Gröbner Basis Cryptosystems

Martin Kreuzer
Fachbereich Mathematik
Universität Dortmund
martin.kreuzer@uni-dortmund.de
(joint work with Peter Ackermann, now AMB/Aachen)

Special Semester on Gröbner Bases
Workshop D1: Gröbner Bases in Cryptography,
Coding Theory, and Algebraic Combinatorics
3. Mai 2006

Outline of the Talk

Outline of the Talk

1. **Gröbner Bases for Modules over Free Monoid Rings**

Outline of the Talk

1. **Gröbner Bases for Modules over Free Monoid Rings**
2. **Gröbner Bases for Modules over Monoid Rings**

Outline of the Talk

1. **Gröbner Bases for Modules over Free Monoid Rings**
2. **Gröbner Bases for Modules over Monoid Rings**
3. **Polly Cracker Cryptosystems**

Outline of the Talk

1. **Gröbner Bases for Modules over Free Monoid Rings**
2. **Gröbner Bases for Modules over Monoid Rings**
3. **Polly Cracker Cryptosystems**
4. **Gröbner Basis Cryptosystems**

Outline of the Talk

1. **Gröbner Bases for Modules over Free Monoid Rings**
2. **Gröbner Bases for Modules over Monoid Rings**
3. **Polly Cracker Cryptosystems**
4. **Gröbner Basis Cryptosystems**
5. **Examples of Gröbner Basis Cryptosystems**

Outline of the Talk

1. **Gröbner Bases for Modules over Free Monoid Rings**
2. **Gröbner Bases for Modules over Monoid Rings**
3. **Polly Cracker Cryptosystems**
4. **Gröbner Basis Cryptosystems**
5. **Examples of Gröbner Basis Cryptosystems**
6. **Efficiency and Security Considerations**

Outline of the Talk

1. **Gröbner Bases for Modules over Free Monoid Rings**
2. **Gröbner Bases for Modules over Monoid Rings**
3. **Polly Cracker Cryptosystems**
4. **Gröbner Basis Cryptosystems**
5. **Examples of Gröbner Basis Cryptosystems**
6. **Efficiency and Security Considerations**
7. **Further Suggestions**

1 – GB for Modules over Free Monoid Rings

1 – GB for Modules over Free Monoid Rings

Let's fix the notation!

1 – GB for Modules over Free Monoid Rings

Let's fix the notation!

$\Sigma = \{x_1, \ldots, x_n\}$ finite alphabet

Σ^* monoid of words (or terms)

K field

$K[\Sigma^*]$ free monoid ring (= free associative algebra, non-commutative polynomial ring)

1 – GB for Modules over Free Monoid Rings

Let's fix the notation!

$\Sigma = \{x_1, \ldots, x_n\}$ finite alphabet

Σ^* monoid of words (or terms)

K field

$K[\Sigma^*]$ free monoid ring (= free associative algebra, non-commutative polynomial ring)

σ term ordering on Σ^*, i.e. a total well-ordering such that $w_1 \leq_\sigma w_2$ implies $w_3 w_1 w_4 \leq_\sigma w_3 w_2 w_4$ for all $w_1, w_2, w_3, w_4 \in \Sigma^*$

Every non-commutative polynomial $f \in K[\Sigma^*]$ has a unique representation $f = c_1 w_1 + \cdots + c_s w_s$ such that $c_i \in K \setminus \{0\}$ and $w_1 >_\sigma \cdots >_\sigma w_s$ in Σ^*.

Every non-commutative polynomial $f \in K[\Sigma^*]$ has a unique representation $f = c_1 w_1 + \cdots + c_s w_s$ such that $c_i \in K \setminus \{0\}$ and $w_1 >_\sigma \cdots >_\sigma w_s$ in Σ^*.

$\mathrm{LT}_\sigma(f) = w_1$ leading term of f

$\mathrm{LC}_\sigma(f) = c_1$ leading coefficient of f

Every non-commutative polynomial $f \in K[\Sigma^*]$ has a unique representation $f = c_1 w_1 + \cdots + c_s w_s$ such that $c_i \in K \setminus \{0\}$ and $w_1 >_\sigma \cdots >_\sigma w_s$ in Σ^*.

$\mathrm{LT}_\sigma(f) = w_1$ leading term of f

$\mathrm{LC}_\sigma(f) = c_1$ leading coefficient of f

Given a right ideal $I \subseteq K[\Sigma^*]$, we let

$\mathrm{LT}_\sigma(I) = \langle \mathrm{LT}_\sigma(f) \mid f \in I \setminus \{0\} \rangle_\varrho$ be its right leading term ideal.

A set $\{f_i \mid i \in \Lambda\}$ is called a (right) Gröbner basis of I if $\mathrm{LT}_\sigma(I) = \langle \mathrm{LT}_\sigma(f_i) \mid i \in \Lambda \rangle_\varrho$.

Theorem 1.1 (Macaulay's Basis Theorem)
The residue classes of the terms in

$$\mathcal{O}_\sigma(I) = \Sigma^* \setminus \mathrm{LT}_\sigma(I)$$

form a K*-basis of* $K[\Sigma^*]/I$*.*

Theorem 1.1 (Macaulay's Basis Theorem)
The residue classes of the terms in

$$\mathcal{O}_\sigma(I) = \Sigma^* \setminus \mathrm{LT}_\sigma(I)$$

form a K*-basis of* $K[\Sigma^*]/I$.

For every $f \in K[\Sigma^*]$, there exists a unique normal form $\mathrm{NF}_{\sigma,I}(f) \in \langle \mathcal{O}_\sigma(I) \rangle_K$ such that $f - \mathrm{NF}_{\sigma,I}(f) \in I$.

