make the observation that puncturing is a morphism of vector spaces. It is clear to see that it is a linear operation on the codewords of a given code, so we can represent it by a matrix. If we puncture a code of length $n$ on the set of indices $I = \{i_1, \ldots, i_s\} \subseteq \{1, \ldots, n\}$, the matrix of this transformation (with respect to the standard basis vectors of $\mathbb{F}_{p^m}^n$ and $\mathbb{F}_{p^m}^{n-s}$) will be

$$[P_I] = \begin{bmatrix} e_{j_1}^\intercal \\ e_{j_2}^\intercal \\ \vdots \\ e_{j_{n-s}}^\intercal \end{bmatrix}$$

such that $\{j_1, \ldots, j_{n-s}\} = \{1, \ldots, n\} \setminus I = \overline{I}$ (with $j_1 < j_2 < \cdots < j_{n-s}$) and $e_j$ denotes the $j^{th}$ standard basis vector of $\mathbb{F}_{p^m}^n$. We then see that our notation $P_I(C)$ reflects that the punctured code is the image of $C$ under the linear map $P_I : \mathbb{F}_{p^m}^n \to \mathbb{F}_{p^m}^{n-s}$ defined by the matrix $[P_I]$.

We will keep this observation in mind for upcoming proofs. For now, we can summarize the basic relationships between a linear code $C$ and the codes we can construct by puncturing it, shortening it, and taking its dual with the following diagram.




The diagram is made commutative by noticing that $P_I(C^\perp)^\perp = S_I(C)$.

We will now present certain results about puncturing in the context of GRS, Goppa, and Alternant codes. We will begin by recalling the polynomial-evaluation characterizations of codes from these families. By this, we mean characterizations of the form given in Definition 3.1.1 for GRS codes and Proposition 4.2.2 for Goppa codes. These characterizations are often written using the evaluation map for a vector $\alpha \in \mathbb{F}_{p^m}^n$ and component-wise product (both introduced in Section 7.1). We will recall the definitions of these maps and the polynomial-evaluation characterizations of these codes. First, the definitions:

$$ev_\alpha : \mathbb{F}_{p^m}[x] \to \mathbb{F}_{p^m}^n \text{ given by } f \mapsto (f(\alpha_i))_{i=1}^n,$$
$$* : \mathbb{F}_{p^m}^n \times \mathbb{F}_{p^m}^n \to \mathbb{F}_{p^m}^n \text{ given by } ((x_i)_{i=1}^n, (y_i)_{i=1}^n) \mapsto (x_i y_i)_{i=1}^n.$$

The characterization for a $(n,k)$ GRS code defined by vectors $\alpha, \beta$ is

$$GRS_{n,k}(\alpha,\beta) = \{\beta * ev_\alpha(f) : f \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})\}.$$

For a $(n, k_\Gamma)$ Goppa code $\Gamma(\alpha, g)$ where $g \in \mathbb{F}_{p^m}[x]$ is a degree-$t$ polynomial, if we take $k = n - t$ and $\beta \in \mathbb{F}_{p^m}^n$ such that $\beta_i = \frac{g(\alpha_i)}{\prod_{j\neq i}(\alpha_j - \alpha_i)}$ for all $i = 1, \ldots, n$, then its polynomial-evaluation characterization is

$$\Gamma(\alpha, g) = \{\beta * ev_\alpha(q) : q \in \mathcal{P}\},$$

where $\mathcal{P}$ is a $\mathbb{F}_p$-linear subspace of $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ that is of dimension $k_\Gamma$ that can be determined from $\alpha$ and $\beta$. This Goppa code is the subfield subcode of $GRS_{n,k}(\alpha,\beta)$.

We will now consider puncturing these codes. For a vector $x \in \mathbb{F}_{p^m}^n$ and a set $I \subseteq \{1, \ldots, n\}$, we will introduce the notation $x|_I := (x_i)_{i\in I}$ for convenience when dealing with vectors punctured on $I$.

**Proposition 9.1.4.** *Let $GRS_{n,k}(\alpha,\beta)$ be a $(n,k)$ GRS code and let $I \subseteq \{1,\ldots,n\}$ be a set such that $|I| \leq n-k$. We have $P_I(GRS_{n,k}(\alpha,\beta)) = GRS_{|\overline{I}|,k}(\alpha|_{\overline{I}}, \beta|_{\overline{I}})$.*

*Proof.* This follows immediately from the definitions of a GRS and puncturing.

$$\begin{aligned} P_I(GRS_{n,k}(\alpha,\beta)) &= \{(\beta_i f(\alpha_i))_{i\in\overline{I}} : f \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})\} \\ &= \{\beta|_{\overline{I}} * ev_{\alpha|_{\overline{I}}}(f) : f \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})\} \\ &= GRS_{|\overline{I}|,k}(\alpha|_{\overline{I}}, \beta|_{\overline{I}}) \end{aligned}$$

The condition $|I| \leq n-k$ is only needed so that when we puncture $GRS_{n,k}(\alpha,\beta)$ on $I$, the resulting code will be of length at least $k$, which is necessary for the punctured code to also have dimension $k$. □

Thus, when we puncture a GRS code on a set $I$ of size less than or equal to its codimension, the resulting code will be a shorter GRS code of the same dimension. We will now prove a similar result for punctured Goppa codes.

**Proposition 9.1.5.** *Let $\Gamma(\alpha, g)$ be a $(n, k_\Gamma)$ Goppa code with degree-$t$ Goppa polynomial $g(x)$. If $I \subseteq \{1, \dots, n\}$ is a set such that $|I| \leq n - k = t$, then $P_I(\Gamma(\alpha, g))$ will be a $k_\Gamma$-dimensional subcode of an Alternant code of length $s = |\overline{I}|$.*

*Proof.* We begin by using the fact that $\Gamma(\alpha, g)$ is the subfield subcode of a GRS code to show the punctured code will be an Alternant code. For $k = n - t$ and the appropriately chosen $\beta$, we have

$$P_I(\Gamma(\alpha, g)) = P_I(GRS_{n,k}(\alpha, \beta) \cap \mathbb{F}_p^n) \subseteq P_I(GRS_{n,k}(\alpha, \beta)) \cap \mathbb{F}_p^n,$$

since for any linear code $C$, we observe $P_I(C \cap \mathbb{F}_p^n) \subseteq P_I(C) \cap \mathbb{F}_p^n$. By the last proposition, the code on the right-hand side is the subfield subcode of a GRS code, so it is Alternant and its length is $s = |\overline{I}|$.

