Improved Gröbner basis computation with applications in cryptography

Christian Eder
joint work with: John Perry, Justin Gash, Bjarke Roune Hammersholt and Jean-Charles Faugère

POLSYS Team, UPMC, Paris, France

June 25, 2013
Improvement 1: Signature-based Gröbner Basis algorithms

Improvement 2: Specialized Gaussian Elimination

Use GB algorithms in algebraic cryptanalysis
Definition

$G = \{g_1, \ldots, g_r\}$ is a **Gröbner Basis** for $I = \langle f_1, \ldots, f_m \rangle$ if

1. $G \subset I$ and
2. $\langle \text{lm}(g_1), \ldots, \text{lm}(g_r) \rangle = \langle \text{lm}(f) \mid f \in I \rangle$. 
Gröbner Basis basics

**Definition**

$G = \{g_1, \ldots, g_r\}$ is a Gröbner Basis for $I = \langle f_1, \ldots, f_m \rangle$ if

1. $G \subseteq I$ and
2. $\langle \text{lm}(g_1), \ldots, \text{lm}(g_r) \rangle = \langle \text{lm}(f) \mid f \in I \rangle$.

**Satz (Buchberger’s Criterion)**

The following are equivalent:

1. $G$ is a Gröbner Basis for $\langle G \rangle$.

2. For all $f, g \in G$ it holds that $\text{spol}(f, g) \xrightarrow{G} 0$, where

$$\text{spol}(f, g) = \text{lc}(g) \frac{\text{lcm}(\text{lm}(f), \text{lm}(g))}{\text{lm}(f)} f - \text{lc}(f) \frac{\text{lcm}(\text{lm}(f), \text{lm}(g))}{\text{lm}(g)} g.$$
Buchberger’s Algorithm

Input: Ideal $I = \langle f_1, \ldots, f_m \rangle$
Output: Gröbner Basis $G$ for $I$

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \ldots, m\}$
3. $P \leftarrow \{(f_i, f_j) \mid f_i, f_j \in G, i > j\}$
Buchberger’s Algorithm

Input: Ideal \( I = \langle f_1, \ldots, f_m \rangle \)

Output: Gröbner Basis \( G \) for \( I \)

1. \( G \leftarrow \emptyset \)
2. \( G \leftarrow G \cup \{f_i\} \) for all \( i \in \{1, \ldots, m\} \)
3. \( P \leftarrow \{(f_i, f_j) \mid f_i, f_j \in G, i > j\} \)
4. While \( P \neq \emptyset \)
   (a) Choose \( (f, g) \in P \), \( P \leftarrow P \setminus \{(f, g)\} \)
   (b) \( h \leftarrow \text{spol}(f, g) \)
5. Return \( G \)
Buchberger’s Algorithm

Input: Ideal $I = \langle f_1, \ldots, f_m \rangle$

Output: Gröbner Basis $G$ for $I$

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \ldots, m\}$
3. $P \leftarrow \{(f_i, f_j) \mid f_i, f_j \in G, i > j\}$
4. While $P \neq \emptyset$
   (a) Choose $(f, g) \in P$, $P \leftarrow P \setminus \{(f, g)\}$
   (b) $h \leftarrow \text{spol}(f, g)$
      (i) If $h \rightarrow 0$

5. Return $G$
Buchberger’s Algorithm

Input: Ideal \( I = \langle f_1, \ldots, f_m \rangle \)

Output: Gröbner Basis \( G \) for \( I \)

1. \( G \leftarrow \emptyset \)

2. \( G \leftarrow G \cup \{ f_i \} \) for all \( i \in \{1, \ldots, m\} \)

3. \( P \leftarrow \{(f_i, f_j) \mid f_i, f_j \in G, i > j\} \)

4. While \( P \neq \emptyset \)
   
   (a) Choose \( (f, g) \in P, P \leftarrow P \setminus \{(f, g)\} \)
   
   (b) \( h \leftarrow \text{spol}(f, g) \)
      
     (i) If \( h \xrightarrow{G} 0 \)
      
     (ii) If \( h \xrightarrow{G} r \neq 0 \)
Buchberger’s Algorithm

Input: Ideal $I = \langle f_1, \ldots, f_m \rangle$
Output: Gröbner Basis $G$ for $I$

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{ f_i \}$ for all $i \in \{1, \ldots, m\}$
3. $P \leftarrow \{(f_i, f_j) \mid f_i, f_j \in G, i > j\}$
4. While $P \neq \emptyset$
   (a) Choose $(f, g) \in P$, $P \leftarrow P \setminus \{(f, g)\}$
   (b) $h \leftarrow \text{spol}(f, g)$
      (i) If $h \xrightarrow{G} 0$
      (ii) If $h \xrightarrow{G} r \neq 0$
   $P \leftarrow P \cup \{(r, g) \mid g \in G\}$
   $G \leftarrow G \cup \{r\}$
5. Return $G$
Buchberger’s Algorithm

Input: Ideal $I = \langle f_1, \ldots, f_m \rangle$
Output: Gröbner Basis $G$ for $I$

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{ f_i \}$ for all $i \in \{1, \ldots, m\}$
3. $P \leftarrow \{ (f_i, f_j) \mid f_i, f_j \in G, i > j \}$
4. While $P \neq \emptyset$
   (a) Choose $(f, g) \in P$, $P \leftarrow P \setminus \{(f, g)\}$
   (b) $h \leftarrow \text{spol}(f, g)$
      (i) If $h \xrightarrow{G} 0 \Rightarrow \text{no new information}$
      (ii) If $h \xrightarrow{G} r \neq 0 \Rightarrow \text{new information}$
         $P \leftarrow P \cup \{(r, g) \mid g \in G\}$
         $G \leftarrow G \cup \{r\}$
5. Return $G$
How to predict zero reductions?

