

Gröbner bases computations and cryptographic implications

Le Van Ly
Fakultät für Mathematik
Ruhr-Universität Bochum

26th April 2001

Abstract

The question of security of some algebraic cryptosystems is highly related with the complexity of Gröbner bases computations. Therefore research in this framework is of main interest in cryptology.

In the first chapter we give a brief description of the Gröbner base theory in the commutative case. The main result in this context is that there exists an effective way to decide the Ideal Membership Problem by the so called Buchberger algorithm, introduced in the year 1965 by Bruno Buchberger [Buc65]. The second chapter considers the noncommutative case of the Gröbner bases theory, and we describe a method to compute such a basis in the noncommutative case. Although both constructions - commutative and noncommutative Gröbner bases- are quite analogous and we have similar results, there are crucial differences, for instance the question of termination, which we specify in the last part of the second chapter. In the third chapter we give some cryptographic implications of Gröbner bases computations, that is, how Gröbner bases can be used in cryptography for a Public Key Cryptosystem or for a attack on HFE. The fourth chapter circumstantiates the fact that Gröbner bases computations are hard by some theoretical results and some experimental computations. In the last section we give a brief outlook, where research can be done and which questions are of further interest.

Contents

1	Gröbner bases in the commutative case	3
1.1	Monomial Orders and Polynomial Reductions	3
1.2	Gröbner bases - existence and uniqueness	6
1.3	The Buchberger algorithm	7
2	Gröbner bases in the noncommutative case	9
2.1	Gröbner bases of two sided ideals	9
2.1.1	Basic Notation. Degree and Order	9
2.1.2	Normal words	10
2.1.3	Gröbner Basis	10
2.1.4	Reduction and Composition.	11
2.1.5	Examples	11
2.2	Gröbner bases of one sided ideals	12
2.3	Comparing the commutative and the noncommutative case	13
3	Cryptographic implications of Gröbner bases computations	14
3.1	Koblitz: Polly Cracker, a Cryptosystem using Gröbner bases directly.	14
3.2	Patarin and HFE: Why HFE does not work, if one can compute Gröbner bases efficiently.	15
3.3	Wagner, Magyarik: A Public Key Cryptosystem based on the Word Problem	17
3.4	A Way to use noncommutative Gröbner bases in Public Key Cryptography	19
4	The complexity of Gröbner bases computations	20
4.1	Theoretical results	20
4.1.1	The Ideal Membership Problem	20
4.1.2	The Ideal Membership Problem for two subclasses of polynomial ideals	21
4.2	Experimental results: The algorithm of Faugère	22
5	Outlook, Questions, Research	29

1 Gröbner bases in the commutative case

Introduction. Suppose first we are given univariate polynomials f, g_1, \dots, g_m over a field, and wish to decide whether f is in the ideal generated by the g_i . This can be done by computing the gcd g of the g_i (Euclidean algorithm) and then perform long division of f by g . The polynomial f will lie in the ideal in question if and only if the remainder of this division equals zero. Moreover if this is the case then one also obtains a polynomial q that satisfies $f = qg$.

Gröbner basis theory generalizes these ideas to multivariate polynomials. The key theorem that makes Gröbner basis theory work states that it is possible to generalize the Euclidean algorithm to a “preprocessing” of the given set $\{g_1, \dots, g_m\}$ in such a way that one obtains another set which still generates the same ideal and has the desired property to yield “remainder” zero for every “division” with a member of the ideal as the “dividend”. Ideal bases with this property are called Gröbner bases. The algorithm that achieves the “preprocessing” is called the Buchberger algorithm [Buc65].

It is a multivariate analogue to the Euclidean algorithm and can also be viewed as a generalization of the Gaussian elimination algorithm to the linear case.

1.1 Monomial Orders and Polynomial Reductions

Throughout this chapter K will be a field, and the polynomial ring $K[X_1, \dots, X_n]$ over K will also be denoted by $K[\underline{X}]$.

A monomial μ is a power product of the form $X_1^{\nu_1} \cdots X_n^{\nu_n}$ with $\nu_1, \dots, \nu_n \in \mathbb{N}$. We denote the set of monomials by $M(X_1, \dots, X_n)$, or simply by M . M forms an Abelian monoid with neutral element 1 under the natural multiplication.

A natural isomorphism between M and \mathbb{N}^n is given by

$$\begin{aligned}\eta : (T, 1, \cdot) &\rightarrow (\mathbb{N}^n, (0, \dots, 0), +), \\ \eta(X_1^{\nu_1} \cdots X_n^{\nu_n}) &= (\nu_1, \dots, \nu_n).\end{aligned}$$

In the following we will represent a monomial μ by its image $\eta(\mu) \in \mathbb{N}^n$. A monomial $\mu = (\mu_1, \dots, \mu_n)$ is divisible by a monomial $\nu = (\nu_1, \dots, \nu_n)$ if $\nu_i \leq \mu_i$ for all $1 \leq i \leq n$. This will be denoted by $\nu \leq' \mu$, the natural partial order on \mathbb{N}^n . Furthermore we denote the total degree of μ by $|\mu| := \mu_1 + \dots + \mu_n$.

Definition 1.1. A monomial order is a linear order on M that satisfies the following conditions.

- (i) $1 \leq \mu$ for all $\mu \in M$.
- (ii) $\mu_1 \leq \mu_2$ implies $\mu_1 + \nu \leq \mu_2 + \nu$ for all $\nu, \mu_1, \mu_2 \in M$.

Example 1.2. Each of the following is a term order on M .

Let $\mu = (\mu_1, \dots, \mu_n)$, $\nu = (\nu_1, \dots, \nu_n)$ be monomials in M .

1. **Lexicographical order.** $\mu \leq \nu$ iff the following hold:

- (i) $\mu = \nu$, or
- (ii) there exists $1 \leq i \leq n$ with $\mu_j = \nu_j$ for $1 \leq j \leq i - 1$ and $\mu_i < \nu_i$.

2. **Inverse lexicographical order.** $\mu \leq \nu$ iff the following hold:

- (i) $\mu = \nu$, or
- (ii) there exists $1 \leq i \leq n$ with $\mu_j = \nu_j$ for $i + 1 \leq j \leq n$ and $\mu_i < \nu_i$.

3. **Graded lexicographical order.** $\mu \leq \nu$ iff the following hold:

- (i) $\mu = \nu$, or
- (ii) $|\mu| < |\nu|$, or
- (iii) $|\mu| = |\nu|$ and there exists $1 \leq i \leq n$ with $\mu_j = \nu_j$ for $1 \leq j \leq i - 1$ and $\mu_i < \nu_i$.

4. **Graded reverse lexicographical order.** $\mu \leq \nu$ iff the following hold:

- (i) $\mu = \nu$, or
- (ii) $|\mu| < |\nu|$, or
- (iii) $|\mu| = |\nu|$ and there exists $1 \leq i \leq n$ with $\mu_j = \nu_j$ for $i + 1 \leq j \leq n$ and $\mu_i > \nu_i$.

Definition 1.3. Given a monomial order \leq on M we define for any non-zero polynomial $f \in K[\underline{X}]$ the leading monomial $\ell(f) \in M$ and the leading coefficient $c(f) \in K$ of f with respect to \leq as follows:

$$\ell(f) := \max(\text{support}(f)) \text{ and}$$

$$c(f) := \text{the coefficient of } \ell(f).$$

For a subset $F \subseteq K[\underline{X}]$ we denote the set of leading monomials in F by

$$\Lambda(F) := \{\ell(f) \mid f \in F, f \neq 0\}.$$

Lemma 1.4. Let $f, g \in K[\underline{X}]$ with $f, g \neq 0$. Then the following hold:

- (i) $\ell(f \cdot g) = \ell(f) + \ell(g)$,
- (ii) $c(f \cdot g) = c(f) \cdot c(g)$, and
- (iii) $\ell(f + g) \leq \max\{\ell(f), \ell(g)\}$.