What remains is to show the dimension of the punctured code is again $k_\Gamma$. Let $\overline{I} = \{j_1, \dots, j_s\}$ with $j_1 < j_2 < \cdots < j_s$. Recall from the polynomial-evaluation characterization of $\Gamma(\alpha, g)$ that

$$\Gamma(\alpha, g) = \{\beta * ev_\alpha(q) : q \in \mathcal{P}\},$$

where $\mathcal{P}$ is a $\mathbb{F}_p$-linear subspace of $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ that is of dimension $k_\Gamma$. Write a basis for $\mathcal{P}$ as $\{q_1, \dots, q_{k_\Gamma}\}$. We'll show that $Q = \{\beta|_{\overline{I}} * ev_{\alpha|_{\overline{I}}}(q_i) : i = 1, \dots, k_\Gamma\}$ is a basis for $P_I(\Gamma(\alpha, g))$. Firstly, it's clear to see that

$$P_I(\Gamma(\alpha, g)) = \{\beta|_{\overline{I}} * ev_{\alpha|_{\overline{I}}}(q) : q \in \mathcal{P}\},$$

so for each codeword $c \in P_I(\Gamma(\alpha, g))$, there exists a polynomial $q \in \mathcal{P}$ such that $c = \beta|_{\overline{I}} * ev_{\alpha|_{\overline{I}}}(q)$. Since $\{q_1, \dots, q_{k_\Gamma}\}$ is a basis for $\mathcal{P}$, there exist $\lambda_1, \dots, \lambda_{k_\Gamma} \in \mathbb{F}_p$ such that $\sum_{i=1}^{k_\Gamma} \lambda_i q_i(x) = q(x)$. Thus, we have

$$\begin{aligned}
c &= \beta|_{\overline{I}} * ev_{\alpha|_{\overline{I}}} \left( \sum_{i=1}^{k_\Gamma} \lambda_i q_j(x) \right) \\
&= \left( \beta_{j_1} \left( \sum_{i=1}^{k_\Gamma} \lambda_i q_i \right)(\alpha_{j_1}), \ \dots \ , \beta_{j_s} \left( \sum_{i=1}^{k_\Gamma} \lambda_i q_i \right)(\alpha_{j_s}) \right) \\
&= \sum_{i=1}^{k_\Gamma} \lambda_i (\beta_{j_1} q_i(\alpha_{j_1}), \dots, \beta_{j_s} q_i(\alpha_{j_s})) \\
&= \sum_{i=1}^{k_\Gamma} \lambda_i \beta|_{\overline{I}} * ev_{\alpha|_{\overline{I}}}(q_i),
\end{aligned}$$

which exactly means $c \in \text{span}_{\mathbb{F}_p}(Q)$. Next, we will show $Q$ is linearly independent. We now let $\lambda_1, \dots, \lambda_{k_\Gamma} \in \mathbb{F}_p$ and we consider the equation

$$\sum_{i=1}^{k_\Gamma} \lambda_i \beta|_{\overline{I}} * ev_{\alpha|_{\overline{I}}}(q_i) = \sum_{i=1}^{k_\Gamma} \lambda_i \begin{bmatrix} \beta_{j_1} q_i(\alpha_{j_1}) \\ \vdots \\ \beta_{j_s} q_i(\alpha_{j_s}) \end{bmatrix} = 0.$$

But since $\alpha, \beta$ defines a GRS code, $\beta_i \neq 0$ for all $i$, so this implies

$$\left( \sum_{i=1}^{k_\Gamma} \lambda_i q_i \right)(\alpha_j) = 0 \quad \text{for all } j \in \overline{I}.$$

Recall that $\mathcal{P} \subseteq \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$, so $deg(q_i) < k$ for all $i = 1, \ldots, k_\Gamma$. There is a unique polynomial of degree at most $k-1$ passing through at least $k$ distinct points. Since $s = n - |I| \geq k$, we know at least $k$ points that the polynomial $\sum_{i=1}^{k_\Gamma} q_i(x)$ passes through; in particular, it must pass through each point in $\{(a_j, 0) : j \in \overline{I}\}$. The zero polynomial satisfies this condition, so because it must be the unique polynomial satisfying this,

$$\sum_{i=1}^{k_\Gamma} \lambda_i q_i(x) = 0 \implies \lambda_1, \ldots, \lambda_{k_\Gamma} = 0 \quad \text{by the linear independence of } \{q_1, \ldots, q_{k_\Gamma}\}$$

Thus, $Q$ is a basis for $P_I(\Gamma(\alpha, g))$, so $\dim_{\mathbb{F}_p}(P_I(\Gamma(\alpha, g))) = k_\Gamma = \dim_{\mathbb{F}_p}(\Gamma(\alpha, g))$. □

**Remark 9.1.6.** The last proposition applies if we considered an Alternant code instead of a Goppa code: that is, puncturing a $(n, k)$ Alternant code on a set of indices $I$ such that $|I| \leq n - k$ will yield a $(|\overline{I}|, k)$ subcode of an Alternant code. The proof is exactly the same as for the Goppa code since Alternant codes possess a polynomial-evaluation characterization of the exact same form as Goppa codes (which can be seen from the proofs of Theorem 1 in [SB] and Proposition 4.2.2).

9.2. **Approach for Using the Galois Closure to Identify Equivalent Code Parameters.** We detail the groundwork for a new approach to identifying the equivalent parameters for a secret Goppa code in a McEliece scheme using the code operations of taking the dual and extending a $\mathbb{F}_p$-linear over $\mathbb{F}_{p^m}$. The work in this section builds on the results collected in sections 2.1 and 2.2, so the reader should be familiar with the aforementioned material before proceeding.

We will now give a first insight into how we intend on tackling the key-recovery problem before filling it in with the full details. Let $\Gamma(\alpha, g)$ be a $(n, k_\Gamma)$ Goppa code and let $GRS_{n,k}(\alpha, \beta)$ be the $(n, k)$ GRS code such that $\Gamma(\alpha, g) = GRS_{n,k}(\alpha, \beta) \cap \mathbb{F}_p^n$. It is clear that $\Gamma(\alpha, g) \otimes \mathbb{F}_{p^m} = GRS_{n,k}(\alpha, \beta) \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m} \subseteq GRS_{n,k}(\alpha, \beta)$. In fact, this is an equality if and only if $\dim_{\mathbb{F}_p}(\Gamma(\alpha, g)) = \dim_{\mathbb{F}_{p^m}}(GRS_{n,k}(\alpha, \beta))$ by Lemma 2.1.15, which means $\Gamma(\alpha, g)$ is a full-rank Goppa code; we know how to identify its parameters by methods from Section 6.2, so we will ignore this easy case and try to find a method that works for when the subset inclusion is strict.