Example

Let \( I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z] \) be given where \( g_1 = xy - z^2 \), \( g_2 = y^2 - z^2 \), and let \( < \) be the graded reverse lexicographical ordering.
How to predict zero reductions?

Example

Let \( I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z] \) be given where \( g_1 = xy - z^2 \), \( g_2 = y^2 - z^2 \), and let \( \prec \) be the graded reverse lexicographical ordering.

\[
\text{spol}(g_2, g_1) = xg_2 - yg_1 = xy^2 - xz^2 - xy^2 + yz^2 = -xz^2 + yz^2,
\]

so it reduces w.r.t. \( G \) to \( g_3 = xz^2 - yz^2 \).
Example

Let \( I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z] \) be given where \( g_1 = xy - z^2 \), \( g_2 = y^2 - z^2 \), and let \(<\) be the graded reverse lexicographical ordering.

\[
\text{spol}(g_2, g_1) = xg_2 - yg_1 = xy^2 - xz^2 - xy^2 + yz^2
\]
\[
= -xz^2 + yz^2,
\]
so it reduces w.r.t. \( G \) to \( g_3 = xz^2 - yz^2 \).

\[
\text{spol}(g_3, g_1) = xyz^2 - y^2z^2 - xyz^2 + z^4 = -y^2z^2 + z^4.
\]
How to predict zero reductions?

Example

Let \( I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z] \) be given where \( g_1 = xy - z^2 \), \( g_2 = y^2 - z^2 \), and let \(<\) be the graded reverse lexicographical ordering.

\[
\text{spol}(g_2, g_1) = xg_2 - yg_1 = xy^2 - xz^2 - xy^2 + yz^2 = -xz^2 + yz^2,
\]

so it reduces w.r.t. \( G \) to \( g_3 = xz^2 - yz^2 \).

\[
\text{spol}(g_3, g_1) = xyz^2 - y^2z^2 - xyz^2 + z^4 = -y^2z^2 + z^4.
\]

We can reduce even further with \( z^2 \cdot g_2 \):

\[
-y^2z^2 + z^4 + y^2z^2 - z^4 = 0.
\]
How to predict zero reductions?

**Example**

Let \( I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z] \) be given where \( g_1 = xy - z^2 \), \( g_2 = y^2 - z^2 \), and let \(<\) be the graded reverse lexicographical ordering.

\[
\text{spol}(g_2, g_1) = xg_2 - yg_1 = xy^2 - xz^2 - xy^2 + yz^2 = -xz^2 + yz^2,
\]

so it reduces w.r.t. \( G \) to \( g_3 = xz^2 - yz^2 \).

\[
\text{spol}(g_3, g_1) = xyz^2 - y^2 z^2 - xyz^2 + z^4 = -y^2 z^2 + z^4.
\]

We can reduce even further with \( z^2 g_2 \):

\[
-y^2 z^2 + z^4 + y^2 z^2 - z^4 = 0.
\]

⇒ How can we discard such zero reductions in advance?
Signatures of polynomials

Let \( I = \langle f_1, \ldots, f_m \rangle \).

**Idea:** Give each \( f \in I \) a bit more structure:

1. Let \( R_m \) be generated by \( e_1, \ldots, e_m \), \( \preceq \) a well-ordering on the monomials of \( R_m \), and let \( \pi: R_m \to R \) such that \( \pi(e_i) = f_i \) for all \( i \).
2. Each \( p \in I \) can be represented by \( s = m \sum_{i=1}^{m} h_i e_i \in R_m \) such that \( p = \pi(s) \).
3. A signature of \( p \) is given by \( \text{sig}(p) = \text{lm}(\prec)(s) \) with \( p = \pi(s) \).
4. A minimal signature of \( p \) exists due to \( \prec \).
Signatures of polynomials

Let \( I = \langle f_1, \ldots, f_m \rangle \).

**Idea:** Give each \( f \in I \) a bit more structure:

1. Let \( R^m \) be generated by \( e_1, \ldots, e_m \), \( \prec \) a well-ordering on the monomials of \( R^m \), and let \( \pi : R^m \rightarrow R \) such that

\[
\pi(e_i) = f_i \text{ for all } i.
\]
Signatures of polynomials

Let \( I = \langle f_1, \ldots, f_m \rangle \).

**Idea:** Give each \( f \in I \) a bit more structure:

1. Let \( R^m \) be generated by \( e_1, \ldots, e_m \), \( \prec \) a well-ordering on the monomials of \( R^m \), and let \( \pi : R^m \rightarrow R \) such that
   \[
   \pi(e_i) = f_i \text{ for all } i.
   \]

2. Each \( p \in I \) can be represented by an
   \[
   s = \sum_{i=1}^{m} h_i e_i \in R^m \text{ such that } p = \pi(s).
   \]
Signatures of polynomials

Let \( I = \langle f_1, \ldots, f_m \rangle \).

