

Contents

1	Generalized Hermitian codes over $GF(2^r)$	2
1.1	Preliminaries on Algebraic-geometric codes	2
1.1.1	Valuation rings	2
1.1.2	Rational function field	3
1.1.3	Divisors	4
1.1.4	Riemann-Roch Theorem	5
1.1.5	Linear codes	5
1.1.6	Algebraic-geometric codes	6
1.2	Preparation material	8
1.3	Structure of a Weierstrass semigroup of the place at infinity .	11
1.4	Application to GH-codes	15
1.5	Duality property	19
1.6	Computational results	21
1.7	Conclusion	22
2	Polynomial-based cryptosystems	23
2.1	Some background on public-key cryptography	23
2.2	Polly Cracker cryptosystem	25
2.2.1	Polly Cracker: initial specialization	25
2.2.2	Gröbner bases point of view	30
2.3	Polly Two cryptosystem	38
2.3.1	Description of Polly Two	39
2.3.2	Potential attacks on Polly Two	41
2.3.3	Attack on Polly Two Challenge 2 and smearing of a plaintext	44
2.4	Noncommutative Polly Cracker cryptosystem	47
2.4.1	Preliminaries on noncommutative Gröbner bases and noncommutative Polly Cracker	47
2.4.2	Chosen-ciphertext attacks	50
2.4.3	Countermeasures	52
	References	53

1 Generalized Hermitian codes over $GF(2^r)$

1.1 Preliminaries on Algebraic-geometric codes

In this section we present basic notions of the theory of algebraic fields and algebraic-geometric codes (AG-codes). We show how the notions of a function field, valuation ring, place, etc. are introduced in this theory. Then we show how the notion of an algebraic-geometric code is introduced. The material of this section is taken from [Sti93]. For coding theory, see [vL92].

1.1.1 Valuation rings

Definition 1.1.1. An *algebraic function field* F/K (or simply, *function field*) of one variable over K is an extension $F \supseteq K$ of K such that F is a finite algebraic extension of $K(x)$, where $x \in F$ is a transcendent element over K .

Definition 1.1.2. A (*discrete*) *valuation ring* of a function field F/K is a ring $O \subseteq F$ with the following properties:

- (1) $K \subsetneq O \subsetneq F$, and
- (2) $\forall z \in F : z \in O$ or $z^{-1} \in O$.

It turns out that any valuation ring is local, moreover its unique maximal ideal is principal.

Definition 1.1.3. A *place* P of F/K is a maximal ideal of some valuation ring O . An element $t \in P : P = tO$ is called *prime element* of P (or local uniformizing parameter).

As the ring O is uniquely determined by its ideal P , we denote $O := O_P$. We also denote $\mathbb{P}_F := \{P \mid P \text{ is a place } F/K\}$.

Definition 1.1.4. A *discrete valuation* of F/K is a function $v : F \rightarrow \mathbb{Z} \cup \{\infty\}$ with the following properties:

- (1) $v(x) = \infty \iff x = 0$;
- (2) $v(xy) = v(x) + v(y) \forall x, y \in F$;
- (3) $v(x + y) \geq \min\{v(x), v(y)\} \forall x, y \in F$;
- (4) $\exists z \in F : v(z) = 1$;
- (5) $v(a) = 0 \forall a \in K, a \neq 0$.

Definition 1.1.5. We associate every place $P \in \mathbb{P}_F$ with the function $v_P : F \rightarrow \mathbb{Z} \cup \{\infty\}$, which is defined as $v_P(z) := n$, where $z = t^n u, \forall z \in F, z \neq 0$, t is a prime element of P , $u \in O_P^*$; $v_P(0) := \infty$. Note that v_P is a discrete valuation.

Theorem 1.1.6. Let F/K be a function field.

(a) $\forall P \in \mathbb{P}_F$:

$$O_P = \{z \in F \mid v_P(z) \geq 0\}, O_P^* = \{z \in F \mid v_P(z) = 0\}, P = \{z \in F \mid v_P(z) > 0\},$$

$x \in F$ is prime for $P \iff v_P(x) = 1$.

(b) Let v be a discrete valuation of $F/K \Rightarrow P := \{z \in F \mid v(z) > 0\}$ is a place of F/K , $O_P = \{z \in F \mid v(z) \geq 0\}$ is a corresponding valuation.

(c) Every valuation ring of F/K is a maximal proper subring of F .

Definition 1.1.7. $P \in \mathbb{P}_F$.

(a) $F_P := O_P/P$ is a residue class field P . A map $x \rightarrow x(P)$ from F to $F_P \cup \{\infty\}$ is called a residue class map w.r.t P . We use notation $x(P) := x + P, x \in O_P$.

(b) $\deg P := [F_P : K]$ is a degree of P (it is well-defined as we have an inclusion $K \hookrightarrow F_P$).

Definition 1.1.8. Let $z \in F, P \in \mathbb{P}_F$. We say that P is a zero of $z \iff v_P(z) > 0$; P is a pole of $z \iff v_P(z) < 0$. If $v_P(z) = m > 0$, then P is a zero z of order m ; if $v_P(z) = -m < 0$, then P is a pole z of order m .

Theorem 1.1.9. Every element $z \in F, z \neq 0$ has finitely many zeroes and poles.

1.1.2 Rational function field

Definition 1.1.10. A function field F/K is called *rational*, if $F = K(x)$.

Next, we describe all valuation rings of $K(x)/K$ and their places.

Let $p(x) \in K[x]$ be a monic irreducible polynomial. Then we have a valuation ring

$$O_{p(x)} := \left\{ \frac{f(x)}{g(x)} \mid f(x), g(x) \in K[x], p(x) \nmid g(x) \right\}$$

and its place is

$$P_{p(x)} = \left\{ \frac{f(x)}{g(x)} \mid f(x), g(x) \in K[x], p(x) \mid f(x), p(x) \nmid g(x) \right\}$$

In the particular case $p(x) = x - \alpha, \alpha \in K$ we denote $P_\alpha := P_{x-\alpha} \in \mathbb{P}_{K(x)}$.

Another valuation ring is

$$O_\infty := \left\{ \frac{f(x)}{g(x)} \mid f(x), g(x) \in K[x], \deg(f(x)) \leq \deg(g(x)) \right\}$$

with the place

$$P_\infty = \left\{ \frac{f(x)}{g(x)} \mid f(x), g(x) \in K[x], \deg(f(x)) < \deg(g(x)) \right\}$$

1.1.3 Divisors

Definition 1.1.11. A *divisor group* of F/K , denoted as \mathcal{D}_F , is a free abelian group generated by the places of F/K . Elements of this group are *divisors*, i.e. $D = \sum_{P \in \mathbb{P}_F} n_P P$, where almost all $n_P = 0$.

We next give some more definitions.

$$\text{supp}(D) := \{P \in \mathbb{P}_F | n_P \neq 0\}.$$

Addition is componentwise, namely if $D = \sum_{P \in \mathbb{P}_F} n_P P, D' = \sum_{P \in \mathbb{P}_F} n'_P P$, then $D + D' = \sum_{P \in \mathbb{P}_F} (n_P + n'_P) P$.

Then, $D = 0 : \iff n_P = 0 \forall P \in \mathbb{P}_F$. For $Q \in \mathbb{P}_F$ and $D = \sum n_P P \in \mathcal{D}_F$ we denote $v_Q(D) := n_Q$.

A partial ordering on \mathcal{D}_F is defined as $D_1 \leq D_2 \iff v_P(D_1) \leq v_P(D_2) \forall P \in \mathbb{P}_F$. A divisor $D \geq 0$ is called *effective*. The degree of a divisor is a number

$$\text{deg}(D) := \sum_{P \in \mathbb{P}_F} v_P(D) \text{deg}(P).$$

We have a homomorphism $\text{deg} : \mathcal{D}_F \rightarrow \mathbb{Z}$.

Definition 1.1.12. $(x)_0 := \sum_{P \in Z} v_P(x) P$ is a *zero divisor* of x , where $Z \subset \mathbb{P}_F$ is a set of zeroes of x .

$(x)_\infty := \sum_{P \in N} v_P(x) P$ is a *pole divisor* of x , where $N \subset \mathbb{P}_F$ is a set of poles of x .

$(x) = (x)_0 - (x)_\infty$ is a *principal divisor* of x .

Definition 1.1.13. $\mathcal{P}_F := \{(x) | 0 \neq x \in F\}$ is a *group of principal divisors* of F/K .

$\mathcal{C}_F := \mathcal{D}_F / \mathcal{P}_F$ is a *divisor class group*.

With the above definition we have a natural notion of an equivalence of divisors.

Definition 1.1.14. Define a set

$$\mathcal{L}(A) := \{x \in F | (x) \geq -A\} \cup \{0\}, A \in \mathcal{D}_F.$$

In fact $\mathcal{L}(A)$ is a vector space over K . Moreover the corresponding vector spaces of equivalent divisors are isomorphic.

1.1.4 Riemann-Roch Theorem

Theorem 1.1.15 (Riemann-Roch). *Let W be a canonical divisor of F/K (cf. [Sti93], I.5). Then $\forall A \in \mathcal{D}_F$:*

$$\dim A = \deg A + 1 - g + \dim(W - A). \quad (1)$$

Here $\dim A := \dim_K \mathcal{L}(A)$, $A \in \mathcal{D}_F$. Then g is a *genus* of F/K , which is defined as $g := \max\{\deg A - \dim A + 1 \mid A \in \mathcal{D}_F\}$.

Corollary 1.1.16. *If $A \in \mathcal{D}_F$: $\deg A \geq 2g - 1$, then*

$$\dim A = \deg A + 1 - g.$$

1.1.5 Linear codes

The codes we will be working with are algebraic-geometric codes. They belong to the class of *linear codes*. Here we give brief introduction to the theory of linear codes. Error-correcting codes (in particular, linear codes) are used for correcting errors after transmission of information via a noisy channel. For more information on that, cf. [vL92].

Let \mathbb{F}_q be a finite field with q elements.

Definition 1.1.17. A *linear code* of dimension k is a subspace $C \leq \mathbb{F}_q^n$ such that $\dim_{\mathbb{F}_q} C = k$.

Notation for such a code is $[n, k]$ -code. It is very convenient to describe linear codes via matrices.

Definition 1.1.18. A *generator matrix* G for a code C is a $k \times n$ matrix, which rows are basis vectors of the code C .

Definition 1.1.19. A code C^\perp is called *dual* to C , if

$$C^\perp = \{y \in (\mathbb{F}_q)^n \mid \forall x \in C : \langle x, y \rangle = 0\},$$

where $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$, $x = (x_1, \dots, x_n)$; $y = (y_1, \dots, y_n)$.

The following property holds:

$$x \in C \iff Hx^T = 0,$$

where H is a generator matrix for C^\perp .

Definition 1.1.20. A generator matrix H for C^\perp is called *parity check matrix* for C .

When studying codes the so-called *Hamming distance* plays crucial role.

Definition 1.1.21. A *distance* $d(x, y)$ between x and y from \mathbb{F}_q^n is a number

$$d(x, y) := |\{i | 1 \leq i \leq n, x_i \neq y_i\}|.$$

A *weight* of $x \in \mathbb{F}_q^n$ is a number

$$w(x) := d(x, 0).$$

It can easily be seen that d is a metric on \mathbb{F}_q^n .

Definition 1.1.22. A *minimal distance* $d(C)$ of the code C is a number

$$d(C) := \min\{d(x, y) | x, y \in C, x \neq y\}$$

A *minimal weight* of the code is:

$$w(C) := \min\{w(x) | x \in C, x \neq 0\}$$

The notation for an $[n, k]$ -code with the distance d is $[n, k, d]$ -code.

For linear codes the following statement holds.

Proposition 1.1.23. *For the linear code C :*

$$d(C) = w(C).$$

Proof. $d(x, y) = d(x - y, 0) = w(x - y)$. If $x \in C, y \in C$, then $x - y \in C$. \square

Next, the following holds

Proposition 1.1.24 (Singleton bound). *For the $[n, k, d]$ -code C holds*

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

Codes, which parameters satisfy $d = n - k + 1$ are called *MDS*-codes.

1.1.6 Algebraic-geometric codes

Here we present only very basics of a fascinating theory of algebraic-geometric codes initially introduced by V.D.Goppa. For a more thorough treatment of the subject we refer again to [Sti93]. We use notions and notation of the preceding sections.

Definition 1.1.25. An *algebraic-geometric code* (AG-code, geometric Goppa code) $C_{\mathcal{L}}(D, G)$ associated with the divisors D and G is the following set

$$C_{\mathcal{L}}(D, G) := \{(x(P_1), \dots, x(P_n)) \mid x \in \mathcal{L}(G)\} \subset \mathbb{F}_q^n.$$

Here $D = P_1 + \dots + P_n$ and P_1, \dots, P_n are pairwise distinct places of F/\mathbb{F}_q of degree 1. Also $\text{supp}(D) \cap \text{supp}(G) = \emptyset$.

We note that the notion above is well-defined: for $x \in \mathcal{L}(G) : v_{P_i}(x) \geq 0$ ($i = 1, \dots, n$), as $\text{supp}(D) \cap \text{supp}(G) = \emptyset$. The residue class $x(P_i)$ for $x \bmod P_i$ is an element of F_{P_i} . As $\deg(P_i) = 1 \Rightarrow [F_{P_i} : \mathbb{F}_q] = 1 \Rightarrow F_{P_i} = \mathbb{F}_q \Rightarrow x(P_i) \in \mathbb{F}_q$.

Consider a map $ev_D : \mathcal{L}(G) \rightarrow (\mathbb{F}_q)^n$ defined as

$$ev_D(x) := (x(P_1), \dots, x(P_n)) \in \mathbb{F}_q^n. \quad (2)$$

This map is \mathbb{F}_q -linear and $C_{\mathcal{L}}(D, G) = ev_D(\mathcal{L}(G))$.

Using (2) we can define some well-known codes. Let us construct Reed-Solomon code (*RS-code*, cf. [vL92], [Sti93]).

Let $n = q - 1$ and $\beta \in \mathbb{F}_q$ is a generating element of a multiplicative group $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$, i.e $\mathbb{F}_q^* = \{\beta, \beta^2, \dots, \beta^n = 1\}$. For all integer $k, 1 \leq k \leq n$ we define a vector space

$$\mathcal{L}_k := \{f \in \mathbb{F}_q[x] \mid \deg(f) \leq k - 1\}$$

and define the map $ev : \mathcal{L}_k \rightarrow (\mathbb{F}_q)^n$ as

$$ev(f) := (f(\beta), f(\beta^2), \dots, f(\beta^n)) \in \mathbb{F}_q^n. \quad (3)$$

This map is \mathbb{F}_q -linear and injective, as a polynomial of degree $k - 1 < n$ has less than n roots. Therefore

$$C_k := \{(f(\beta), f(\beta^2), \dots, f(\beta^n)) \mid f \in \mathcal{L}_k\}$$

is an $[n, k]$ -code over \mathbb{F}_q , which is called *RS-code*. Next $\forall 0 \neq c = ev(f) \in C_k$:

$$w(c) = n - |\{i \in \{1, \dots, n\} \mid f(\beta^i) = 0\}| \geq n - \deg(f) \geq n - (k - 1).$$

So, we have $d(C_k) \geq n + 1 - k$. On the other hand the Singleton bound is $d \leq n + 1 - k \Rightarrow d = n - k + 1$. Therefore *RS-code* is an instance of an *MDS-code*. We note similarity of (2) and (3). In fact, we can choose divisors G and D of F/\mathbb{F}_q in a proper way to obtain an *RS-code* as an *AG-code*.

Now we will state without proofs several results on *AG-codes* that will give us an opportunity to estimate their parameters.

Theorem 1.1.26. $C_{\mathcal{L}}(D, G)$ is an $[n, k, d]$ -code with the parameters

$$k = \dim(G) - \dim(G - D), d \geq n - \deg(G).$$

Corollary 1.1.27. Let $\deg(G) < n$. Then the map $ev_D : \mathcal{L}(G) \rightarrow C_{\mathcal{L}}(D, G)$ is surjective, moreover we have:

(a) $C_{\mathcal{L}}(D, G)$ is an $[n, k, d]$ -code with

$$k = \dim(G) \geq \deg(G) + 1 - g, d \geq n - \deg(G).$$

where g is the genus of F/\mathbb{F}_q .

(b) Moreover if $2g - 2 < \deg(G) < n$, then $k = \deg(G) + 1 - g$.

(c) If $\{x_1, \dots, x_k\}$ is a basis of $\mathcal{L}(G)$ then the matrix

$$M = \begin{pmatrix} x_1(P_1) & x_1(P_2) & \dots & x_1(P_n) \\ \vdots & \vdots & \ddots & \vdots \\ x_k(P_1) & x_k(P_2) & \dots & x_k(P_n) \end{pmatrix}$$

is a generator matrix for the code $C_{\mathcal{L}}(D, G)$.

Note that if $g = 0$ (this is true for instance for $\mathbb{F}_q(z)/\mathbb{F}_q$), then (a) yields $k + d \geq n + 1 - g \Rightarrow k + d \geq n + 1$. On the other hand $d \leq n - k + 1$ (Singleton bound). Therefore $C_{\mathcal{L}}(D, G)$ in this case is an *MDS*-code.

1.2 Preparation material

The main focus of the present research is on construction of codes on function fields, which we called Generalized Hermitian function fields (GH-fields). The term follows from the fact that well-known Hermitian function fields are a special case of the considered family. We first will introduce some preparation material (notions, theorems etc.). The next subsection 1.3 is a technical core of this work, which will give us an opportunity to calculate or at least estimate parameters of codes, constructed on GH-fields (Generalized Hermitian codes; GH-codes) in subsection 1.4. These codes turn out to have nice properties similar to those of Hermitian codes, but over larger alphabet. In fact some of these codes over \mathbb{F}_8 attain record values for given parameters; one code delivers a new record. Also their generator matrices can be effectively constructed. Subsection 1.5 is devoted to investigating a duality property of GH-codes. It turns out that the duality property of GH-codes is analogous to that of Hermitian codes. Subsection 1.6 gives some

specific computational results. We finish with conclusions in section 1.7.

So, first of all recall that Hermitian function fields are from the family of elementary abelian p -extensions of $K(x)$, where $\text{char}K = p > 0$. The main properties of these function fields that are of importance for coding theory are collected in the following (Lemma VI.4.4, [Sti93]):

Proposition 1.2.1. *The Hermitian function field over \mathbb{F}_{q^2} , q is a prime power, can be defined by*

$$H = \mathbb{F}_{q^2}(x, y) \text{ with } y^q + y = x^{q+1}. \quad (4)$$

It has the following properties:

- (a) *The genus of H is $g = q(q-1)/2$.*
- (b) *H has $q^3 + 1$ places of degree one over \mathbb{F}_{q^2} , namely*
 - (1) *the common pole Q_∞ of x and y , and*
 - (2) *for each $\alpha \in \mathbb{F}_{q^2}$, there are q elements $\beta \in \mathbb{F}_{q^2}$ such that $\beta^q + \beta = \alpha^{q+1}$, and for all such pair (α, β) there is a unique place $P_{\alpha, \beta} \in \mathbb{P}_H$ of degree one with $x(P_{\alpha, \beta}) = \alpha$ and $y(P_{\alpha, \beta}) = \beta$.*
- (c) *H/\mathbb{F}_{q^2} is a maximal function field.*
- (d) *For $r \geq 0$, the elements $x^i y^j$ with $0 \leq i, 0 \leq j \leq q-1$ and $iq + j(q+1) \leq r$ form a basis of $\mathcal{L}(rQ_\infty)$.*

Now we present a family of function fields, which we call *GH-fields*. The following theorem is from [AH99].