Since both $\Gamma(\alpha, g) \otimes \mathbb{F}_{p^m}$ and $GRS_{n,k}(\alpha, \beta)$ are $\mathbb{F}_{p^m}$-linear codes, it is easy to verify that $GRS_{n,k}(\alpha, \beta)^\perp \subset (\Gamma(\alpha, g) \otimes \mathbb{F}_{p^m})^\perp$. The code on the right is one for which we can easily find a generator matrix given the public key (ignoring the permutation matrix in the McEliece scheme) and the code on the left is a GRS subcode of $(\Gamma(\alpha, g) \otimes \mathbb{F}_{p^m})^\perp$ whose parameters can be used to find the Goppa parameters $(\alpha, g)$. Since we can apply the Sidelnikov-Shestakov attack to a GRS code to recover its parameters efficiently, we need to describe a method to systematically identify this GRS subcode (or another one with equivalent parameters) to recover the Goppa code's parameters. This is the perspective our approach will take.

Seeing that the code $(\Gamma(\alpha, g) \otimes \mathbb{F}_{p^m})^\perp$ will be the starting point for our approach, we present some results that reveal it to be related to the GRS subcode we want to identify by means of the Galois closure.

**Proposition 9.2.1.** *If $C$ be a $(n, k)$ $\mathbb{F}_p$-linear code, then $(C \otimes \mathbb{F}_{p^m})^\perp = C^\perp \otimes \mathbb{F}_{p^m}$.*

*Proof.* Certainly, we will have

$$C^\perp \otimes \mathbb{F}_{p^m} = \text{span}_{\mathbb{F}_{p^m}}\{x \in \mathbb{F}_p^n : x^\intercal c = 0 \ \ \forall c \in C\} \subseteq \{x \in \mathbb{F}_{p^m}^n : x^\intercal c = 0 \ \ \forall c \in C\}.$$

It is easy to verify that $\{x \in \mathbb{F}_{p^m}^n : x^\intercal c = 0 \ \ \forall c \in C\} = (C \otimes \mathbb{F}_{p^m})^\perp$ so we must have $C^\perp \otimes \mathbb{F}_{p^m} \subseteq (C \otimes \mathbb{F}_{p^m})^\perp$. Next, by the fact that $\dim_{\mathbb{F}_p}(C) = \dim_{\mathbb{F}_{p^m}}(C \otimes \mathbb{F}_{p^m})$, which appears as Corollary 2.1.14, we will have

$$\dim_{\mathbb{F}_{p^m}}((C \otimes \mathbb{F}_{p^m})^\perp) = n - \dim_{\mathbb{F}_{p^m}}(C \otimes \mathbb{F}_{p^m}) = n - \dim_{\mathbb{F}_p}(C) = n - k$$

and

$$\dim_{\mathbb{F}_{p^m}}(C^\perp \otimes \mathbb{F}_{p^m}) = \dim_{\mathbb{F}_p}(C^\perp) = n - k.$$

Both $(C \otimes \mathbb{F}_{p^m})^\perp$ and $C^\perp \otimes \mathbb{F}_{p^m}$ being of the same dimension then forces them to be the same by the above inclusion. □

This gives us the new formulation $(\Gamma(\alpha, g) \otimes \mathbb{F}_{p^m})^\perp = \Gamma(\alpha, g)^\perp \otimes \mathbb{F}_{p^m}$. We will use Delsarte Duality, which gives the relationship between subfield subcodes and trace codes, and Theorem 2.2.13, which gives the relationship between trace codes and the Galois closure, to classify $\Gamma(\alpha, g)^\perp \otimes \mathbb{F}_{p^m}$ as the Galois closure of $GRS_{n,k}(\alpha, \beta)^\perp$. Applying these theorems, we find

$$\begin{aligned}
\Gamma(\alpha, g)^\perp \otimes \mathbb{F}_{p^m} &= (GRS_{n,k}(\alpha, \beta) \cap \mathbb{F}_p^n)^\perp \otimes \mathbb{F}_{p^m} \\
&= Tr(GRS_{n,k}(\alpha, \beta)^\perp) \otimes \mathbb{F}_{p^m} \text{ by Delsarte Duality} \\
&= (GRS_{n,k}(\alpha, \beta)^\perp)^* \cap \mathbb{F}_p^n \otimes \mathbb{F}_{p^m} \text{ by Theorem 2.2.13} \\
&= (GRS_{n,k}(\alpha, \beta)^\perp)^* \text{ by Theorem 2.1.16.}
\end{aligned}$$

We recall that our perspective is to start from $\Gamma(\alpha, g)^\perp \otimes \mathbb{F}_{p^m} = (GRS_{n,k}(\alpha, \beta)^\perp)^*$ and then identify the GRS subcode $GRS_{n,k}(\alpha, \beta)^\perp$ from it, at which point the recovery of the Goppa code parameters follows immediately from the recovery of the GRS code's parameters. We can now see the correspondence between this task and the more general problem of identifying a linear code that generates a particular Galois closure. This raises some natural questions.

(1) Given a particular code that we know is Galois closure of another, can we count how many codes there are whose Galois closure is precisely this code?
(2) Is there an informative equivalence relation that exists on the set of codes whose Galois closure is some given code?
(3) Is it possible to devise an attack to tease out the underlying code $C$ from its Galois closure $C^*$?

All of these questions remain interesting open problems (as far as I'm aware). In what follows, we will an answer to the second in the context of GRS codes, recall a tool that can be used to give an answer to the third question, and present a modification to the approach we described so far.

We begin our answer to the second by first showing that, in the cases of interest to us, if $C^*$ is the Galois closure of a linear code $C$, there will exist another linear code $D$ such that $D \neq C$ and $D^* = C^*$. If $C$ is fixed under $\phi_n$, then $C = \sum_{i=1}^m \phi_n^i(C) = C^*$, so $C$ is $\text{Gal}(p^m, p)$-invariant. Since we are interested in cases when $C$ is a GRS code whose subfield subcode is not of full rank, which is equivalent to $C$ not being $\text{Gal}(p^m, p)$-invariant, we see that this restricts us to considering codes not fixed under $\phi_n$. Thus, for $\phi_n(C) \neq C$, we will show that both of these codes have the

same Galois closure by showing all codes in the orbit of $C$ under $\langle \phi_n \rangle$ have the same Galois closure.

**Lemma 9.2.2.** *Let $C$ be a $\mathbb{F}_{p^m}$-linear code. For all $j \in \{1, \ldots, m\}$, we have $\phi_n^j(C)^* = C^*$.*

*Proof.* Let $j \in \{1, \ldots, m\}$ be given.

$$\begin{aligned}
\phi_n^j(C)^* &= \sum_{i=1}^{m} \phi_n^i(\phi_n^j(C)) \\
&= \sum_{i=1}^{m} \phi_n^{i+j}(C) \\
&= \sum_{i=1}^{m} \phi_n^i(C) \quad \text{since } \langle \phi_n \rangle \text{ is a cyclic group of order } m \\
&= C^*
\end{aligned}$$

□

While the orbit of $C$ under $\langle \phi_n \rangle$ does not only consist of $C$, in the case that $C$ is a GRS code, the parameters of the codes in this orbit are in fact related to each other through the Frobenius map.