**Idea:** Give each \( f \in I \) a bit more structure:

1. Let \( R^m \) be generated by \( e_1, \ldots, e_m \), \( \prec \) a well-ordering on the monomials of \( R^m \), and let \( \pi : R^m \to R \) such that
   \[
   \pi(e_i) = f_i \text{ for all } i.
   \]

2. Each \( p \in I \) can be represented by an
   \[
   s = \sum_{i=1}^{m} h_i e_i \in R^m \text{ such that } p = \pi(s).
   \]

3. **A signature** of \( p \) is given by
   \[
   \text{sig}(p) = \text{lm}_{\prec}(s) \text{ with } p = \pi(s).
   \]
Signatures of polynomials

Let \( I = \langle f_1, \ldots, f_m \rangle \).

**Idea:** Give each \( f \in I \) a bit more structure:

1. Let \( R^m \) be generated by \( e_1, \ldots, e_m, \prec \) a well-ordering on the monomials of \( R^m \), and let \( \pi : R^m \to R \) such that
   \[ \pi(e_i) = f_i \text{ for all } i. \]

2. Each \( p \in I \) can be represented by an
   \[
   s = \sum_{i=1}^{m} h_i e_i \in R^m \text{ such that } p = \pi(s).
   \]

3. A **signature** of \( p \) is given by
   \[
   \text{sig}(p) = \text{lm}_{\prec}(s) \text{ with } p = \pi(s).
   \]

4. A **minimal signature** of \( p \) exists due to \( \prec \).
Our example – now with signatures and $\preceq_{\text{pot}}$

We have already computed the following data:

$$g_1 = xy - z^2, \quad \text{sig}(g_1) = e_1,$$

$$g_2 = y^2 - z^2, \quad \text{sig}(g_2) = e_2,$$

$$g_3 = \text{spol}(g_2, g_1) = xg_2 - yg_1$$

$$\Rightarrow \text{sig}(g_3) = x \text{sig}(g_2) = xe_2.$$
Our example – now with signatures and $\prec_{\text{pot}}$

We have already computed the following data:

\[
\begin{align*}
g_1 &= xy - z^2, \quad \text{sig}(g_1) = e_1, \\
g_2 &= y^2 - z^2, \quad \text{sig}(g_2) = e_2, \\
g_3 &= \text{spol}(g_2, g_1) = xg_2 - yg_1 \\
\Rightarrow \text{sig}(g_3) &= x \text{sig}(g_2) = xe_2.
\end{align*}
\]

\[
\begin{align*}
\text{spol}(g_3, g_1) &= yg_3 - z^2g_1: \\
\text{sig}(\text{spol}(g_3, g_1)) &= y \text{sig}(g_3) = xye_2.
\end{align*}
\]
Our example – now with signatures and $\preceq_{\text{pot}}$

We have already computed the following data:

\[ g_1 = xy - z^2, \quad \text{sig}(g_1) = e_1, \]
\[ g_2 = y^2 - z^2, \quad \text{sig}(g_2) = e_2, \]
\[ g_3 = \text{spol}(g_2, g_1) = xg_2 - yg_1 \]
\[ \Rightarrow \text{sig}(g_3) = x \text{sig}(g_2) = xe_2. \]

\[ \text{spol}(g_3, g_1) = yg_3 - z^2g_1: \]
\[ \text{sig} \left( \text{spol}(g_3, g_1) \right) = y \text{sig}(g_3) = yxe_2. \]

Note that $\text{sig} \left( \text{spol}(g_3, g_1) \right) = xye_2$ and $\text{lm}(g_1) = xy$. 

Our example – now with signatures and $\prec_{pot}$

We have already computed the following data:

\[
g_1 = xy - z^2, \quad \text{sig}(g_1) = e_1, \\
g_2 = y^2 - z^2, \quad \text{sig}(g_2) = e_2, \\
g_3 = \text{spol}(g_2, g_1) = xg_2 - yg_1 \\
\Rightarrow \text{sig}(g_3) = x \text{sig}(g_2) = xe_2.
\]

\[
\text{spol}(g_3, g_1) = yg_3 - z^2g_1:
\]

\[
\text{sig} (\text{spol}(g_3, g_1)) = y \text{sig}(g_3) = xye_2.
\]

Note that $\text{sig} (\text{spol}(g_3, g_1)) = xye_2$ and $\text{lm}(g_1) = xy$.

\[
\Rightarrow \textbf{We know that } \text{spol}(g_3, g_1) \textbf{ will reduce to zero w.r.t. } G.
\]
Why do we know this?

The general idea is to check the signatures of the generated s-polynomials.

If \( \text{sig}(\text{spol}(f, g)) \) is not minimal for \( \text{spol}(f, g) \) then \( \Rightarrow \) \( \text{spol}(f, g) \) is discarded.

Our goal
Find and discard as many s-polynomials as possible for which the algorithm computes a non-minimal signature.

Our task
We need to take care of the correctness of the signatures throughout the computations.