Definition 1.5. Let $f, g, p \in K[\underline{X}]$ with $f, p \neq 0$, and let P be a subset of $K[\underline{X}]$. Then we say:

(i) f reduces to g modulo p by eliminating μ , if $\mu \in \text{support}(f)$, there exists $\nu \in M$ with $\nu + \ell(p) = \mu$, and

$$g = f - \frac{c_\mu(f)}{c(p)} \cdot X^\nu \cdot p,$$

where $c_\mu(f)$ denotes the coefficient of μ in f .

Notation: $f \xrightarrow[p]{} g [\mu]$.

(ii) f reduces to g modulo p , if $f \xrightarrow[p]{} g [\mu]$ for some $\mu \in \text{support}(f)$.

Notation: $f \xrightarrow[p]{} g$.

(iii) f reduces to g modulo P , if $f \xrightarrow[p]{} g$ for some $p \in P$.

Notation: $f \xrightarrow[P]{} g$.

(iv) f is reducible modulo p , if there exists $g \in K[\underline{X}]$ such that $f \xrightarrow[p]{} g$.

(v) f is reducible modulo P , if there exists $g \in K[\underline{X}]$ such that $f \xrightarrow[P]{} g$.

(vi) f is top-reducible modulo p if there exists $g \in K[\underline{X}]$ such that $f \xrightarrow[p]{} g [\mu]$ and $\mu = \ell(f)$.

Moreover we denote the reflexive-transitive closure of $\xrightarrow[p]{} g$ by

$$\xrightarrow[P]^* g,$$

i.e. $f \xrightarrow[P]^* g$ means, that there exists a sequence $h_0, \dots, h_m \in K[\underline{X}]$ with $h_0 = f$, $h_m = g$ and $h_i \xrightarrow[p]{} h_{i+1}$ for $1 \leq i \leq m - 1$.

Definition 1.6. Let $f, g, p \in K[\underline{X}]$ with $f, p \neq 0$, and let P be a subset of $K[\underline{X}]$. If f is not reducible modulo p (modulo P), then we say f is in normal form modulo p (modulo P). A normal form of f modulo P is a polynomial g that is in normal form modulo P and satisfies $f \xrightarrow[P]^* g$.

1.2 Gröbner bases - existence and uniqueness

In the following let I be an ideal of $K[\underline{X}]$ generated by a subset F of $K[\underline{X}]$.

A monomial $\mu \in M$ is called *normal modulo I* , if it is not a leading monomial of any element in I , i.e. $m \in M \setminus \{\ell(f) \mid f \in I\}$. We denote by N the K -linear hull of the normal monomials (modulo I), the *normal complement* of I . This name is justified by the fact that N is isomorphic to $K[\underline{X}]/I$. Furthermore we call a polynomial $f \in K[\underline{X}]$ *normal modulo I* if it is in N .

Generally spoken, given an subset F of $K[\underline{X}]$ and an element $f \in K[\underline{X}]$ there exist many different normal forms modulo F depending on the used reduction sequence. But we are interested in a finite generating set G of $Id(F)$ (i.e. ideal of $K[\underline{X}]$ generated by F), such that the normal form of every element in $K[\underline{X}]$ is unique. A subset with this properties is called Gröbner basis. We now give the exact definition:

Definition 1.7. *A subset G of $K[\underline{X}]$ is called a Gröbner basis (with respect to the monomial order \leq), if it is finite, $0 \notin G$ and the polynomials $h \in K[\underline{X}]$ that are in normal form w.r.t. \xrightarrow{G} form a system of unique representatives for $K[\underline{X}]/I$.*

Proposition 1.8. *Let I be an ideal of $K[\underline{X}]$ and G a finite subset of I with $0 \notin G$. Then each of the following is equivalent to G being a Gröbner basis of I .*

- (i) $f \xrightarrow{G}^* 0$ for all $f \in I$.
- (ii) Every $0 \neq f \in I$ is reducible modulo G .
- (iii) For every $\mu \in \Lambda(I)$ there exists $\nu \in \Lambda(G)$ with $\nu \leq' \mu$.
- (iv) $\Lambda(I) \subseteq \Lambda(G) + \mathbb{N}^n$.
- (v) Every normal form w.r.t. \xrightarrow{G} is normal modulo I .

We can now give a simple non-constructive existence proof for Gröbner bases.

Theorem 1.9. *Let I be an ideal of $K[\underline{X}]$. Then there exists a Gröbner basis G of I w.r.t. \leq .*

Proof. By Dickson's Lemma we know that every non-empty subset S of M has a finite subset B , such that for all $\nu \in S$ there exists $\mu \in B$ with $\mu \leq' \nu$. So the set $\Lambda(I)$ has a finite basis B w.r.t. \leq' . For each $\mu \in B$, there exists $f_\mu \in I$ such that $\ell(f_\mu) = \mu$. Let now

$$G = \{f_\mu \mid \mu \in B\}.$$

Then G satisfies condition (iii) of the previous proposition, and so G is a Gröbner basis. \square

Note that a Gröbner basis of an ideal I is of course not uniquely determined by I . We are now going to define the term of “reduced” Gröbner basis and quote that for every ideal I a reduced Gröbner bases always exists and is uniquely determined by I .

Definition 1.10. *Let I be an ideal of $K[\underline{X}]$ and G a Gröbner Basis of I . Then G is called reduced if every $g \in G$ is monic and in normal form modulo $G \setminus \{g\}$.*

Theorem 1.11. *Let I be an ideal of $K[\underline{X}]$. Then there exists a unique reduced Gröbner basis G of I w.r.t. \leq .*

1.3 The Buchberger algorithm

We keep the conventions of the last section: $K[\underline{X}]$ is a polynomial ring over the field K , and \leq is a monomial order on M . In the preceding section we quote the existence and uniqueness of a reduced Gröbner basis. If we want to use Gröbner bases in algorithms we need to know, how a reduced Gröbner basis can be constructed from a arbitrary generating set of an ideal. To achieve that, we first need a practicable criterion for a set being a Gröbner basis of the ideal it generates.

Definition 1.12. *Let $g_1, g_2 \in K[\underline{X}] \setminus \{0\}$ with $\ell(g_1) = \mu = (\mu_1, \dots, \mu_n)$ and $\ell(g_2) = \nu = (\nu_1, \dots, \nu_n)$. Then the least common multiple of μ, ν is defined as*

$$\text{lcm}(\mu, \nu) = (\kappa_1, \dots, \kappa_n), \text{ where } \kappa_i = \max(\mu_i, \nu_i) \text{ for } 1 \leq i \leq n.$$

Furthermore the S-polynomial of g_1 and g_2 is defined as

$$\text{spol}(g_1, g_2) = c(g_2) \cdot g_1 \cdot X^{\mu'} - c(g_1) \cdot g_2 \cdot X^{\nu'},$$

where $\nu', \mu' \in M$ such that $\mu + \mu' = \nu + \nu' = \text{lcm}(\mu, \nu)$.

Theorem 1.13. *Let G be a finite subset of $K[\underline{X}]$ with $0 \notin G$. Then the following are equivalent:*

- (i) G is a Gröbner basis.
- (ii) Whenever $g_1, g_2 \in G$ and $h \in K[\underline{X}]$ is a normal form of $\text{spol}(g_1, g_2)$ modulo G , then $h = 0$.
- (iii) $\text{spol}(g_1, g_2) \xrightarrow[G]{*} 0$ for all $g_1, g_2 \in G$.

A important consequence of this theorem is the following algorithm for the construction of a Gröbner basis from an arbitrary ideal basis. The algorithm that achieves this is also called the **Buchberger algorithm**.