Theorem 1.2.2. *Let $r \geq 2$. Then the curve*

$$y^{q^{r-1}} + \dots + y^q + y = x^{1+q} + x^{1+q^2} + \dots + x^{q^{r-2}+q^{r-1}} \quad (5)$$

over \mathbb{F}_{q^r} is absolutely irreducible. The corresponding function field F/\mathbb{F}_{q^r} of this curve has genus

$$g = q^{r-1}(q^{r-1} - 1)/2,$$

and the number of rational places is

$$N = 1 + q^{2r-1}.$$

Remark 1.2.3. (i) Note that we can write equation (5) as $s_{r,1}(y) = s_{r,2}(x)$, where $s_{r,1}(y)$ and $s_{r,2}(x)$ are the first and the second symmetric polynomials of $(y, y^q, \dots, y^{q^{r-1}})$ and $(x, x^q, \dots, x^{q^{r-1}})$ respectively.

(ii) For $r = 2$ this curve is the Hermitian curve over \mathbb{F}_{q^2} , and the result is well known (Proposition 1.2.1).

Our aim is to present an analogue of Proposition 1.2.1 for GH-fields from Theorem 1.2.2.

Basically, (a) from Proposition 1.2.1 has its analogue in Theorem 1.2.2, (c) does not hold for GH-curves starting with $r \geq 3$. So, we have to find the analogue for (b) and (d). The answer to (d) is very important for construction of GH-codes and is given in Theorem 1.3.4 from subsection 1.3, but in order to justify the proof of this theorem we need some preparation.

Proposition 1.2.4. *Let F/\mathbb{F}_{q^r} be a function field of the curve defined by (5). Then the following holds:*

(a) *The pole $P_\infty \in \mathbb{P}_{\mathbb{F}_{q^r}(x)}$ (by \mathbb{P}_F we denote the set of places of a function field F) of x in $\mathbb{F}_{q^r}(x)$ has a unique extension $Q_\infty \in \mathbb{P}_F$, and $Q_\infty|P_\infty$ is totally ramified (i.e. ramification index $e(Q_\infty|P_\infty) = q^{r-1}$). Hence Q_∞ is a place of F/\mathbb{F}_{q^r} of degree one.*

(b) *The pole divisor of x is $(x)_\infty = q^{r-1}Q_\infty$, and of y is $(y)_\infty = (q^{r-1} + q^{r-2})Q_\infty$.*

(c) *For each $\alpha \in \mathbb{F}_{q^r}$, there are q^{r-1} elements $\beta \in \mathbb{F}_{q^r}$ such that $\beta^{q^{r-1}} + \dots + \beta = \alpha^{q^{r-1}+q^{r-2}} + \dots + \alpha^{1+q} =: f(\alpha)$, and for all such pairs (α, β) there is a unique place $P_{\alpha,\beta} \in \mathbb{P}_F$ of degree one with $x(P_{\alpha,\beta}) = \alpha$ and $y(P_{\alpha,\beta}) = \beta$.*

Proof. (a) It follows from the proof of Theorem 4.1, [AH99] (it is Theorem 1.2.2 in our text).

(b) $(x)_\infty = q^{r-1}Q_\infty$ follows from (a) (cf. Theorem I.4.11, [Sti93]). By (5) x and y have the same poles, hence Q_∞ is the only pole of y as well. As $q^{r-1}v_{Q_\infty}(y) = v_{Q_\infty}(y^{q^{r-1}} + \dots + y) = v_{Q_\infty}(x^{q^{r-1}+q^{r-2}} + \dots + x^{1+q}) = q^{r-1}(q^{r-1} + q^{r-2})$ we obtain $(y)_\infty = (q^{r-1} + q^{r-2})Q_\infty$.

(c) For the first part of the statement see the proof of Theorem 4.1. from [AH99]. Now, suppose there is some $\beta \in \mathbb{F}_{q^r}$ such that $\beta^{q^{r-1}} + \dots + \beta = f(\alpha)$. It follows that $(\beta+\gamma)^{q^{r-1}} + \dots + (\beta+\gamma) = f(\alpha)$ for all γ with $\gamma^{q^{r-1}} + \dots + \gamma = 0$, so

$$T^{q^{r-1}} + \dots + T - f(\alpha) = \prod_{j=1}^{q^{r-1}} (T - \beta_j)$$

with pairwise distinct elements $\beta_i \in \mathbb{F}_{q^r}$. By Corollary III.3.8(c) from [Sti93], there exists for $j = 1, \dots, q^{r-1}$ a unique place $P_j \in \mathbb{P}_F$ such that $P_j|P_\alpha$ and $y - \beta_j \in P_j$, and the degree of P_j is one, which means $x(P_j) = \alpha, y(P_j) = \beta_j$. \square

In the proof we have used some methods from the proof of Proposition VI.4.1 ([Sti93]). Now using this proposition we go on to the question of the structure of vector spaces $\mathcal{L}(sQ_\infty)$ for given s .

1.3 Structure of a Weierstrass semigroup of the place at infinity

Our aim now is to determine the basis of $\mathcal{L}(sQ_\infty)$ for given s . This problem is closely connected with finding a Weierstrass semigroup of Q_∞ up to a given parameter s .

In the case of Hermitian curves the situation is quite simple as Weierstrass semigroup is generated by the orders at infinity of x and y . This gives rise to the fact that $\mathcal{L}(sP_\infty)$ is generated by functions of the form $x^i y^j$, where $qi + (q + 1)j \leq s$, where $q, (q + 1)$ are orders of x and y respectively. Our situation is more complicated. We will restrict ourselves to the case $q = 2$. So we are considering a curve

$$y^{2^{r-1}} + \dots + y^2 + y = x^{2^{r-1}+2^{r-2}} + \dots + x^3 \quad (6)$$

over $\mathbb{F}_{2^r}, r \geq 3$.

Remark 1.3.1. Our case does not include all curves from the considered family over the field of characteristics 2. Indeed, we can consider $\mathbb{F}_{2^{rk}}$ as $\mathbb{F}_{(2^r)^k}$, so the equation will be

$$y^{(2^r)^{k-1}} + \dots + y^{2^r} + y = x^{(2^r)^{k-1}+(2^r)^{k-2}} + \dots + x^{1+2^r}$$

as oppose to

$$y^{2^{rk-1}} + \dots + y^2 + y = x^{2^{rk-1}+2^{rk-2}} + \dots + x^3$$

in the case $q = 2$ (constant field is $\mathbb{F}_{2^{rk}}$).

If we consider all q and r such that q^r is fixed and q is a power of 2, then the maximal number of rational points will be when $q = 2$. This follows from the fact that $N = 1 + q^{2r-1} = 1 + \frac{(q^r)^2}{q}$. So as q grows N decreases. The same argument, of course, works for arbitrary characteristics.

However, the ratio

$$\frac{N}{g} = \frac{2(1 + q^{2r-1})}{q^{r-1}(q^{r-1} - 1)} = \frac{2(1 + \frac{(q^r)^2}{q})}{\frac{q^r}{q}(q^r - 1)}$$

is the lowest when $q = 2$, and grows as q goes up. That is why studying the curves over \mathbb{F}_{q^r} , where q is a power of a prime is of interest.

As a prelude to finding the Weierstrass semigroup of Q_∞ and corresponding basis of $\mathcal{L}(sQ_\infty)$, let us first find orders of some functions at infinity (i.e. at Q_∞). The following lemma will give us an opportunity to find numbers that generate the whole Weierstrass semigroup of Q_∞ .

Lemma 1.3.2. *If we denote $ord(f) := -v_{Q_\infty}(f)$, $\epsilon := x^3 + y^2$, $\theta := \epsilon + xy$, then the following hold:*

- $ord(x) = 2^{r-1}$;
- $ord(y) = 2^{r-1} + 2^{r-2}$;
- $ord(\epsilon) = ord(xy)$;
- $ord(\theta) = 2^r + 1$.

Proof. As was noted before, we have that x and y have orders at infinity respectively $ord(x) = -v_{Q_\infty}(x) = 2^{r-1}$ and $ord(y) = -v_{Q_\infty}(y) = 2^{r-1} + 2^{r-2}$. When $r \geq 3$ we have that $ord(x)$ and $ord(y)$ are both even, so we cannot hope on them to generate the Weierstrass semigroup. So we have to search for other generator(s).

As we see $ord(x^3) = ord(y^2) = 2^r + 2^{r-1} = 3 \cdot 2^{r-1}$. If we consider their sum $x^3 + y^2$ we may hope that $ord(x^3 + y^2)$ will be lower than $ord(x^3) = ord(y^2)$ and differ from those orders that could be generated by $ord(x)$ and $ord(y)$. So let us put $\epsilon := x^3 + y^2$. By squaring both sides of (6) we have

$$y^{2^r} + \dots + y^4 + y^2 = x^{2^r+2^{r-1}} + \dots + x^6. \quad (7)$$

Now, considering that $\alpha = -\alpha$ in a field of characteristics 2 we have

$$\begin{aligned} \epsilon^{2^{r-1}} &= x^{3 \cdot 2^{r-1}} + y^{2^r} = x^{2^r+2^{r-1}} + y^{2^r} = \text{using (5)} \\ &= y^{2^{r-1}} + \dots + y^4 + y^2 + x^{2^r+2^{r-2}} + \dots + x^6. \end{aligned}$$

We then use the fact that

$$y^{2^{r-1}} + \dots + y^4 + y^2 = y + x^{2^{r-1}+2^{r-2}} + \dots + x^3.$$

So,

$$\epsilon^{2^{r-1}} = y + (x^{2^{r-1}+2^{r-2}} + \dots + x^3) + (x^{2^r+2^{r-2}} + \dots + x^6).$$

All summands of the form $x^{2^i+2^j}$, $i, j > 0$ from the first bracket will be canceled, as $x^{2^i+2^j} = (x^{2^{i-1}+2^{j-1}})^2$. So the first bracket reduces to $x^{2^{r-1}+1} + x^{2^{r-2}+1} + \dots + x^3$. Analogously, in the second bracket all summands of the form $x^{2^i+2^j}$; $0 \leq j < i \leq r-1$ will be crossed out (as they are present in the first bracket), so the second bracket reduces to $x^{2^r+2^{r-2}} + x^{2^r+2^{r-3}} + \dots + x^{2^r+1}$. After reducing we have

$$\epsilon^{2^{r-1}} = y + x^{2^r+2^{r-2}} + \dots + x^{2^r+2} + x^{2^{r-1}+1} + x^{2^{r-2}+1} + \dots + x^3.$$

Orders (at Q_∞) of all summands here are pairwise distinct, so Strict Triangle Inequality works. Thus, $2^{r-1} \cdot ord(\epsilon) = (2^r + 2^{r-2})ord(x)$ (which is the highest). So

$$ord(\epsilon) = 2^r + 2^{r-2}.$$

Now note that $ord(xy) = ord(x) + ord(y) = 2^{r-1} + 2^{r-1} + 2^{r-2} = 2^r + 2^{r-2} = ord(\epsilon)$.

Remark 1.3.3. $ord(xy) \neq ord(\epsilon)$ if $q \neq 2$, so our considerations are essentially valid only for $q = 2$.

Let us do the summing up again. Consider $\theta := \epsilon + xy = x^3 + y^2 + xy$.

$$\begin{aligned} \theta^{2^{r-1}} &= \epsilon^{2^{r-1}} + x^{2^{r-1}}y^{2^{r-1}} = |using (5)| = \\ &= \epsilon^{2^{r-1}} + x^{2^{r-1}}(y^{2^{r-2}} + \\ &\quad + \dots + y^2 + y + x^{2^{r-1}+2^{r-2}} + \dots + x^3) = \\ &= y + x^{2^r+2^{r-2}} + \dots + x^{2^r+2} + x^{2^{r-1}+1} + x^{2^{r-2}+1} \\ &+ \dots + x^3 + x^{2^{r-1}}y^{2^{r-2}} + \dots + x^{2^{r-1}}y^2 + x^{2^{r-1}}y + \\ &\quad + x^{2^r+2^{r-2}} + \dots + x^{2^r+2} + x^{2^r+1} + \\ &\quad + x^{2^{r-1}+2^{r-2}+2^{r-3}} + \dots + x^{2^{r-1}+3} = \\ &= y + x^{2^{r-1}+1} + x^{2^{r-2}+1} + \dots + x^3 + x^{2^{r-1}}y^{2^{r-2}} + \\ &\quad + \dots + x^{2^{r-1}}y^2 + x^{2^{r-1}}y + x^{2^r+1} + \\ &\quad + x^{2^{r-1}+2^{r-2}+2^{r-3}} + \dots + x^{2^{r-1}+3}. \end{aligned}$$

For Strict Triangle Inequality to work we need that the highest orders are not duplicated (if some lower orders are duplicated, we can sum corresponding functions and a resulting function will have either the same order, i.e. duplication is removed, or the lower order, so we can repeat our procedure, and so on). In order to find the highest order among orders of our summands we need to compare the orders of x^{2^r+1} and $x^{2^{r-1}}y^{2^{r-2}}$: $ord(x^{2^r+1}) = (2^r + 1)2^{r-1}$; $ord(x^{2^{r-1}}y^{2^{r-2}}) = 2^{r-1} \cdot 2^{r-1} + 2^{r-2}(2^{r-1} + 2^{r-2}) = 2^{r-2}(2^r + 2^{r-2})$. Now $2^{r-1}(2^r + 1) = 2^{r-2}(2^{r+1} + 2) = 2^{r-2}(2^r + 2^{r-1} + 2^{r-1} + 2) > 2^{r-2}(2^r + 2^{r-2})$. So $ord(\theta^{2^{r-1}}) = (2^r + 1) \cdot 2^{r-1} \Rightarrow ord(\theta) = 2^r + 1$. \square

The question of calculating of orders of functions was also studied in [AH91]. Obviously, $ord(\theta) = 2^r + 1$ is not a linear combination of 2^{r-1} and $2^{r-1} + 2^{r-2}$. It turns out that this is all what we need in order to construct the Weierstrass semigroup of Q_∞ (we denote it as $WS(Q_\infty)$). Namely, the following holds

Theorem 1.3.4. $WS(Q_\infty) = \mathbb{N} \cdot 2^{r-1} + \mathbb{N} \cdot (2^{r-1} + 2^{r-2}) + \mathbb{N} \cdot (2^r + 1)$.

This theorem can be proven via direct computations (cf. [S.B04]). But we will use so-called telescopic semigroups, which will yield a short and elegant proof of the theorem. First, let us define what a telescopic semigroup is and give a result that we will use in the proof.

Definition 1.3.5. (Definition 5.31, [TvLR98]) Let (a_1, \dots, a_k) be a sequence of positive integers with greatest common divisor 1. Define

$$d_i = \gcd(a_1, \dots, a_i) \text{ and } A_i = \{a_1/d_i, \dots, a_i/d_i\}$$

for $i = 1, \dots, k$. Let $d_0 = 0$. Let Λ_i be the semigroup generated by A_i . If $a_i/d_i \in \Lambda_{i-1}$ for $i = 2, \dots, k$, then the sequence (a_1, \dots, a_k) is called *telescopic*. A semigroup is called telescopic if it is generated by a telescopic sequence.

Definition 1.3.6. (Section 5.1, [TvLR98]) Let Λ be a semigroup. The number of gaps is denoted by $g = g(\Lambda)$. If $g < \infty$, then there exists an $n \in \Lambda$ such that if $x \in \mathbb{N}_0$ and $x \geq n$, then $x \in \Lambda$. The conductor of Λ is the smallest $n \in \Lambda$ such that $\{x \in \mathbb{N}_0 | x \geq n\}$ is contained in Λ , denoted by $c = c(\Lambda)$. So $c - 1$ is the largest gap of Λ if $g > 0$. A semigroup is called *symmetric* if $c = 2g$.

Now we are ready to give the result.

Proposition 1.3.7. (Proposition 5.35, [TvLR98]) Let Λ_k be the semigroup generated by the telescopic sequence (a_1, \dots, a_k) . Then

$$\begin{aligned} c(\Lambda_k) - 1 &= d_{k-1}(c(\Lambda_{k-1}) - 1) + (d_{k-1} - 1)a_k = \\ &= \sum_{i=1}^k (d_{i-1}/d_i - 1)a_i, \\ g(\Lambda_k) &= d_{k-1}g(\Lambda_{k-1}) + (d_{k-1} - 1)(a_k - 1)/2 = c(\Lambda_k)/2. \end{aligned}$$

So telescopic semigroups are symmetric. Here we put $d_0 = 0$.

Proof. (of Theorem 1.3.4) Let $\Lambda(r) = \langle 2^{r-1}, 2^{r-1} + 2^{r-2}, 2^r + 1 \rangle, r \geq 3$ be a semigroup generated by $2^{r-1} =: a_1, 2^{r-1} + 2^{r-2} =: a_2$, and $2^r + 1 =: a_3$ for given $r \geq 3$. It is clear that $\gcd(a_1, a_2, a_3) = 1$. Let us check the definition of a telescopic semigroup:

$$\begin{aligned} d_1 &= \gcd(a_1) = 2^{r-1}, A_1 = \{1\}, \Lambda_1 = \mathbb{N}; \\ d_2 &= \gcd(a_1, a_2) = 2^{r-2}, A_2 = \{2, 3\}, \Lambda_2 = \langle 2, 3 \rangle; \\ d_3 &= \gcd(a_1, a_2, a_3) = 1, A_3 = \{a_1, a_2, a_3\}, \Lambda_3 = \Lambda(r); \end{aligned}$$

It is clear that $A_2 = \{2, 3\} \subseteq \mathbb{N} = \Lambda_1$. Also $2^{r-1} \in \Lambda_2 = \langle 2, 3 \rangle$, and $2^{r-1} + 2^{r-2} \in \Lambda_2$. Finally, $2^r + 1 = 2 \cdot (2^{r-1} - 1) + 3 \cdot 1 \in \langle 2, 3 \rangle$. This means that $\Lambda(r)$ is a telescopic semigroup. Let us apply Proposition 1.3.7 to $\Lambda(r)$. We obtain:

$$c(\Lambda(r)) = (d_0/d_1 - 1)a_1 + (d_1/d_2 - 1)a_2 + (d_2/d_3 - 1)a_3 + 1.$$

So that

$$c(\Lambda(r)) = -a_1 + a_2 + (2^{r-2} - 1)a_3 + 1 = 2^{2r-2} - 2^{r-1}.$$

As telescopic semigroups are symmetric, we have:

$$g(\Lambda(r)) = c(\Lambda(r))/2 = 2^{2r-3} - 2^{r-2}.$$

Note, that $g(\Lambda(r)) = g(WS(Q_\infty))$ per Theorem 1.2.2. Considering the fact that $\Lambda(r) \subseteq WS(Q_\infty)$ we conclude that $\Lambda(r) = WS(Q_\infty)$. \square

As a straightforward, but very important corollary, we have

Theorem 1.3.8. $\mathcal{L}(sQ_\infty) = \langle x^i y^j \theta^k \rangle_{i,j,k}$, where $\theta = x^3 + y^2 + xy$ and $i \cdot 2^{r-1} + j(2^{r-1} + 2^{r-2}) + k \cdot (2^r + 1) \leq s$; $i, k \geq 0, j \in \{0, 1\}$.