Note
We order \( P \) by increasing signatures, so we always take the s-polynomial of minimal signature.
Why do we know this?

The general idea is to check the signatures of the generated s-polynomials.

If \( \text{sig}(\text{spol}(f, g)) \) is not minimal for \( \text{spol}(f, g) \) then
\[ \Rightarrow \text{spol}(f, g) \text{ is discarded.} \]
Why do we know this?

The general idea is to check the signatures of the generated s-polynomials.

If $\text{sig}(\text{spol}(f, g))$ is not minimal for $\text{spol}(f, g)$ then
$\Rightarrow$ $\text{spol}(f, g)$ is discarded.

**Our goal**

Find and discard as many s-polynomials as possible for which the algorithm computes a non-minimal signature.
Why do we know this?

The general idea is to check the signatures of the generated s-polynomials.

If $\text{sig}(\text{spol}(f, g))$ is not minimal for $\text{spol}(f, g)$ then $\Rightarrow \text{spol}(f, g)$ is discarded.

Our goal

Find and discard as many s-polynomials as possible for which the algorithm computes a non-minimal signature.

Our task

We need to take care of the correctness of the signatures throughout the computations.
Why do we know this?

The general idea is to check the signatures of the generated s-polynomials.

If \( \text{sig}(\text{spol}(f, g)) \) is not minimal for \( \text{spol}(f, g) \) then
\[ \Rightarrow \text{spol}(f, g) \] is discarded.

**Our goal**
Find and discard as many s-polynomials as possible for which the algorithm computes a non-minimal signature.

**Our task**
We need to take care of the correctness of the signatures throughout the computations.

**Note**
We order \( P \) by increasing signatures, so we always take the s-polynomial of minimal signature.
Signature-based criteria

Non-minimal signature (NM)

$\text{sig}(h)$ not minimal for $h$? $\Rightarrow$ Remove $h$. 

Sketch of proof

1. There exists a syzygy $s \in R^m$ such that $lm(s) = \text{sig}(h)$.
   $\Rightarrow$ We can represent $h$ with a lower signature.

2. Pairs are handled by increasing signatures.
   $\Rightarrow$ All relations of lower signature are already taken care of.

Our example with $\prec \text{pot}$ revisited

$\text{sig}(\text{spol}(g_3, g_1)) = xye^2$
$g_1 = xy - z^2$
$g_2 = y^2 - z^2$

$\Rightarrow \text{psyz}(g_2, g_1) = g_1 e^2 - g_2 e^1 = xye^2 + ...$
Signature-based criteria

Non-minimal signature (NM)

sig(h) not minimal for h? ⇒ Remove h.

Sketch of proof

1. There exists a syzygy $s \in R^m$ such that $\text{lm}(s) = \text{sig}(h)$.
   ⇒ We can represent $h$ with a lower signature.

2. Pairs are handled by increasing signatures.
   ⇒ All relations of lower signature are already taken care of.
Signature-based criteria

Non-minimal signature (NM)

sig(h) not minimal for h? $\Rightarrow$ Remove h.

**Sketch of proof**

1. There exists a syzygy $s \in R^m$ such that $\text{lm}(s) = \text{sig}(h)$.
   $\Rightarrow$ We can represent $h$ with a lower signature.

2. Pairs are handled by increasing signatures.
   $\Rightarrow$ All relations of lower signature are already taken care of.

**Our example with $\prec_{\text{pot}}$ revisited**

\[
\text{sig} \left( \text{spol}(g_3, g_1) \right) = xye_2
\]
\[
g_1 = xy - z^2
\]
\[
g_2 = y^2 - z^2 \quad \Rightarrow \text{psyz}(g_2, g_1) = g_1 e_2 - g_2 e_1 = xye_2 + \ldots
\]
Rewritable signature (RW)

\[ \text{sig}(g) = \text{sig}(h) \implies \text{Remove either } g \text{ or } h. \]
Signature-based criteria

Rewritable signature ( RW )

\[ \text{sig}(g) = \text{sig}(h)? \Rightarrow \text{Remove either } g \text{ or } h. \]

Sketch of proof

1. \[ \text{sig}(g - h) \prec \text{sig}(g), \text{sig}(h). \]
2. Pairs are handled by increasing signatures.
   \Rightarrow \text{All necessary computations of lower signature have already taken place.}

\Rightarrow \text{We can represent } h \text{ by}

\[ h = g + \text{ elements of lower signature.} \]
A good decade on signature-based algorithms
A good decade on signature-based algorithms

- **F5**
  - Faugère (2002)

- **F4/5**
  - Albrecht, Perry (2010)

- **Matrix F5**
  - Bardet (2002)

- **F5 with BC**
  - Ars (2005)

- **Bihomog F5**
  - Faugère, Safey El-Din, Spaenlehauer (2011)

- **Extended F5 Criteria**
  - Ars, Hashemi (2009)

- **SAGBI F5**
  - Faugère, Rahmany (2009)

- **SB**
  - Roune, Stillmann (2012)

- **AP**
  - Arri Perry (2009)

- **AP1**
  - Arri, Perry, E. (2011)

- **AP2**
  - Arri, Perry, E. (2012)