Algorithm BUCHBERGER

INPUT: A finite subset F of $K[\underline{X}]$.
OUTPUT: A Gröbner basis G of $Id(F)$.

```
begin
 $G \leftarrow F$ ;  $B \leftarrow \{\{g_1, g_2\} \mid g_1, g_2 \in G \text{ with } g_1 \neq g_2\}$ ;
while  $B \neq \emptyset$  do
  select  $\{g_1, g_2\}$  from  $B$ ;
   $B \leftarrow B \setminus \{\{g_1, g_2\}\}$ ;
   $h \leftarrow spol(g_1, g_2)$ ;
   $h_0 \leftarrow$  some normal form of  $h$  modulo  $G$ ;
  if  $h_0 \neq 0$  then
     $B \leftarrow B \cup \{\{g, h_0\} \mid g \in G\}$ ;
     $G \leftarrow G \cup \{h_0\}$ 
  endif;
endwhile;
Return  $G$ ;
end;
```

Theorem 1.14 (Buchberger Algorithm). *Let F be a finite subset of $K[\underline{X}]$. Suppose the ground field is computable, and the term order on M is decidable. Then the algorithm BUCHBERGER computes a Gröbner basis G in $K[\underline{X}]$ such that $F \subseteq G$ and $Id(F) = Id(G)$.*

Algorithm REDUCEDGB

INPUT: A Gröbner Basis G in $K[\underline{X}]$.
OUTPUT: The reduced Gröbner basis H of $Id(G)$.

```
begin
 $H \leftarrow \emptyset$ ;  $F \leftarrow G$ ;
while  $F \neq \emptyset$  do
  select  $f_0$  from  $F$ ;
   $F \leftarrow F \setminus \{f_0\}$ ;
  if  $\ell(f) \leq' \ell(f_0)$  for all  $f \in F$  and  $\ell(h) \leq' \ell(f_0)$  for all  $h \in H$  then
     $H \leftarrow H \cup \{f_0\}$  endif;
endwhile;
for  $h$  in  $H$  do  $h \leftarrow$  the normal form of  $h$  modulo  $H \setminus \{h\}$  endfor;
Return  $H$ ;
end;
```

Proposition 1.15. *Let G be a Gröbner basis in $K[\underline{X}]$. Suppose K is computable and the monomial order on M is decidable. Then the algorithm REDUCEDGB computes the reduced Gröbner basis of $Id(G)$.*

Theorem 1.16. *Let F be a finite subset of $K[\underline{X}]$, let $I = Id(F)$, and assume K is computable. Then the following hold:*

- (i) *The equivalence problem for the ideal I is decidable. In particular, one can decide membership in I (i.e. the ideal membership problem for I is algorithmically solvable).*
- (ii) *The residue class ring $K[\underline{X}]/I$ is computable.*

2 Gröbner bases in the noncommutative case

Introduction. Considering ideals in a free associative algebra we get a quite analogous calculus of Gröbner bases with similar results as in the commutative case, see also [Ufn90]. The main differences are situated in the construction of the Gröbner bases. But first we got to distinguish two cases, two-sided and one-sided ideals.

2.1 Gröbner bases of two sided ideals

2.1.1 Basic Notation. Degree and Order

Let K be a field, $K\langle X \rangle$ a free associative algebra with the unity and W the semigroup of all words in the alphabet X (including the empty word 1).

If $u, v \in W$, then we denote by $deg_u v$ the number of different occurrences of the u inside the word v . If $U \subseteq W$ is some set of words, then we denote

$$deg_U v = \sum_{u \in U} deg_u v.$$

Furthermore let us assume that the set of words W has an admissible well-ordering $>$, for instance the graded-lexicographical order.

Now, with every non-zero element $f \in K\langle X \rangle$ we can associate its leading word $\ell(f)$ as the largest word in the support of f and we can extend $>$ to a partial order on $K\langle X \rangle$: $f > g \Leftrightarrow \ell(f) > \ell(g)$. Furthermore, if $F \subseteq K\langle X \rangle$, then $\Lambda(F) = \{\ell(f) : f \in F, f \neq 0\}$ and if $g \in K\langle X \rangle$, then we set

$$deg_F g = deg_{\Lambda(F)} \ell(g).$$

If $f \in K\langle X \rangle$, $f \neq 0$, we denote the coefficient of the leading word $\ell(f)$ with $c(f)$, the leading coefficient of f .

2.1.2 Normal words

Let I be a two-sided ideal of the free algebra $K\langle X \rangle$.

Definition 2.1. A word $u \in W$ is called normal (modulo ideal I), if u is not the leading term of any element I , equivalently $\text{deg}_I u = 0$. Let us denote by N the linear hull of the set of normal words.

For every $f \in K\langle X \rangle$, its normal form \bar{f} is defined to be its image by the natural projection $K\langle X \rangle \rightarrow N$.

Remark 2.2. Clearly, $\bar{f} = 0 \Leftrightarrow f \in I$. If we define a new operation on N by setting $f * g = \overline{fg}$, then $(N, *) \cong K\langle X \rangle / I$.

Thus the main question is: How to reduce an arbitrary word to its normal form. Unfortunately, this problem is generally speaking, algorithmically unsolvable. Nonetheless, we will be coming across solving such a problem, and in the majority of important cases it will be possible.

2.1.3 Gröbner Basis

Definition 2.3. A subset G of the ideal I is called a Gröbner Basis, if for all $f \in I$ the following is satisfied:

$$\text{deg}_G f > 0.$$

The value of Gröbner basis is shown through the following.

Theorem 2.4. A word $v \in W$ is normal if and only if

$$\text{deg}_G v = 0.$$

There are, generally speaking, many Gröbner bases, however there is always a *minimal* one in the sense, that no subset of it is a Gröbner basis. The minimality condition is equivalent to the statement that, for every $g \in G$ $\text{deg}_{G \setminus \{g\}} g = 0$ holds.

Definition 2.5. A minimal Gröbner Basis G is called reduced, if every element $g \in G$ has the form $\ell(g) - f$, where $f \in N$.

Theorem 2.6. The reduced Gröbner basis of an ideal is uniquely determined.

It is easy to construct a reduced basis starting with a minimal Gröbner basis G by the following approach:

1. Normalize the elements in G .
2. Reduce all the non-leading words with the aid of the basis itself into the normal form.

Therefore our immediate goal is to learn how to construct a minimal basis starting with a given generating set of an ideal.

2.1.4 Reduction and Composition.

Let us assume that the ideal I is generated by a set R . In order to get a Gröbner basis, starting with R , we need to transform R using three operations-stages.

1. **Normalization** Substitute all elements $g \in R$ by $c(g)^{-1}g$.
2. **Reduction** If f and g are normalized elements, such that $\deg_g f > 0$, then let $\ell(f) = u\ell(g)w$ be the occurrence of the leading word of g in the leading word of f . Then by reduction we mean:

$$f \xrightarrow[g]{} \frac{1}{c(f - vgw)} \cdot (f - vgw).$$

Analogous to the commutative case we denote the transitive hull of $\xrightarrow[R]{} \xrightarrow[R]{} \dots$ with $\xrightarrow[R]{*}$.

3. **Composition** The composition of a pair f, g of normalized elements is a word $u \in W$, such that $u = xyz$, where $x, y, z \in W$, $xy = \ell(f)$, $yz = \ell(g)$, $y \neq 1$. The result of composition is

$$\text{comp}(f, g, u) = \frac{1}{c(xg - fz)} \cdot (xg - fz).$$

We adjoin the result of all compositions unaccounted earlier to R and return to the reduction stage.

A infinite number of repetitions of the second and the third stages results in a minimal Gröbner basis since the following holds:

Lemma 2.7 (Lemma of Composition). *If the set F is such that for all $f \in F$ $\deg_{F \setminus \{f\}} f = 0$ and the result of any composition reduces to zero after a few steps, then F is a minimal Gröbner Basis.*

2.1.5 Examples

Example 2.8. Consider $A = \langle x, y \mid x^2 + y^2 = 0 \rangle$, $x > y$.

- $G_0 = \{x^2 + y^2\}$. $f := x^2 + y^2$ allows for composition with itself.
 $\text{comp}(f, f, x \cdot x \cdot x) = xf - fx = xy^2 - y^2x =: g$.
- $G_1 = \{f, g\}$.
 $\text{comp}(f, g, x \cdot x \cdot y^2) = xg - fy^2 = -(xy^2x + y^4) \xrightarrow[g]{} -(xy^2x + y^4) - gx = y^2x^2 + y^4 \xrightarrow[f]{} y^2x^2 + y^4 - y^2f = 0$.

There are no more reductions and unaccounted compositions, thus our process is finished and we have obtained a Gröbner basis $G = \{f, g\}$.

Example 2.9. Consider $A = \langle x, y \mid x^2 - yx = 0 \rangle$, $x > y$.