Proof. This is easily seen as $\dim \mathcal{L}(sQ_\infty) = |WS(Q_\infty) \cap \{0, 1, \dots, s\}|$, and functions of the form $x^i y^j \theta^k$ as above are linearly independent, because they have different orders at Q_∞ . \square

In the next section we are going to show how this theorem applies to codes.

Remark 1.3.9. It can be shown (cf. [S.B04], proof of Theorem 1.3) that the numbers $i \cdot 2^{r-1} + j(2^{r-1} + 2^{r-2}) + k \cdot (2^r + 1)$ are all different provided that $i, k \geq 0, j \in \{0, 1\}$.

1.4 Application to GH-codes

In coding theory ([Sti93], [vL92]) Hermitian codes have taken a special place, as this class of codes provides interesting and non-trivial examples of Goppa codes. These codes are over \mathbb{F}_{q^2} , they are not too short compared with the size of the alphabet, and their parameters k (dimension) and d (minimum distance) are fairly good. In addition there is an efficient way to produce generator matrices for these codes.

Definition 1.4.1. ([Sti93], Definition VII.4.1) For $s \in \mathbb{N}$ we define

$$H_s := C_{\mathcal{L}}(D, sQ_{\infty}),$$

where

$$D := \sum_{\beta^q + \beta = \alpha^{q+1}} P_{\alpha, \beta}$$

is the sum of all places of degree one except Q_{∞} of the Hermitian function field H/\mathbb{F}_{q^2} (cf. Proposition 1.2.1). The codes H_s are called *Hermitian codes*.

All the basic facts on performance of Hermitian code can be found in [Sti93].

We will now treat the Generalized Hermitian codes.

Definition 1.4.2. For $s \in \mathbb{N}$ we define

$$GH_s := C_{\mathcal{L}}(D, sQ_{\infty}),$$

where

$$D := \sum_{\beta^{q^{r-1} + \dots + \beta} = \alpha^{q^{r-1} + q^{r-2} + \dots + \alpha^{1+q}}} P_{\alpha, \beta}$$

is the sum of all places of degree one except Q_{∞} of the Generalized Hermitian function field GH/\mathbb{F}_{q^r} (cf. Theorem 1.2.2). We call the codes GH_s *Generalized Hermitian codes*. They are Hermitian codes for $r = 2$.

GH-codes are codes of length $n = q^{2r-1}$ over \mathbb{F}_{q^r} . For $t \leq s$ we have $GH_t \subseteq GH_s$. Now if $s - q^{2r-1} > 2g - 2 \Rightarrow s > q^{2r-1} + q^{2r-2} - q^{r-1} - 2$. Riemann-Roch Theorem and Theorem 1.1.26 yield $\dim GH_s = \dim(sQ_{\infty}) - \dim(sQ_{\infty} - D) = (s + 1 - g) - (s + 1 - g - q^{2r-1}) = q^{2r-1} = n$, which is trivial. So GH-codes are interesting for $0 < s \leq q^{2r-1} + q^{2r-2} - q^{r-1} - 2$.

Denote $\mathcal{S}(s) := WS(Q_{\infty}) \cap \{0, 1, \dots, s\}$. We have $|\mathcal{S}(s)| = s + 1 - g = s + 1 - \frac{q^{r-1}(q^{r-1}-1)}{2}$ for $s \geq 2g - 1 = q^{r-1}(q^{r-1} - 1) - 1$. As before, we will restrict ourselves to the case $q = 2$. From section 2 we have

$$\begin{aligned} \mathcal{S}(s) = \{l \leq s \mid l = i \cdot 2^{r-1} + j \cdot (2^{r-1} + 2^{r-2}) + k \cdot (2^r + 1); \\ i, k \geq 0, j = 0, 1\}. \end{aligned}$$

Some insight on parameters of the code GH_s over \mathbb{F}_{2^r} gives the following:

Proposition 1.4.3. *Suppose $0 < s \leq 2^{2r-1}$. Then*

(a) *The dimension of GH_s is given by*

$$\dim GH_s = |\mathcal{S}(s)|. \tag{8}$$

For $2^{2r-2} - 2^{r-1} - 1 < s < 2^{2r-1}$, we have

$$\dim GH_s = s + 1 - 2^{r-2}(2^{r-1} - 2). \quad (9)$$

(b) The minimum distance d of GH_s satisfies

$$d \geq 2^{2r-1} - s. \quad (10)$$

Proof. (a) For $0 < s < 2^{2r-1}$, Corollary II.2.3 ([Sti93]) gives $\dim GH_s = \dim \mathcal{L}(sQ_\infty) = |\mathcal{S}(s)|$. The formula (9) is straightforward. \square

(b) Inequality (10) follows from Theorem 1.1.26. \square

Of course, this proposition remains valid for arbitrary q , but for $s \leq 2^{2r-2} - 2^{r-1} - 1$ the description of $|\mathcal{S}(s)|$, which we obtained in subsection 1.3, is crucial.

Let us now present a generator matrix for the GH-codes over \mathbb{F}_{2^r} . We fix an ordering of the set $T := \{(\alpha, \beta) \in \mathbb{F}_{2^r} \times \mathbb{F}_{2^r} \mid \beta^{2^{r-1}} + \dots + \beta = \alpha^{2^{r-1}+2^{r-2}} + \dots + \alpha^3\}$. For $l = i \cdot 2^{r-1} + j(2^{r-1} + 2^{r-2}) + k \cdot (2^r + 1)s$; $i, k \geq 0, j = 0, 1$ we define a vector

$$u_l := (\alpha^i \beta^j (\alpha^3 + \beta^2 + \alpha\beta)^k)_{(\alpha, \beta) \in T} \in (\mathbb{F}_{2^r})^{2^{2r-1}}.$$

As a corollary of Theorem 1.3.4 and Corollary 1.1.29 we have:

Proposition 1.4.4. *Suppose that $0 < s < 2^{2r-1}$ and let $k := |\mathcal{S}(s)|$. Then the $k \times 2^{2r-1}$ matrix*

$$GHM_s := (u_l)_{l \in \mathcal{S}(s)} \quad (11)$$

is a generator matrix for GH_s .

Now we will show how an estimate from Proposition 1.4.3 can be improved by applying results from [CR95]. For this we define

$$C'_s = (C_{\mathcal{L}}(D, \rho_s Q_\infty))^\perp = C_\Omega(D, \rho_s Q_\infty), \quad (12)$$

where $WS(Q_\infty) = (\rho_i)_{i \in \mathbb{N}}$ is a non-gap sequence of Q_∞ .

For these codes a designed Feng-Rao distance $\delta_{FR}(s)$ can be defined (for definition cf. [CR95]). Without going deeply into details we only state that:

$$d(C'_s) \geq \delta_{FR}(s)$$

(Theorem 2.5, [CR95]), and

$$\delta_{FR}(s) \geq \delta_\Gamma(s)$$

where $\delta_\Gamma(s)$ is a Goppa designed distance of C'_s (Corollary 3.9, [CR95]). Note that the estimate in Proposition 3.3 is given via this designed distance.

In [CR95] C.Kirfel and R.Pellikaan give some estimates on δ_{FR} for the case when a Weierstrass semigroup is telescopic. As this is the case in our situation we can apply these results. First we quote:

Theorem 1.4.5. (Theorem 6.10, [CR95]) *Let the semigroup of non-gaps at P ($P = Q_\infty$ in (12)) be generated by the telescopic sequence (a_1, \dots, a_k) . Suppose $a_k = \max(A_k)$ and $d_{k-1} = \gcd(a_1, \dots, a_{k-1}) > 1$. Let (ρ_i) be the non-gap sequence at P . For codes $C(r) = C_\Omega(D, \rho_r P)$ we have*

$$\delta_{FR}(r) = \min\{\rho_t | \rho_t \geq r + 1 - g\},$$

if $3g - 2 - (d_{k-1} - 1)a_k < r \leq 3g - 2$ and $g \leq r$.

Theorem 1.4.6. (Theorem 6.11, [CR95]) *Let the semigroup of non-gaps at P be generated by the telescopic sequence (a_1, \dots, a_k) . Suppose $a_k = \max(A_k)$. If*

$$(j - 1)a_k < \rho_{r+1} \leq ja_k \leq (d_{k-1} - 1)a_k$$

then

$$\delta_{FR}(r) = j + 1.$$

A direct application to our situation yields:

Proposition 1.4.7. *Let C'_s be defined as above. The the following holds:*

$$\delta_{FR}(s) = \min\{\rho_t | \rho_t \geq s + 1 - g\},$$

if $3g - 2 - (2^{r-2} - 1)(2^r + 1) < s \leq 3g - 2$ and $g \leq s$, where $g = 2^{r-2}(2^{r-1} - 1)$, $r \geq 3$.

Proof. We have (cf. the proof of Theorem 1.3.4): $k = 3, d_{k-1} = d_2 = 2^{r-2} > 1, a_3 = \max(A_3) = 2^r + 1$. \square

Proposition 1.4.8. *In the notation as above, if*

$$(j - 1)(2^r + 1) < \rho_{s+1} \leq j(2^r + 1) \leq (2^{r-2} - 1)(2^r + 1)$$

then

$$\delta_{FR}(s) = j + 1.$$

Example 1.4.9. Let us consider the case $q = 2, r = 3$, then $g = 6$. From Proposition 1.4.7 we have $\delta_{FR}(s) = \min\{\rho_t | \rho_t \geq s - 5\}$, if $7 < s \leq 16$. The table 1 lists $s, \delta_{FR}(s)$, and $\delta_\Gamma(s)$ for $s = 8, \dots, 16$. Where $\delta_{FR}(s) > \delta_\Gamma(s)$ a bold font is used.

The case $s = 16$ is of particular interest, see subsection 1.6.

Table 1: Goppa and Feng-Rao bounds for the Example 1.4.9

s	$\delta_{FR}(s)$	$\delta_{\Gamma}(s)$
8	4	3
9	4	4
10	6	5
11	6	6
12	9	7
13	9	8
14	9	9
15	10	10
16	12	11

1.5 Duality property

In this section we want to establish a duality property for GHC, which turns out to generalize the one of HC. First of all, let us recall the corresponding result for Hermitian codes.

Proposition 1.5.1. *The dual code of H_s is*

$$H_s^\perp = H_{q^3+q^2-q-2-s}.$$

Hence H_s is self-orthogonal if $2s \leq q^3 + q^2 - q - 2$, and H_s is self-dual if and only if $s = (q^3 + q^2 - q - 2)/2$.

Now we will formulate an analogous result in a more general setting and then apply it to GHC.

Consider a curve \mathcal{G} over \mathbb{F}_{q^r} given by an equation

$$(f(y))^q + y = g(x),$$

$f(T), g(T) \in \mathbb{F}_q[T]$. Suppose that \mathcal{G} is absolutely irreducible. Denote a function field of \mathcal{G} by \mathcal{F} . Let $N = N(\mathcal{F})$ and $g = g(\mathcal{F})$ denote the number of rational points and the genus of \mathcal{F} resp. Suppose further that the pole P_∞ of x in $\mathbb{F}_{q^r}(x)$ has a unique extension $Q_\infty \in \mathbb{P}_{\mathcal{F}}$, and $Q_\infty|P_\infty$ is totally ramified. From this it follows that Q_∞ is a place of $\mathcal{F}/\mathbb{F}_{q^r}$ of degree one and $(x)_\infty = q \cdot \text{deg}f(T)$ (cf. the proof of Proposition 1.2.4). Finally, assume that for each $\alpha \in \mathbb{F}_{q^r}$, there are $q \cdot \text{deg}f(T)$ elements $\beta \in \mathbb{F}_{q^r}$ such that $(f(\beta))^q + \beta = g(\alpha)$, and for all such pairs (α, β) there is a unique place $P_{\alpha, \beta}$

of degree one with $x(P_{\alpha,\beta}) = \alpha, y(P_{\alpha,\beta}) = \beta$.

Consider a family of codes

$$\mathcal{C}_l = C_{\mathcal{L}}(D, lQ_{\infty}),$$

where $D = \sum_{(f(\beta))^q + \beta = g(\alpha)} P_{\alpha,\beta}$. We will be interested in the case, when $0 \leq l \leq N + 2g - 3$. Note that HC and GHC are \mathcal{C}_l -codes for a special choice of the curve \mathcal{G} .

Theorem 1.5.2. *The dual code of \mathcal{C}_l is*

$$\mathcal{C}_l^{\perp} = \mathcal{C}_{N+2g-3-l}.$$

Proof. First of all we will need the following lemma.

Lemma 1.5.3. *The divisor of the differential dx is*

$$(dx) = (2g - 2)Q_{\infty}.$$

(for an analogous result for Hermitian codes cf. Lemma VI.4.4(d), [Sti93])

Proof. From Remark IV.3.7(c), [Sti93] we have

$$(dx) = -2(x)_{\infty} + Diff(\mathcal{F}/\mathbb{F}_{q^r}(x)), \quad (13)$$

where $Diff(\mathcal{F}/\mathbb{F}_{q^r}(x))$ is a *different* of $\mathcal{F}/\mathbb{F}_{q^r}(x)$. We know that $(x)_{\infty} = q \deg f(T) \cdot Q_{\infty}$, so we have to calculate the different. We need another lemma (Theorem III.5.10(a), [Sti93]):

Lemma 1.5.4. *Suppose $F' = F(y)$ is a finite separable extension of a function field F of degree $[F' : F] = n$. Let $P \in \mathbb{P}_F$ be such that the minimal polynomial $\phi(T)$ of y over F has coefficients in the valuation ring \mathcal{O}_P of P (i.e. y is integral over \mathcal{O}_P), and let $P_1, \dots, P_m \in \mathbb{P}_{F'}$ be all places of F' lying over P . Then*

$$d(P_i|P) \leq v_{P_i}(\phi'(y)) \text{ for } 1 \leq i \leq m,$$

where $d(P_i|P)$ is a different exponent of P_i over P .

Now, $\forall \mathbb{P}_{\mathbb{F}_{q^r}(x)} \ni P \neq P_{\infty} : g(x) \in \mathcal{O}_P$, so $\phi(T) = (f(T))^q + T - g(x) \in \mathcal{O}_P[T]$. Next, $\phi'(y) = 1$. So $\forall \mathbb{P}_{\mathcal{F}} \ni P'|P : d(P'|P) \leq v_{P'}(1) = 0$. By definition of $d(P'|P)$ we have that $d(P'|P) \geq 0$. It follows that $d(P'|P) = 0$. As P_{∞} is totally ramified, we obtain that $Diff(\mathcal{F}/\mathbb{F}_{q^r}(x)) = a \cdot Q_{\infty}$. By Hurwitz genus formula (Theorem III.4.12, [Sti93])

$$a = \deg Diff(\mathcal{F}/\mathbb{F}_{q^r}(x)) = 2g - 2 + 2q \cdot \deg f(T).$$

Collecting all the above, (13) yields:

$$(dx) = (-2q \cdot \text{deg}f(T) + 2q \cdot \text{deg}f(T) + 2g - 2)Q_\infty = (2g - 2)Q_\infty.$$

□

Now we can rewrite the proof of Proposition VII.4.2, [Sti93] for our situation. Consider the element

$$z := \prod_{\alpha \in \mathbb{F}_{q^r}} (x - \alpha) = x^{q^r} - x.$$

z is a prime element for all places $P_{\alpha,\beta} \leq D$, and its principal divisor is $(z) = D - (N - 1)Q_\infty$. Since $dz = d(x^{q^r} - x) = -dx$, the differential dz has the divisor $(dz) = (dx) = (2g - 2)Q_\infty$ due to Lemma 1.5.3. Now Theorem 1.1.26 and Proposition VII.1.2, [Sti93] imply

$$\begin{aligned} \mathcal{C}_l^\perp &= C_\Omega(D, lQ_\infty) = C_{\mathcal{L}}(D, D - lQ_\infty + (dz) - (z)) = \\ &= C_{\mathcal{L}}(D, ((N - 1) + 2g - 2 - l)Q_\infty) = \mathcal{C}_{N+2g-3-l}. \end{aligned}$$

□

It is clear the Proposition 1.5.1 is a corollary of Theorem 1.5.2. A result for GHC looks as follows:

Corollary 1.5.5. *The dual code of GH_s is*

$$GH_s^\perp = GH_{q^{2r-1}+2g-2-s} = GH_{q^{2r-1}+q^{r-1}(q^{r-1}-1)-2-s}.$$

Hence GH_s is self-orthogonal if $2s \leq q^{2r-1} + q^{r-1}(q^{r-1} - 1) - 2$, and GH_s is self-dual (this case can only occur if q is a power of 2) if and only if $s = (q^{2r-1} + q^{r-1}(q^{r-1} - 1) - 2)/2$.

1.6 Computational results

Here we demonstrate some computational results on GH-codes over \mathbb{F}_{2^3} . The codes (their generator matrices) were computed using SINGULAR computer algebra system [GPS05], [ML05] the minimum distance was computed in GAP computer algebra system [GAP02].

In the table 2 d_{rec} is a record value for d for given $n = 32$ and k . These are taken from Brouwer's table ([A.B]) for the linear codes over \mathbb{F}_8 . When discussing estimates on Feng-Rao designed distance in section 3, we saw for $k=16$, $\delta_{FR}(16) \geq 12$. Thus we obtained $[32, 16, \geq 12]$ -code over \mathbb{F}_8 (in a view of the duality property). This yields a new record, which is cited in Brouwer's table.