- **Involutive F5**
  - Gerdt, Hashemi, Alizadeh (2013)

- **nF5**
  - E. (2012)

- **F5C**
  - Perry, E. (2009)

- **iF5C**
  - E. (2012)

- **iF5A**
  - E. (2012)

- **iG2V**
  - E. (2012)

- **G2V**
  - Gao, Guan, Volny (2010)

- **GVW**
  - Gao, Volny, Wang (2011)

- **F5 using sym**
  - Faugère, Svartz (2013)

- **Quasihomog F5**
  - Faugère, Safey El-Din, Verron (2013)

- **F5A**
  - Perry, E. (2011)

- **iF5A**
  - E. (2012)

- **nF5**
  - E. (2012)

- **SB**
  - Roune, Stillmann (2012)

- **AP**
  - Arri Perry (2009)

- **AP1**
  - Arri, Perry, E. (2011)

- **AP2**
  - Arri, Perry, E. (2012)

- **Involutive F5**
  - Gerdt, Hashemi, Alizadeh (2013)

- **nF5**
  - E. (2012)

- **F5C**
  - Perry, E. (2009)

- **iF5C**
  - E. (2012)

- **iF5A**
  - E. (2012)

- **nF5**
  - E. (2012)

- **SB**
  - Roune, Stillmann (2012)
- Improvement 1: Signature-based Gröbner Basis algorithms

- Improvement 2: Specialized Gaussian Elimination

- Use GB algorithms in algebraic cryptanalysis
Use **Linear Algebra** for reduction steps in GB computations.
Use **Linear Algebra** for reduction steps in GB computations.

\[
\begin{array}{ccccccc}
1 & 3 & 0 & 0 & 7 & 1 & 0 \\
1 & 0 & 4 & 1 & 0 & 0 & 5 \\
0 & 1 & 6 & 0 & 8 & 0 & 1 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 1 \\
\end{array}
\]
Use **Linear Algebra** for reduction steps in GB computations.

```
1 3 0 0 7 1 0
1 0 4 1 0 0 5
0 1 6 0 8 0 1
0 5 0 0 0 2 0
0 0 0 0 1 3 1
```
Use **Linear Algebra** for reduction steps in GB computations.

### s-polynomial

\[
\begin{align*}
\text{s-polynomial} & = \{ \begin{array}{ccccccc}
1 & 3 & 0 & 0 & 7 & 1 & 0 \\
1 & 0 & 4 & 1 & 0 & 0 & 5 \\
0 & 1 & 6 & 0 & 8 & 0 & 1 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 1 \\
\end{array} \end{align*}
\]

**reducer** \[\leftarrow 0 \ 0 \ 0 \ 0 \ 1 \ 3 \ 1\]

Knowledge of underlying GB structure
Use **Linear Algebra** for reduction steps in GB computations.

\[
\begin{align*}
\text{s-polynomial} & \quad \begin{cases}
1 & 3 & 0 & 0 & 7 & 1 & 0 \\
1 & 0 & 4 & 1 & 0 & 0 & 5
\end{cases} \\
\text{s-polynomial} & \quad \begin{cases}
0 & 1 & 6 & 0 & 8 & 0 & 1 \\
0 & 5 & 0 & 0 & 0 & 2 & 0
\end{cases} \\
\text{reducer} & \quad \leftarrow 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 3 \quad 1
\end{align*}
\]

Knowledge of underlying GB structure
Use **Linear Algebra** for reduction steps in GB computations.

<table>
<thead>
<tr>
<th>s-polynomial</th>
<th>1 3 0 0 7 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 0 4 1 0 0 5</td>
</tr>
<tr>
<td></td>
<td>0 1 6 0 8 0 1</td>
</tr>
<tr>
<td>s-polynomial</td>
<td>0 5 0 0 0 2 0</td>
</tr>
<tr>
<td>reducer</td>
<td>← 0 0 0 0 1 3 1</td>
</tr>
</tbody>
</table>

**Knowledge of underlying GB structure**

**Idea**

Do a static **reordering before** the Gaussian Elimination to achieve a better initial shape. **Reorder afterwards.**
**1st step**: Sort pivot and non-pivot columns

<table>
<thead>
<tr>
<th>Pivot column</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 3 0 0 7 1 0</td>
</tr>
<tr>
<td>1 0 4 1 0 0 5</td>
</tr>
<tr>
<td>0 1 6 0 8 0 1</td>
</tr>
<tr>
<td>0 5 0 0 0 2 0</td>
</tr>
<tr>
<td>0 0 0 0 1 3 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-Pivot column</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 4 1 0 0 5</td>
</tr>
<tr>
<td>0 1 6 0 8 0 1</td>
</tr>
<tr>
<td>0 5 0 0 0 2 0</td>
</tr>
<tr>
<td>0 0 0 0 1 3 1</td>
</tr>
</tbody>
</table>
**1st step**: Sort pivot and non-pivot columns

<table>
<thead>
<tr>
<th>1</th>
<th>3</th>
<th>0</th>
<th>0</th>
<th>7</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
1st step: Sort pivot and non-pivot columns

<table>
<thead>
<tr>
<th>1</th>
<th>3</th>
<th>0</th>
<th>0</th>
<th>7</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
1st step: Sort pivot and non-pivot columns