- $G_0 = \{x^2 - yx\}$. $f_0 := x^2 - yx$ allows for the composition with itself.
 $comp(f_0, f_0, x \cdot x \cdot x) = f_0x - xf_0 = xyx - y^2x =: f_1$.
- $G_1 = \{f_0, f_1\}$.
 $comp(f_0, f_1, x \cdot x \cdot yx) = f_0yx - xf_1 = xy^2x - yxyx \xrightarrow{f_1} xy^2x - yxyx - yf_1 = xy^2x - y^3x =: f_2$.
- ...
- $G_n = \{f_0, f_1, \dots, f_n\}$.
 $comp(f_0, f_n, x \cdot x \cdot y^n x) = f_0y^n x - xf_n = xy^{n+1}x - yxy^n x \xrightarrow{f_n} xy^{n+1}x - yxy^n x - yf_n = xy^{n+1}x - yf_n = xy^{n+1}x - y^{n+2}x =: f_{n+1}$.

According to the lemma of composition the set $F = \{f_n \mid n = 0, 1, \dots\}$ is a Gröbner basis, since for every composition $comp(f_k, f_l, xy^k xy^l x) = f_k y^l x - xy^k f_l = xy^{k+l+1}x - y_{k+1}xy^l x \xrightarrow{f_{k+l+1}} y^{k+l+2}x - y_{k+1}xy^l x \xrightarrow{f_l} y^{k+l+2}x - y^{k+l+2}x = 0$.

2.2 Gröbner bases of one sided ideals

As in the preceding section we consider the free associative algebra $K\langle X \rangle$ and use the same basic notation. In order to define a Gröbner basis of a right sided ideal I (i.e. for all $f \in I$, $g \in K\langle X \rangle$: $f \cdot g \in I$) we introduce the following notation: If $u, w \in W$, then

$$pref_u w = \begin{cases} 1 & \text{if } u \text{ is a prefix of } w, \\ 0 & \text{else.} \end{cases}$$

Moreover we define $pref_U w = \sum_{u \in U} pref_u w$, where $U \subseteq W$, and $pref_F g = pref_{\Lambda(F)} \ell(g)$, where $F \subseteq K\langle X \rangle$ and $g \in K\langle X \rangle$.

Analogous to the two sided case we now are able to define

Definition 2.10. Let I be an right sided Ideal of the the free algebra $K\langle X \rangle$.

- (i) A word $w \in W$ is called normal if $pref_I w = 0$. Let us denote by N the linear hull of the set of normal words.
- (ii) A subset G of the ideal I is called a Right Gröbner basis of I if, for all $f \in I$, the following is satisfied:

$$pref_G f > 0.$$

Remark 2.11. As before we get similar definitions and results as in the two sided case:

1. There are analogous terms like *minimal* and *reduced*.
2. The reduced Right Gröbner basis is uniquely determined, too.
3. To construct a Right Gröbner basis we need the operations-stages Normalization and Reduction, but no Compositions. In the noncommutative case we mean by reduction the following: Let $f, g \in \langle K \rangle$ and $pref_f g = 1$, then there exists a word $u \in W$ such that $\ell(g) = \ell(f)u$ and f reduces g as follows

$$g \xrightarrow[f]{} g - c(g)c(f)^{-1}fu.$$

4. The computation of a Right Gröbner basis is efficient. Particularly, if R is the generating set of I and G is the reduced Gröbner basis of I , the following hold: $\#G \leq \#R$ and $\maxdeg(G) \leq \maxdeg(R)$.

Remarks on my Diplomarbeit

In my Diplomarbeit I consider matrices over free noncommutative algebras. Especially I dealt with the question, how to decide whether a matrix is invertible or not. In the commutative case this is quite easy to decide, one just compute the determinant of the matrix, and if its a unit, the matrix is invertible. When we work with matrices over noncommutative structures, we do not have the determinant calculus.

With the theory of one sided Gröbner bases I found a algorithm to decide, whether a matrix is invertible, and to invert it where applicable. One main result is, that the invertible matrices over a free associative algebra are generated by elementary matrices, in other words the general linear group is equal to the elementary group. This implies that the ring of matrices over a free associative algebra is euclidean.

2.3 Comparing the commutative and the noncommutative case

1. The Buchberger algorithm always terminates, since the ring $K[X_1, \dots, X_n]$ is noetherian. In the noncommutative case the described algorithm to compute a minimal Gröbner base runs infinitely for many generating sets, returning a Gröbner base with infinite many elements.
2. In both cases the algorithm consist of the three operations-stages Normalization, Reduction and Compositions.

3. Whereas in the commutative case the composition (i.e. the s-polynomial) of two elements in $K[X]$ is uniquely determined, in the noncommutative case two elements in $K\langle X \rangle$ can have different composition. Therefore in the noncommutative case one has to make even more choices during the algorithm. (In addition to the choice of the critical pair and the reducing polynomials you got to select a critical word for the composition.)
4. In all cases the reduced Gröbner bases is uniquely determined.

3 Cryptographic implications of Gröbner bases computations

Introduction. We now have the theoretical background to understand the coherences between Cryptography and Gröbner bases computations. At first we describe a PKC-scheme introduced by Neal Koblitz [Kob98] which uses Gröbner bases directly to construct a one-way-function. After that we explain why the security of the HFE-cryptosystem of Patarin depend on the non-efficiency of the Gröbner basis construction. In the last two subsections we propose a PKC-scheme based on Gröbner bases in the noncommutative case.

3.1 Koblitz: Polly Cracker, a Cryptosystem using Gröbner bases directly.

In the book “Algebraic Aspects of Cryptography” Neal Koblitz proposed a public key cryptosystem using Gröbner bases directly, a generalization of the cryptosystem Polly Cracker. As in the preceding chapter let K be a field and $K[\underline{X}]$ the commutative polynomial ring over K . Given an ideal I of $K[\underline{X}]$ we denote the set of normal polynomials modulo I in $K[\underline{X}]$ by N and define something similar to an “one-way-function” as follows

$$\varphi_r : N \rightarrow K[\underline{X}], f \mapsto f + g,$$

where $g \in I$ is a random polynomial. (The r in φ_r should underline that this function has an random component.) This is in the following sense a one-way-function, that in most cases the normal polynomial $\varphi_r^{-1}(f)$ is not efficiently computable. We quote that the Ideal Membership Problem is NP-hard (see chapter 4) and this fact implies that computing the normal form of an polynomial modulo an ideal is NP-hard, too.

A PKC-scheme:

1. Alice:

- Alice chooses an ideal I of $K[\underline{X}]$.
- She publishes a subset $S \subseteq I$ and a subset $T \subseteq N$.
- Her trapdoor is the reduced Gröbner basis G of I .

2. Bob:

- Bob represents his message by an element $m \in N$ in the K -linear hull of T .
- After that he selects a element $g \in I$ by choosing polynomials $p_1, \dots, p_s \in K[\underline{X}]$ and $q_1, \dots, q_s \in S$ and computing $g = \sum_{i=1}^s p_i \cdot q_i$.
- The encrypted message is then $c = m + g$.

3. Alice:

- Alice decrypts the message with her trapdoor G .
- Since $h \xrightarrow[G]{*} h$ for all $h \in N$ and $f \xrightarrow[G]{*} 0$ for all $f \in I$, she gets:

$$c \xrightarrow[G]{*} m.$$

4. Catherine:

An eavesdropper Catherine has two possibilities to decrypt the message m :

- She computes a Gröbner basis G' of $Id(S)$ and reduces c with G' and gets: $c \xrightarrow[G']{*} m$, since d is in $Id(S)$.
- She reduces c without a Gröbner basis.

Both ways to decrypt m are in general inefficient, since it can be shown that computing a normal form is at least NP-hard.

3.2 Patarin and HFE: Why HFE does not work, if one can compute Gröbner bases efficiently.

At Eurocrypt 96 Jacques Patarin [Pat96] proposed a cryptosystem named HFE, which stands for **H**idden **F**ields **E**quations. The principle of HFE is the following:

1. It is possible to find a solution of univariate polynomial over a big finite field provided the degree d of the polynomial is not too big.

2. For some polynomial functions it is possible to represent them as n quadratic equations with n variables which hide the polynomial structure and makes it look quite as (almost) any other polynomial of any degree.