Table 2: Minimum distance: computed for GH-codes over $\text{GF}(8)$ and corresponding record values

k	d_{rec}	d
6	22	22
7	20	20
8	20	19
9	18	18
10	17	17
11	16	16

1.7 Conclusion

In this paper we studied generalization of Hermitian function field proposed by A.Garcia and H.Stichtenoth. We calculated a Weierstrass semigroup of the point at infinity for the case $q = 2, r \geq 3$. It turned out that unlike Hermitian case, we have already three generators for the semigroup. We then applied this result to codes, constructed on generalized Hermitian function fields. Further, we applied results of C.Kirfel and R.Pellikaan to estimating a Feng-Rao designed distance for GH-codes, which improved on Goppa designed distance. Next, we studied the question of codes dual to GH-codes. We identified that the duals are also GH-codes and gave an explicit formula. We concluded with some computational results. In particular, a new record-giving $[32, 16, \geq 12]$ -code over \mathbb{F}_8 was presented. As a further work we see studying a structure of the Weierstrass semigroup for other values of q . It could also be interesting to apply a theory of generalized weights to GH-codes.

2 Polynomial-based cryptosystems

2.1 Some background on public-key cryptography

In this introductory section we give some basics of public-key cryptography. A more detailed insight on the subject can be found in [AvOS96].

Public-key cryptography emerged in 1976 [WM76] as a way to provide security in groups with a large number of participants, where every participant wants to communicate with many other members of the group. This need was imposed by massively growing electronic communications, where using traditional symmetric cryptosystems hit a wall, because the key management became intractable.

The idea of the public-key cryptography is as follows. As in the case of symmetric-key cryptosystems we have: $\{E_e : e \in \mathcal{K}\}$ is the set of encryption transformations, $\{D_d : d \in \mathcal{K}\}$ is the set of corresponding decryption transformations, where \mathcal{K} is the key space used. Consider any pair of associated encryption/decryption transformations (E_e, D_d) and suppose that each pair has the property that knowing E_e it is computationally infeasible, given a random ciphertext $c \in \mathcal{C}$, to find the message $m \in \mathcal{M}$ such that $E_e(m) = c$. This property implies that given e it is infeasible to determine the corresponding decryption key d . E_e is being viewed here as a trapdoor one-way function with d being the trapdoor information necessary to compute the inverse function and hence allow decryption. This is unlike symmetric-key ciphers where e and d are essentially the same.

Under these assumptions, consider the two-party communication between Alice (a person who wants to send a message) and Bob (a person who wants to receive a message) illustrated in the Figure 1.

Bob selects the key pair (e, d) . He then places the encryption key e (called the *public key*) in the public directory, but keeps the decryption key d (called *private key*) secure and secret. If Alice want to send a message $m \in \mathcal{M}$ to Bob, she needs to extract Bob's public key e from the public directory and encrypt the message $E_e(m) = c$. Bob decrypts the ciphertext c by applying the inverse transformation D_d uniquely determined by d .

Note, that unlike in symmetric-key cryptosystems the encryption key e is being transmitted via an unsecured channel. Public-key encryption, as described here, assumes that knowledge of the public key e does not allow computation of the private key d . In other words, this assumes the existence of *trapdoor one-way functions*. We say that a one-to-one function $f : X \rightarrow Y$ is "one-way" if it is easy to compute $f(x)$ for any $x \in X$, but hard to compute $f^{-1}(y)$ for most randomly selected y in the range of f . A trapdoor one-way function is a function whose one-way status depends on keeping some piece

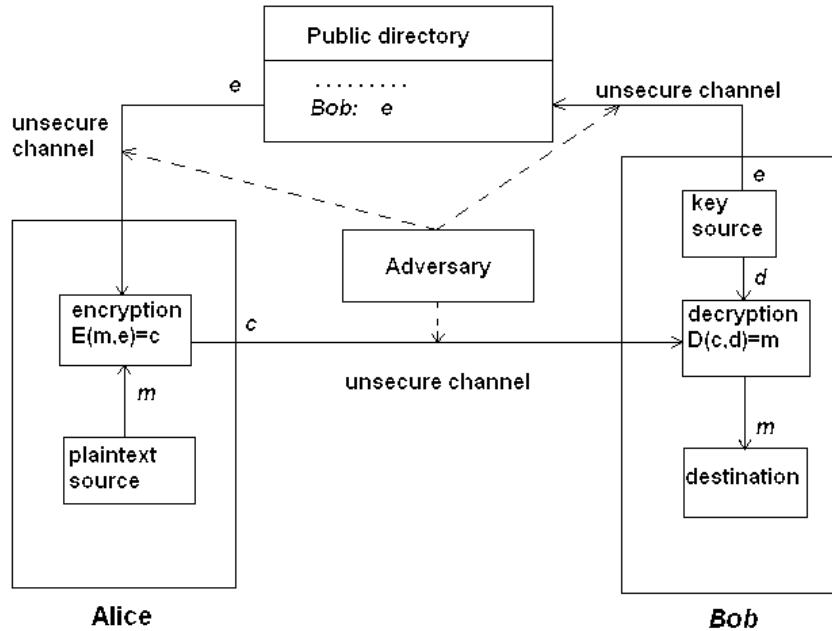


Figure 1: Public-key cryptosystem

of information secret.

Definition 2.1.1. Consider an encryption scheme consisting of the sets of encryption and decryption transformations $\{E_e : e \in \mathcal{K}\}$ and $\{D_d : d \in \mathcal{K}\}$, respectively. The encryption method is said to be a *public-key encryption scheme* if for each associated encryption/decryption pair (e, d) , one key e (the *public key*) is made publicly available, while the other d (the *private key*) is kept secret. For the scheme to be *secure*, it must be computationally infeasible to compute d from e .

In public-key cryptosystems key management is much easier to establish, than in symmetric-key ones. Two communicating parties do not have to agree upon an encryption/decryption pair. Instead the public directory can be used. Despite this advantage public-key encryption schemes are slower, than symmetric-key ones. Thus the usual idea is to use public-key encryption for transmitting the secret key for symmetric key encryption for the further communication between the two parties.

2.2 Polly Cracker cryptosystem

2.2.1 Polly Cracker: initial specialization

Since the failure of knapsack-based cryptosystems in 1980's, a common conviction of cryptographic society was that NP-complete problems were not suitable for a construction of secure trapdoor one-way functions. M.Fellows and N.Koblitz in 1993 ([MN93], [N.K98]) proposed a generic public-key cryptosystem, which could be derived from a collection of combinatorial NP-complete problems. They called it the Polly Cracker cryptosystem. We will not go into details about how a specific cryptosystem can be obtained from a combinatorial NP-complete problem. Although, we will see two concrete instances, namely the Koblitz's "graph perfect code instance" (GPCI) and an instance based on 3-satisfiability problem from logic.

Let us now describe a general framework for the Polly-Cracker cryptosystem. Let $\mathbb{F}_q[x_1, \dots, x_n]$ denote a ring of polynomials in n indeterminates over a finite field \mathbb{F}_q with q elements.

Public key: $f_1, \dots, f_s \in \mathbb{F}_q[x_1, \dots, x_n]$.

Private key: a common zero $\sigma \in \mathbb{F}_q^n$ of f_1, \dots, f_s , i.e.

$$f_1(\sigma) = \dots = f_s(\sigma) = 0.$$

Encryption: to encrypt a plaintext $m \in \mathbb{F}_q$, Bob chooses $h_1, \dots, h_s \in \mathbb{F}_q[x_1, \dots, x_n]$ and sends $c := m + \sum_{j=1}^s h_j f_j$ to Alice.

Decryption: Alice recovers m by evaluating c at σ :

$$c(\sigma) = m + \sum_{j=1}^s h_j(\sigma) \cdot f_j(\sigma) = m.$$

Here, security with respect to the private key is based on NP-hardness of determining a solution of a system of algebraic (polynomial) equations. Despite the fact that the problem chosen for security of the cryptosystem is undoubtable, as it is equivalent to NP-hard Ideal-Membership Problem, finding good practical instances of Polly Cracker turns out to be a challenging task. It was shown that some straightforward instances of Polly Cracker can easily be broken.

One of the immediate ideas on attacking a Polly Cracker cryptosystem without revealing of the private key resides on linear algebra. Here two main cases arise: dense polynomials of small degree and sparse polynomials of high

degree.

D.Naccache et al. [BCE⁺94] describes a linear algebra attack in the dense case. In order to recover a plaintext m from a corresponding ciphertext c one can put up a system of linear equations by comparing coefficients in $c := m + \sum_{j=1}^s h_j f_j$, where coefficients of h_j 's and m itself are unknowns. The ideology behind Naccache's attack is that in dense case polynomials can be represented as vectors, and encryption function is just a map of vector spaces. So linear algebra allows to invert this encryption map efficiently.

In the sparse case one presents polynomials by their coefficients and the support of monomials. Direct linear algebra approach as above leads to a system of linear equations, where the number of unknowns is very large, but, here an unartful choice of h_j 's in the encryption step can still enable a linear algebra attack. This attack was first proposed by Lenstra (in [N.K98]).

The Lenstra's intelligent linear algebra attack is based on considering the set \mathcal{M} , which consists of those terms (cf. 2.2.2 below) t for which there exist terms t_f and t_c occurring in (one of the) public key polynomials f_1, \dots, f_s and in the ciphertext c resp., such that $t_c = t \cdot t_f$. That is, \mathcal{M} is the set of terms, which Bob can potentially use for encryption. Denote by $\mathcal{M}(h_j)$ the set of terms of h_j . Then if

$$\cup_{j=1}^s \mathcal{M}(h_j) \subseteq \mathcal{M},$$

(i.e. every h_j divides at least one term in the ciphertext c), then it is possible to perform the above mentioned linear algebra attack with $|\mathcal{M}|$ unknowns. Namely, if such a set \mathcal{M} is known, then we can write

$$c = A_0 + \sum_{j=1}^s \left(\sum_{\alpha \in \mathcal{M}} A_{j\alpha} \alpha \right) \cdot f_j,$$

where A_0 is an indeterminate responsible for the plaintext m , α runs through the set \mathcal{M} with a corresponding coefficient $A_{\alpha n}$.

In order to dodge such an attack Koblitz proposed in [N.K98]; "..., Bob must artfully build at least one monomial d' into at least one h_j such that d' times *any* term in f_j is canceled in the entire sum (so that it does not occur in the set of monomials occurring in the ciphertext c). Also, the monomials d' with that property should not be too few and/or too easy to guess,..." Such monomials d' are called "hidden". The following "differential" attack by D.Hofheinz and R.Steinwandt ([DR02]) aims at revealing such hidden monomials. we will then see how it can be applied to Koblitz's GPCI.

The main tool in the "differential" attack is a function Δ :

$$\Delta : \mathbb{F}_q[x_1, \dots, x_n] \longrightarrow 2^{\mathbb{F}_q[x_1, \dots, x_n, x_1^{-1}, \dots, x_n^{-1}]}$$

$$\sum_{\nu \in \mathbb{N}_0^n} \gamma_\nu \cdot x^\nu \longmapsto \left\{ \frac{\gamma_\mu}{\gamma_\eta} \cdot x^{\mu-\eta} \mid \mu > \eta, \gamma_\mu \cdot \gamma_\eta \neq 0 \right\},$$

where we denote $x^\nu = x_1^{\nu_1} \dots x_n^{\nu_n}$, for $\nu = (\nu_1, \dots, \nu_n \in \mathbb{N}_0^n)$, $>$ denotes a monomial order (see 2.2.2), for example it can be chosen to be *lex* order. So, Δ maps given polynomial to a set of its "differences". Some simple properties of Δ include:

1. $\Delta(a) = \emptyset \Leftrightarrow a$ is a term or $a = 0$.
2. $|\Delta(a)| \leq \frac{|\mathcal{M}(a)|^2 - |\mathcal{M}(a)|}{2}$, where $\mathcal{M}(a)$ is the set of monomials occurring in a .
3. $\Delta(a) = \Delta(\gamma_\nu x^\nu \cdot a)$;
4. $\Delta(a + b) \supseteq \Delta(a) \cup \Delta(b)$, for $a, b : \mathcal{M}(a) \cap \mathcal{M}(b) = \emptyset$.

Now recall that $c = m + \sum_{j=1}^s h_j \cdot f_j$. Having properties 3-4 we can hope that with some (not negligible) probability there exist f_i :

$$\Delta(f_i) \cap \Delta(c) \neq \emptyset.$$

Further assume that for some $i \in \{1, \dots, s\}$ there exist terms $\gamma_{\mu_i} x^{\mu_i}, \gamma_{\nu_i} x^{\nu_i}$ in f_i such that

1. there is a "characteristic difference"

$$\delta_i := \gamma_{\mu_i} x^{\mu_i} / \gamma_{\nu_i} x^{\nu_i} \in \Delta(f_i) \setminus (\cup_{j \neq i} \Delta(f_j)), \text{ and}$$

2. there is a term $\gamma_{\eta_i} x^{\eta_i}$ in h_i such that $x^{\eta_i} x^{\mu_i}$ and $x^{\eta_i} x^{\nu_i}$ do not occur among monomials of $c - \gamma_{\eta_i} x^{\eta_i} \cdot f_j, j \neq i$.

Assumption 2 excludes "collisions" of coefficients of monomials $x^{\eta_i} x^{\mu_i}$ and $x^{\eta_i} x^{\nu_i}$ during the computation of c . In addition 2 ensures that

$$\delta_i = \frac{\gamma_{\eta_i} x^{\eta_i} \cdot \gamma_{\mu_i} x^{\mu_i}}{\gamma_{\eta_i} x^{\eta_i} \cdot \gamma_{\nu_i} x^{\nu_i}} \in \Delta(c).$$

If an attacker then is able to find terms t_1, t_2 , such that $x^{\mu_i} | t_1$ for some i and $t_1/t_2 = \delta_i$, then he may assume 1 and 2 and make a guess for a possible term in h_i :

$$t_h := \frac{t_1}{\gamma_{\mu_i} x^{\mu_i}} = \frac{t_2}{\gamma_{\nu_i} x^{\nu_i}}$$

He can verify his guess by considering $c' = c - t_h f_i$. If the number of monomials in c' decreases, then this can be taken as an evidence of the fact that t_h was a correct guess. In particular, if $c' \in \mathbb{F}_q$, then the encrypted plaintext has been recovered successfully. This process above should be iterated with "characteristic differences" in $\Delta(f_i) \setminus (\cup_{j \neq i} \Delta(f_j)), \forall i = 1, \dots, s$, until the encrypted message is recovered.

The described procedure can reveal hidden monomials. Specifically, if some t_{h_j} , a term in h_j , is hidden in c , then $\forall i, 1 \leq s \leq s \exists t_{h_i}$ in h_i, t_{f_i} in f_i :

$$t_{h_j} f_j + \sum_{i=1}^s t_{h_i} t_{f_i} = 0.$$

So, if an attacker is able to find a monomial $t_{h_i} \in \{t_{h_1}, \dots, t_{h_s}\}$ then he knows that $c' = c - t_{h_i} f_i$ contains some $t_{h_j} t_{f_j}$. Therefore, the monomial, which was hidden in the ciphertext c is no longer hidden in successive c' . Later we will discuss an attack inspired by the "differential" attack. It no longer computes "differences", but supposedly does reductions in the size of a current c .

This "differential" attack was successfully applied by Hofheinz and Steinwandt to Koblitz's GPCI. Here we outline the instance itself and an outcome of the proposed attack. In Koblitz's GPCI for a derivation of a public key/private key pair, Alice chooses a 3-regular (undirected) graph $G = (V, E)$ such that there is a subset $V' \subseteq V$ of the vertices that forms a *perfect code*, i.e. each vertex $v \in V$ of G is in the neighborhood $N(v')$ of one and only one $v' \in V'$ (here $N(v')$ the set of vertices that are incident to v'). Public polynomials can then be derived from such a graph ([N.K98]). In order to comprehend results of the attack it is sufficient to know that Alice's public polynomials f_i are in $\mathbb{F}_2[x_1, \dots, x_n]$, where $n = |V|$. The security of the resulting cryptosystem is based on inability of the attacker to find the perfect code V' from G . As Koblitz provided no concrete procedures on how specific keys are to be obtained, Hofheinz and Steinwandt used random graphs.

Koblitz suggested the value of $n = 500$, but it turns out that for such n message expansion becomes huge, so the cryptosystem for $n = 500$ is not practical whatsoever. Reasonable size of a ciphertext can be obtained in the range $100 \leq n \leq 200$. For such n the number of monomials in a ciphertext is large, but manageable, e.g. $\sim 60,000$ monomials. The attack proceeds as follows:

1. Pick one of the (public) polynomials f_i of the form $1 - \sum_{u \in N[v]} x_u$ (cf. [N.K98]) randomly.
2. For each term t_c in c and each term t_f in f_i with $t_f | t_c$ check whether

subtracting $R(f_i; \frac{t_c}{t_f})$ reduces c in size (see [N.K98], Ch.5, 7] for a definition of the reduction operator R).

3. If so, iterate the attack with the "simplified" ciphertext from step 2 until we end up with a constant c .
4. Otherwise proceed with step 2, and eventually skip back to step 1.

Note that in this attack $\Delta(c)$ was not explicitly computed, but rather potential terms t_c/t_f in h_i were guessed.

When the encrypted message was $0 \in \mathbb{F}_2$ the attack yielded the following results:

- For $n = 100, d = 13$ ($d = \deg(c)$) in 1,000 encryptions of the plaintext $m = 0$ Hofheinz and Steinwandt could 951 times reveal m successfully.
- For $n = 128, d = 14$ in 100 encryptions of $m = 0$, 94 times m was revealed successfully.
- For $n = 160, d = 15$ in 30 encryptions of $m = 0$, 28 times m was revealed successfully.
- For $n = 200, d = 15$ from a ciphertext with 575,182 terms the plaintext $m = 0$ could be recovered successfully in ≈ 8.75 hours.

The attacks described above, namely different linear algebra attacks and "differential" attack qualify as ciphertext-only attacks. The following attack due to R.Steinwandt and W.Geiselmann [RW02] is a chosen-ciphertext attack.

The attack is based on the fact that decryption device in Polly Cracker does not have any mechanism against different kinds of "fake" ciphertexts. So, an attacker can send a "ciphertext" $\tilde{c}_i := x_i + \sum_{j=1}^s f_j h_{ij}, 1 \leq i \leq n$ instead of "legitimate" $c = m + \sum_{j=1}^s f_j h_j$. Next $\tilde{c}_i(\sigma) = \sigma_i$, where $\sigma = (\sigma_1, \dots, \sigma_n) \in \mathbb{F}_q^n$ is a common zero of $\{f_1, \dots, f_s\}$. So, by sending n "fake" ciphertexts $\tilde{c}_1, \dots, \tilde{c}_n$ the attacker is able to disclose the private key. The attacker can also "mask" his fake ciphertexts by first applying linear coordinate change:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{pmatrix} := A \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix},$$

and the send

$$c'_i := x'_i + \sum_{j=1}^s f_j h_{ij}.$$

In order to recover the private key σ , the attacker now needs to perform an inverse coordinate change A^{-1} after evaluation of c'_i 's at σ .

In [RW02] the following modification of the decryption step is considered. Namely, Alice constructs her public polynomials f_1, \dots, f_s in such a way that she knows *two* common zeroes $\sigma_1, \sigma_2 \in \mathbb{F}_q^n$ of f_1, \dots, f_s . For each $1 \leq i \leq n$ the i -th coordinate of σ_1 is to be different from the i -th coordinate of σ_2 . Upon reception of a ciphertext polynomial c , Alice checks whether $c(\sigma_1) = c(\sigma_2)$. If this is not fulfilled, then the ciphertext is considered to be invalid.