**Proposition 9.2.3.** *Let $GRS_{n,k}(\alpha, \beta)$ be the $(n, k)$ GRS code defined by $\alpha, \beta \in \mathbb{F}_{p^m}^n$ with the usual conditions on $\alpha$ $\beta$. For any $j \in \{1, \ldots, m\}$, we have*

$$\phi_n^j(GRS_{n,k}(\alpha, \beta)) = GRS_{n,k}(\phi_n^j(\alpha), \phi_n^j(\beta)).$$

*Proof.* Let $j \in \{1, \ldots, m\}$ be given and lets consider the code $\phi_n^j(GRS_{n,k}(\alpha, \beta))$.

$$\begin{aligned}
\phi_n^j(GRS_{n,k}(\alpha, \beta)) &= \{\phi_n^j(\beta * ev_\alpha(f)) : f \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})\} \\
&= \left\{ \phi_n^j(\beta) * \left( \left( \sum_{t=1}^{k} f_t \alpha_i^{t-1} \right)^{p^j} \right)_{i=1}^{n} : f_t \in \mathbb{F}_{p^m} \ \ \forall t = 1, \ldots, k \right\}
\end{aligned}$$

Examining this more closely, we have the following for any choice of $i \in \{1, \ldots, n\}$,

$$\begin{aligned}
\left( \sum_{t=1}^{k} f_t \alpha_i^{t-1} \right)^{p^j} &= \sum_{d_1 + \cdots + d_k = p^j} \binom{p^j}{d_1, \ldots, d_k} \prod_{t=1}^{k} (f_t \alpha_i^{t-1})^{d_t} \\
&= \sum_{t=1}^{k} (f_t \alpha_i^{t-1})^{p^j} \quad \text{since } \mathrm{char}(\mathbb{F}_{p^m}) = p \\
&= \sum_{t=1}^{k} \phi^j(f_t) \phi^j(\alpha_i)^{t-1},
\end{aligned}$$

We define the operator $T : \mathbb{P}_{k-1}(\mathbb{F}_{p^m}) \to \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ such that it maps

$$f = \sum_{t=1}^{k} f_t x^{t-1} \ \text{ to } \ T(f) = \sum_{t=1}^{k} \phi^j(f_t) x^{t-1}.$$

The bijectivity $T$ follows immediately from the bijectivity of $\phi$. We may then conclude $T(\mathbb{P}_{k-1}(\mathbb{F}_{p^m})) = \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$ since the former set is clearly included in the latter. Hence, we conclude

$$\phi_n^j(GRS_{n,k}(\alpha,\beta)) = \{\phi_n^j(\beta) * ev_{\phi_n^j(\alpha)}(f) : f \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})\} = GRS_{n,k}(\phi_n^j(\alpha), \phi_n^j(\beta)).$$

□

We can use this proposition to give an answer to our second open question. Let $C^*$ be a $\mathbb{F}_{p^m}$-linear code and let $\mathcal{G}$ denote the set of GRS subcodes of $C^*$ whose Galois closure is $C^*$ itself. Define an equivalence relation $\sim$ on $\mathcal{G}$ by

$$G \sim G' \iff \exists i \in \{1,\dots,m\} \text{ such that } \phi_n^i(G) = G'.$$

By Lemma 9.2.2, all codes in the equivalence class $[G]$ have the same Galois closure, so we observe that $[G] \subseteq \mathcal{G}$. But by Proposition 9.2.3, there is a stronger relationship between all of these codes, which is that the locators of these codes belong in the same orbit under $\langle \phi_n \rangle$ and likewise for the multipliers. This is powerful because it's easy to compute $\phi_n^i(\alpha)$ and $\phi_n^i(\beta)$ for any choice of $i$, so given any one code in the equivalence class $[G]$, it's computationally efficient to identify the rest.

Therefore, if we wish to identify the $\mathbb{F}_{p^m}$-linear code $C$ whose Galois closure is $C^*$, we need to at worst identify the quotient $\mathcal{G}/\sim$. This will at most reduce the number of codes we need to identify by a factor of $m$ over identifying all of $\mathcal{G}$ and solving for $\mathcal{G}/\sim$ instead of $\mathcal{G}$ will be faster unless all codes in $\mathcal{G}$ are $\mathrm{Gal}(p^m, p)$-invariant.

This defines an equivalence relation that answers question (2), but we might wonder if there are other relationships between the parameters of the codes in $\mathcal{G}$ that can be used to define another equivalence relation $R$ on $\mathcal{G}$ such that $\mathcal{G}/R$ is small. Indeed, we might also wonder what are necessary conditions on $\alpha$ and $\beta$ for the set $\mathcal{G}$ of GRS subcodes of $GRS_{n,k}(\alpha,\beta)^*$ whose Galois closure is $GRS_{n,k}(\alpha,\beta)^*$ to simply be the orbit of a single code under $\langle \phi_n \rangle$. If we could devise a way of efficiently finding a single element of $\mathcal{G}$, then subject to these conditions, $\alpha$ and $\beta$ can efficiently be recovered. We also leave these as open questions for future work.

9.3. **Modification to the Approach Based on the Galois Closure.** We present a modification to the key-recovery approach introduced in the last section through the use of puncturing. We discuss initial motivations for this modification and why they ultimately do not agree with our assumption of $GRS_{n,k}(\alpha,\beta)$ being $\mathrm{Gal}(p^m, p)$-invariant. This discussion culminates in describing sufficient conditions for the $\mathrm{Gal}(p^m, p)$-invariance of a GRS code motivated by the observation that a linear code's dimension is invariant under an appropriate puncturing. We finally describe briefly what advantage we can still hope to achieve with this modification.

Returning to the particular code $C^* = (GRS_{n,k}(\alpha,\beta)^\perp)^*$ specified in our setup, we introduce a modification to our approach in the interest of increasing the efficiency at which we can recover the code parameters. The inclusion we are working from is $\Gamma(\alpha, g) \subset GRS_{n,k}(\alpha,\beta)$, where $\Gamma(\alpha, g)$ is a $(n, k_\Gamma)$ Goppa code and $GRS_{n,k}(\alpha,\beta)$ is a $(n,k)$ GRS code. We notice by propositions 9.1.4 and 9.1.5, if we puncture both codes on a set $I$ such that $|I| \leq n-k$, then the inclusion will be preserved (although potentially not strictly), meaning $P_I(\Gamma(\alpha, g)) \subseteq$

$P_I(GRS_{n,k}(\alpha,\beta))$. If we again extend $P_I(\Gamma(\alpha,g))$ by $\mathbb{F}_{p^m}$ and take the dual of both codes, then we get the familiar inclusion

$$P_I(GRS_{n,k}(\alpha,\beta))^\perp \subseteq P_I(\Gamma(\alpha,g))^\perp \otimes \mathbb{F}_{p^m} \text{ by Proposition 9.2.1.}$$