The HFE polynomial is of the form

$$f(a) = \sum_i \alpha_i \cdot a^{q^{s_i} + q^{t_i}}.$$

Given $K = \mathbb{F}_q$, $q = p^m$ with p prime, the α_i are random coefficients in K^n taken for all possible powers of the form $a^{q^{s_i} + q^{t_i}}$ are not greater than a given maximum power a^d . (d is one of the security parameters.)

Generally, if we consider polynomials over the extension field K^n , every polynomial f of the univariate degree d can also be written as a multivariate set of polynomial equations in variables a_0, \dots, a_{n-1} written in K , as follows:

$$b_i = f_i(a_{n-1}, \dots, a_0),$$

where $f(a) = f(a_{n-1}X^{n-1} + \dots + a_1X + a_0) = b_{n-1}X^{n-1} + \dots + b_1X + b_0$. These equations have a distinct degree, in general not greater than d , which is called the multivariate degree or the nonlinear order of f .

In the case of the HFE polynomials the special structure provides that the multivariate degree is always 2, since in fields of characteristic p some powers are linear transformations. For instance we have

$$(a + b)^{p^t} = a^{p^t} + b^{p^t}$$

for all a, b in K^n . Therefore the f_i are quadratic (over K) for all the monomials of the form $f(a) = a^{q^s + q^t}$, and these are the monomials the HFE polynomial is a sum of.

The HFE PKC-scheme:

1. Alice:

- Alice chooses an a finite field K , an extension field K^n and a HFE-polynomial.
- Moreover she generates two random affine transformations

$$S : a_i = \gamma_{ij}x_j + \beta \text{ and } T : y_i = \gamma'_i x_i + \beta'.$$

- After that she computes the composition $g = T \circ f \circ S$, and publishes the field K and the quadratic forms of g , that is:

$$y_1 = p_1(x_1, \dots, x_n)$$

$$y_2 = p_2(x_1, \dots, x_n)$$

$$\vdots$$

$$y_n = p_n(x_1, \dots, x_n)$$

- Her secret key is the triplet (S, f, T) .

2. Bob:

- Bob represents his message by an tuple $x = (x_1, \dots, x_n) \in K^n$.
- After that he computes the $(y_1, \dots, y_n) \in K^n$ with the quadratic forms of the public key.
- The encrypted message is then $y = (y_1, \dots, y_n)$.

3. Alice:

- Alice decrypts the message with her secret key (S, f, T) .
- Since there exists many efficient algorithms to compute f^{-1} , if the degree d is not too large, Alice is able to compute $g^{-1}(y)$.
- As the HFE polynomial is not one-to-one in general, some redundancy must be added before encryption so that Alice could choose the good solution. This can be done with some public hash functions, for instance MD5 or SHA.

4. Catherine:

An eavesdropper Catherine has two possibilities to decrypt the message x :

- She tries to find f , but this seems to be impossible.
- She solves the system of multivariate quadratic polynomial equations.

The classical algorithm for solving a system of multivariate polynomial equations is Buchberger's algorithm for constructing Gröbner bases, and its many variants. The algorithm orders the monomials (typically in lexicographic order), and eliminates the leading monomial by combining two equations with appropriate polynomial coefficients. This process is repeated until all but one of the variables are eliminated, and then solves the remaining univariate polynomial equation, for instance by using Berlekamp's algorithm.

Therefore, if one is able to compute the Gröbner basis efficiently, there is an efficient algorithm to attack the HFE cryptosystem. Note that HFE is patented by Jacques Patarin and is planned to be used commercially for Smart Cards and Terminals. Therefore there is much research on it by Adi Shamir, Nicolas Courtois and Louis Goubin, for instance Shamir develops an attack called "Relinearization", see also [SK99, CKPS00].

3.3 Wagner, Magyarik: A Public Key Cryptosystem based on the Word Problem

In the year 1984 Neal R. Wagner and Marianne R. Magyarik [WM84] proposed a cryptosystem defined on noncommutative structures. It is based on the word problem, which is defined as follows:

The free group F on generators x_1, x_2, \dots, x_n is defined as the set of all words in the x_i and x_i^{-1} that are reduced by repeatedly cancelling out $x_i x_i^{-1}$ and $x_i^{-1} x_i$ until no further cancellations are possible. The empty string e is also a word, the identity of the group. Let $G = \langle x_1, x_2, \dots, x_n \mid r_1 = e, r_2 = e, \dots, r_m = e \rangle$ with r_1, r_2, \dots, r_m relators be a finitely presented group in F , i.e. G is the quotient group F/R , where R is the normal subgroup generated by the words r_1, r_2, \dots, r_m , that is R is the intersection of all normal subgroups containing the r_i .

Then the **Word Problem** for a group G is the decision problem that asks for each word w , whether w is equivalent to the identity of G . (Equivalently one can ask whether two given words are equivalent.)

It turns out that there exist specific groups for which the word problem is undecidable. Like any undecidable problem, the word problem can only be undecidable as a question asked about infinitely many words. Any finite collection of words must have a decidable word problem.

The word problem can also be defined for semigroups. We start with generators and words in the generators as before but without the inverses. Instead of relators we have a list of equations of the form $a_1 = b_1, a_2 = b_2, \dots, a_m = b_m$. In defining equivalent words we can only replace any occurrence of a_i by b_i and vice versa. The **Word Problem for semigroups** asks whether two given words are equivalent. We will refer to it in the next section.

The word problem is similar to the knapsack problem in that both immediately and directly allow public encryption. The difficulty is to insert a trapdoor that will allow decryption. The trapdoor then becomes a point of weakness for cryptanalytic attacks. We feel that a harder problem may make direct attacks more difficult and allow more leeway for such trapdoor insertions.

A PKC-scheme:

1. Alice:

- Alice chooses a finitely presented group

$$G = \langle x_1, x_2, \dots, x_n \mid r_1 = e, r_2 = e, \dots, r_m = e \rangle$$

and two words w_0 and w_1 known to be inequivalent in G .

- She publishes the group G and the words w_0, w_1 .
- Her trapdoor is a quotient group

$$G' = \langle x_1, x_2, \dots, x_n \mid r_1 = e, r_2 = e, \dots, r_m = e, s_1 = e, s_2 = e, \dots, s_p = e \rangle.$$

of G with the following properties:

- (i) The images of w_0 and w_1 under the natural residue class homomorphism $\varphi : G \rightarrow G'$ are still inequivalent in G' .
- (ii) The relators s_1, \dots, s_p should provide that the word problem in G' is decidable.

2. Bob:

- Bob sends Alice a bit by choosing w_0 , if the bit is 0, w_1 otherwise.
- After that he “randomly” applies a sequence of the following Rules to the word w_0 respectively w_1 , resulting in a word v :
 - (i) changing $x_i x_i^{-1}$ or $x_i^{-1} x_i$ to e ,
 - (ii) introducing $x_i x_i^{-1}$ or $x_i^{-1} x_i$ at any point,
 - (iii) changing r_j or r_j^{-1} to e ,
 - (iv) introducing r_j or r_j^{-1} at any point.
- The encrypted message is the word v .

3. Alice:

- Alice decrypts the message with her trapdoor G' .
- Since the word problem in G' is efficiently decidable, for instance because it is commutative, Alice can easily decide whether v is equivalent to w_0 or w_1 by computing $\varphi(v)$.

4. Catherine:

An eavesdropper Catherine has two possibilities to decrypt the message m :

- She decides directly which of the w_0 and w_1 is equivalent to v , for instance trying all possible sequences of replacements in parallel on w_0 and w_1 . Alice hopes that with a good choice of G there will be no tractable algorithms for direct attacks of this sort.
- She tries to find extra relators (not necessary those of Alice), such that the word problem in this group is decidable. This general method is a standard way in group theory to show that two elements are not equivalent.

3.4 A Way to use noncommutative Gröbner bases in Public Key Cryptography

The main difficulty in Wagner’s and Magyarik’s cryptosystem is the insertion of the trapdoor. The authors proposed to introduce such relators that G' becomes commutative. In this case on one hand the Word problem can be solved efficiently,

but on the other hand there is nothing known about the complexity of the Word problem for such special groups.