This modification counters the above attack, but still adaptive chosen-ciphertext attack can be performed via sending ciphertexts $\tilde{c}_{\alpha,\beta}(\alpha, \beta \in \mathbb{F}_q : \alpha \neq \beta)$ of the form:

$$\tilde{c}_{\alpha,\beta} := (x_i - \alpha)(x_i - \beta) + \sum_{j=1}^s f_j h_{j,\alpha,\beta}.$$

If $\tilde{c}_{\alpha,\beta}(\sigma_1) = \tilde{c}_{\alpha,\beta}(\sigma_2) = 0$, then the attacker concludes that $\{(\sigma_1)_i, (\sigma_2)_i\} = \{\alpha, \beta\}$. In such manner the attacker is able to construct n pairs $\{\alpha_1, \beta_1\}, \dots, \{\alpha_n, \beta_n\} \subseteq \mathbb{F}_q$ for all n coordinate positions. He further is able to distinguish the first two coordinates of σ_1 and σ_2 by sending

$$(x_1 - \alpha_1)(x_2 - \alpha_2) + (x_1 - \beta_1)(x_2 - \beta_2) + \sum_{j=1}^s f_j h_j.$$

By iterating this process $n - 1$ times, the attacker can fully recover σ_1 and σ_2 .

Although this attack is not practical for large q , it exposes certain structural flaws of the initial Polly Cracker.

2.2.2 Gröbner bases point of view

In this section we want to present a point of view on Polly Cracker-like cryptosystem from Gröbner bases theory. To this end we need a notion of a Gröbner basis. After we give some preliminaries, we are going to present F.Levy-dit-Vehel's and L.Perret's Polly Cracker based on satisfiability problem from logic. Then we will remark on generalizations of the Gröbner basis theory that can be used in order to unify study of many known public-key cryptosystems including polynomial-based ones. Here we provide only sketchy introduction to Gröbner bases. For the full treatment of the subject and on examples, the reader is referred to [GMG02], [AKC05].

In order to perform computations in the ring $K[x_1, \dots, x_n]$, K a field, one needs a notion of a monomial ordering. So, let us start with the terminology.

Definition 2.2.1. Let $K[x] = K[x_1, \dots, x_n]$.

- $x^\alpha := x_1^{\alpha_1} \dots x_n^{\alpha_n}$ is called a *monomial* in x_1, \dots, x_n with *exponent vector* $\alpha \in \mathbb{N}^n$;
- $|\alpha| := \alpha_1 + \dots + \alpha_n$ is called the *total degree* of the monomial x^α ;
- $Mon(x) := \{x^\alpha | \alpha \in \mathbb{N}^n\}$.

If $f = \sum_{\alpha} c_{\alpha} x^{\alpha} \in K[x] \setminus \{0\}$ then we call

- c_{α} the *coefficient* of x^{α} in f ;
- $c_{\alpha} x^{\alpha}$ with $c_{\alpha} \neq 0$ a *term*¹ of f , and
- $deg(f) := \max\{|\alpha| | c_{\alpha} \neq 0\}$ the *total degree* of f .

We set $deg(0) := -\infty$.

Definition 2.2.2. A *monomial ordering* on $K[x]$ is any relation $>$ on $Mon(x)$ (or equivalently on \mathbb{N}^n) such that

- $>$ is a *total ordering* on $Mon(x)$, i.e. $\forall x^{\alpha}, x^{\beta} \in Mon(x)$ one and only one of the following holds: $x^{\alpha} > x^{\beta}, x^{\alpha} = x^{\beta}, x^{\alpha} < x^{\beta}$.
- $>$ is *multiplicative*, i.e. $x^{\alpha} > x^{\beta} x^{\alpha} \cdot x^{\gamma} > x^{\beta} \cdot x^{\gamma} \forall \gamma \in \mathbb{N}^n$.
- $>$ is a *well-ordering*, i.e. $x^{\alpha} > 1 \forall \alpha \in \mathbb{N}^n \setminus \{0\}$.

Example 2.2.3. The following are three most frequently used monomial orderings on $K[x]$:

- *Lexicographic order* (w.r.t $x_1 > x_2 > \dots > x_n$)

$$x^{\alpha} >_{lp} x^{\beta} : \Leftrightarrow \exists s : \alpha_1 = \beta_1, \dots, \alpha_{s-1} = \beta_{s-1}, \alpha_s > \beta_s.$$

- *Graded lexicographic order*

$$x^{\alpha} >_{Dp} x^{\beta} : \Leftrightarrow |\alpha| > |\beta| \text{ or } (|\alpha| = |\beta|, x^{\alpha} >_{lp} x^{\beta}).$$

- *Graded reverse lexicographic order*

$$x^{\alpha} >_{dp} x^{\beta} : |\alpha| > |\beta| \text{ or } (|\alpha| = |\beta|, \exists s : \alpha_n = \beta_n, \dots, \alpha_{n-s+1} = \beta_{n-s+1}, \alpha_{n-s} < \beta_{n-s}).$$

¹Some authors use terminology, where "monomial" is called "term", and "term" is called "monomial"

Definition 2.2.4. (Leading data) Let $f = \sum_{\alpha} c_{\alpha} x^{\alpha} \in K[x] \setminus \{0\}$, and let $>$ be any monomial ordering on $K[x]$. Set $\alpha_0 := \max_{>} \{\alpha \in \mathbb{N}^n \mid c_{\alpha} \neq 0\}$. Then

- $lc(f) := c_{\alpha_0}$ is called the *leading coefficient* of f ;
- $lm(f) := x^{\alpha_0}$ is called the *leading monomial* of f ;
- $lt(f) := c_{\alpha_0} x^{\alpha_0}$ is called the *leading term* of f .

Now we are ready to give a definition of a Gröbner basis of an ideal $I \subseteq K[x_1, \dots, x_n]$, but before that we give some motivation for this notion. It is known (Hilbert's basis theorem) that any ideal $I \subseteq K[x_1, \dots, x_n]$ can be generated by a finite number of elements: $I = \langle f_1, \dots, f_s \rangle$. The interesting and practical question is the study of the set of common zeroes of $f_1, \dots, f_s \in K[x]$. We know that in the case, when f_i 's are linear, Gaussian elimination works quite well. On the other hand, when $s = 1$, different factorization techniques, e.g. Berlekamp's algorithm, can be efficiently used. Gröbner basis computations were aimed to generalize both Gaussian elimination and extended Euclidian algorithm. So that a Gröbner basis for an ideal I gives an opportunity to solve the system $f_1 = \dots = f_s = 0$. We note also that many other interesting properties of objects from algebraic geometry can be studied via Gröbner bases.

Definition 2.2.5. (1) The *leading ideal* of I (w.r.t $>$) is

$$L(I) := L_{>}(I) := \langle lt_{>}(f) \mid f \in I \rangle \subset K[x].$$

(2) A finite subset $G = \{g_1, \dots, g_r\}$ of I is called a *Gröbner basis* for (w.r.t $>$) if

$$L(I) = \langle lt(g_1), \dots, lt(g_r) \rangle.$$

Proposition 2.2.6. Let $I \subset K[x]$ be an ideal, and let $G \subset I$ be a Gröbner basis for I (w.r.t $>$). Then:

(1) If $J \subset K[x]$ is an ideal such that $I \subset J$, then $(I = J \iff L(I) = L(J))$.

(2) $\langle G \rangle = I$.

Definition 2.2.7. A Gröbner basis G is called *reduced*, if for all $f \neq g \in G$ no term of f is divisible by the leading term of g .

It can be shown that an ideal I always has a Gröbner basis. Moreover, reduced Gröbner basis of I is unique. The next important notion, which always accompanies the notion of a Gröbner basis is the notion of a (reduced) normal form.

Proposition 2.2.8. (*Division with Remainder*) Let $f_1, \dots, f_k \in K[x] \setminus \{0\}$. For every $f \in K[x]$ there exist $a_1, \dots, a_k \in K[x]$ and $h \in K[x]$ such that

- $f = a_1 f_1 + \dots + a_k f_k + h$,
- $lm(f) \geq lm(a_i f_i)$ whenever f and $a_i f_i$ are non-zero, and
- if $h \neq 0$, then $lt(h)$ is not divisible by any of $lt(f_1), \dots, lt(f_k)$.

Any such expression is called a standard expression for f in terms of the f_i (with remainder h).

Definition 2.2.9. For $f \in K[x]$, and any finite ordered set $\mathcal{F} \subset K[x]$, a normal form for f w.r.t \mathcal{F} , denoted as $NF(f|\mathcal{F})$, is the remainder of some standard expression for f in terms of the elements in \mathcal{F} .

We have the following nice characterization

Proposition 2.2.10. Let $I \subset K[x]$ be an ideal, let $G \subset I$ be a Gröbner basis for I (w.r.t $>$), and let $NF(\cdot|G)$ be a normal form w.r.t G . Moreover, let $f \in K[x]$. Then

$$f \in I \iff NF(f|G) = 0.$$

If the remainder h in the Definition 2.2.9 is chosen to be such that if $h \neq 0$, then no term of h is divisible by any of the $lt(f), f \in \mathcal{F}$, then the corresponding normal form is called *reduced*. Reduced normal form w.r.t a Gröbner basis is uniquely defined. So a notion of Gröbner basis gives nice uniqueness condition on division issues. Next, we want to formulate Buchberger's algorithm for computing Gröbner bases. For this we need:

Definition 2.2.11. Let $f, g \in K[x] \setminus \{0\}$, and denote their leading monomials by $x^\alpha := lm(f), x^\beta := lm(g)$. Setting $x^\gamma := lcm(x^\alpha, x^\beta)$, the *s-polynomial* of f and g is defined as

$$spoly(f, g) := x^{\gamma-\alpha} \cdot f - \frac{lc(f)}{lc(g)} \cdot x^{\gamma-\beta} \cdot g.$$

The *spoly* can be regarded as a generalization of the elementary row operations in Gaussian elimination.

Algorithm 2.2.12. (Buchberger's algorithm)

Input: $f_1, \dots, f_k \in K[x]$, and a normal form NF .

Output: $G \subset K[x]$, a Gröbner basis for $\langle f_1, \dots, f_k \rangle \subset K[x]$, w.r.t $>$.

(1) Set $G := \{f_1, \dots, f_k\}$;

$\mathcal{P} := \{(f_i, f_j) | 1 \leq i < j \leq k\}$;

WHILE ($\mathcal{P} \neq \emptyset$) DO

- choose $(f, g) \in \mathcal{P}$;
- set $\mathcal{P} := \mathcal{P} \setminus \{(f, g)\}$;
- set $h := NF(\text{spoly}(f, g)|G)$;
- IF $(h \neq 0)$ THEN $\mathcal{P} := \mathcal{P} \cup \{(h, f) | f \in G\}$; $G := G \cup \{h\}$;

(3) RETURN(G).

The termination of this algorithm is due to the fact that $K[x_1, \dots, x_n]$ is a Noetherian ring. The correctness of the algorithm relies on Buchberger's Criterion.

Theorem 2.2.13. (*Buchberger's Criterion*) *Let $I \subset K[x]$ be an ideal, let $G = \{g_1, \dots, g_r\} \subset I$ and let NF be a normal form w.r.t G . Then the following are equivalent:*

- G is a Gröbner basis for I .
- $NF(f|G) = 0 \forall f \in I$.
- I is generated by G , and, for each $i, j = 1, \dots, r$ we have

$$NF(\text{spoly}(g_i, g_j)|G_{ij}) = 0$$

for some subset G_{ij} of G .

Remark 2.2.14. The complexity of the Buchberger's Algorithm is doubly exponential in the number of variables, but in practice it performs quite well. Numerous refinements of the initial algorithm were proposed. The most recent and prominent one's are due to J.-C.Faugère ([Fau99]).

Now we will take a closer look at an instance of the Polly Cracker cryptosystem proposed by F.Levy-dit-Vehel and L.Perret [dVL04]. Security of this system relies on hardness of a satisfiability problem from propositional logic. More precisely, the authors consider 3-satisfiability problem (3-SAT). Although this problem is proved to be NP-complete, it admits "easy" instances, for which effective deterministic algorithms exist. The authors have chosen the so-called doubly-balanced 3-SAT. It is known that this instance is much harder to handle than a random one.

So, let us recall some notions from logic that we will need. Let $x = \{x_1, \dots, x_n\}$ be a set of variables; \wedge, \vee, \neg denote logical *and, or, not* resp. A truth assignment on x is a function $t : x \rightarrow \{\text{True}, \text{False}\}$. For all $j, 1 \leq j \leq n$ a *literal* u_j is either x_j or \bar{x}_j . For a variable $x_j \in X$, a literal x_j (resp. \bar{x}_j) is

true if $t(x_j) = \text{True}$ (resp. $t(x_j) = \text{False}$). A *clause* over x is the disjunction (\vee -operation) of a set of literals over x . A clause containing only three literals is called *3-clause*. A *Conjunctive Normal Form (CNF)-formula* \mathcal{C} is the conjunction (\wedge -operation) of arbitrary many clauses C_1, \dots, C_m . It is said to be *satisfiable* iff there exist some truth assignment for x that simultaneously satisfies all the clauses in \mathcal{C} . Such a truth assignment is called a *model* for the formula \mathcal{C} . If \mathcal{C} contains only 3-clauses, then we say that \mathcal{C} is a *3-CNF formula*. For instance, $\mathcal{C} = \bigwedge_{j=1}^m C_j$, where $C_j = u_{j_1} \vee u_{j_2} \vee u_{j_3}$, is such a formula.

The 3-satisfiability problem can then be stated as follows.

INSTANCE: a collection $\mathcal{C} = \{C_1, \dots, C_m\}$ of 3-clauses.

QUESTION: is there a satisfying model for \mathcal{C} ?

There are procedures for obtaining random 3-SAT instances, given a number of variables n , and the number of clauses m .

In the double-balanced 3-SAT, every variable appears (almost) equally often, and (almost) as often negated as not negated.

The next step is to obtain polynomials out of clauses. Let $K[x_1, \dots, x_n]$ be a ring of polynomials in n variables over a field K . Fix $T, F \in K$, representing *True* and *False* resp. If we have a 3-clause C involving the three literals $u_j, u_k, u_l, 1 \leq j, k, l \leq n$, we can associate a polynomial in $K[x]$ as follows: if $u_j = x_j$, then we replace u_j by $(x_j - T)$; if $u_j = \bar{x}_j$, then we replace it by $(x_j - F)$. Then replace \vee by multiplication. For example a polynomial $p_c(x) = p_c(x_1, \dots, x_n) \in K[x]$ corresponding to the clause $C = x_j \vee \bar{x}_k \vee x_l$ is $p_c(x) = (x_j - T)(x_k - F)(x_l - T)$. We then have that a model of X for c corresponds to a zero of the polynomial $p_c(x)$. In general we have:

Theorem 2.2.15. *A 3-CNF formula $\mathcal{C} = \bigwedge_{i=1}^m C_i$ admits a model iff the corresponding system of polynomial equations $\{p_1(x) = \dots = p_m(x) = 0\}$ has a solution over the algebraic closure of K .*

Further, let $k \in \{1, \dots, m\}, \{i_1, \dots, i_k\} \subset \{1, \dots, m\}$ and $\{C_{i_j}\}_{1 \leq j \leq k}$ be a set of clauses. We will say that $\{Var(C_{i_j})\}_{1 \leq j \leq k}$ is a *disjoint set* if for all $a, b \in \{i_1, \dots, i_k\}, a \neq b, Var(C_a) \cap Var(C_b) = \emptyset$, where $Var(C)$ is a set of variables from $x = x_1, \dots, x_n$, which occur in $p_c(x)$. The next result connects sets of clauses and a Gröbner basis.

Proposition 2.2.16. *Let $\mathcal{C} = \bigwedge_{i=1}^m C_i$ be a 3-CNF formula, $\{p_1, \dots, p_m\} \subseteq K[x]$, where $p_i = p_{C_i}(x), 1 \leq i \leq m$ as above; $T, F \in K$. If $\{Var(C_{i_j})\}_{1 \leq j \leq k}$ is a disjoint set, then $\{p_{i_j}\}_{1 \leq j \leq k}$ is the reduced Gröbner basis of $\langle p_{i_j} \rangle_{1 \leq j \leq k}$ for the graded lexicographic order.*

Now, let us take a look at how the Polly Cracker for this instance looks like. We fix $K = \mathbb{F}_q, q \geq 3$ and $T, F \in \mathbb{F}_q \setminus \{0\}$.

Private key: a random vector $y \in \{T, F\}^n$.

Public key: q, n, m , an instance of doubly-balanced 3-SAT admitting y as a model: $\mathcal{C} = \bigwedge_{i=1}^m C_i$ (there are generation methods for this); also T and F . Equivalently Alice can publish m polynomials p_1, \dots, p_m , which correspond to the clauses C_1, \dots, C_m .

Encryption: We denote $I = \langle p_1, \dots, p_m \rangle \subset \mathbb{F}_q[x]$. The following algorithm returns an element e_I from I .

Algorithm 2.2.17. (Element of the ideal)

Input: $f \in \mathbb{F}_q[x], l \geq 2, \{\lambda_1, \dots, \lambda_l\} \subseteq \mathbb{F}_q : \sum_{i=1}^l \lambda_i = 0$ and $D = (d_1, \dots, d_l)$ a set of subset indexes such that $\forall 1 \leq i \leq l : \{Var(C_{i_j})\}_{j \in d_i}$ is a disjoint set.

Output: $e_I \in I$.

FOR $i = 1$ TO l DO

 compute $NF(f|\{p_{i_j}\}_{j \in d_i}) =: N_i(f)$

END FOR

RETURN($e_I = \sum_{i=1}^l \lambda_i N_i(f)$)

For $e_I(x) = \sum_{\alpha \in \mathbb{N}^n} a_\alpha x^\alpha$ (only finitely many $a_\alpha \neq 0$) Bob chooses $\beta = (\beta_1, \dots, \beta_n) \in \text{supp}(e_I) = \{\alpha \in \mathbb{N}^n : a_\alpha \neq 0\}, \beta \neq (0, \dots, 0)$. He encrypts a plaintext $M \in \mathbb{F}_q$ with a ciphertext as:

$$c(x) = e_I(x) + M \cdot x_1^{\beta_1} \dots x_m^{\beta_m} \in \mathbb{F}_q[x],$$

and sends $(c(x), \beta)$ to Alice.

Decryption: Alice evaluates

$$\frac{c(y)}{y^\beta} = \frac{e_i(y) + My^\beta}{y^\beta} = m.$$

Note that as $T, F \neq 0 \Rightarrow y^\beta \neq 0 \forall \beta$.