Since $P_I(\Gamma(\alpha,g)) = P_I(GRS_{n,k}(\alpha,\beta) \cap \mathbb{F}_p^n) \subseteq P_I(GRS_{n,k}(\alpha,\beta)) \cap \mathbb{F}_p^{|\overline{I}|}$, by the development following Proposition 9.2.1, we again conclude

$$(P_I(GRS_{n,k}(\alpha,\beta)) \cap \mathbb{F}_p^{|\overline{I}|})^\perp \otimes \mathbb{F}_{p^m} = (P_I(GRS_{n,k}(\alpha,\beta))^\perp)^*.$$

Summarizing the above inclusions, we have

$$P_I(GRS_{n,k}(\alpha,\beta))^\perp \subseteq (P_I(GRS_{n,k}(\alpha,\beta))^\perp)^* \subseteq P_I(\Gamma(\alpha,g))^\perp \otimes \mathbb{F}_{p^m}.$$

While the notation is getting a bit cumbersome, we again have a scenario where we want to find a GRS subcode given some additional knowledge of its Galois closure. However, this time we do not know the Galois closure, but we instead know a code that contains it, this being $P_I(\Gamma(\alpha,g))^\perp \otimes \mathbb{F}_{p^m}$. To clarify, a generator matrix for this code can be obtained from a generator matrix for $\Gamma(\alpha,g)$, $\mathbf{M}$, by puncturing it by $I$ and then finding a basis for the kernel of $P_I(\mathbf{M})^\intercal$, which can be done with polynomial complexity.

Our motivation for puncturing is that while we suppose $GRS_{n,k}(\alpha,\beta)$ is not $\mathrm{Gal}(p^m,p)$-invariant because we can easily recover the code parameters of $\Gamma(\alpha,g)$ if it were, it might be possible for the dual of an appropriate puncturing of $GRS_{n,k}(\alpha,\beta)$ to be $\mathrm{Gal}(p^m,p)$-invariant or for an appropriately punctured closure $(P_I(GRS_{n,k}(\alpha,\beta))^\perp)^*$ to itself be a GRS code. In the former case, both $P_I(GRS_{n,k}(\alpha,\beta))^\perp$ and its Galois closure will be equal and the $\mathrm{Gal}(p^m,p)$-invariance of $P_I(GRS_{n,k}(\alpha,\beta))^\perp$ might imply certain restrictions on $P_I(\Gamma(\alpha,g))^\perp \otimes \mathbb{F}_{p^m}$ such that using it to recover the punctured GRS subcode will be easy. In the latter case, if the Galois closure is a GRS code, it will either be equal to or contain $P_I(GRS_{n,k}(\alpha,\beta))^\perp$ . If it is equal, then this is a reduction to the preceding case; if it is not equal, then we might be able to a filtration-style attack similar to the one introduced by Couvreur et al. in [C2] to recover the parameters of the punctured GRS subcode. Despite these hopes, we will show that the first case will not occur if $GRS_{n,k}(\alpha,\beta)$ is assumed not to be $\mathrm{Gal}(p^m,p)$-invariant and the second will not occur under certain conditions on $\alpha$ and $\beta$. Through this, we will give another sufficient condition for a GRS code being $\mathrm{Gal}(p^m,p)$-invariant.

We will first address the case where $P_I(GRS_{n,k}(\alpha,\beta))^\perp$ is $\mathrm{Gal}(p^m,p)$-invariant with the following lemma.

**Lemma 9.3.1.** *Let $GRS_{n,k}(\alpha,\beta)$ be a $(n,k)$ GRS code defined by vectors $\alpha,\beta \in \mathbb{F}_{p^m}^n$ with the usual restrictions. If $I \subseteq \{1,\ldots,n\}$ is a set of indices such that $|I| \leq n-k$, then we have the following:*

$$P_I(GRS_{n,k}(\alpha,\beta)) = P_I(GRS_{n,k}(\alpha,\beta))^* \iff GRS_{n,k}(\alpha,\beta) \text{ is } Gal(p^m,p)\text{-invariant.}$$

*Proof.* We note that $GRS_{n,k}(\alpha,\beta)$ is $\mathrm{Gal}(p^m,p)$-invariant if and only if this code equals its Galois closure, so we need only show $P_I(GRS_{n,k}(\alpha,\beta)) = P_I(GRS_{n,k}(\alpha,\beta))^*$

if and only if $GRS_{n,k}(\alpha,\beta) = GRS_{n,k}(\alpha,\beta)^*$. This can be done as follows.

$$\begin{aligned}
P_I(GRS_{n,k}(\alpha,\beta))^* &= \sum_{j=1}^{m} \phi^j_{|\overline{I}|}(P_I(GRS_{n,k}(\alpha,\beta))) \\
&= \sum_{j=1}^{m} GRS_{|\overline{I}|,k}(\phi^j_{|\overline{I}|}(\alpha|_{\overline{I}}), \phi^j_{|\overline{I}|}(\beta|_{\overline{I}})) \text{ by propositions 9.1.4 and 9.2.3} \\
&= \sum_{j=1}^{m} GRS_{|\overline{I}|,k}(\phi^j_n(\alpha)|_{\overline{I}}, \phi^j_n(\beta)|_{\overline{I}}) \\
&= \sum_{j=1}^{m} P_I(GRS_{n,k}(\phi^j_n(\alpha), \phi^j_n(\beta))) \text{ by Proposition 9.1.4} \\
&= \left\{ \sum_{j=1}^{m} P_I(\phi^j_n(\beta) * ev_{\phi^j_n(\alpha)}(f_j)) : f_j \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})\, \forall j \right\} \\
&= P_I\left( \left\{ \sum_{j=1}^{m} \phi^j_n(\beta) * ev_{\phi^j_n(\alpha)}(f_j) : f_j \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m})\, \forall j \right\} \right) \text{ by the linearity of } P_I \\
&= P_I(GRS_{n,k}(\alpha,\beta)^*)
\end{aligned}$$

Thus, we have

$$P_I(GRS_{n,k}(\alpha,\beta)) = P_I(GRS_{n,k}(\alpha,\beta))^* \iff P_I(GRS_{n,k}(\alpha,\beta)) = P_I(GRS_{n,k}(\alpha,\beta)^*).$$

Although we only proved that puncturing a code on a set $I$ preserves its dimension in the case of GRS and Goppa codes, this holds for all linear codes given an appropriate choice of $I$ as noted in Ch.1 §9 of [MS]. Thus, we conclude

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

But since $GRS_{n,k}(\alpha,\beta)$ is a subcode of its Galois closure, this implies $GRS_{n,k}(\alpha,\beta) = GRS_{n,k}(\alpha,\beta)^*$. This completes the only if part of the proof. The converse direction follows immediately from the equality $P_I(GRS_{n,k}(\alpha,\beta))^* = P_I(GRS_{n,k}(\alpha,\beta)^*)$. □

It is not difficult to show that if $C$ is a $\mathbb{F}_{p^m}$-linear code, then $C$ is $\mathrm{Gal}(p^m,p)$-invariant if and only if $C^\perp$ is $\mathrm{Gal}(p^m,p)$-invariant. With this, we have that $P_I(GRS_{n,k}(\alpha,\beta))$ is $\mathrm{Gal}(p^m,p)$-invariant if and only if its dual is $\mathrm{Gal}(p^m,p)$-invariant. By applying the lemma to the punctured GRS code, we see that the dual will $\mathrm{Gal}(p^m,p)$-invariant if and only if the unpunctured GRS code is $\mathrm{Gal}(p^m,p)$-invariant. Since we assume the latter condition is not true, we conclude that no matter how we choose $I$ so long as $|I| \leq n-k$, there is no puncturing such that $P_I(GRS_{n,k}(\alpha,\beta))^\perp$ is $\mathrm{Gal}(p^m,p)$-invariant.