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>0</th>
<th>0</th>
<th>7</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Pivot column

Non-Pivot column
**1st step**: Sort pivot and non-pivot columns

<table>
<thead>
<tr>
<th>Pivot column</th>
<th>Non-Pivot column</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 3 0 0 7 1 0</td>
<td>1 0 4 1 0 0 5</td>
</tr>
<tr>
<td>0 1 6 0 8 0 1</td>
<td>0 5 0 0 0 2 0</td>
</tr>
<tr>
<td>0 0 0 0 1 3 1</td>
<td></td>
</tr>
</tbody>
</table>
**Faugère-Lachartre Idea**

1st step: Sort pivot and non-pivot columns

\[
\begin{array}{cccccc}
1 & 3 & 0 & 0 & 7 & 1 & 0 \\
1 & 0 & 4 & 1 & 0 & 0 & 5 \\
0 & 1 & 6 & 0 & 8 & 0 & 1 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 1 & 3 & 1 \\
\end{array}
\]

\[
\begin{array}{cccccc}
1 & 3 & 7 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 4 & 1 & 0 & 5 \\
0 & 1 & 8 & 6 & 0 & 0 & 9 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 1 & 0 & 0 & 3 & 1 \\
\end{array}
\]
2nd step: Sort pivot and non-pivot rows

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2nd step: Sort pivot and non-pivot rows

1 3 7 0 0 1 0
1 0 0 4 1 0 5
0 1 8 6 0 0 9
0 5 0 0 0 2 0
0 0 1 0 0 3 1
Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

Pivot row

Non-Pivot row
**2nd step**: Sort pivot and non-pivot rows

\[
\begin{array}{cccccccccc}
1 & 3 & 7 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 4 & 1 & 0 & 5 \\
0 & 1 & 8 & 6 & 0 & 0 & 9 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 1 & 0 & 0 & 3 & 1 \\
\end{array}
\]
**2nd step:** Sort pivot and non-pivot rows

<table>
<thead>
<tr>
<th>Pivot row</th>
<th>Non-Pivot row</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 3 7 0 0 1 0</td>
<td>1 0 0 4 1 0 5</td>
</tr>
<tr>
<td>1 0 0 4 1 0 5</td>
<td>0 5 0 0 0 2 0</td>
</tr>
<tr>
<td>0 1 8 6 0 0 9</td>
<td>0 0 1 0 0 3 1</td>
</tr>
<tr>
<td>0 5 0 0 0 2 0</td>
<td>1 3 7 0 0 1 0</td>
</tr>
<tr>
<td>0 0 1 0 0 3 1</td>
<td>0 1 8 6 0 0 9</td>
</tr>
</tbody>
</table>

Faugère-Lachartre Idea
**3rd step**: Reduce lower left part to zero

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>4</th>
<th>1</th>
<th>0</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>
3rd step: Reduce lower left part to zero
4th step: Reduce lower right part

```
1 0 0 4 1 0 5
0 5 0 0 0 2 0
0 0 1 0 0 3 1
0 0 0 7 10 3 10
0 0 0 6 0 2 1
```
**4th step**: Reduce lower right part

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>
Faugère-Lachartre Idea

4th step: Reduce lower right part

\[
\begin{array}{cccc}
1 & 0 & 0 & 4 \\
0 & 5 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 6 \\
0 & 0 & 0 & 0 \\
\end{array}
\quad \rightarrow \quad
\begin{array}{cccc}
1 & 0 & 0 & 4 \\
0 & 5 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\]

5th step: Remap columns of lower right part

\[
\begin{array}{cccc}
0 & 0 & 0 & 7 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\quad \rightarrow \quad
\begin{array}{cccc}
0 & 0 & 0 & 4 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\]
How our matrices look like
Faugère-Lachartre Idea

**Improvements:**

- Use knowledge of underlying GB structures
- Parallelization of Linear Algebra
- Divide sparse and dense data as much as possible

Recent research:

- Improve parallelization
- Better usage of cache: Use small blocks inside matrix per thread
- Use more of the polynomials structure
- Relax idea of signature-based GB algorithms
Faugère-Lachartre Idea

**Improvements:**
- Use knowledge of underlying GB structures
- Parallelization of Linear Algebra
- Divide sparse and dense data as much as possible

**Recent research:**
- Improve parallelization
- Better usage of cache:
  - Use small blocks inside matrix per thread
- Use more of the polynomials structure
- Relax idea of signature-based GB algorithms
1. Improvement 1: Signature-based Gröbner Basis algorithms

2. Improvement 2: Specialized Gaussian Elimination

3. Use GB algorithms in algebraic cryptanalysis
General idea of asymmetric cryptography

- complete key (set of data)
- public key (subset of complete key)
- private key (complete key)
- message \( M \)
- ciphertext \( C \)
- original message \( M \)
General idea of asymmetric cryptography

- **Complete key**: set of data
- **Public key**: subset of complete key
- **Private key**: complete key \ public key

Message: $M$

Ciphertext: $C$
General idea of asymmetric cryptography

- **Message** \( M \)
- **Ciphertext** \( C \)
- **Public key**: (subset of complete key)
- **Private key**: (complete key \ public key)
- **Complete key**: (set of data)
General idea of asymmetric cryptography

- **Complete key**: Set of data
- **Public key**: Subset of complete key
- **Private key**: Complete key \ public key

Message $M$ -> Ciphertext $C$ -> Original message $M$
Choice of HFE Polynomial

Choose **private polynomial** $p$ such that

- $p \in F_{q^n}(x)$ (mostly $q = 2$),
- $\deg(p) = d$,
- $p$ is “easily” invertible over $F_{q^n}$, i.e. find any solution of $p(x) = y$. 