As was stated before, security of 3-SAT Polly Cracker on one hand resides on harness of solving a satisfiability problem. The authors claim that for $700 \leq n \leq 900, m = 3.5n$ doubly-balanced 3-SAT instances far beyond reach of the current best solvers for satisfiability problems. As they point out this doubly-balanced 3-SAT is too "random" to be solved by solvers that work with "regular" formulae, and too "regular" to be solved by ones that work with "random" formulae.

On the other hand, security of this scheme relies on the difficulty of finding a solution of a system of m polynomial equations (of maximal degree 3) in n variables over a finite field \mathbb{F}_q . Here different Gröbner bases techniques can be used. In particular F4 algorithm by Faugere [Fau99]. From the paper [dVL04] it is not convincing enough that such systems cannot be solved

by F4, as for computations the authors used web interface for F4, which cannot be considered as a serious computational experiment.

Now, let us take a look at possible weaknesses of Bob's encryption strategy, rather than Alices' private key generation. The following result proves security of 3-SAT Polly Cracker against intelligent linear algebra attack by Lenstra (see 2.2.1 above).

Theorem 2.2.18. *Let $\{p_1, \dots, p_m\}$ be the polynomials of the public key and x^α be a monomial of total degree d . We set $d \leq q$ and define $f'(x) = ax^\alpha + g(x) \in \mathbb{F}_q[x]$ with $(a, \alpha) \in \mathbb{F}_q^* \times \mathbb{N}^n$, such that the terms of the polynomial $g(x) \in \mathbb{F}_q[x]$ are of total degree strictly smaller than $d - 3$. We also set:*

- $l \geq 2$,
- $\{\lambda_1, \dots, \lambda_l\} \subseteq \mathbb{F}_q : \sum_{i=1}^l \lambda_i = 0$,
- $D = (d_1, \dots, d_l)$ a set of indexes such that $\forall 1 \leq i \leq l : \{p_{i_j}\}_{j \in d_i}$ is constructed from a disjoint set of clauses; $\forall 1 \leq i \leq l : d_i$ is set of indexes of cardinality $3 < |d_i| \leq \lfloor \frac{n}{3} \rfloor$ ($\lfloor \frac{n}{3} \rfloor$ is the maximum cardinality of a disjoint set).

If $e_I \in \langle p_1, \dots, p_m \rangle = I$ is computed by Algorithm 2.2.17 with these parameters, then we have:

$$E \not\subseteq \mathcal{M}(e_i) = \mathcal{M}(c).$$

E being the set of terms of the decomposition of e_I under $\langle p_i \rangle_{1 \leq i \leq m}$.

Next, we want to take a look at an improvement of "differential" attack proposed in the same paper [dVL04]. The attack proceeds as follows. Given a ciphertext $c = e_I + M \cdot x^\beta = \sum_{j=1}^m h_j p_j + M \cdot x^\beta$, we first compute - for a term t_i occurring in a decomposition of the form $t_c = t_i \cdot t_{p_i}$, with t_c being a term of c and for some term t_{p_i} of p_i - a polynomial $c' = c - t_i p_i$. This polynomial can validate the choice of the guess (we do not know if t_i is really a term of h_i). Indeed, if $|c'| = |c| - |t_i p_i|$, then this can be taken as an evidence that t_i is a term of h_i . If this equality on the number of terms is not true, the polynomial c' can also be useful to reveal hidden monomials: if there exists a term t'_c in c' which is not a term of c , and which occurs in a decomposition of the form $t_{c'} = t'_j \cdot t_{p_j}$ for some term t_{p_j} of p_j (indeed, we then have $t_{c'} = t'_j t_{p_j} = t_i t'_{p_i}$) then, in addition to the fact that t_i is probably a term of h_i , it is also very likely that t_j was a term of h_j that was hidden in the ciphertext c . In all other cases, t_i is not a term of h_i , and we then set $c' = c$.

At the second step, we select a term $t_k \neq t_i$ having a decomposition of

the form $t_{c'} = t_k \cdot t_{p_k}$, with $t_{c'}$ a term of c' and for some term t_{p_k} of p_k . We compute $c'' = c' - t_k p_k$ and verify as previously whether t_k is a correct guess. We iterate this process while the simplified ciphertext is not a term of the form $a_\beta x^\beta$ (when it equals $a_\beta x^\beta$, then a_β is the plaintext corresponding to c). Notice that even if there are hidden monomials in the ciphertext, it is very likely that these monomials can supposedly be guessed by considering simplified ciphertext. As authors noted, the above attack is quite generic, and thus applies to 3-SAT Polly Cracker too. Although, lacking experimental results the authors failed to convince that all stages of the above iteration process go smoothly, at least "on average".

Remark 2.2.19. We observe two things. First, there is an example, when for $c' = c - t_i \cdot p_i$ there is a term $t_{c'}$ from c' , which is not in c , but in the decomposition $t_{c'} = t'_j \cdot t_{p_j}$, a term t_{p_j} is actually from p_i , and $t_i = -t'_j$. We certainly would like to avoid situation like that. So, some additional checks should be put in the attack. Second, at the second step it can be the case that different h_i 's have coinciding terms. So, choosing just $t_k \neq t_i$ could be unwary.

As a final remark we note that in [dVL04] it is stated that for $n = 700$ and $m = 2450$ it takes 4.3sec to generate a public key of size 6.9Kb, 3.22sec for encryption. The ciphertext has 1527 terms and decryption time is 0.135. We will see that Polly Two cryptosystem (see 2.3) offers less sizes of parameters than the above.

We would like to mention a paper by P.Ackermann and M.Kreuzer [PMar]. In this paper a rather general concept of Gröbner basis cryptosystem was introduced. The tools for the latter are generalizations of the Gröbner basis theory from polynomial rings to right and two-sided modules over free monoid rings (i.e. non-commutative polynomial rings) and modules over monoid rings. The setting the authors considered is very general, so more practical results on effectiveness of computations are needed. Public-key cryptosystems like Polly Cracker, Polly Two (section 2.3), non-commutative Polly Cracker (section 2.4), RSA, ElGamal turn out to be particular instances of the proposed Gröbner basis cryptosystem. The authors point out that this shift to modules and non-commutative algebras may prevent future instances of Gröbner basis cryptosystems from known attacks on Polly Cracker-type cryptosystems, in particular linear algebra attacks.

2.3 Polly Two cryptosystem

In this section we briefly describe Polly Two cryptosystem proposed by Le Van Ly in 2002 ([L.V02]). This system was designed primarily in order

to demonstrate that there exist a Polly Cracker-like cryptosystem robust to linear algebra attacks. The description of Polly Two is taken from an overview paper [L.V05].

2.3.1 Description of Polly Two

The background for construction Polly Two is to observe the following:

- The linear algebra attacks mainly exploit linear independence of monomials in $\mathbb{F}_q[x_1, \dots, x_n]$. Hence, generators should not be free so that relations among the monomials can exist.
- There are various ways to describe ideals. There is no need to represent them by their generators.
- In Polly Cracker the ciphertext strongly depends on the public key. But ciphertexts should look random, hence "more" encryption is necessary.

The notation we need is $P := \mathbb{F}_q[x_1, \dots, x_n], Q := \mathbb{F}_q[y_1, \dots, y_t]$; q is supposed to be a large power of a small prime number p (e.g. $p = 2$).

Given polynomials $g_1, \dots, g_t \in P$ of small degree we consider the subalgebra of P generated - in general not freely - by g_1, \dots, g_t and denote it by $R := \mathbb{F}_q[g_1, \dots, g_t]$. This algebra is the image of an algebra homomorphism $\phi : Q \rightarrow P, y_i \mapsto g_i, i = 1, \dots, t$. We denote the kernel of ϕ , which is a prime ideal, by $\mathfrak{p} := \ker \phi$. All these data are a part of the system and are public. Observe that we have:

$$R \cong Q/\mathfrak{p}.$$

We will identify monomials with elements in \mathbb{N}^n . Given $\alpha \in \mathbb{N}$ we call a polynomial α -sparse if $|Supp(f)| \leq \alpha$.

Having this notation we can present the Polly Two cryptosystem.

Key Generation: Alice chooses random sparse $\hat{f}_1, \dots, \hat{f}_s \in Q$ of the same total degree and sets $f_i := \phi(\hat{f}_i)$ for all i . She is able to construct the \hat{f}_i in such a way that she knows a point $\xi \in \mathbb{F}_q^n$ that is a root of every f_i and not a root of any g_i (see [L.V02], p.45).

Public Key: The set $\{\hat{f}_1, \dots, \hat{f}_s\} \subseteq Q$.

Private key: $\xi \in \mathbb{F}_q^n$.

Encryption: To encrypt a message $m \in \mathbb{F}_q$ Bob randomly chooses polynomials $\hat{h}_1, \dots, \hat{h}_s \in Q$ so that the sum $\hat{h} = \sum_{i=1}^s \hat{h}_i \hat{f}_i$ is sparse. Then Bob disguises the monomials of \hat{h} by a polynomial $h \in \mathfrak{p} \subseteq Q$ and obtains a polynomial

$$\tilde{c} := \hat{h} + h \in Q$$

such that $\phi(\hat{h}) = \phi(\tilde{c})$. From \tilde{c} Bob selects a monomial κ from the support of \tilde{c} and sends the ciphertext (c, κ) with $c = \tilde{c} + m \cdot y^\kappa$.

Decryption: To decrypt the message Alice evaluates

$$\frac{c(g_1(\xi), \dots, g_t(\xi))}{y^\kappa(g_1(\xi), \dots, g_t(\xi))} = \frac{\phi(c)(\xi)}{\phi(y^\kappa(\xi))} = m.$$

Alice recovers the correct message because $\phi(c) = \phi(\hat{h} + m \cdot y^\kappa)$ and all \hat{f}_i 's vanish at $\zeta = (g_1(\xi), \dots, g_t(\xi))$, since ξ is a zero of the f_i 's.

Now let us take a look at a toy example of Polly Two.

Example 2.3.1. We are working over the field \mathbb{F}_{17} . Let $t = 4, n = 2$. The mapping ϕ is given by the polynomials $g_i \in P = \mathbb{F}_{17}[x_1, x_2]$ as follows:

$$\begin{aligned} g_1 &= 13x_1^2 + 12x_1 + 13, g_2 = 12x_1x_2 + 2x_2 + 5, \\ g_3 &= 5x_1x_2 + 13, g_4 = 10x_1^2 + 10. \end{aligned}$$

The public key consists of the following polynomials $\hat{f}_i \in \mathbb{F}_{17}[y_1, \dots, y_4]$:

$$\hat{f}_1 = y_1y_2^3y_3y_4 + 3 \text{ and } \hat{f}_2 = y_1y_3^3y_4^2 + 4,$$

so that the corresponding polynomials $f_i = \phi(\hat{f}_i) = \hat{f}_i(g_1, \dots, g_4)$ are dense with large supports (33 and 28 elements resp.) and vanish at the private key $\zeta = (12, 16) \in \mathbb{F}_{17}$.

To encrypt the message $m = 13 \in \mathbb{F}_{17}$ we select polynomials $(\hat{h}_1, \hat{h}_2) = (10y_1^2y_3^3y_4, 7y_1^2y_2^3y_3)$ so that we get a sparse polynomial

$$\hat{h} = \hat{h}_1\hat{f}_1 + \hat{h}_2\hat{f}_2 = y_1^2y_2^3y_3 + 9y_1^2y_3^3y_4.$$

To cancel the monomials of \hat{h} we add the following polynomial

$$\begin{aligned} h &= 16y_1^2y_2^3y_3 + 14y_1^3y_2y_3^2 + 6y_1y_2^3y_3y_4 + 8y_1^2y_3^3y_4 + 8y_2^3y_4^2 + 15y_1^3y_3^2 \\ &\quad + 5y_1^2y_2y_3^2 + 16y_1^2y_3^3 + 11y_2^3y_3y_4 + 15y_1^2y_3^2y_4 + 9y_2^3y_3 + 5y_1^2y_3^2. \end{aligned}$$

from $\ker(\phi)$. We choose the monomial $y_2^3y_3$ in the support of $\tilde{c} = \hat{h} + h$ to be y^κ , and the resulting ciphertext corresponding to m is $(c, y_2^3y_3)$, where

$$\begin{aligned} c &= 14y_1^3y_2y_3^2 + 6y_1y_2^3y_3y_4 + 8y_2^3y_4^2 + 15y_1^3y_3^2 \\ &\quad + 5y_1^2y_2y_3^2 + 16y_1^2y_3^3 + 11y_2^3y_3y_4 + 15y_1^2y_3^2y_4 + 5y_2^3y_3 + 5y_1^2y_3^2. \end{aligned}$$

The decryption is simply the evaluation of c at $\zeta = (6, 12, 4, 5)$, i.e. $(c/y_2^3y_3)(\zeta) = 13$.

Recall that a polynomial h should be chosen in such a way that all monomials of \hat{h} change, so that the monomials of the \hat{h}_i 's do not "shine through" the ciphertext anymore. One way to do this, is to select h in such a way that it cancels all monomials in the support of \hat{h} . Clearly, in this situation it is necessary that the polynomial h is hard to guess. The crucial point is now that the ideal \mathfrak{p} is not given by generators, but by a mapping. This offers us many ways to construct various elements of the ideal \mathfrak{p} , so-called *relations*, so that a concrete h cannot easily be deduced from $\tilde{c} = \hat{h} + h$. Without going deeply into details, we only mention that there are several methods of obtaining such relations: a Gröbner-basis techniques and an approach by equations are some of them ([L.V05]).

2.3.2 Potential attacks on Polly Two

Attack on the private key. Let us denote $\mathfrak{a} := (f_1, \dots, f_s) \subseteq P$, $\mathfrak{b} := (\hat{f}_1, \dots, \hat{f}_s) \subseteq Q$. Similarly to Polly Cracker, finding an equivalent secret key means finding a zero of \mathfrak{a} . Since this means solving a system of algebraic equations with s equations in n variables, we should choose $s = n$ to provide that there are only few zeros. Moreover, in order to prevent a brute force attack, $|\mathbb{F}_q^n|$ should be greater than 2^{80} .

In contrast to Polly Cracker, it is not clear how to construct a Polly-Two instance from an NP-problem combinatorial problem, because the f_i 's emerge from the \hat{f}_i 's. So, in the case of Polly Two the security w.r.t the private key has to reside on the difficulty to solve a system of algebraic equations (f_1, \dots, f_s) coming from polynomials \hat{f}_i via a mapping ϕ . A straightforward way to find zeroes in this situation is as follows:

1. Compute $f_i \in P$ from $\hat{f}_i \in Q$ by applying ϕ , i.e., $f_i = \phi(\hat{f}_i)$.
2. Find a zero by calculating resultants or a Gröbner basis of the ideal \mathfrak{a}

As computation of resultants or Gröbner bases needs an amount of time that is exponential in n and in $\deg \phi(\hat{f}_i)$, this attack does not work if we choose these parameters sufficiently large. If we take, for instance, $n = 4$, $\deg \hat{f}_i = 2^7$, $\deg g_i = 2$, we have a system of algebraic equations of dense polynomials in 4 variables of degree 2^8 . With these parameters even step 1 is very time and space consuming as the supports of the polynomials are of magnitude $|Supp(f_i)| \approx 2^{28}$.

Remark 2.3.2. M.Grassl and R.Steinwandt (August 2004) were able to solve the Polly Two challenge (Challenge 1, see 2.3.3 for Challenge 2) from [L.V02] by computing a reduced Gröbner basis of the radical of \mathfrak{b} . They found out

that it contains 4 polynomials of degree 2. Substituting the y_i 's by the g_i 's they got a quite small system of algebraic equations, which they were able to solve.

Another approach to find an equivalent private key exploits the very special representation of $f_i = \phi(\hat{f}_i)$ as follows:

1. Compute the zero set $Z(\mathbf{b}) = \{\zeta \in \mathbb{F}_q^t : f(\zeta) = 0 \forall f \in \mathbf{b}\} \subseteq \mathbb{F}_q^t$.
2. For a zero $\zeta = (\zeta_1, \dots, \zeta_t) \in Z(\mathbf{b})$ test whether the equation system

$$g_1 - \zeta_1 = 0, \dots, g_t - \zeta_t = 0$$

over $\mathbb{F}_q[x_1, \dots, x_n]$ has a solution $\xi \in \mathbb{F}_q^n$. If yes, then ξ is a root of \mathbf{a} .

If we take $q = 2^{23}, n = s = 4, t \geq 8$, an attacker has to check more than about $|\mathbb{F}_q^4| = 2^{92}$ points in step 2. In [L.V05] it is shown that this is more than a brute-force attack.

Dense linear algebra attacks over P . To apply a linear attack over P an attacker has to consider the monomials of f_i 's on x_1, \dots, x_n . Since the degree of the \hat{f}_i 's is large and the g_i 's have small degree, almost every polynomial $f_i = \phi(\hat{f}_i)$ is a dense polynomial of large degree in P . Hence, an attacker has to apply a dense linear algebra attack and needs exponential time in the number of variables and the degree. If we set $\deg g_i = 2, n = s = 4, \deg \hat{f}_i = 2^7$, we obtain $\deg c = 2^9$. Hence an attacker has to consider a system of linear equations with about $\binom{2^9+4}{4} \approx 2^{32}$ equations (the number of monomials in the ciphertext, cf. Lemma 1.15, [L.V02]) and $4 \cdot \binom{2^8+4}{4} \approx 2^{30}$ unknowns (the number of monomials in all h_i). Since the complexity of Gaussian elimination is a little less than cubic in the number of equations and variables, we get a trivial lower time bound of $(2^{30})^{2.8} = 2^{84}$.

Linear algebra attacks over R . In Polly Two representation of elements in R is not unique. Hence a straightforward linear algebra attack over R fails. Nevertheless, as we do not exclude a situation when a Gröbner basis of the ideal \mathfrak{p} is computable, we get unique representation in $R \cong Q/\mathfrak{p}$ by reduction. More precisely, assume that G is a Gröbner basis of \mathfrak{p} with respect to a total degree order. Then the reduction $Red_G f$ of a polynomial $f \in Q$ w.r.t G is unique, and for a degree $e \in \mathbb{N}$ the vector space

$$N_G^e := \{Red_G f : \deg(Red_G f) \leq e, f \in Q\}$$

is isomorphic to the vector space $R_e := \{\bar{f} \in R : \exists f' \in Q \text{ such that } \deg f' \leq e, f' \equiv f \pmod{\mathfrak{p}}\}$. Hence, an attacker is able to set a linear algebra attack over N_G^d with $d := \deg Red_G c$.

In order to escape this attack we have to choose the g_i 's in the following way. If we select them so that the generators of \mathfrak{p} are not sparse reduction costs exponential time, as it destroys sparsity of the polynomials. Moreover, reduction does not evidently decrease degrees. Thus d is high and $\dim N_G^d$ is exponential in d , if we choose ϕ so that the vector space $N_G := \cup_{e=0}^{\infty} N_G^e$ is not finite dimensional. Fulfilling all these requirements we are able to assume that doing arithmetic in Q/\mathfrak{p} is highly inefficient, and therefore the preceding approach fails. In [L.V05] a concrete example of such parameters is given.