Next, we will consider the case when $(P_I(GRS_{n,k}(\alpha,\beta))^\perp)^*$ is a GRS code. This occurs if and only if exist $\alpha',\beta' \in \mathbb{F}_{p^m}^{|\overline{I}|}$ and an integer $k'$ such that the coordinates of $\alpha'$ are distinct, the coordinates of $\beta'$ are non-zero, and $(P_I(GRS_{n,k}(\alpha,\beta))^\perp)^* = GRS_{|\overline{I}|,k'}(\alpha',\beta')$. Since we know both the forms of the dual of a GRS code and of a punctured GRS code, we have that

$$P_I(GRS_{n,k}(\alpha,\beta))^\perp = GRS_{|\overline{I}|,|\overline{I}|-k}(\alpha|_{\overline{I}},\gamma'),$$

where $\gamma'$ is a vector in $\mathbb{F}_{p^m}^{|\overline{I}|}$ that can be found by Proposition 3.1.7. One way to meet this condition is if the orbits of $\alpha|_{\overline{I}}$ and $\gamma'$ under $\langle \phi_{|\overline{I}|} \rangle$ consist of only $\alpha'$ and $\beta'$, respectively. We will show this implies $GRS_{n,k}(\alpha, \beta)$ is $\mathrm{Gal}(p^m, p)$-invariant.

**Lemma 9.3.2.** *Let $GRS_{n,k}(\alpha, \beta)$ be a $(n, k)$ GRS code defined by vectors $\alpha, \beta \in \mathbb{F}_{p^m}^n$ with the usual restrictions. Suppose there are vectors $\alpha', \beta' \in \mathbb{F}_{p^m}^n$ such that $\langle \phi_n \rangle(\alpha) = \{\alpha'\}$ and $\langle \phi_n \rangle(\beta) = \{\beta'\}$. Consequently, we have*

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

*Proof.* We begin by developing the expression for the Galois closure.

$$\sum_{j=1}^{m} \phi_n^j(GRS_{n,k}(\alpha, \beta)) = \left\{ \sum_{j=1}^{m} \phi_n^j(\beta) * ev_{\phi_n^j(\alpha)}(f_j) : f_j \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m}) \, \forall j \right\}$$

By the hypothesis, $\phi_n^j(\alpha) = \alpha'$ and $\phi_n^j(\beta) = \beta'$ for all $j = 1, \ldots, m$, so we can simplify the above expression.

$$GRS_{n,k}(\alpha, \beta)^* = \left\{ \beta' * ev_{\alpha'} \left( \sum_{j=1}^{m} f_j \right) : f_j \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m}) \, \forall j \right\}$$

We will show $F := \left\{ \sum_{j=1}^{m} f_j : f_j \in \mathbb{P}_{k-1}(\mathbb{F}_{p^m}) \, \forall j \right\} = \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$. $F$ is a vector space over $\mathbb{F}_{p^m}$ and it contains the standard basis of $\mathbb{P}_{k-1}(\mathbb{F}_{p^m})$. Thus, $\mathbb{P}_{k-1}(\mathbb{F}_{p^m}) = \mathrm{span}_{\mathbb{F}_{p^m}}\{1, \ldots, x^{k-1}\} \subseteq F$. The other inclusion is obvious, so we get $F = \mathbb{P}_{k-1}(\mathbb{F}_{p^m})$. Using this in the above expression for $GRS_{n,k}(\alpha, \beta)^*$ gives us the following:

$$GRS_{n,k}(\alpha, \beta)^* = \{\beta' * ev_{\alpha'}(f) : f \in F\} = GRS_{n,k}(\alpha', \beta').$$

Since all linear codes are subcodes of their Galois closure, we have that $GRS_{n,k}(\alpha, \beta) \subseteq GRS_{n,k}(\alpha', \beta')$. But because both of these codes are of the same dimension, this must be an equality. □

Applying this lemma to the GRS code $P_I(GRS_{n,k}(\alpha, \beta))^\perp = GRS_{|\overline{I}|,|\overline{I}|-k}(\alpha|_{\overline{I}}, \gamma')$ gives us that $P_I(GRS_{n,k}(\alpha, \beta))^\perp$ is $\mathrm{Gal}(p^m, p)$-invariant if the orbits of its locator and multiplier consist of only a single element each. But the $\mathrm{Gal}(p^m, p)$-invariance of this code is equivalent to the $\mathrm{Gal}(p^m, p)$-invariance of $GRS_{n,k}(\alpha, \beta)$ by Lemma 9.3.1 and the discussion immediately proceeding it. Again, since we assume the latter condition is not true, we can conclude that there is no puncturing by a set of at most $n - k$ indices such that the GRS parameters of the punctured code $P_I(GRS_{n,k}(\alpha, \beta))^\perp$ are fixed under $\langle \phi_n \rangle$. In fact, stating it this way makes the following characterization for this condition more obvious.

**Lemma 9.3.3.** *Let $GRS_{n,k}(\alpha, \beta)$ be a $(n, k)$ GRS code defined by vectors $\alpha, \beta \in \mathbb{F}_{p^m}^n$ with the usual restrictions. The orbits of $\alpha$ and $\beta$ are fixed under $\langle \phi_n \rangle$ if and only if $\alpha, \beta$ belong in $\mathbb{F}_p^n$.*

*Proof.* Let $\alpha', \beta' \in \mathbb{F}_{p^m}^n$ such that $\langle \phi_n \rangle(\alpha) = \{\alpha'\}$ and $\langle \phi_n \rangle(\beta) = \{\beta'\}$. We will complete the proof for the locator as the proof for the multiplier is exactly the same

so long as we replace every instance of $\alpha$ with $\beta$. The condition $\langle\phi_n\rangle(\alpha) = \{\alpha'\}$ holds if and only if

$$\phi^j(\alpha_i) = \alpha'_i \quad \forall i = 1, \ldots, n, \ \ \forall j = 1, \ldots, m.$$

In particular, this implies

$$\alpha_i = \alpha'_i, \ \ \phi(\alpha_i) = \alpha'_i \quad \forall i = 1, \ldots, n.$$

This then lets us conclude $\phi(\alpha'_i) = \alpha'_i$ for all $i = 1, \ldots, n$. But by Lemma 2.1.2, this can happen if and only if $\alpha' \in \mathbb{F}_p^n$. Since we found that $\alpha = \alpha'$, this complete the proof of the forwards direction. The converse direction follows immediately from Lemma 2.1.2, which gives us that the coordinates of $\alpha$ will be fixed under $\langle\phi\rangle$ if $\alpha \in \mathbb{F}_p^n$. □

In combining these last three lemmas, we get another sufficient condition for a GRS code to be $\mathrm{Gal}(p^m, p)$-invariant.