The normal form can be computed by using the term rewriting system $\xrightarrow{G}$ defined by a σ-Gröbner basis G of I.

Theorem 1.1 (Macaulay's Basis Theorem)
The residue classes of the terms in

$$\mathcal{O}_\sigma(I) = \Sigma^* \setminus \mathrm{LT}_\sigma(I)$$

form a K-basis of $K[\Sigma^*]/I$.

For every $f \in K[\Sigma^*]$, there exists a unique normal form $\mathrm{NF}_{\sigma,I}(f) \in \langle \mathcal{O}_\sigma(I) \rangle_K$ such that $f - \mathrm{NF}_{\sigma,I}(f) \in I$.

The normal form can be computed by using the term rewriting system $\xrightarrow{G}$ defined by a σ-Gröbner basis G of I.

A σ-Gröbner basis of I can be enumerated using the Buchberger procedure (Knuth-Bendix completion).

And What About Modules?

Everything generalizes easily to right submodules of free right modules over $K[\Sigma^*]$.

And What About Modules?

Everything generalizes easily to right submodules of free right modules over $K[\Sigma^*]$.

$F_\varrho = \bigoplus_{i=1}^r e_i\, K[\Sigma^*]$ free right $K[\Sigma^*]$-module with basis $e_1, \dots, e_r$

A term in F_ϱ is of the form $e_i\, t$ with $t \in \Sigma^*$.

$\mathbb{T}(F_\varrho)$ is the set of all terms in F_ϱ.

And What About Modules?

Everything generalizes easily to right submodules of free right modules over $K[\Sigma^*]$.

$F_\varrho = \bigoplus_{i=1}^r e_i\, K[\Sigma^*]$ free right $K[\Sigma^*]$-module with basis $e_1, \ldots, e_r$

A term in F_ϱ is of the form $e_i\, t$ with $t \in \Sigma^*$.

$\mathbb{T}(F_\varrho)$ is the set of all terms in F_ϱ.

A module term ordering on $\mathbb{T}(F_\varrho)$ is a total well-ordering τ such that $t_1 \leq_\tau t_2$ implies $t_1 w \leq_\tau t_2 w$ for all $t_1, t_2 \in \mathbb{T}(F_\varrho)$ and $w \in \Sigma^*$.

For every vector $v \in F_\varrho$ we define its leading term $\mathrm{LT}_\tau(v)$ and its leading coefficient $\mathrm{LC}_\tau(v)$ in the obvious way.

Given a right submodule $U \subseteq F_\varrho$, we let

$\mathrm{LT}_\tau(U) = \langle \mathrm{LT}_\tau(v) \mid v \in U \setminus \{0\} \rangle_\varrho$ be its (right) leading term module.

A set of non-zero vectors $\{v_i \mid i \in \Lambda\}$ is called a (right) τ-Gröbner basis of U if $\mathrm{LT}_\tau(U) = \langle \mathrm{LT}_\tau(v_i) \mid i \in \Lambda \rangle_\varrho$.

For every vector $v \in F_\varrho$ we define its leading term $\mathrm{LT}_\tau(v)$ and its leading coefficient $\mathrm{LC}_\tau(v)$ in the obvious way.

Given a right submodule $U \subseteq F_\varrho$, we let

$\mathrm{LT}_\tau(U) = \langle \mathrm{LT}_\tau(v) \mid v \in U \setminus \{0\} \rangle_\varrho$ be its (right) leading term module.

A set of non-zero vectors $\{v_i \mid i \in \Lambda\}$ is called a (right) τ-Gröbner basis of U if $\mathrm{LT}_\tau(U) = \langle \mathrm{LT}_\tau(v_i) \mid i \in \Lambda \rangle_\varrho$.

Theorem 1.2 (Macaulay Basis Theorem for Modules)
The residue classes of the terms in $\mathcal{O}_\tau(U) = \mathbb{T}(F_\varrho) \setminus \mathrm{LT}_\tau(U)$ form a K-basis of the module F_ϱ/U.

Also for modules we can compute normal forms of vectors and have a Buchberger procedure to enumerate a Gröbner basis.

2 – GB for Modules over Monoid Rings

$M = \Sigma^* / \sim_W$ finitely presented monoid, i.e. $\sim_W$ is the equivalence relation generated by finitely many relations $w_i \sim w'_i$ with $w_i, w'_i \in \Sigma^*$ for $i = 1, \ldots, r$.

$K[M] = K[\Sigma^*]/I_M$ monoid ring over M where I_M is the two-sided ideal $I_M = \langle w_1 - w'_1, \ldots, w_r - w'_r \rangle$.

2 – GB for Modules over Monoid Rings

$M = \Sigma^*/\sim_W$ finitely presented monoid, i.e. $\sim_W$ is the equivalence relation generated by finitely many relations $w_i \sim w_i'$ with $w_i, w_i' \in \Sigma^*$ for $i = 1, \ldots, r$.

$K[M] = K[\Sigma^*]/I_M$ monoid ring over M where I_M is the two-sided ideal $I_M = \langle w_1 - w_1', \ldots, w_r - w_r' \rangle$.

Assumption: There is a term ordering σ such that $w_i >_\sigma w_i'$ for $i = 1, \ldots, r$ and such that the term rewriting system $\xrightarrow{W}$ is convergent (i.e. Noetherian/terminating and confluent).

So, $W = \{w_1 - w_1', \ldots, w_r - w_r'\}$ is a two-sided Gröbner basis of I_M.