The intelligent linear attack over Q . Over Q the ciphertext decomposes to

$$c = \hat{h} + h + my^\kappa,$$

where $\hat{h} = \sum_{i=1}^s \hat{h}_i \hat{f}_i \in \mathfrak{b}$ and $h = \sum_{j=1}^r h_j r_j \in \mathfrak{p}$ with r_j sparse polynomials in \mathfrak{p} and $\hat{h}_i, h_j \in Q$. In order to set up a sparse linear attack an attacker needs to be able to guess at least the set of monomials $A := \cup_{i=1}^s \text{Supp}(\hat{h}_i)$. But in the second encryption step we create an element h so that κ and every monomial of \hat{h} changes, and we yield by construction

$$A \not\subseteq \{\mu - \nu : \mu \in \text{Supp}(c), \nu \in \text{Supp}(\hat{f}_i)\}$$

(here we identify monomials with exponents from \mathbb{N}^t) and with high probability even $A \cap \{\mu - \nu : \mu \in \text{Supp}(c), \nu \in \text{Supp}(\hat{f}_i)\} = \emptyset$ (see 2.2.1). The difference to original Polly Cracker is that we do not cancel the "treasonous" monomials by element of the ideal \mathfrak{b} , but by elements of the ideal \mathfrak{p} . The crucial fact is now that \mathfrak{b} and \mathfrak{p} are quite different in handling. See [L.V05] for comments on this issue.

Dense linear algebra attacks over Q . As an attacker is able to compute a generating set G of the ideal $\mathfrak{p} = \ker \phi$, he might try to set up a dense linear attack using the equation

$$c = \sum_{i=1}^s \hat{h}_i \hat{f}_i + \sum_{g \in G} h_g g + my^\kappa$$

with m and the coefficients of \hat{h}_i and h_g as unknowns. But again the number of equations and the number of unknowns are exponential in the parameters $\deg c$ and t . If we use $q = 2^{23}$, $n = s = 4$, $t = 11$, $\deg \hat{f}_i = \deg \hat{h}_i = 2^7$, then we have $\deg c = 2^8$. Hence, we get $\binom{2^8+11}{11} \approx 2^{63}$ equations and $|G+s| \cdot \binom{2^7+11}{11} \approx (76+4) \cdot 2^{52} \approx 2^{58}$ unknowns. As complexity of Gaussian elimination is cubic, we get a lower time bound of about 2^{162} .

2.3.3 Attack on Polly Two Challenge 2 and smearing of a plaintext

In [L.V05] Ly proposed a realistic example, which could be taken as a second attempt to propose a challenge (see above for Challenge 1). The challenge has the following parameter characteristics.

- **Domain parameters:** $t = 11$, and all g_i 's are binomials over $\mathbb{F}_{2^{23}}$.
- **Public key:** $n = s = 4$, all \hat{f}_i 's are trinomials of total degree 128.
- **Ciphertext:** c has 126 terms and has total degree 256 (intermediate ciphertext c'' has ≤ 6 terms).

At the Workshop on Algebraic Methods in Cryptography (Bochum, Germany, Nov. 2005) R.Steinwandt proposed a ciphertext-only attack on Challenge 2 ([R.S05]). The goal of this attack is to reconstruct the encryption step, without recovery of the secret (or equivalent) key. Below we will sketch the attack.

Observe that with some luck all terms of the $\ker(\phi)$ -elements, which cancel terms in $\sum \hat{h}_i \hat{f}_i$ should occur in c up to the canceled terms y^{μ_j} and a term involving y^κ . The attacker then tries to recover syzygies (relations among $\{g_1, \dots, g_{11}\} = \{\phi(y_1), \dots, \phi(y_{11})\}$, i.e. $\ker(\phi)$ -elements) "up to one term", so that he then can subtract them from c , thus simplify the ciphertext. The attacker assumes that likely construction for $\ker(\phi)$ -element used in encryption is to multiply low-degree syzygy with a term $a_j \cdot y^{\xi_j}$ for some a_j and η corresponding to y^{μ_j} (here y^{ξ_j} is supposed to be of high degree).

Then the attack proceeds as follows. To identify candidates for a monomial y^{ξ_j} , a term $\beta_\mu y^\mu$ of \hat{c} (where \hat{c} is obtained by ignoring y^κ -part of the ciphertext) is fixed and the following multiset is computed

$$A_\mu := \{\gcd(y^\mu, y^\pi) : y^\pi \neq y^\mu \text{ a monomial in } \hat{c}\}.$$

High multiplicity of an element in this multiset (say > 10) yields y^{ξ_j} -candidate. In the challenge there are 137 candidates for y^η . As we expect monomials y^{ξ_j} to be of high degree, we consider only maximal w.r.t divisibility elements in the set of all y^{ξ_j} -candidates. There are 22 maximal monomials in our set of 137 candidates. Then given such a maximal y^{ξ_j} -candidate y^η we can find the terms

$$\{\beta \cdot y^\sigma : \beta \cdot y^\sigma \text{ is a term of } \hat{c} \text{ divisible by } y^\eta\}.$$

Then summing (almost) all of these terms up should yield "a $\ker(\phi)$ -element up to one term". The next question is how to check whether a polynomial is a "syzygy up to one term". Denote the "trial syzygy" by r . In principle, we need to evaluate r at $g(x) = (g_1(x), \dots, g_{11}(x))$ and check whether

$r(g_1(x), \dots, g_{11}(x))$ is (up to a constant) a power product of the g_i 's (which supposedly corresponds to the canceled y^{ξ_j} term). In practice, we specialize some x_i 's to constants before trial division. In this way we in principle can find the canceled term too (and we can validate it through repeated evaluation). An iteration of this method reveals four syzygies. In order to find the fifth syzygy an attacker has to relax heuristics above and consider some non-maximal element y^n from the set of y^{ξ_j} -candidates. Subtracting these five syzygies from the original \hat{c} yields a simplified ciphertext with only 27 terms.

Now we need to recover secret terms \hat{h}_i (the attacker assumes that each \hat{h}_i consists of only one term). Applying "differential" attack (see 2.2.1) to the simplified \hat{c} yields only one term \hat{h}_2 . Another simple approach turns out to suffice. Namely, remaining (all except \hat{f}_2) public key polynomials contain a term with only two multiples in simplified \hat{c} . This yields a recovery of all secret terms \hat{h}_i . Subtracting found $\sum \hat{h}_i \hat{f}_i$ from simplified \hat{c} yields (short) polynomial that up to the term $-m \cdot y^\kappa$ is a syzygy. In this way the attacker recovers m .

As can be seen from above, the attack heavily relies on the fact that only one term containing y^κ is "spoiled" by adding the plaintext. We can bound feasibility of the above attack by smearing the plaintext among several terms of the ciphertext c . This can be done as follows. Suppose that we are able to construct $\hat{f}_1, \dots, \hat{f}_s$ such that they have ξ_1, \dots, ξ_l as common zeros, which are not the zeros of any of g_i (see below for such a construction). Let m_1, \dots, m_l be (pieces of) the plaintext. Note that "Encryption" step does not depend on ξ in the original setting, so does in ours. Now having m_1, \dots, m_l we encrypt $c = \tilde{c} + m_1 y^{\kappa_1} + \dots + m_l y^{\kappa_l}$, where $y^{\kappa_i} \in \text{Supp}(\tilde{c}), i = 1, \dots, l$. So the ciphertext now is $(c; \kappa_1, \dots, \kappa_l)$. For decryption we need to solve a system of linear equations:

$$\begin{cases} m_1 \cdot a_{11} + \dots + m_l \cdot a_{1l} = c_1, \\ \vdots \\ m_1 \cdot a_{l1} + \dots + m_l \cdot a_{ll} = c_l. \end{cases}$$

where $a_{ij} = \phi(y^{\kappa_j})(\xi_i)$, and $c_i = \phi(c)(\xi_i)$, m_i are unknowns, $i, j = 1, \dots, l$. This system obviously has a unique solution which we take as a decrypted ciphertext.

The question to discuss now is how to produce $\hat{f}_1, \dots, \hat{f}_s$ such that they have ξ_1, \dots, ξ_l as common zeroes. The key generation procedure from [L.V05] (compare with Algorithm 3.58 from [L.V02]):

1. Select a random point $\xi \in \mathbb{F}^n$. Compute the corresponding point $\zeta = (g_1(\xi), \dots, g_t(\xi)) \in \mathbb{F}^t$, so that $\zeta \neq (0, \dots, 0)$.

2. Choose random exponents $\mu_1, \dots, \mu_s, \nu_1, \dots, \nu_s \in \mathbb{N}^t$ of degree d . Also choose random $\alpha_1, \dots, \alpha_s \in \mathbb{F}$. Set $\beta_i := (y^{\nu_i} + \alpha_i y^{\mu_i})(\zeta)$, then put $\hat{f}_i = y^{\nu_i} + \alpha_i y^{\mu_i} + \beta_i \in Q$ for all i .
3. The public key is the set $\{\hat{f}_1, \dots, \hat{f}_s\}$ and the private key the point ξ .

For our needs we modify the procedure as follows:

1. Select random points $\xi_1, \dots, \xi_l \in \mathbb{F}^n$. Compute the corresponding points $\zeta_i = (g_1(\xi_i), \dots, g_t(\xi_i)) \in \mathbb{F}^t, i = 1, \dots, l$, so that $\zeta_i \neq (0, \dots, 0)$ for all i .
2. Choose random exponents $\mu_{ij}, \nu_{ij} \in \mathbb{N}^t, i = 1, \dots, s, j = 1, \dots, l$ of degree d . Also choose random $\alpha_{ij} \in \mathbb{F}$. Set $\beta_{ij} := (y^{\nu_{ij}} + \alpha_{ij} y^{\mu_{ij}})(\zeta_j)$, then put $\hat{f}_i = \prod_{j=1}^l (y^{\nu_{ij}} + \alpha_{ij} y^{\mu_{ij}} + \beta_{ij}) \in Q$.
3. The public key is the set $\{\hat{f}_1, \dots, \hat{f}_s\}$ and the private key the points ξ_1, \dots, ξ_l .

In this case instead of 3-sparsity (as in the original setting) we get 3^l -sparsity. So, obviously, the size of the public key increases. Of course, in order to really have "sparsity" we should not take l too large.

What we gain from this approach are the following two things:

1. The message size can be increased in l times.
2. By "smearing" the plaintext m_1, \dots, m_l among l terms of the ciphertext we reduce vulnerability of the Polly Two scheme to different attacks that aim at recovering terms of the polynomial h .

Remark 2.3.3. We may choose that some m_i 's do not carry actual information. Instead, we can have some m_i 's equal to random values. We even can arrange that only m_1 carries actual plaintext and other m_i 's are random or some hashed values of m_1 . This can be done in order to dodge the situation, when for some reason an attacker was able to recover some pieces m_i of the plaintext.

Still R.Steinwandt claims that this protective mechanism can be overcome by some sparse interpolation techniques.

2.4 Noncommutative Polly Cracker cryptosystem

2.4.1 Preliminaries on noncommutative Gröbner bases and noncommutative Polly Cracker

The noncommutative Polly Cracker cryptosystems were developed by T.Rai in his Ph.D. dissertation ([T.R04]), and rely on the fact that most ideals of noncommutative algebras over finite fields have infinite reduced Gröbner bases.

First let us briefly present notations that will be used further in the text (compare the following definitions with those from 2.2.2). Everything in this subsection is based on [T.R04]. We will be working with a noncommutative algebra $\mathbb{F}_q \langle X \rangle$, where $X = \{x_1, \dots, x_n\}$, which is an algebra of noncommutative polynomials. By a monomial, we mean a finite noncommutative word in the alphabet X . We use the letter B to denote the set of monomials. We define multiplication in the set B of monomials by concatenation. The next important thing is the notion of an admissible ordering. A well-ordering $>$ on B is said to be *admissible* if it satisfies the following conditions for all $p, q, r, s \in B$:

- if $p < q$ then $pr < qr$;
- if $p < q$ then $sp < sq$;
- if $p = qr$ then $p > q$ and $p > r$.

Let $>$ be an admissible ordering on the monomials and $f \in \mathbb{F}_q \langle X \rangle$. We say that a monomial b_i occurs in f if the coefficient of b_i in $f = \sum \alpha_i b_i$ is not zero. We say that b_i is the *tip* of f , denoted $tip(f)$, if b_i occurs in f and $b_i > b_j$ for all b_j occurring in f . We denote the coefficient of $tip(f)$ by $Ctip(f)$. If $S \subseteq \mathbb{F}_q \langle X \rangle$, then we write $Tip(S) = \{b \in B : b = tip(f) \text{ for some nonzero } f \in S\}$ and $NonTip(S) = B \setminus Tip(S)$.

Another thing we need is the notion of division of a polynomial $g \in \mathbb{F}_q \langle X \rangle$ by polynomials $f_1, \dots, f_k \in \mathbb{F}_q \langle X \rangle$. To perform such a division means to find nonnegative integers t_1, t_2, \dots, t_k and elements $u_{ij}, v_{ij}, r \in \mathbb{F}_q \langle X \rangle$, for $1 \leq i \leq k$ and $1 \leq j \leq t_i$ such that:

- $g = \sum_{i=1}^k \sum_{j=1}^{t_i} u_{ij} f_i v_{ij} + r$;
- $tip(g) \geq tip(u_{ij} f_i v_{ij})$ for all i and j ;
- $tip(f_i)$ does not divide any monomial that occurs in r , for $1 \leq i \leq k$.

Note that if $r \neq 0$, then $tip(r) \leq tip(g)$; r is the *remainder* of the division.

Definition 2.4.1. Let $>$ be an admissible order on $K \langle x_1, \dots, x_n \rangle$, and I is a two-sided ideal of $K \langle x_1, \dots, x_n \rangle$. We say that $G \subset I$ is a *Gröbner basis* for I w.r.t $>$ if $\langle Tip(G) \rangle = \langle Tip(I) \rangle$. Equivalently, $G \subset I$ is a Gröbner basis of I if for every $b \in Tip(I)$, there is some $g \in G$ such that $tip(g)$ divides b , i.e. for every $f \in I$ there exists $g \in G$, and $p, q \in B$ such that $p \cdot tip(g) \cdot q = tip(f)$.

For any ideal I we have:

$$K \langle x_1, \dots, x_n \rangle = I \oplus Span(NonTip(I)),$$

as vector spaces. In particular, every $f \in K \langle x_1, \dots, x_n \rangle \setminus \{0\}$ can be written uniquely as $f = f_I + N_I(f)$, where $f_I \in I$ and $N_I(f) \in Span(NonTip(I))$ (recall division with remainder). $N_I(f)$ is called the *normal form of f w.r.t I* .

Definition 2.4.2. Let $I \subset K \langle x_1, \dots, x_n \rangle$ be an ideal, let $I_{MON} = \langle Tip(I) \rangle$, and let T be the unique minimal monomial generating set² of I_{MON} . Then the *reduced Gröbner basis* for I is $G = \{t - N(t) : t \in T\}$

Reduced Gröbner basis as above possesses usual properties of "reducedness" that we expect:

1. G is a Gröbner basis for I .
2. If $g \in G$, then the coefficient of $tip(g)$ is 1.
3. If $g_i, g_j \in G$ with $g_i \neq g_j$, and b_i is any monomial that occurs in g_i , then $tip(g_j)$ does not divide b_i .
4. $g \in G$, then $g - tip(g) \in Span(NonTip(I))$.
5. $Tip(G)$ is the minimal monomial generating set of I_{MON} .

The reduced Gröbner basis may not be finite. This is one of the background points for the noncommutative Polly Cracker. Also we note, that as in the commutative case the order of f_1, \dots, f_k in the division procedure affects the remainder r , but if $G = \{f_1, \dots, f_k\}$ is a Gröbner basis, then r does not depend on the order of f_1, \dots, f_k .

Now we present the noncommutative Polly Cracker from [T.R04]. It can be summarized as follows.

Private Key: A Gröbner basis, $G = \{g_1, \dots, g_t\}$ for a two-sided ideal I of a noncommutative algebra $\mathbb{F}_q \langle X \rangle$ over a finite field \mathbb{F}_q of q elements.

²It does exist and it is unique, but it need not be finite

Public Key: A set, $Q = \{q_r : q_r = \sum_{i=1}^t \sum_{j=1}^{d_{ir}} f_{rij} g_i h_{rij}\}_{r=1}^s \subseteq I$, chosen so that computing a Gröbner basis of $\langle Q \rangle$ is infeasible (we even can demand it to be infinite).

Message Space: $M = \text{Span}(\text{NonTip}(I))$ or a subset of $\text{Span}(\text{NonTip}(I))$.

Encryption: $c = p + m$, where $m \in M$ is a message and $p = \sum_{i=1}^s \sum_{j=1}^{k_i} F_{ij} q_i H_{ij}$ is a polynomial in $J = \langle Q \rangle \subseteq I$.

Decryption: Reduction of c modulo G yields the message, m .

For practical reasons T.Rai proposes to use G containing only one element g . We cite some examples of such cryptosystems from [T.R04].

Example 2.4.3. Let $K \langle x_1, \dots, x_6 \rangle$ be a free algebra over a finite field K in six non-commuting variables. Let $Z = \prod_{i=1}^6 x_i$ and $c_0, \dots, c_6 \in K \setminus \{0\}$ be arbitrary constants. Set $g := Z + \sum_{i=1}^6 c_i x_i + c_0 \in K \langle x_1, \dots, x_6 \rangle$ as the private key. The public key $Q = \{q_1, q_2\}$ consists of the polynomials $q_1 = fgh + hg$, $q_2 = hgf + gh$, where $f = X + \sum_{i=1}^6 a_i x_i + a_0$, $h = Y + \sum_{i=1}^6 b_i x_i + b_0 \in K \langle x_1, \dots, x_6 \rangle$, $X = x_1 \cdot \prod_{i=2}^5 \rho(x_i) \cdot x_6$, $Y = x_1 \cdot \prod_{i=2}^5 \sigma(x_i) \cdot x_6$, where ρ, σ are distinct, nontrivial permutations of $\{x_2, \dots, x_5\}$, and $a_0, \dots, a_6, b_0, \dots, b_6 \in K \setminus \{0\}$ are constants. In this setting, the message space $M \subset \text{NonTip}(\langle g \rangle)$ could consist of linear polynomials in $K \langle x_1, \dots, x_6 \rangle$

Note that here $\langle Q \rangle = \{q_1, q_2\}$ does not have a finite Gröbner basis for any admissible order.

Example 2.4.4. Let $K \langle x, y \rangle$ be a free algebra over a finite field K in two non-commuting variables. Let $\alpha, \beta, \gamma, \delta \in K \setminus \{0\}$, and set $g := \alpha xy + \beta x + \gamma y + \delta$ as the private key. We refer to [T.R04] for expressions of the public key polynomials. As in the previous example, the message space $M \subset \text{NonTip}(\langle g \rangle)$ could consist of linear polynomials.