**Theorem 9.3.4.** *Let $GRS_{n,k}(\alpha, \beta)$ be a $(n, k)$ GRS code defined by vectors $\alpha, \beta \in \mathbb{F}_{p^m}^n$ with the usual restrictions. If any $k$ of the same entries of $\alpha$ and $\beta$ belong in $\mathbb{F}_p$, then $GRS_{n,k}(\alpha, \beta)$ is Gal$(p^m, p)$-invariant.*

*Proof.* Let $J \subseteq \{1, \ldots, n\}$ be the set of the indices for which $\alpha_j, \beta_j \in \mathbb{F}_p$ for all $j \in J$. Since $|J| \geq k$, Proposition 9.1.4 gives us $P_J(GRS_{n,k}(\alpha, \beta)) = GRS_{|J|,k}(\alpha|_J, \beta|_J)$, which then lets us apply the last two lemmas to conclude that $P_J(GRS_{n,k}(\alpha, \beta))^* = P_J(GRS_{n,k}(\alpha, \beta))$. Finally, applying Lemma 9.3.1 gives us the result. □

While our initial motivation for considering puncturing in the approach outlined in Section 9.2 does not agree with our assumption that $GRS_{n,k}(\alpha, \beta)$ is not $\mathrm{Gal}(p^m, p)$-invariant, our consideration of puncturing in the approach led us to this last surprising result about when GRS codes are $\mathrm{Gal}(p^m, p)$-invariant. That is to say, while puncturing may not be useful in the context of Section 9.2 as we might have hoped, it gave us a way of saying when a McEliece scheme will be vulnerable to other attacks like the S-S attack detailed in Section 6.2 through this last theorem. Obviously, this poses no threat to McEliece schemes based on binary codes (i.e. where $p = 2$), but this gives some room to worry for schemes based on $q$-ary Goppa codes where $q \geq k$. Whether this is any cause for alarm, meaning when the proportion $\frac{F}{T}$ is sufficiently large where $F$ counts the number of locator-multiplier pairs that possess $k$ of $n$ common entries in $\mathbb{F}_q$ and where $T$ counts the total number of locator-multiplier pairs in $\mathbb{F}_{q^m} \times \mathbb{F}_{q^m}$, will be considered in further work.

There may still be some value in the use of puncturing in the approach from Section 9.2. For instance, there are fewer GRS subcodes of a fixed dimension $k$ of the space $\mathbb{F}_{p^m}^s$ than there are of the space $\mathbb{F}_{p^m}^n$ for $s$ such that $k \leq s < n$ since the number of ways to choose the locator and multiplier as vectors in $\mathbb{F}_{p^m}^s$ is smaller. It may therefore be easier to identify $P_I(GRS_{n,k}(\alpha, \beta))^\perp$ as a subcode of $P_I(\Gamma(\alpha, g))^\perp \otimes \mathbb{F}_{p^m}$ than it would be without puncturing. There is also an advantage to this approach coming from Proposition 8.1.1, which gives us an attack of polynomial complexity that recovers the private key given partial knowledge of the key. If we can develop this approach so that it can recover the punctured locator $\alpha|_{\overline{I}}$ in a way that beats randomly guessing, then this will allow us to begin chipping away at the security of the McEliece cryptosystem. Again, we leave this for future work.

9.4. **Squaring as a GRS Distinguisher.** We lastly note that code squaring (introduced in Section 7.1) can be used to distinguish between GRS codes and random linear codes. This gives us a tool that can be used in the approach of Section 9.2 for identifying GRS subcodes of a given Galois closure, which is a starting point for justifying the feasibility of this approach.

Recall that for a $\mathbb{F}_{p^m}$-linear code $C$, we define its square as

$$C^{(*2)} := \text{span}_{\mathbb{F}_{p^m}}\{c * d : c, d \in C\}.$$

We also recall Proposition 7.1.3, which tells us that the square of a GRS code is again a GRS code and, more more importantly for us now, what its dimension will be. If we consider $GRS_{n,k}(\alpha, \beta)$ where $2k - 1 \leq n$, we have

$$GRS_{n,k}(\alpha, \beta)^{(*2)} = GRS_{n,2k-1}(\alpha, \beta * \beta).$$

Thus, we note that the dimension of a squared GRS code will be nearly twice the dimension of the original code. It's not hard to see that if we had $2k - 1 > n$, then the squared code will be the maximal subspace of $\mathbb{F}_{p^m}^n$, which is $\mathbb{F}_{p^m}^n$ itself. In fact, squaring a $\mathbb{F}_{p^m}$-linear (as long as its dimension isn't too small) will often result in the square being the maximal subspace. GRS codes are unique in that the cutoff on their dimensions needed for this to occur is higher than for an arbitrary linear code. We will begin to see this by considering generic bounds on the dimension of a squared code as presented in [C2]

**Proposition 9.4.1.** *Let $C$ be a $(n, k)$ $\mathbb{F}_{p^m}$-linear code. The dimension of its square is bounded as follows:*

$$\dim_{\mathbb{F}_{p^m}}(C^{(*2)}) \leq min\left\{n, \binom{k+1}{2}\right\}.$$

*Proof.* Let $\{b_1, \ldots, b_k\}$ be a basis for $C$. We therefore have

$$\begin{aligned} C^{(*2)} &= \text{span}_{\mathbb{F}_{p^m}}\{c * d : c, d \in C\} \\ &= \text{span}_{\mathbb{F}_{p^m}}\left\{\sum_{i=1}^{k} \lambda_i b_i * \sum_{j=1}^{k} \gamma_j b_j : \lambda_i, \gamma_j \in \mathbb{F}_{p^m}\ \forall i, j\right\} \\ &= \text{span}_{\mathbb{F}_{p^m}}\left\{\sum_{i=1}^{k}\sum_{j=1}^{k} \lambda_i \gamma_j b_i * b_j : \lambda_i, \gamma_j \in \mathbb{F}_{p^m}\ \forall i, j\right\} \\ &= \text{span}_{\mathbb{F}_{p^m}}\{b_j * b_j : i, j = 1, \ldots, k\}. \end{aligned}$$