Then every $f \in K[\Sigma^*]$ can be effectively reduced via $\xrightarrow{W}$ to a unique normal form $\mathrm{NF}_{I_M}(f)$.

Φ finite or countable infinite set

$\overline{F}_{\varrho}$ free right $K[M]$-module with basis $\{\bar{e}_i \mid i \in \Phi\}$

$\overline{U} \subseteq \overline{F}_{\varrho}$ finitely generated right submodule

τ module term ordering on $\mathbb{T}(F_{\varrho})$ that is compatible with σ (i.e. $w_1 <_{\sigma} w_2$ implies $e_i w_1 <_{\tau} e_i w_2$)

By representing every element of M using the normal form of the corresponding word in Σ^*,we can view τ as an ordering on

$$\mathbb{T}(\overline{F}_{\varrho}) = \{\bar{e}_i m \mid i \in \Phi,\, m \in M\}$$

Φ finite or countable infinite set

$\overline{F}_\varrho$ free right $K[M]$-module with basis $\{\bar{e}_i \mid i \in \Phi\}$

$\overline{U} \subseteq \overline{F}_\varrho$ finitely generated right submodule

τ module term ordering on $\mathbb{T}(F_\varrho)$ that is compatible with σ (i.e. $w_1 <_\sigma w_2$ implies $e_i w_1 <_\tau e_i w_2$)

By representing every element of M using the normal form of the corresponding word in Σ^*,we can view τ as an ordering on

$$\mathbb{T}(\overline{F}_\varrho) = \{\bar{e}_i m \mid i \in \Phi,\, m \in M\}$$

Problem: $\bar{e}_i w_1 \leq_\tau \bar{e}_i w_2$ does (in general) not imply $\bar{e}_1 m_1 m_3 \leq_\tau \bar{e}_i m_2 m_3$ for $m_1, m_2, m_3 \in M$ because reductions via $\xrightarrow{W}$ may destroy the inequality for the representing words.

Definition 2.1 $v, w \in \overline{F}_\varrho \setminus \{0\}$

If there exist a term $\bar{e}_i m_1 \in \mathrm{Supp}(w)$ and $m_2 \in M$ such that $\mathrm{LT}_\tau(v) \circ m_2 \equiv \bar{e}_i m_1$, we say that v prefix reduces w to $w' = w - \mathrm{LC}_\tau(v)^{-1}\, v\, m_2$. We write $w \xrightarrow{v}_\pi w'$.

Here $\circ$ denotes the concatenation of the representing words and $\equiv$ is the identity for words.

In this situation we have $\mathrm{LT}_\tau(v m_2) = \mathrm{LT}_\tau(v) \circ m_2$ *a fortiori*.

Definition 2.1 $v, w \in \overline{F}_\varrho \setminus \{0\}$

If there exist a term $\bar{e}_i m_1 \in \mathrm{Supp}(w)$ and $m_2 \in M$ such that $\mathrm{LT}_\tau(v) \circ m_2 \equiv \bar{e}_i m_1$, we say that v prefix reduces w to $w' = w - \mathrm{LC}_\tau(v)^{-1}\, v\, m_2$. We write $w \xrightarrow{v}_\pi w'$.

Here $\circ$ denotes the concatenation of the representing words and $\equiv$ is the identity for words.

In this situation we have $\mathrm{LT}_\tau(vm_2) = \mathrm{LT}_\tau(v) \circ m_2$ *a fortiori*.

$S \subseteq \overline{F}_\varrho$ is called prefix saturated if $vm \xrightarrow{S}_\pi 0$ in one step for all $v \in S$ and $m \in M$.

If S is prefix saturated then $v \xleftrightarrow{S}_\pi 0$ for all $\langle S \rangle_\varrho$.

There exists a procedure for enumerating the prefix saturation of a set $S = \{v\}$.

Definition 2.2 A set G in a right submodule $\overline{U} \subseteq \overline{F}_{\varrho}$ is called a prefix Gröbner basis of $\overline{U}$ if we have $u \xleftrightarrow{G}_{\pi} 0$ for all $u \in \overline{U}$ and if $\xrightarrow{G}$ is confluent.

Definition 2.2 A set G in a right submodule $\overline{U} \subseteq \overline{F}_\varrho$ is called a prefix Gröbner basis of $\overline{U}$ if we have $u \overset{G}{\longleftrightarrow}_\pi 0$ for all $u \in \overline{U}$ and if $\overset{G}{\longrightarrow}$ is confluent.

One can formulate a Buchberger criterion for prefix Gröbner bases and a Buchberger procedure for enumerating a prefix Gröbner basis of a given right submodule of $\overline{F}_\varrho$.

Definition 2.2 A set G in a right submodule $\overline{U} \subseteq \overline{F}_{\varrho}$ is called a prefix Gröbner basis of $\overline{U}$ if we have $u \overset{G}{\longleftrightarrow}_{\pi} 0$ for all $u \in \overline{U}$ and if $\overset{G}{\longrightarrow}$ is confluent.

One can formulate a Buchberger criterion for prefix Gröbner bases and a Buchberger procedure for enumerating a prefix Gröbner basis of a given right submodule of $\overline{F}_{\varrho}$.

Applications:

- submodule membership can be solved effectively
- the subgroup membership problem is equivalent to a right ideal membership problem in $K[M]$
- the conjugator search problem can be solved using a two-sided syzygy computation

3 – Polly Cracker Cryptosystems

In 1994, Fellows and Koblitz suggested the following cryptosystem.