Note:
- Greater $d = \Rightarrow$ greater security
- Complexity of computing $p^{-1}$ depends quadratically on $d = \Rightarrow d \leq 512$. 

22 / 27
Choice of HFE Polynomial

Choose **private polynomial** $p$ such that

- $p \in F_{q^n}(x)$ (mostly $q = 2$),
- $\deg(p) = d$,
- $p$ is "easily" invertible over $F_{q^n}$, i.e. find any solution of $p(x) = y$.

**Common choice:**

$$p(x) = \sum_{i,j} \alpha_{i,j} x^{q^{u_{i,j}} + q^{v_{i,j}}} + \sum_k \beta_k x^{q^{w_k}} + \gamma.$$
Choice of HFE Polynomial

Choose **private polynomial** $p$ such that

- $p \in F_{q^n}(x)$ (mostly $q = 2$),
- $\deg(p) = d$,
- $p$ is “easily” invertible over $F_{q^n}$, i.e. find any solution of $p(x) = y$.

**Common choice:**

$$p(x) = \sum_{i,j} \alpha_{i,j} x^{q^{u_{i,j}} + q^{v_{i,j}}} + \sum_k \beta_k x^{q^{w_k}} + \gamma.$$ 

**Note**

- Greater $d \implies$ greater security
Choice of HFE Polynomial

Choose **private polynomial** \( p \) such that

- \( p \in F_{q^n}(x) \) (mostly \( q = 2 \)),
- \( \deg(p) = d \),
- \( p \) is “easily” invertible over \( F_{q^n} \), i.e. find any solution of \( p(x) = y \).

**Common choice:**

\[
p(x) = \sum_{i,j} \alpha_{i,j} x^{q^{u_{i,j}} + q^{v_{i,j}}} + \sum_k \beta_k x^{q^{w_k}} + \gamma.
\]

**Note**

- Greater \( d \) \( \Longrightarrow \) greater security
- Complexity of computing \( p^{-1} \) depends quadratically on \( d \).
Choice of HFE Polynomial

Choose **private polynomial** $p$ such that

- $p \in F_{q^n}(x)$ (mostly $q = 2$),
- $\deg(p) = d$,
- $p$ is “easily” invertible over $F_{q^n}$, i.e. find any solution of $p(x) = y$.

**Common choice:**

$$p(x) = \sum_{i,j} \alpha_{i,j} x^{q^{u_{i,j}} + q^{v_{i,j}}} + \sum_k \beta_k x^{q^{w_k}} + \gamma.$$  

**Note**

- Greater $d \implies$ greater security
- Complexity of computing $p^{-1}$ depends quadratically on $d$.  

$$\implies d \leq 512.$$
Generate public key

Represent $p$ publicly such that original structure and inversion are hidden:
Generate public key

Represent $p$ publicly such that original structure and inversion are hidden:

- Represent $F_{q^n}$ as $F_q$ vector space.
- Choose 2 linear transformations $S$ and $T$. 
Generate public key

Represent $p$ publicly such that original structure and inversion are hidden:

- Represent $F_{q^n}$ as $F_q$ vector space.
- Choose 2 linear transformations $S$ and $T$.

$\Rightarrow$ **public key** $T \circ p \circ S$. 
Generate public key

Represent $p$ publicly such that original structure and inversion are hidden:

- Represent $F_{q^n}$ as $F_q$ vector space.
- Choose 2 linear transformations $S$ and $T$.

$$\quad \Rightarrow \text{public key } T \circ p \circ S.$$

Assume $q = 2$

Frobenius map on $F_{2^n}$ is a linear transformation over $F_2$ on $F_{2^n}$:

$$\alpha_{i,j}x^{2^{u_{i,j}}+2^{v_{i,j}}} \quad \rightarrow \quad \text{quadratic term}$$
$$\sum_k \beta_k x^{2^{w_k}} \quad \rightarrow \quad \text{linear term}$$
$$\gamma \quad \rightarrow \quad \text{constant term}$$
Generate public key

Represent $p$ publicly such that original structure and inversion are hidden:

- Represent $F_{q^n}$ as $F_q$ vector space.
- Choose 2 linear transformations $S$ and $T$.

$\Rightarrow$ **public key** $T \circ p \circ S$.

Assume $q = 2$

Frobenius map on $F_{2^n}$ is a linear transformation over $F_2$ on $F_{2^n}$:

$$\alpha_{i,j}x^{2^{u_{i,j}} + 2^{v_{i,j}}} \quad \Rightarrow \quad \text{quadratic term}$$

$$\sum_k \beta_k x^{2^{w_k}} \quad \Rightarrow \quad \text{linear term}$$

$$\gamma \quad \Rightarrow \quad \text{constant term}$$

**system of $n$ quadratic equations in $n$ variables over $F_2$**
**Public key**: $n$ multivariate polynomials $(p_1, \ldots, p_n)$ over $F_q$. 
**Public key:** $n$ multivariate polynomials $(p_1, \ldots, p_n)$ over $F_q$.