To overcome this difficulty we propose to use Gröbner bases, since the following holds:

Theorem 3.1. *Given a finitely presented group $G = \langle x_1, x_2, \dots, x_n \mid r_1 = e, r_2 = e, \dots, r_m = e \rangle$. Then two words $w_0, w_1 \in F$ are equal if and only if $w_0 - w_1$ is in $Id(r_1 - 1, r_2 - 1, \dots, r_m - 1) \subseteq K\langle x_1, x_2, \dots, x_n \rangle$.*

Therefore, if Alice has a reduced Gröbner basis of the $Id(r_1 - 1, r_2 - 1, \dots, r_m - 1)$, she has an effective algorithm to compute whether two words are equal or not. Furthermore one advantage could be, that she can find a group G with relators such that the Gröbner basis of the corresponding ideal in the free associative algebra is infinite and a group G' with additional relators where the corresponding Gröbner basis is finite. This is one main focus of my research.

4 The complexity of Gröbner bases computations

4.1 Theoretical results

4.1.1 The Ideal Membership Problem

One basic decision problem of classical algebra is the **The Ideal Membership Problem**, Notation: IMP, which is defined as follows:

Let \underline{X} be some finite set of indeterminates and p, p_1, \dots, p_n polynomials in $\mathbb{Q}[\underline{X}]$. Then the question is: Is p in $Id(p_1, \dots, p_n)$?

In order to study the complexity of IMP we need to know the Uniform Word Problem, UWP, which is defined as follows: Let S be some finite set of symbols, P some finite commutative semigroup presentation over S , and u, w two words over S . Then the question is: Are u, w equivalent in the semigroup presented by P ?

Since there is a reduction of UWP to the complexity theoretic analogue of the Halting Problem, which is known to be exponential space complete, and conversely, we get the following result:

Theorem 4.1. *UWP is exponential space complete (with respect to log-lin reducibility).*

There is an straightforward connection between the UWP and IMP, since the following holds:

Proposition 4.2. *Let $A = \langle \underline{X} \mid X^{\mu_1} = X^{\nu_1}, \dots, X^{\mu_s} = X^{\nu_s} \rangle$ be a finitely presented semigroup over \underline{X} . Then: X^μ, X^ν are equivalent iff $X^\mu - X^\nu \in Id(X^{\mu_1} - X^{\nu_1}, \dots, X^{\mu_s} - X^{\nu_s})$.*

This implies that UWP is reducible to IMP. Hence we get the following result:

Theorem 4.3 (Ernst W. Mayer and Albert R. Meyer [MM82]). *The Ideal Membership Problem for polynomial ideals IMP is exponential space hard.*

Even though we have by the Buchberger algorithm an effective way to decide the Ideal Membership Problem, the property that IMP requires exponential space to decide present the same intractability as undecidable problems, as undecidability of a problem means that every procedure, which gives only correct decisions, must fail to produce a decision for infinitely many instances. In the case of exponential complexity for infinitely many instances we have to wait more than a life time for an answer, this is quite the same as never getting an answer.

4.1.2 The Ideal Membership Problem for two subclasses of polynomial ideals

We have seen that IMP in general is in ExSpace. But for a concrete public key cryptosystem we have to use very special cases of IMP, for instance we have to determine the number of indeterminates. Therefore we will now consider two subclasses of polynomial ideals:

1. ideals of polynomials in four variables,
2. ideals of polynomials of the form $Y - c \cdot X^\mu$, where Y is a variable, $c \in \mathbb{Q}$ and μ is a monomial.

We quote that even this very special cases of IMP are NP-hard. We now define these Problems more precisely:

1. IMP_1

Input: Polynomials $p, p_1, \dots, p_s \in \mathbb{Q}[X_1, X_2, X_3, X_4]$.

Question: Is $p \in \text{Id}(p_1, \dots, p_s)$?

2. IMP_2

Input: Polynomials $Y - c \cdot X^\mu, Y_1 - c_1 \cdot X^{\mu_1}, \dots, Y_s - c_s \cdot X^{\mu_s} \in \mathbb{Q}[\underline{X}]$ where $Y, Y_1, \dots, Y_s \in \underline{X}$, $c, c_1, \dots, c_s \in \mathbb{Q}$ and $\mu, \mu_1, \dots, \mu_s \in M$.

Question: Is $Y - c \cdot X^\mu \in \text{Id}(Y_1 - c_1 \cdot X^{\mu_1}, \dots, Y_s - c_s \cdot X^{\mu_s})$?

By reduction of the satisfiability problem for Boolean formulas in conjunctive normal form, SAT, which is known to be NP-complete, to IMP_1 and IMP_2 , we get the following result:

Theorem 4.4 (Huynh [Huy86]). *IMP_1 and IMP_2 are NP-hard.*

4.2 Experimental results: The algorithm of Faugère

Introduction. In this section we describe an algorithm for computing Gröbner bases introduced by Jean-Charles Faugère in 1999 [Fau99]. It avoids as much intermediate computation as possible by computing successive truncated Gröbner bases and it replaces the classical polynomial reduction (found in the Buchberger algorithm) by the simultaneous reduction of several polynomials. This powerful reduction mechanism is achieved by means of a symbolic precomputation and by extensive use of sparse linear algebra methods.

Even though this algorithm does not improve the worst case complexity it is several times faster than previous implementations both for integers and modulo p computations. (Shamir and Patarin said that Faugère's algorithm is the most efficient variant of the Buchberger algorithm which they are aware of.)

There are two possibilities for improvements of the classical Buchberger algorithm. First, since 90% of the times is spent computing zero, it is useful to have a powerful criterion to remove useless critical pairs. The second improvement is concerned with strategies, i.e. during a Gröbner basis computation one has to make several choices: select critical pairs, choose a reductors.

The algorithm we describe in this section is more efficient than previous implementations, since it uses a strategy that relies on linear algebra techniques. Nevertheless, the heuristics which it relies on could not be satisfactorily explained.

Linear algebra and polynomials

Let K be the ground field and $K[\underline{X}]$ the polynomial ring over K . We use the notations of the first section for basis definitions.

Definition 4.5. Let a be a $s \times m$ -matrix and $a_{i,j}$ the j th element of the i th row of a . If $T_a = [\mu_1, \dots, \mu_m]$ is an ordered set of monomials, let V_{T_a} be the subvector space of $K[\underline{X}]$ generated by T_a . We consider the linear map

$$\varphi_{T_a} : V_{T_a} \rightarrow K^m \text{ such that } \varphi_{T_a}(t_i) = \varepsilon_i,$$

where $(\varepsilon_i)_{i=1, \dots, m}$ be the canonical basis of K^m .

The reciprocal function will be denoted by ψ_{T_a} . The application ψ_{T_a} allows to interpret vectors of K^n as polynomials.

We note by (a, T_a) a matrix with such an interpretation (i.e. (a, T_a) represent a subset of $V_{T_a} \subseteq K[\underline{X}]$ with $\leq s$ polynomials). This subset is denoted by

$$\text{Rows}(a, T_a) := \{\psi_{T_a}(\text{row}(a, i)) \mid i = 1, \dots, s\} \setminus \{0\},$$

where $\text{row}(M, i)$ is the i th row of a (an element of R^m).

Conversely, if $l = (l_1, \dots, l_s)$ is a list of polynomials and $T_l = [\mu_1, \dots, \mu_m]$ an ordered set of monomials we can construct an $s \times m$ -matrix a as follows

$$a_{i,j} := c_{\mu_j}(l_i), \text{ for } i = 1, \dots, s, j = 1, \dots, m.$$

We note the matrix $(a_{i,j})$ by $a^{(l, T_l)}$.

Definition 4.6. Let a be a $s \times m$ -matrix and $\underline{Y} = [Y_1, \dots, Y_m]$ new variables. Then $F = \text{Rows}(M, \underline{Y})$ is a set of equations, so we can compute \tilde{F} a reduced Gröbner basis of F for a lexicographical ordering such that $Y_1 > \dots > Y_m$. From this basis we can reconstruct a matrix $\tilde{M} = a^{(\tilde{F}, \underline{Y})}$. We call \tilde{M} the (unique) row echelon form of M and \tilde{F} a row echelon basis of F .