$P = K[x_1, \ldots, x_n]$ commutative polynomial ring

$f_1, \ldots, f_s \in P$ polynomials having a common zero $(a_1, \ldots, a_n) \in K^n$

Public: $f_1, \ldots, f_s$

Secret: $(a_1, \ldots, a_n)$

Encryption: a plaintext unit $m \in K$ is encrypted as
$w = m + f_1 g_1 + \cdots + f_s g_s$ with $g_i \in P$ suitably chosen

Decryption: evaluation yields $w(a_1, \ldots, a_n) = m$

3 – Polly Cracker Cryptosystems

In 1994, Fellows and Koblitz suggested the following cryptosystem.

$P = K[x_1, \ldots, x_n]$ commutative polynomial ring

$f_1, \ldots, f_s \in P$ polynomials having a common zero $(a_1, \ldots, a_n) \in K^n$

Public: $f_1, \ldots, f_s$

Secret: $(a_1, \ldots, a_n)$

Encryption: a plaintext unit $m \in K$ is encrypted as
$w = m + f_1 g_1 + \cdots + f_s g_s$ with $g_i \in P$ suitably chosen

Decryption: evaluation yields $w(a_1, \ldots, a_n) = m$

Security: The attacker can break the cryptosystem if he can compute a Gröbner basis of $I = \langle f_1, \ldots, f_s \rangle$ because $m = \mathrm{NF}_{\sigma,I}(w)$.

Ideals can be constructed which encode hard combinatorial problems so that it is believed to be difficult to compute their Gröbner bases.

Ideals can be constructed which encode hard combinatorial problems so that it is believed to be difficult to compute their Gröbner bases.

Polly Cracker Is Under Attack!

Ideals can be constructed which encode hard combinatorial problems so that it is believed to be difficult to compute their Gröbner bases.

Polly Cracker Is Under Attack!

1. Basic Linear Algebra Attack: The attacker knows $w = m + f_1 g_1 + \cdots + f_s g_s$. Consider the coefficients of $g_1, \ldots, g_s$ as unknowns. All coefficients of the non-constant terms in $f_1 g_1 + \cdots + f_s g_s$ are known. Thus we get a system of linear equations.

Ideals can be constructed which encode hard combinatorial problems so that it is believed to be difficult to compute their Gröbner bases.

Polly Cracker Is Under Attack!

1. Basic Linear Algebra Attack: The attacker knows $w = m + f_1 g_1 + \cdots + f_s g_s$. Consider the coefficients of $g_1, \ldots, g_s$ as unknowns. All coefficients of the non-constant terms in $f_1 g_1 + \cdots + f_s g_s$ are known. Thus we get a system of linear equations.

2. "Intelligent" Linear Algebra Attack: One may guess the terms t occurring in $\mathrm{Supp}(g_i)$ because some of the terms in $t \cdot \mathrm{Supp}(f_j)$ should occur in $\mathrm{Supp}(w)$ if there is not too much cancellation.

3. Differential Attack: Quotients of terms in $\mathrm{Supp}(w)$ allow conclusions about possible terms in $\mathrm{Supp}(g_i)$.

3. Differential Attack: Quotients of terms in $\mathrm{Supp}(w)$ allow conclusions about possible terms in $\mathrm{Supp}(g_i)$.
4. Attack by Characteristic Terms: If there are terms which occur in just one f_i we can recognize multiples of these terms in w and compute the corresponding terms in g_i.

3. Differential Attack: Quotients of terms in $\mathrm{Supp}(w)$ allow conclusions about possible terms in $\mathrm{Supp}(g_i)$.
4. Attack by Characteristic Terms: If there are terms which occur in just one f_i we can recognize multiples of these terms in w and compute the corresponding terms in g_i.
5. Attack by Truncated GB: In order to compute $\mathrm{NF}_{\sigma,I}(w)$, it may be sufficient to find a partial Gröbner basis of I.

3. Differential Attack: Quotients of terms in $\mathrm{Supp}(w)$ allow conclusions about possible terms in $\mathrm{Supp}(g_i)$.

4. Attack by Characteristic Terms: If there are terms which occur in just one f_i we can recognize multiples of these terms in w and compute the corresponding terms in g_i.

5. Attack by Truncated GB: In order to compute $\mathrm{NF}_{\sigma,I}(w)$, it may be sufficient to find a partial Gröbner basis of I.

A more refined version of the cryptosystem suggested by L. Ly and called Polly 2 has been broken recently by R. Steinwandt using a side channel attack.

4 – Gröbner Basis Cryptosystems

$M = \Sigma^* / \sim_W$ finitely presented monoid

$\overline{F}_\varrho = \bigoplus_{i \in \Phi} \bar{e}_i K[M]$ free right module over the monoid ring

σ, τ compatible term orderings

$\overline{U} \subseteq \overline{F}_\varrho$ right submodule

4 – Gröbner Basis Cryptosystems

$M = \Sigma^* / \sim_W$ finitely presented monoid

$\overline{F}_\varrho = \bigoplus_{i\in\Phi} \bar{e}_i K[M]$ free right module over the monoid ring

σ, τ compatible term orderings

$\overline{U} \subseteq \overline{F}_\varrho$ right submodule

Public: $\mathcal{O}_\tau(\overline{U}) = \mathbb{T}(\overline{F}_\varrho) \setminus \mathrm{LT}_\tau(\overline{U})$ (or a subset thereof) and finitely many vectors $u_1, \ldots, u_s \in \overline{U}$

Secret: a prefix Gröbner basis G of $\overline{U}$

Encryption: a plaintext unit is of the form
$m = \bar{e}_{\lambda_1} c_1 w_1 + \cdots + \bar{e}_{\lambda_r} c_r w_r \in \langle \mathcal{O}_\tau(\overline{U}) \rangle_K$ with $\lambda_i \in \Phi$, $c_i \in K$, and $w_i \in M$.