In [T.R04] the author claims that his cryptosystem should not be susceptible to linear algebra attacks. This is due to the fact that encryption is achieved by $c = p + m$, where $\{q_j\}_{j=1}^s$ is the public key, $p = \sum_{i=1}^s \sum_{j=1}^{k_{ir}} F_{rij} q_i H_{rij}$, and m is the message. As a result, the coefficient of any monomial W that occurs in c is quadratic in the coefficients of $F_{rij}, H_{rij}, i = 1, \dots, s, j = 1, \dots, k_{ir}$. So if we treat the coefficients of $F_{rij}, H_{rij}, i = 1, \dots, s, j = 1, \dots, k_{ir}$, as unknowns, we get a non-linear system of equations, which is far harder to solve than a linear one.

However, as noted in [T.R04], a ‘‘linear algebra type’’ attack might be successful in the case of poorly constructed ciphertext. This attack, which we describe below, is a blend of the linear algebra attack on the commutative Polly Cracker cryptosystem and a failed cryptanalysis of Patarin’s

Little Dragon that is described in Koblitz [N.K98]. Let c be the ciphertext polynomial. Then $c = \sum_{i=1}^s \sum_{j=1}^{k_{ir}} F_{rij} q_i H_{rij} + m$ and the coefficient of any monomial W that occurs in c is quadratic in the coefficients of $F_{rij}, H_{rij}, i = 1, \dots, s, j = 1, \dots, k_{ir}$. Thus, by treating the coefficients of the polynomials $F_{rij}, H_{rij}, i = 1, \dots, s, j = 1, \dots, k_{ir}$ as unknowns, we get a system of quadratic equations. Now, the coefficient of any monomial W in c consists of sums of constants of the form $\alpha_i \gamma_j \beta_k$ and m_W , where α_i is the coefficient of a monomial that occurs in some F_{rij} , γ_j is the coefficient of a monomial that occurs in some q_i , β_k is the coefficient of a monomial that occurs in some H_{rij} , and m_W is the coefficient of W in m . Next, we introduce new variables δ_l for all the products $\alpha_i \beta_k$ that appear in the coefficient of W , i.e. we replace each product of the form $\alpha_i \beta_k$ with a new variable δ_l . This results in a system of linear equations in the variables δ_l and m_W . If the number of such equations equals (or exceeds) the number of such unknowns, it has a unique solution that can be determined using Gaussian elimination. Since the cryptanalyst is only interested in the coefficients of the terms that occur in m , solving this system of equations would serve his/her purpose.

In order to ensure that a noncommutative Polly Cracker cryptosystem is not vulnerable to such an attack, we need to ensure that the number of the products $\alpha_i \beta_k$ that occur in c exceeds the number of such equations. Since each monomial in c contributes a set of equations, one way of ensuring this is to make k_{ir} sufficiently large for each i . In [T.R04] there is some concrete example on how one could do this.

It is still unclear whether other more intelligent linear algebra attacks could brake the noncommutative Polly Cracker. Also, it could be an interesting question to perform an attack similar to one performed by R.Steinwandt on Polly Two Challenge 2 (see 2.3.3). More study should be done in this direction.

2.4.2 Chosen-ciphertext attacks

In [RW02] and [VR04] it was shown that (commutative) Polly Cracker and its various modifications are susceptible to chosen ciphertext attacks. We have seen one such attack from [RW02]. We will now show that noncommutative Polly Cracker is also susceptible to a chosen-ciphertext attack. In fact, we will only need one "fake" ciphertext in order to be able to decrypt all further ciphertexts correctly. The attack is from [S.B05].

In the following we assume that we know the form of g (e.g. $g = \alpha xy + \beta x + \gamma y + \delta$, where $\alpha, \beta, \gamma, \delta \in \mathbb{F}_q$, cf. e.g section 5.1.3 of [T.R04], also Examples 2.4.1, 2.4.2).

The main idea relies on the following observation. Let $I = \langle g \rangle$, and

consider $tip(g)$. We have:

$$tip(g) = Ctip(g)^{-1}g - Ctip(g)^{-1} \cdot tail(g),$$

where $tail(g) = g - Ctip(g) \cdot tip(g)$. Note, that $tip(g)$ does not divide any monomial in $tail(g)$. This means that $-Ctip(g)^{-1} \cdot tail(g)$ is the remainder of division of $tip(g)$ by g , or equivalently, it is the result of decryption of the "fake" ciphertext $tip(g)$.

Now, let us go on to the chosen-ciphertext attack itself. Let us construct a "fake" ciphertext $c' = t \cdot tip(g) \cdot s + \sum F_{ij}q_iH_{ij}$, where $t, s \in \mathbb{F}_q \langle X \rangle$ are such that any monomial of $t \cdot tail(g) \cdot s$ is not divisible by $tip(g)$. Polynomials t and s are chosen for masking the "fake" ciphertext and, in principle, can be dropped out. We have:

$$t \cdot tip(g) \cdot s = Ctip(g)^{-1}t \cdot g \cdot s - Ctip(g)^{-1}t \cdot tail(g) \cdot s,$$

and using this latter assumption we obtain that $-Ctip(g)^{-1}t \cdot tail(g) \cdot s$ is the remainder of division of $t \cdot tip(g) \cdot s$ by g , and thus this is the remainder of division of c' by g (as $\sum F_{ij}q_iH_{ij}$ reduces to 0 modulo $G = \{g\}$).

A next simple example shows that requirements on t and s can be easily satisfied. For instance, let us take $g = x_1 \cdots x_6 + c_1x_1 + \cdots + c_6x_6 + c_0, c_0, \dots, c_6 \in \mathbb{F}_q \setminus \{0\}$ as in section 5.1.2 of [T.R04]. Then $tail(g) = c_1x_1 + \cdots + c_6x_6 + c_0$ (under any admissible ordering) and we can take $t := x_2x_4 + x_2x_3x_6 + x_4x_1x_5; s := x_5x_1x_3 + x_6x_2x_4$. One easily sees that no monomial of $t \cdot tail(g) \cdot s$ is divisible by $tip(g) = x_1 \cdots x_6$. It is also clear that many more variants of t and s can be proposed.

So, going back to our construction we see that if we send a "ciphertext" c' , we obtain a "plaintext" $p' = -Ctip(g)^{-1}t \cdot tail(g) \cdot s$. We know t and s , so we can easily deduce $-Ctip(g)^{-1} \cdot tail(g)$ from p' . Now construct $g' = tip(g) + Ctip(g)^{-1} \cdot tail(g)$. We have $Ctip(g) \cdot g' = g$, so $I = \langle g \rangle = \langle g' \rangle$, and thus we can use g' in order to decrypt ciphertexts to correct plaintexts, which is equivalent to knowing the private key $G = \{g\}$. Indeed, if for a ciphertext c we had $c = g_1 \cdot g \cdot g_2 + r$, where r is the remainder, then for the same ciphertext we have $c = Ctip(g)g_1 \cdot g' \cdot g_2 + r$, where r is again the remainder, and it coincides with the remainder of division of c by the initial g .

For even more confusion for decrypting system we may send "fake" ciphertext of the form $c'' = c' + h$, where c' is as above, and $h \in \mathbb{F}_q \langle X \rangle$, such that $tip(g)$ does not divide any monomial in h . Note that such polynomials h "incorporate" monomials from $NonTip(I)$, i.e. valid messages. A "plaintext" corresponding to c'' will be $p' + h$, which again gives rise to g' as above. So, in our attack the "fake" ciphertext c'' contains either monomials

divisible by $tip(g)$ and non-divisible. In addition, we note that the variety of such c'' 's is very broad.

Note that right from the definition of a reduced Gröbner basis we get that also private keys of the form $G = \{g_1, \dots, g_s\}$, where G is the reduced Gröbner basis for I are also susceptible to such an attack. It proceeds as follows. We start with a "fake" ciphertext $c_1 = \sum_{i=1}^s \sum_{j=1}^{k_{ir}} F_{rij} q_i H_{rij} + tip(g_1)$ as in the previous attack (without masking t and s , see below). As we have already seen c_1 decrypts to $-Ctip(g_1)^{-1} \cdot tail(g_1)$. This is justified by the fact that G is reduced, so none of the $tip(g_2), \dots, tip(g_s)$ divides any monomial in $tail(g_1)$. As before, we construct $g'_1 = tip(g_1) + Ctip(g_1)^{-1} \cdot tail(g_1)$. Then we repeat this procedure for all $i = 2, \dots, s$, and obtain $G' = \{g'_1, \dots, g'_s\}$, where $g'_i = tip(g_i) + Ctip(g_i)^{-1} \cdot tail(g_i), i = 1, \dots, s$. Again as in the attack above, $\langle G \rangle = \langle G' \rangle$, and G' is a Gröbner basis for $\langle G \rangle$. These follow from $Ctip(g_i) \cdot g'_i = g_i, i = 1, \dots, s$. So again we are able to decrypt all messages with $\langle G' \rangle$.

One may also perform a disguising procedure, similar to one above for $G = \langle g \rangle$. For a given $tip(g_i)$, we can choose polynomials t_i and s_i such that $tip(g_j)$ does not divide any monomial occurring in $t_i \cdot tail(g_i) \cdot s_i$, for any $j \neq i$. We can then send $c'_i = \sum_{a=1}^s \sum_{b=1}^{k_{ar}} F_{rab} q_a H_{rab} + t_i \cdot tip(g_i) \cdot s_i$. Proceeding as above, we obtain a "plaintext" $f'_i = -Ctip(g_i)^{-1} t_i \cdot tail(g_i) \cdot s_i$. And again, as before, we conclude with $G' = \{g'_1, \dots, g'_s\}$. Note that such t_i and s_i could exist in theory.

One may think that using non-reduced G could be an escape from such an attack. But, in [T.R05] it was shown that a modified version of the above attacks can also be applied to non-reduced Gröbner bases. In the described attacks we assumed that we knew $Tip(G)$, which is quite a reasonable assumption, especially considering the absence of a broad set of suitable instances for noncommutative Polly Cracker. In [T.R05] it was also shown that one does not need to know $Tip(G)$ in order to perform a chosen-ciphertext attack. One instead needs to know a monomial order used during the decryption.

2.4.3 Countermeasures

As an answer to our chosen-ciphertext attack T.Rai in his paper [T.R05] presents countermeasures against such attacks as described in 2.4.2.

Ironically, the countermeasures presented in [T.R05] are as simple as the attacks we discussed above. The following procedure ensures that Alice's decryption device identifies "fake" ciphertexts of the type described above in the attacks. Namely

1. Restrict the message space M , such that $M \not\subseteq NonTip(G)$.

2. Ensure that at least one monomial b_i occurs in each $g_i \in G$, such that $b_i \in \text{NonTip}(G) \setminus M$, and $u \cdot b_i \cdot vM$, for all $u, v \in B$.
3. Program the decryption algorithm to check whether any element of $\text{NonTip}(G) \setminus M$ occurs in a normal form of a ciphertext polynomial after it has been reduced modulo the private key.
4. If the decryption algorithm encounters an element of $\text{NonTip}(G) \setminus M$ in the normal form of a ciphertext polynomial, program it to return an error message (or the original ciphertext polynomial without reduction).

For instance, in Example 2.4.4 $g := \alpha xy + \beta x + \gamma y + \delta$, and the message space could be taken to be linear polynomials in y . So that if Alice encounters an x term in the normal form, she returns an error, rather than a reduced element. Similarly, in Example 2.4.3 $g := Z + \sum_{i=1}^6 c_i x_i + c_0 \in K \langle x_1, \dots, x_6 \rangle$, and the message space could be taken to be e.g. polynomials in x_2, \dots, x_6 form $\text{NonTip}(G)$. So that if Alice encounters an x_1 term in the normal form, she returns an error.

Now, in the attacks we had a "ciphertext" of the form $c = \text{tip}(g) + \sum F_{ij} q_i H_{ij}$ (without masking) and the corresponding "plaintext" $p = -C \text{tip}(g)^{-1} \cdot \text{tail}(g)$. Now, since there is at least one monomial $b \in \text{NonTip}(G) \setminus M$, which occurs in g , then it will also occur in p . So, Alice will detect b , and return an error. Note, that countermeasure (2) above detects also "fake" ciphertexts that use masking.

Remark 2.4.5. Let us ones more take a look at Example 2.4.4, where $g := \alpha xy + \beta x + \gamma y + \delta$. We have

$$\text{tip}(g) + \alpha^{-1} \beta x = \alpha^{-1} g - \alpha^{-1} (\text{tail}(g) - \beta x).$$

Note, that $\text{tail}(g) - \beta x$ does not contain an x term any more. So, the "brute-force" attack here could be to send $c_\xi = \text{tip}(g) + \xi x + \sum F_{ij} q_i H_{ij}$, $\xi \in K$ until the decryption device drops out a reduced message, rather than an error. This would imply that $\xi = \alpha^{-1} \beta$, and we proceed as in the previous attacks. Of course, if the size of K is large this attack is impractical. Nevertheless, observe that breaking a system with the private key in four unknown coefficients (or with three up to equivalence) reduced to finding just one parameter ξ . Clearly, we can increase the number of parameters needed for the attack by "forbidding" more monomials in g_i 's, and thus by reducing the message space size. So, some problems still do exist here. Primarily this is due to a lack of the broad list of instances for the noncommutative Polly Cracker, as we have already mentioned earlier. We believe that new developments in the theory of noncommutative Gröbner bases could fix this lack.

References

- [A.B] A.Brouwer, *Bounds on the minimum distance of linear codes*, <http://www.win.tue.nl/~aeb/voorlincod.html>.
- [AH91] A.Garcia and H.Stichtenoth, *Elementary abelian p -extensions of algebraic function fields*, *Manuscr. Math.* **72** (1991), 67–79.
- [AH99] ———, *A class of polynomials over finite fields*, *Finite Fields and Their Applications* **5** (1999), 424–435.
- [AKC05] A.Frühbis-Krüger and C.Lossen, *Introduction to computer algebra (solving systems of polynomial equations)*, 2005.
- [AvOS96] A.Menezes, P. van Oorschot, and S.Vanstone, *Handbook of applied cryptography*, CRC Press, 1996.
- [BCE⁺94] Boo Barkee, Deh Cac Can, Julia Ecks, Theo Moriarty, and R.F.Ree, *Why you cannot even hope yo use Gröbner bases in public key cryptography: An open letetr to a scientists who failed and a challene to those who have not yet failed*, *Journal Symbolic Computations* (1994), no. 18, 497–501.
- [CR95] C.Kirfel and R.Pellikaan, *The minimum distance of codes in an array coming from telescopic semigroups*, *IEEE Trans. on Inform. Theory* **41** (1995), no. 6, 1720–1732.
- [DR02] D.Hofheinz and R.Steinwandt, *A "Differential" attack on Polly Cracker*, *Proceedings of 2002 IEEE International Symposium on Information Theory ISIT 2002*, extended abstract, 2002, p. 211.
- [dVL04] F.Levy dit Vehel and L.Perret, *A Polly Cracker system based on satisfiability*, *Progress in Computer Science and Applied Logic* (2004), no. 23, 177–192.
- [Fau99] J.-C. Faugère, *A new efficient algorithm for computing Gröbner bases (F_4)*, *J. Pure Appl. Algebra* **139** (1999), no. 1-3, 61–88.
- [GAP02] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.3*, 2002, <http://www.gap-system.org>.
- [GMG02] G.-M.Greuel and G.Pfister, *A SINGULAR introduction to commutative algebra*, Springer-Verlag, 2002.

- [GPS05] G.-M. Greuel, G. Pfister, and H. Schönemann, *SINGULAR 3.0*, A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern, 2005, <http://www.singular.uni-kl.de>.
- [L.V02] L.V.Ly, *Polly Two - a public-key cryptosystem based on Polly Cracker*, Ph.D. thesis, Ruhr-Universität, Bochum, Germany, october 2002, <http://www-brs.ub.ruhr-uni-bochum.de/netahtml/HSS/Diss/LyLeVan/diss.pdf>.
- [L.V05] ———, *Polly Two - a new algebraic polynomial-based public-key scheme*, AAEECC (2005), to appear.
- [ML05] J. Martin and C. Lossen, *brnoeth.lib. A SINGULAR 3.0 library for computing Brill-Noether Algorithm, Weierstrass-SG and AG-codes*, A computer algebra system for polynomial computations, Centre for Computer Algebra, University of Kaiserslautern, 2005, <http://www.singular.uni-kl.de>.
- [MN93] M.Fellows and N.Koblitz, *Combinatorial systems galore!*, Finite fields: theory, applications and algorithms, 1993.
- [N.K98] N.Koblitz, *Algebraic aspects of cryptography*, Algorithms and Computation in Mathematics, 3, Springer-Verlag, Berlin, 1998.
- [PMar] P.Ackermann and M.Kreuzer, *Gröbner basis cryptosystems*, AAEECC (2005, to appear).
- [R.S05] R.Steinwandt, *A ciphertext-only attack on Polly Two*, manuscript (2005).
- [RW02] R.Steinwandt and W.Geiselmann, *Cryptanalysis of Polly Cracker*, IEEE Trans. on Inform Theory **48** (2002), no. 11, 2990–2991.
- [S.B04] S.Bulygin, *On some applications of algebraic function fields to the problems of coding theory*, Bachelor's thesis, Kyiv National University, Kyiv, Ukraine, 2004.
- [S.B05] ———, *Chosen-ciphertext on noncommutative Polly Cracker*, Los Alamos XXX eprint archive, 2005, <http://xxx.lanl.gov/abs/cs.IT/0508015>.
- [Sti93] H. Stichtenoth, *Algebraic function fields and codes*, Springer-Verlag, Berlin-Heidelberg-New York, 1993.

- [T.R04] T.Rai, *Infinite Gröbner bases and noncommutative Polly Cracker cryptosystem*, Ph.D. thesis, Virginia Tech, Blacksburg, VA, USA, 2004.
- [T.R05] ———, *Countering chosen-ciphertext attacks against noncommutative polly cracker-type cryptosystems*, Cryptology ePrint Archive:Report 2005/344, 2005, <http://eprint.iacr.org/2005/344>.
- [TvLR98] T.Hoholdt, J.H. van Lint, and R.Pellikaan, *Algebraic geometry codes*, Handbook of Coding Theory (V.S.Pless, W.C.Huffman, and R.A.Brualdi, eds.), Elsevier, Amsterdam, 1998.
- [vL92] J.H. van Lint, *Introduction to coding theory*, Springer-Verlag, Berlin-Heidelberg-New York, 1992.
- [VR04] M.I.G. Vasco and R.Steinwandt, *Chosen ciphertext attacks as common vulnerability of some group- and polynomial-based encryption schemes*, WartaCrypt, 2004.
- [WM76] W.Diffie and M.E.Hellamn, *New directions in cryptography*, IEEE Trans. on Inform. Theory **22** (1976), 644–654.