However, we know that the component-wise product is symmetric, so we have $\{b_i * b_j : i, j = 1, \ldots, k\} = \{b_i * b_j : i \leq j, j = 1, \ldots, k\}$. Hence, the number of distinct elements in the generating set $\{b_i * b_j : i \leq j, j = 1, \ldots, k\}$ is at most $\binom{k+1}{2}$. If each choice of $j$ and $i \leq j$ yields a distinct element $b_i * b_j$ and this set is linearly independent, then $\binom{k+1}{2}$ is the dimension of the square. Else, the dimension is lower. The upper bound of $n$ on the dimension is trivial since the square is a subspace of $\mathbb{F}_{p^m}^n$. □

It has been shown in [Ca] and [Ra] that almost all linear codes of a given length and dimension reach these bounds. However, as previously discussed, squaring a

GRS code inflates its dimension by less than this. When it does not map a $\mathbb{F}_{p^m}$-linear code to $\mathbb{F}_{p^m}^n$, squaring nearly doubles the dimension of a GRS code and it nearly squares the dimension of an arbitrary linear code. In general, the square of a GRS code is of much lower dimension than we would expect, so this can be used to distinguish a GRS code from an arbitrary linear code.

## References


[B] 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.

[Bi] B. Biswas, "Implementational aspects of code-based cryptography," Cryptography and Security [cs.CR]. Ecole Polytechnique X, 2010. English. pastel-00523007

[BMT] E. Berlekamp, R. .J. McEliece, H. van Tilborg, "On the inherent intractability of certain coding problems," IEEE Transactions on Information Theory, **24**(3):384–386, 1978.

[Ca] I. Cascudo, R. Cramer, D. Mirandola, G. Zémor, "Squares of random linear codes," IEEE Transactions Information Theory, **61**(3): 1159–1173, 2015.

[C1] A. Couvreur, 2019, *Introduction to Coding Theory*, lecture notes, Informatics laboratory of l'École Polytechnique, delivered 6 December 2019.

[C2] A. Couvreur, A. Otmani, J. Tillich, "Polynomial Time Attack on Wild McEliece Over Quadratic Extensions," IEEE Transactions on Information Theory, **63**(1): 404-427, 2017.

[D] P. Delsarte, "On subfield subcodes of modified Reed–Solomon codes," IEEE Transactions on Information Theory, **21**(5): 575-576, 1975.

[GP] M. Giorgetti, A. Previtali, "Galois invariance, trace codes and subfield subcodes", Finite Fields and Their Applications, **16**(2): 96-99, 2010.

[H] W. C. Huffman, V. Pless, *Fundamentals Error-Correcting Codes*, Cambridge University Press, 2003.

[J] E. Jochemsz. "Goppa Codes & the McEliece Cryptosystem," Ph.D Thesis, Vrije Universiteit Amsterdam, 2002.

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

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

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

[P] R. Pellikaan. "Polynomial Codes," in *Codes, Cryptography and Curves with Computer Algebra*, pages 200-242. Cambridge University Press, 2018.

[Ra] H. Randriambololona, "Linear independence of rank 1 matrices and the dimension of products of codes," in Proceedings of IEEE International Symposium on Information Theory (ISIT): 196–200, 2016.

[R] J. Rotman, *Advanced Modern Algebra*, Prentice Hall, 2002.

[Ro] R. Roth. *Introduction to Coding Theory*, Cambridge University Press, 2006.

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

[St] H. Stichtenoth, *Algebraic Function Fields and Codes*, Springer-Verlag, Berlin-Heidelberg, 2009.

[SB] C. Senger, R. Bohara, "A Linear Algebraic Approach to Subfield Subcodes of GRS Codes," Mar. 2018, [online] Available: http://arxiv.org/abs/1803.04028.

[SS] V. M. Sidelnikov, S. O. Shestakov, "On the insecurity of cryptosystems based on generalized Reed-Solomon codes," Discrete Mathematics and its Applications, **2**(4): 439-444, 1992.

[TS] R. C. Torres, N. Sendrier, "Analysis of Information Set Decoding for a Sub-linear Error Weight," PQCrypto 2016, Feb 2016, Fukuoka, Japan. hal-01244886v2

[W] C. Wieschebrink,"Cryptanalysis of the Niederreiter Public Key Scheme Based on GRS Subcodes," in Sendrier, N. (ed.) Post-Quantum Cryptography, PQCrypto 2010. Lecture Notes in Computer Science, **6061**: 61-72. Springer, Berlin-Heidelberg, 2010.

## Appendix

**Modification to the algorithm presented in Proposition 8.1.1.** Suppose column $i$ of $\mathbf{M}^\intercal$ is the zero vector. If the rest of the columns are not the zero vector, then we can solve for them by the algorithm presented in Proposition 8.1.1. Since column $i$ of $\mathbf{M}^\intercal$ is zero, all vectors in any basis of $\mathcal{P}$ are divisible by $x - a_i$. In particular, this means the basis of row polynomials, $\{q_{R_j} : j = 1, \dots, k_\Gamma\}$, must have that

$$q_{R_j}(x) = \left( \prod_{l \in \{1,\dots,k_\Gamma\} \setminus \{i,j\}} (x - a_l) \right) (x - a_i)\rho_j(x) \quad \forall j = 1, \dots, k_\Gamma.$$

Define $\rho_{i,j}(x) := (x - a_i)\rho_j(x)$ for all $j \in \{1, \dots, k_\Gamma\} \setminus \{i\}$. By dividing the non-zero entries $l$ of both $R_j$ and $R_{k_\Gamma}$, we get the familiar equations

$$\frac{(R_j)_l}{(R_{k_\Gamma})_l} = \frac{(a_l - a_{k_\Gamma})\rho_j(a_l)}{(a_l - a_j)\rho_{k_\Gamma}(a_l)} = \frac{\tilde{P}_j(a_l)}{\tilde{Q}_j(a_l)}.$$

We identify $\tilde{Q}_j$ by following the same algorithm as presented in Proposition 8.1.1, which also identifies $\rho_{k_\Gamma}$ through the greatest common divisor of the set $\{\tilde{Q}_j : j = 1, \dots, k_\Gamma - 1\}$. We can then identify $a_i$ from the constant term in the quotient $\frac{\rho_{i,k_\Gamma}(x)}{\rho_{k_\Gamma}(x)}$.

If there were more than one zero column, then the linear factors of the quotient $\frac{\rho_{i,k_\Gamma}(x)}{\rho_{k_\Gamma}(x)}$ identify the values of the missing entries of $a$, but not their order. Recovering these, assigning them an arbitrary order, and then proceeding with the rest of the algorithm finds parameters to a code that's permutation equivalent to the secret Goppa code, which is equivalent to what the Sidelnikov-Shestakov attack finds for GRS codes.