The plaintext unit m is encrypted as $w = m + \bar{u}_1 f_1 + \cdots + \bar{u}_s f_s$ with suitably chosen $f_i \in K[M]$.

Decryption: Using $\xrightarrow{G}$, compute $m = \mathrm{NF}_{\sigma,\overline{U}}(w)$.

The plaintext unit m is encrypted as $w = m + \bar{u}_1 f_1 + \cdots + \bar{u}_s f_s$ with suitably chosen $f_i \in K[M]$.

Decryption: Using $\xrightarrow{G}$, compute $m = \mathrm{NF}_{\sigma,\overline{U}}(w)$.

Security: • The attacker can break the cryptosystem if he can compute a Gröbner basis of $\langle \bar{u}_1, \ldots, \bar{u}_s \rangle_\varrho$.

The plaintext unit m is encrypted as $w = m + \bar{u}_1 f_1 + \cdots + \bar{u}_s f_s$ with suitably chosen $f_i \in K[M]$.

Decryption: Using $\xrightarrow{G}$, compute $m = \mathrm{NF}_{\sigma,\overline{U}}(w)$.

Security: • The attacker can break the cryptosystem if he can compute a Gröbner basis of $\langle \bar{u}_1, \ldots, \bar{u}_s \rangle_\varrho$.

• The advantage of using modules is that the action of M on the set $\{\bar{e}_i \mid i \in \Phi\}$ can encode hard combinatorial or number theoretic problems.

The plaintext unit m is encrypted as $w = m + \bar{u}_1 f_1 + \cdots + \bar{u}_s f_s$ with suitably chosen $f_i \in K[M]$.

Decryption: Using $\xrightarrow{G}$, compute $m = \mathrm{NF}_{\sigma,\overline{U}}(w)$.

Security: • The attacker can break the cryptosystem if he can compute a Gröbner basis of $\langle \bar{u}_1, \ldots, \bar{u}_s \rangle_\varrho$.

• The advantage of using modules is that the action of M on the set $\{\bar{e}_i \mid i \in \Phi\}$ can encode hard combinatorial or number theoretic problems.

• The free module $\overline{F}_\varrho$ is not required to be finitely generated. Any concrete calculation will involve only finitely many components.

5 – Examples of Gröbner Basis Cryptosystems

5 – Examples of Gröbner Basis Cryptosystems

Example 5.1 (Polly Cracker Cryptosystems)
If we use the monoid $M = \mathbb{N}^n$, the free module $\overline{F}_\varrho = K[M] = K[x_1, \ldots, x_n]$, and the submodule $\overline{U} = \langle x_1 - a_1, \ldots, x_n - a_n \rangle$, we obtain the original Polly Cracker Cryptosystem.

The set $\mathcal{O}_\tau(\overline{U})$ is equal to $\{1\}$. Thus a plaintext unit is just an element of K.

The secret Gröbner basis is $\{x_1 - a_1, \ldots, x_n - a_n\}$.

The decryption yields the same result because $\mathrm{NF}_{\tau,\overline{U}}(w) = w(a_1, \ldots, a_n)$.

Example 5.2 $K = \mathbb{F}_2$ and $M = \mathbb{N}^2$ yields $K[M] = \mathbb{F}_2[x, y]$

$p, q \gg 0$ distinct prime numbers, $n = pq$, and $\Pi = (\mathbb{Z}/n\mathbb{Z})^\times$

$\overline{F}_\varrho = \bigoplus_{i=0}^{n-1} e_i K[x, y]$ and $\tau =$ `DegRevLexPos`

Choose $\varepsilon \in (\mathbb{Z}/(p-1)(q-1)\mathbb{Z})^*$ and compute $d = \varepsilon^{-1}$.

Example 5.2 $K = \mathbb{F}_2$ and $M = \mathbb{N}^2$ yields $K[M] = \mathbb{F}_2[x, y]$

$p, q \gg 0$ distinct prime numbers, $n = pq$, and $\Pi = (\mathbb{Z}/n\mathbb{Z})^\times$

$\overline{F}_\varrho = \bigoplus_{i=0}^{n-1} e_i K[x, y]$ and $\tau = \texttt{DegRevLexPos}$

Choose $\varepsilon \in (\mathbb{Z}/(p-1)(q-1)\mathbb{Z})^*$ and compute $d = \varepsilon^{-1}$.

Public: $\overline{F}_\varrho$ (and thus n), $\mathcal{O}_\tau(\overline{U}) = \{e_0, \ldots, e_{n-1}\}$, the number ε, and the vectors

$$\{u_1, \ldots, u_s\} = \{\bar{e}_i x - e_{i^\varepsilon \bmod n},\ e_i xy - e_i \mid i = 0, \ldots, n-1\}$$

Example 5.2 $K = \mathbb{F}_2$ and $M = \mathbb{N}^2$ yields $K[M] = \mathbb{F}_2[x, y]$

$p, q \gg 0$ distinct prime numbers, $n = pq$, and $\Pi = (\mathbb{Z}/n\mathbb{Z})^\times$

$\overline{F}_\varrho = \bigoplus_{i=0}^{n-1} e_i K[x, y]$ and $\tau =$ `DegRevLexPos`

Choose $\varepsilon \in (\mathbb{Z}/(p-1)(q-1)\mathbb{Z})^*$ and compute $d = \varepsilon^{-1}$.