In the case of polynomials we have a similar definition:

Definition 4.7. Let F be a finite subset of $K[\underline{X}]$ and $<$ an admissible ordering. We define

$$T_{<}(F) := \text{Sort}(\{\Lambda(f) \mid f \in F\}, <),$$

(i.e. $T_{<}(F) = (\mu_1, \dots, \mu_m)$, where $\{\mu_1, \dots, \mu_m\} = \Lambda(F)$ and $\mu_i < \mu_j$ for $i < j$). Let $a := a^{(F, T_{<}(F))}$ and \tilde{a} the row echolon form of a . We say that $\tilde{F} = \text{Rows}(\tilde{a}, T_{<}(F))$ is the row echolon form of F w.r.t. $<$.

Elementary properties of row matrices are summarized by the following theorem:

Theorem 4.8. Let a be a $s \times m$ -matrix and $\underline{Y} = [Y_1, \dots, Y_m]$ new variables. Furthermore let $F = \text{Rows}(a, \underline{Y})$, \tilde{M} the row echolon form of M , $\tilde{F} = \text{Rows}(\tilde{a}, \underline{Y})$. We define

$$\begin{aligned} \tilde{F}^+ &= \{g \in \tilde{F} \mid \ell(g) \notin \Lambda(F)\}, \\ \tilde{F}^- &= \tilde{F} \setminus \tilde{F}^+. \end{aligned}$$

For any subset F_- of F such that $|F_-| = |\Lambda(F)|$ and $\Lambda(F_-) = \Lambda(F)$, the set $G = \tilde{F}^+ \cup F_-$ is a triangular basis of the K -Subvectorspace V_a generated by F (i.e. for all $f \in V_a$ there exists $(c_k)_k$ elements of R and $(g_k)_k$ elements of G such that $f = \sum_k c_k g_k$, $\ell(g_1) = \ell(f)$ and $\ell(g_k) > \ell(g_{k+1})$).

We can transpose immediately the theorem for polynomials.

Corollary 4.9. Let F be a finite subset of E and $<$ an admissible ordering, and \tilde{F} the row echolon form of F w.r.t. $<$. We define

$$\tilde{F}^+ = \{g \in \tilde{F} \mid \ell(g) \notin \Lambda(F)\}.$$

For all subset F_- of F such that $|F_-| = |\Lambda(F)|$ and $\Lambda(F_-) = \Lambda(F)$, the set $G = \tilde{F}^+ \cup F_-$ is a triangular basis of the K -Subvectorspace V_a generated by F .

The F_4 algorithm.

It is well known, that during the execution of the Buchberger algorithm one has a lot of choices:

1. select a critical pair in the list of critical pairs;

2. choose one reductor among a list of reducers when reducing a polynomial by a list of polynomials.

Buchberger proves that these choices are not important for the correctness of the algorithm, but it is well known that these choices are crucial for the total time computation. Faugère solves this problem by making no choice, more precisely he select a subset of critical pairs at the same time. He delays the necessary choices in a second step of the algorithm, the linear algebra part of the algorithm.

Definition 4.10. A critical pair of two polynomials (f, g) , $f \neq g$ is an element of $M^2 \times K[\underline{X}] \times M \times K[\underline{X}]$:

$$Pair(f, g) := (lcm(\ell(f), \ell(g)), \mu, f, \nu, g)$$

such that $\mu + \ell(f) = \nu + \ell(g) = lcm(\ell(f), \ell(g))$. We denote the set of all critical pairs of a set $F \subseteq K[\underline{X}]$ by $Pairs(F) := \{Pair(f, g) \mid f, g \in F \text{ with } f \neq g\}$.

The degree of a critical pair $p = Pair(f, g) = (lcm(\ell(f), \ell(g)), \mu, f, \nu, g)$ is defined as

$$deg(p) := deg(lcm(\ell(f), \ell(g))).$$

Furthermore we define the two projections $Left(p) := (\mu, f)$ and $Right(p) := (\nu, g)$.

We now are able to describe the basic version of the algorithm. All the matrices occurring in the following algorithms are the representation of a list polynomials through the set of all their terms.

Algorithm F_4

INPUT: A finite subset $F \subseteq K[\underline{X}]$
and a function $Selection : \mathcal{P}(Pairs(F)) \rightarrow \mathcal{P}(Pairs(F))$,
such that $Selection(l) \subseteq l$, and $Selection(l) \neq \emptyset$ if $l \neq \emptyset$.
OUTPUT: A finite subset G of $K[\underline{X}]$.

```

begin
 $G \leftarrow F$ ;  $\tilde{F}_0^+ \leftarrow F$ ;  $d \leftarrow 0$ ;  $P \leftarrow Pairs(G)$ ;
while  $P \neq \emptyset$  do
   $d \leftarrow d + 1$ ;
   $P_d \leftarrow Selection(P)$ ;
   $P \leftarrow P \setminus P_d$ ;
   $L_d \leftarrow Left(P_d) \cup Right(P_d)$ ;
   $\tilde{F}_d^+ \leftarrow Reduction(L_d, G)$ ;
  for  $h \in \tilde{F}_d^+$  do
     $P \leftarrow P \cup \{Pair(h, g) \mid g \in G\}$ ;
     $G \leftarrow G \cup \{h\}$ 

```

```

    endfor;
    endwhile;
return G;
end;

```

Function *Reduction*

INPUT: A finite subset L of $M \times K[\underline{X}]$ and a finite subset G of $K[\underline{X}]$.
OUTPUT: A finite subset of $K[\underline{X}]$.

```

begin
 $F \leftarrow \text{SymbolicPreprocessing}(L, G)$ ;
 $\tilde{F} \leftarrow$  Reduction to Row Echolon Form of  $F$ ;
 $\tilde{F}^+ \leftarrow \{f \in \tilde{F} \mid \ell(f) \notin \Lambda(F)\}$ ;
return  $\tilde{F}^+$ ;
end;

```

Function *SymbolicPreprocessing*

INPUT: A finite subset L of $M \times K[\underline{X}]$ and a finite subset G of $K[\underline{X}]$.
OUTPUT: A finite subset of $K[\underline{X}]$.

```

begin
 $F \leftarrow \{X^\mu \cdot f \mid (\mu, f) \in L\}$ ;
 $Done \leftarrow \Lambda(F)$ ;
while  $\text{support}(F) \neq Done$  do
    choose  $\mu$  an element of  $\text{support}(F) \setminus Done$ ;
     $Done \leftarrow Done \cup \{\mu\}$ ;
    if  $\mu$  top reducible modulo  $G$  then
        there exist  $f \in G$  and  $\mu' \in M$  such that  $\mu = \mu' + \ell(f)$ ;
         $F \leftarrow F \cup \{X^{\mu'} \cdot f\}$ 
    endif;
endwhile;
Return  $F$ ;
end;

```

Theorem 4.11. *The algorithm F_4 computes a Gröbner basis G in $K[\underline{X}]$ such that $F \subseteq G$ and $\text{Id}(G) = \text{Id}(F)$.*

Remark 4.12. If $|Selection(l)| = 1$ for all $l \neq \emptyset$, then the algorithm F_4 is the Buchberger algorithm. In that case the *Selection*-function corresponds to the selection strategy used in the Buchberger algorithm.

Faugère develops improvements of this algorithm based on a more efficient version of the procedure *SymbolicPreprocessing*.

Selection strategy

The choice of a good selection strategy, i.e. the choice of the function *Selection*, is very important for the performance of the algorithm. We now give some possible implementation of *Selection*.

1. The easiest way to implement *Selection* is to take the identity, i.e. reducing all the critical pairs at the same time.
2. By experiments (!) Faugère found out that taking all the critical pairs with a minimal total degree seems to be the best function, i.e.

$$Selection(P) := \{p \in P \mid deg(p) = d\},$$

where $d := \min_{p \in P} \{deg(p)\}$. We call this strategy the *normal strategy* for F_4 .

If the input polynomials are homogeneous, we already have a Gröbner basis up to degree $d - 1$ and *Selection* selects exactly all the critical pairs which are needed for computing a Gröbner basis up to degree d .