Public: $\overline{F}_\varrho$ (and thus n), $\mathcal{O}_\tau(\overline{U}) = \{e_0, \ldots, e_{n-1}\}$, the number ε, and the vectors

$$\{u_1, \ldots, u_s\} = \{\bar{e}_i x - e_{i^\varepsilon \bmod n},\ e_i xy - e_i \mid i = 0, \ldots, n-1\}$$

Secret: The secret key consists of the primes p, q and the number d. Equivalently, it is the τ-Gröbner basis

$$G = \{u_1, \ldots, u_s\} \cup \{e_i y - e_{i^d \bmod n} \mid i = 0, \ldots, n-1\} \quad \text{of} \quad \overline{U} = \langle G \rangle$$

Encryption: A plaintext unit is a vector $e_m \in \mathcal{O}_\tau(\overline{U})$. To encrypt it, we form

$$w = e_m + (e_m xy - e_m) - (e_m x - e_{m^\varepsilon \bmod n})y = e_{m^\varepsilon \bmod n} y$$

Encryption: A plaintext unit is a vector $e_m \in \mathcal{O}_\tau(\overline{U})$. To encrypt it, we form

$$w = e_m + (e_m xy - e_m) - (e_m x - e_{m^\varepsilon \bmod n})y = e_{m^\varepsilon \bmod n} y$$

Decryption: Compute $\mathrm{NF}_{\tau,\overline{U}}(w) = e_{m^{\varepsilon d} \bmod n} = e_m$.

Encryption: A plaintext unit is a vector $e_m \in \mathcal{O}_\tau(\overline{U})$. To encrypt it, we form

$$w = e_m + (e_m xy - e_m) - (e_m x - e_{m^\varepsilon \bmod n})y = e_{m^\varepsilon \bmod n} y$$

Decryption: Compute $\mathrm{NF}_{\tau,\overline{U}}(w) = e_{m^{\varepsilon d} \bmod n} = e_m$.

Security: The attacker can compute the Gröbner basis if and only if he can factor $n = pq$ and find d.

Encryption: A plaintext unit is a vector $e_m \in \mathcal{O}_\tau(\overline{U})$. To encrypt it, we form

$$w = e_m + (e_m xy - e_m) - (e_m x - e_{m^\varepsilon \bmod n})y = e_{m^\varepsilon \bmod n} y$$

Decryption: Compute $\mathrm{NF}_{\tau,\overline{U}}(w) = e_{m^{\varepsilon d} \bmod n} = e_m$.

Security: The attacker can compute the Gröbner basis if and only if he can factor $n = pq$ and find d.

This is nothing but the GB version of the **RSA cryptosystem**!

Example 5.3 $K = \mathbb{F}_2$, $M = \mathbb{N}$, and $K[M] = \mathbb{F}_2[x]$

$p \gg 0$ prime number, g generator of $(\mathbb{Z}/p\mathbb{Z})^\times$

$\overline{F}_\varrho = \bigoplus_{i=1}^{p-1} \varepsilon_i K[x] \oplus \bigoplus_{j=1}^{p-1} e_j K[x]$ and $\tau = \mathtt{DegPos}$ with $\varepsilon_i > e_j$

Choose a number $a \in \{1, \ldots, p-1\}$ and compute $b = g^a \bmod p$.

Example 5.3 $K = \mathbb{F}_2$, $M = \mathbb{N}$, and $K[M] = \mathbb{F}_2[x]$

$p \gg 0$ prime number, g generator of $(\mathbb{Z}/p\mathbb{Z})^\times$

$\overline{F}_\varrho = \bigoplus_{i=1}^{p-1} \varepsilon_i K[x] \oplus \bigoplus_{j=1}^{p-1} e_j K[x]$ and $\tau = \mathtt{DegPos}$ with $\varepsilon_i > e_j$

Choose a number $a \in \{1, \ldots, p-1\}$ and compute $b = g^a \bmod p$.

Public: $\overline{F}_\varrho$ (and thus p), $\mathcal{O}_\tau(\overline{U}) = \{e_1, \ldots, e_{p-1}\}$, the number b, and the vectors

$$\{u_1, \ldots, u_s\} = \{\varepsilon_1 - e_1\} \cup \{\varepsilon_i x - \varepsilon_{gi},\ e_j x - e_{bj} \mid i, j = 1, \ldots, p-1\}$$

where all indices are computed modulo p.

Example 5.3 $K = \mathbb{F}_2$, $M = \mathbb{N}$, and $K[M] = \mathbb{F}_2[x]$

$p \gg 0$ prime number, g generator of $(\mathbb{Z}/p\mathbb{Z})^\times$

$\overline{F}_\varrho = \bigoplus_{i=1}^{p-1} \varepsilon_i K[x] \oplus \bigoplus_{j=1}^{p-1} e_j K[x]$ and $\tau = \texttt{DegPos}$ with $\varepsilon_i > e_j$

Choose a number $a \in \{1, \ldots, p-1\}$ and compute $b = g^a \bmod p$.

Public: $\overline{F}_\varrho$ (and thus p), $\mathcal{O}_\tau(\overline{U}) = \{e_1, \ldots, e_{p-1}\}$, the number b, and the vectors

$$\{u_1, \ldots, u_s\} = \{\varepsilon_1 - e_1\} \cup \{\varepsilon_i x - \varepsilon_{gi},\ e_j x - e_{bj} \mid i, j = 1, \ldots, p-1\}$$

where all indices are computed modulo p.

Secret: The number a, or equivalently the τ-Gröbner basis

$$G = \{u_1, \ldots, u_s\} \cup \{\varepsilon_i - e_{i^a} \mid i = 1, \ldots, p-1\} \quad \text{of} \quad \overline{U} = \langle G \rangle$$

Encryption: A plaintext unit is of the form $e_1 + e_m$ with $m \in \{1, \ldots, p-1\}$. Use the following variant of the GB cryptosystem: choose a random number k, form $(e_1 + e_m)x^k$, and send $w = \varepsilon_{g^k} + e_{mb^k} \in (\varepsilon_1 + e_m)x^k + \langle u_1, \ldots, u_s \rangle_\varrho$.