3. Experiments showed that a algorithm using so called “sugar degrees” instead of total degrees are less efficient.

Conclusion

Faugère transformed the dregree of freedrom in the Buchberger algorithm into strategies for efficiently solving linear algebra systems. This is more efficient, because after a matrix a is constructed one can decide to begin the reduction of one row before another with a “good reason”. Furthermore there are already algorithms for solving large sparse linear systems. Negative aspects of this approach are that the matrix a is singular and that a is often huge.

Experiments

The theoretical results on the complexity of Gröbner basis are often useless, if we want to make a statement on the practical behaviour of effective implementation. The quality of the computer implementation may have a dramatic effect on their performance. Therefore one have to study and use programs, which implements these algorithms, besides the theoretical analysis of the algorithms. In order to compare the different algorithms and implementations there are benchmark examples like the **Cyclic** n -set or the **Hom Cyclic** n -set.

Definition 4.13. Let $n \in \mathbb{N} \setminus \{0\}$ be an integer. Then we define subsets of $K[X_1, \dots, X_n]$ as follows:

- **Cyclic n :**

$$\{p_k := \sum_{j=1}^n \prod_{i=j}^{j+k-1} X_i \mid 1 \leq k < n\} \cup \{p_n := \prod_{i=1}^n X_i - 1\},$$

- **Hom Cyclic $n - 1$:**

$$\{p_k := \sum_{j=1}^{n-1} \prod_{i=j}^{j+k-1} X_i \mid 1 \leq k < n - 1\} \cup \{p_{n-1} := \prod_{i=1}^{n-1} X_i - X_n^{n-1}\}.$$

Example 4.14. Cyclic 4:

$$\begin{aligned} & \{x_1 + x_2 + x_3 + x_4, \\ & x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1, \\ & x_1x_2x_3 + x_2x_3x_4 + x_3x_4x_1 + x_4x_1x_2, \\ & x_1x_2x_3x_4 - 1\} \end{aligned}$$

Hom cyclic 4:

$$\begin{aligned} & \{x_1 + x_2 + x_3 + x_4, \\ & x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1, \\ & x_1x_2x_3 + x_2x_3x_4 + x_3x_4x_1 + x_4x_1x_2, \\ & x_1x_2x_3x_4 - x_5^4\} \end{aligned}$$

As described before the generated matrices we have to solve are sparse. Nevertheless the matrices can have a very different structure and the number of non-zero elements varies greatly. Here we give some examples:

1. For Cyclic 7 the matrix a_9 is a 475×786 -matrix with 13,8 % of the entries are non-zero.
2. In an application example the matrix a_5 is a 1425×2561 -matrix with 0,47 % of the entries are non-zero.
3. For an engineering problem the matrix a_7 is a 1557×3838 -matrix with 0,2 % of the entries are non-zero.

Table 1: Academic examples mod p

Set	F_4	Gb	Singular	Magma	Maple
Cyclic 6	≤ 1 s	3 s	5 s	6 s	16 min
Cyclic 7	4.6 s	1 min 15 s	2 min 34 s	2 min 15 s	
Hom Cyclic 7	5.2 s	1 min 2 s			
Cyclic 8	1 min 55 s	26 min 17 s	1 h 56 min		
Hom Cyclic 8	3 min 4.4 s	39 min 17 s			
Cyclic 9	4 h 32 min	∞	∞		
Hom Cyclic 9	11 h 10 min ^a	∞	∞		

The computations are made on a PC Pentium Pro 200 MHz with 128 MB RAM.
 (^a This was computed by a 500 MHz Alpha workstation with 1 GB RAM.)

The conclusion of Table 1 is that the old Gb is still faster than other systems, and that FGb is much more faster than Gb. For small sets the difference is small but for big sets like Cyclic 9 there is a huge difference. For Cyclic 9 the algorithm generates a 292051×317850 -matrix.

Table 2: Academic examples \mathbb{Z}

Set	F_4	Gb	Singular	Magma	$ maxcoeff $
Cyclic 6	0.3 s	3.2 s	5.3 s	2.6 s	96
Hom Cyclic 7	54.2 s	1 h 32 min	10 h 35 min	36 min 42 s	96
Cyclic 7	39.7 s	5 h 17 min	∞	=	96
Cyclic 8	24 min 4 s	∞	∞	∞	202
Cyclic 9	18 days	∞	∞	∞	800

Cyclic 9 for big integers is an example of huge computation:

- 3 Processors Pentium Pro 200 MHz, 512 MB RAM, + 1 Alpha 400 MHz 570 MB RAM.
- Total sequential CPU time: 18 days.
- Size of the file of coefficients in the output (binary): 1660 MB.
- The result contains 1344 polynomials with 1000 monomials and 800 digits numbers.

This success is also a failure in some sense: the size of the output is so big that we cannot do anything with this result. That is to say we are now near to the intrinsic complexity of Gröbner bases computations. On the other side, the output is very big because the coefficients are big and floating point computation would not suffer from this exponential growth.

5 Outlook, Questions, Research

My research focuses on the following points and questions:

1. Koblitz's Generalization of Polly Cracker.
 - How can this idea be realized in an applicable PKC-scheme?
 - What are the security parameters?
 - Are the attacks proposed by Lenstra and Mora correct and efficient?
2. The generalization of Wagner's and Magyarik's cryptosystem to the free associative algebra.
 - How can this idea be realized in an applicable PKC-scheme?
 - What are the security parameters?
 - Are there properties of Gröbner bases in the noncommutative which support the intractability of the one-way function?
3. An implementation of Faugère's ideas in the noncommutative case.
 - First, the implementation of the noncommutative structures.
 - The implementation of the original procedure to construct a Gröbner basis in the noncommutative case.
 - How to handle the case where the procedure does not terminate?
 - Is there a way to apply Faugère's ideas for this construction.

References

- [Buc65] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, Leopold-Franzens-Universität Innsbruck, 1965.
- [BW93] Thomas Becker and Volker Weisfennig. *Gröbner Bases - A Computational Approach to Commutative Algebra*. Springer-Verlag, New York, 1993.
- [CKPS00] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In B. Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000*, Lecture Notes in Computer Science LNCS 1070, page 392 ff. Springer-Verlag LNCS 1070, 2000.
- [Fau99] Jean-Charles Faugère. A new efficient algorithm for computing gröbner bases (\mathbf{f}_4). *Journal of Pure and Applied Algebra* 139, pages 61–88, 1999.
- [Huy86] Dung T. Huynh. The complexity of the ideal membership problem for two subclasses of polynomial ideals. *SIAM J. Computations* 15, pages 581–594, 1986.
- [Kob98] Neal Koblitz. *Algebraic aspects of cryptography*, volume 3 of *Algorithms and Computations in Mathematics*. Springer-Verlag, New York, 1998.
- [MM82] Ernst W. Mayr and Albert R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Adv. Math.* 46, pages 305–329, 1982.
- [Mor] T. Mora. Why you cannot even hope to use gröbner bases in public key cryptography. An open letter.
- [Pat96] Jacques Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): two new families of asymmetric algorithms. In *Advances in Cryptology - EUROCRYPT '96*, Lecture Notes in Computer Science LNCS 1807, pages 33–48. Springer-Verlag, 1996.
- [SK99] Adi Shamir and Aviad Kipnis. Cryptanalysis of the hfe public key cryptosystem. In M. Wiener, editor, *Advances in Cryptology - CRYPTO '99*, Lecture Notes in Computer Science LNCS 1666, page 19 ff. Springer-Verlag, 1999.
- [Ufn80] V.A. Ufnarovskij. Poincaré series of graded algebras. *Math. Notes* 27, pages 12–18, 1980.

- [Ufn90] V.A. Ufnarovskij. Combinatorial and asymptotic methods in algebra. In I.R. Shafarevich A.I. Kostrikin, editor, *Algebra VI*, volume 57 of *Encyclopaedia of Mathematical Sciences*. Springer-Verlag, Berlin-Heidelberg, 1990.
- [WM84] Neal R. Wagner and Marianne R. Magyarik. A public key cryptosystem based on the word problem. In David Chaum G.R. Blakley, editor, *Advances in Cryptology - CRYPTO '84*, page 19 ff. Springer-Verlag LNCS 0196, 1984.