Encryption: A plaintext unit is of the form $e_1 + e_m$ with $m \in \{1, \ldots, p-1\}$. Use the following variant of the GB cryptosystem: choose a random number k, form $(e_1 + e_m)x^k$, and send $w = \varepsilon_{g^k} + e_{mb^k} \in (\varepsilon_1 + e_m)x^k + \langle u_1, \ldots, u_s \rangle_\varrho$.

Decryption: First compute $\mathrm{NF}_{\tau,\overline{U}} = e_{b^k} + e_{mb^k}$. Since $e_{b^k} + e_{mb^k} \overset{G}{\longleftrightarrow} (e_1 + e_m)x^k$, we have to "divide" this vector by x^k. To this end, it suffices to compute $m = (mb^k)/b^k$ and to form e_m.

Encryption: A plaintext unit is of the form $e_1 + e_m$ with $m \in \{1, \ldots, p-1\}$. Use the following variant of the GB cryptosystem: choose a random number k, form $(e_1 + e_m)x^k$, and send $w = \varepsilon_{g^k} + e_{mb^k} \in (\varepsilon_1 + e_m)x^k + \langle u_1, \ldots, u_s \rangle_\varrho$.

Decryption: First compute $\mathrm{NF}_{\tau,\overline{U}} = e_{b^k} + e_{mb^k}$. Since $e_{b^k} + e_{mb^k} \overset{G}{\longleftrightarrow} (e_1 + e_m)x^k$, we have to "divide" this vector by x^k. To this end, it suffices to compute $m = (mb^k)/b^k$ and to form e_m.

Security: This cryptosystem can be broken if the attacker is able to compute the discrete logarithm a of $b = g^a$ or k of g^k. In the GB version, the reduction $\varepsilon_{g^k} \xrightarrow{u_i} \cdots \xrightarrow{u_j} x^k \varepsilon_1 \xrightarrow{u_1} x^k e_1$ would take $k \gg 0$ steps. If one knows a, one can get rid of ε_{g^k} by using just one reduction step $\varepsilon_{g^k} \longrightarrow e_{g^{ka}} = e_{b^k}$.

Encryption: A plaintext unit is of the form $e_1 + e_m$ with $m \in \{1, \ldots, p-1\}$. Use the following variant of the GB cryptosystem: choose a random number k, form $(e_1 + e_m)x^k$, and send $w = \varepsilon_{g^k} + e_{mb^k} \in (\varepsilon_1 + e_m)x^k + \langle u_1, \ldots, u_s \rangle_\varrho$.

Decryption: First compute $\mathrm{NF}_{\tau,\overline{U}} = e_{b^k} + e_{mb^k}$. Since $e_{b^k} + e_{mb^k} \overset{G}{\longleftrightarrow} (e_1 + e_m)x^k$, we have to "divide" this vector by x^k. To this end, it suffices to compute $m = (mb^k)/b^k$ and to form e_m.

Security: This cryptosystem can be broken if the attacker is able to compute the discrete logarithm a of $b = g^a$ or k of g^k. In the GB version, the reduction $\varepsilon_{g^k} \xrightarrow{u_i} \cdots \xrightarrow{u_j} x^k \varepsilon_1 \xrightarrow{u_1} x^k e_1$ would take $k \gg 0$ steps. If one knows a, one can get rid of ε_{g^k} by using just one reduction step $\varepsilon_{g^k} \longrightarrow e_{g^{ka}} = e_{b^k}$.

This is nothing but the GB version of the **ElGamal** cryptosystem!

Further Examples of GB Cryptosystems

Further Examples of GB Cryptosystems

- Le van Ly's cryptosystem Polly 2 is a variant using commutative polynomials

Further Examples of GB Cryptosystems

- Le van Ly's cryptosystem Polly 2 is a variant using commutative polynomials
- Tapan Rai's cryptosystem uses two-sided Gröbner bases of ideals in $K[\Sigma^*]$, but is otherwise identical.

Further Examples of GB Cryptosystems

- Le van Ly's cryptosystem Polly 2 is a variant using commutative polynomials
- Tapan Rai's cryptosystem uses two-sided Gröbner bases of ideals in $K[\Sigma^*]$, but is otherwise identical.
- Also the braid group based cryptosystems of Ko-Lee *et al.* and of Anshel-Anshel-Goldfeld can be viewed as Gröbner basis cryptosystems, where the group elements act on the standard basis vectors by conjugation on the index.

6 – Efficiency and Security Considerations

Efficiency. One difficulty in constructing an efficient example of a GB cryptosystem is the possibility of exponential support growth during the normal form computation. Possible countermeasures include:

6 – Efficiency and Security Considerations

Efficiency. One difficulty in constructing an efficient example of a GB cryptosystem is the possibility of exponential support growth during the normal form computation. Possible countermeasures include:

- many generators are binomials

6 – Efficiency and Security Considerations

Efficiency. One difficulty in constructing an efficient example of a GB cryptosystem is the possibility of exponential support growth during the normal form computation. Possible countermeasures include:

- many generators are binomials
- determine individual coefficients of the normal form by applying suitable linear functionals

6 – Efficiency and Security Considerations

Efficiency. One difficulty in constructing an efficient example of a GB cryptosystem is the possibility of exponential support growth during the normal form computation. Possible countermeasures include:

- many generators are binomials
- determine individual coefficients of the normal form by applying suitable linear functionals

Linear Algebra Attacks. The various types of linear algebra attacks can be rendered infeasible in the following ways:

- use a module of very large rank

- use a large set $\mathcal{O}_\tau(\overline{U})$ to make the ciphertext statistically similar to the plaintext

- use a large set $\mathcal{O}_\tau(\overline{U})$ to make the ciphertext statistically similar to the plaintext
- over a (not too big) group ring many products $(e_i t)t'$ will give the same term; the corresponding coefficients cannot be recovered

• use a large set $\mathcal{O}_\tau(\overline{U})$ to make the ciphertext statistically similar to the plaintext

• over a (not too big) group ring many products $(e_i t)t'$ will give the same term; the corresponding coefficients cannot be recovered

• in a group ring every term is a multiple of any other term

• use a large set $\mathcal{O}_\tau(\overline{U})$ to make the ciphertext statistically similar to the plaintext

• over a (not too big) group ring many products $(e_i t)t'$ will give the same term; the corresponding coefficients cannot be recovered

• in a group ring every term is a multiple of any other term

Chosen Ciphertext Attacks. In the proposed system the receiver cannot detect invalid cyphertexts. Moreover, the decryption is K-linear. Using a hash function we can overcome this problem:

• use a large set $\mathcal{O}_\tau(\overline{U})$ to make the ciphertext statistically similar to the plaintext

• over a (not too big) group ring many products $(e_i t)t'$ will give the same term; the corresponding coefficients cannot be recovered

• in a group ring every term is a multiple of any other term

Chosen Ciphertext Attacks. In the proposed system the receiver cannot detect invalid cyphertexts. Moreover, the decryption is K-linear. Using a hash function we can overcome this problem:

• append suitable random values to the message ("message padding")

• compute a hash value of the padded message

• transmit the cyphertext of the message, the ciphertext of the padding, and the hash value

7 – Further Suggestions

Increasing the Security.

7 – Further Suggestions

Increasing the Security.

• The Gröbner basis of the module $\langle u_1, \ldots, u_s \rangle_\varrho$ generated by the public vectors need not be finite. A truncated GB computation should yield no "simple" elements in the module.

7 – Further Suggestions

Increasing the Security.

• The Gröbner basis of the module $\langle u_1, \ldots, u_s \rangle_\varrho$ generated by the public vectors need not be finite. A truncated GB computation should yield no "simple" elements in the module.

• If we work with two-sided ideals and modules, the linear algebra attack will yield a system of quadratic equations for the unknown coefficients.

7 – Further Suggestions

Increasing the Security.

• The Gröbner basis of the module $\langle u_1, \ldots, u_s \rangle_\varrho$ generated by the public vectors need not be finite. A truncated GB computation should yield no "simple" elements in the module.

• If we work with two-sided ideals and modules, the linear algebra attack will yield a system of quadratic equations for the unknown coefficients.

• We should try to give a security certificate: if you can solve this instance, then you can also solve the following (supposedly difficult) computational problem ...

Generating New Hard Instances.

Generating New Hard Instances.

• Find monoid or group rings having ideals whose Gröbner bases are difficult to compute.

Generating New Hard Instances.

- Find monoid or group rings having ideals whose Gröbner bases are difficult to compute.

- Encode a hard instance of an action of a group on a set by letting the group act on the standard basis vectors of a free module

Generating New Hard Instances.

• Find monoid or group rings having ideals whose Gröbner bases are difficult to compute.

• Encode a hard instance of an action of a group on a set by letting the group act on the standard basis vectors of a free module

• Use ideals or submodules for which $\mathcal{O}_\tau(\overline{U})$ is "large enough" to allow the encryption of sizable plaintext units. This decreases the message expansion ratio.

Generating New Hard Instances.

• Find monoid or group rings having ideals whose Gröbner bases are difficult to compute.

• Encode a hard instance of an action of a group on a set by letting the group act on the standard basis vectors of a free module

• Use ideals or submodules for which $\mathcal{O}_\tau(\overline{U})$ is "large enough" to allow the encryption of sizable plaintext units. This decreases the message expansion ratio.

• Manufacture the encryption procedure such that the likelihood of cancellations in the computation of $w = m + u_1 f_1 + \cdots + u_s f_s$ is maximized. Use finite groups of "medium size".

References

1. P. Ackermann and M. Kreuzer, Gröbner basis cryptosystems, AAECC (2006), available *"online first"*

2. Boo Barkee et al., Why you cannot even hope to use Gröbner bases in public key cryptography: an open letter to a scientist who failed and a challenge to those who have not yet failed, J. Symb. Comput. 18 (1994), 497–501

3. M. Fellows and N. Koblitz, Combinatorial cryptosystems galore!, Contemp. Math. 168 (1994), 51–61

4. L. Ly, Polly two – a new algebraic polynomial-based public-key scheme, AAECC (2006), to appear

5. K. Madlener and B. Reinert, String rewriting and Gröbner bases – a general approach to monoid and group rings, in: Progress Comp. Sci. Appl. Logic **15**, Birkäuser 1998, 127–150

6. T. Mora, Gröbner bases in non-commutative algebras, in: Lect. Notes Comp. Sci. 358, Springer 1989, 150–161

5. K. Madlener and B. Reinert, String rewriting and Gröbner bases – a general approach to monoid and group rings, in: Progress Comp. Sci. Appl. Logic **15**, Birkäuser 1998, 127–150

6. T. Mora, Gröbner bases in non-commutative algebras, in: Lect. Notes Comp. Sci. 358, Springer 1989, 150–161

Thank You for Your Attention!