$\implies$ Transform message $M \in F_{q^n}$ to $F_q^n$, i.e. $M = (x_1, \ldots, x_n)$. 
HFE Encryption

**Public key**: $n$ multivariate polynomials $(p_1, \ldots, p_n)$ over $F_q$.

$\implies$ Transform message $M \in F_q^n$ to $F_q^n$, i.e. $M = (x_1, \ldots, x_n)$.

**Encryption**: Evaluate each $p_i$ at $M$.

$\implies$ Ciphertext $C = (p_1(x_1, \ldots, x_n), \ldots, p_n(x_1, \ldots, x_n)) \in F_q^n$. 

HFE Encryption

**Public key**: \( n \) multivariate polynomials \( (p_1, \ldots, p_n) \) over \( F_q \).

\[ \Rightarrow \text{Transform message } M \in F_{q^{n}} \text{ to } F_q^n, \text{i.e. } M = (x_1, \ldots, x_n). \]

**Encryption**: Evaluate each \( p_i \) at \( M \).

\[ \Rightarrow \text{Ciphertext } C = (p_1(x_1, \ldots, x_n), \ldots, p_n(x_1, \ldots, x_n)) \in F_q^n. \]

**Or in terms of** \( p, S \) and \( T \) (those are not available to the public):
**Public key:** $n$ multivariate polynomials $(p_1, \ldots, p_n)$ over $F_q$.

\[ \Rightarrow \text{Transform message } M \in F_{q^n} \text{ to } F_q^n, \text{ i.e. } M = (x_1, \ldots, x_n). \]

**Encryption:** Evaluate each $p_i$ at $M$.

\[ \Rightarrow \text{Ciphertext } C = (p_1(x_1, \ldots, x_n), \ldots, p_n(x_1, \ldots, x_n)) \in F_q^n. \]

**Or in terms of $p$, $S$ and $T$** (those are not available to the public):

- Apply $S$ to $M$: $S(x_1, \ldots, x_n) \Rightarrow x' \in F_{q^n}.$
HFE Encryption

**Public key**: $n$ multivariate polynomials $(p_1, \ldots, p_n)$ over $F_q$.

$\implies$ Transform message $M \in F_{q^n}$ to $F_q^n$, i.e. $M = (x_1, \ldots, x_n)$.

**Encryption**: Evaluate each $p_i$ at $M$.

$\implies$ Ciphertext $C = (p_1(x_1, \ldots, x_n), \ldots, p_n(x_1, \ldots, x_n)) \in F_q^n$.

**Or in terms of $p$, $S$ and $T$** (those are not available to the public):

- Apply $S$ to $M$: $S(x_1, \ldots, x_n) \implies x' (\in F_{q^n})$.
- Evaluate $p(x') = y' \implies (y'_1, \ldots, y'_n) \in F_q^n$. 
HFE Encryption

**Public key**: $n$ multivariate polynomials $(p_1, \ldots, p_n)$ over $F_q$.

$\implies$ Transform message $M \in F_q^n$ to $F_q^n$, i.e. $M = (x_1, \ldots, x_n)$.

**Encryption**: Evaluate each $p_i$ at $M$.

$\implies$ Ciphertext $C = (p_1(x_1, \ldots, x_n), \ldots, p_n(x_1, \ldots, x_n)) \in F_q^n$.

Or in terms of $p$, $S$ and $T$ (those are not available to the public):

- Apply $S$ to $M$: $S(x_1, \ldots, x_n) \implies x' \ (\in F^n_q)$.
- Evaluate $p(x') = y' \implies (y'_1, \ldots, y'_n) \in F_q^n$.
- Apply $T \implies C = Ty' \in F_q^n$. 
Simply put: Take $C$ and apply $T^{-1}$, $p^{-1}$ and $S^{-1}$. 
**Simply put**: Take $C$ and apply $T^{-1}$, $p^{-1}$ and $S^{-1}$.

- Computing $S^{-1}$ and $T^{-1}$ is easy.
**Simply put:** Take $C$ and apply $T^{-1}$, $p^{-1}$ and $S^{-1}$.

- Computing $S^{-1}$ and $T^{-1}$ is easy.
- Finding solutions for $p(x') = y'$ is crucial:
  - $\deg(p) = d \implies$ at most $d$ different solutions for one $y'$ ($p$ not nec. one-to-one).
  - Redundancy $r$ is added to message $M$ to get a unique solution.
**Simply put**: Take $C$ and apply $T^{-1}$, $p^{-1}$ and $S^{-1}$.

- Computing $S^{-1}$ and $T^{-1}$ is easy.
- Finding solutions for $p(x') = y'$ is crucial:
  - $\deg(p) = d \implies$ at most $d$ different solutions for one $y'$. ($p$ not nec. one-to-one).
  - Redundancy $r$ is added to message $M$ to get a unique solution.

**How to break the system?**
Simply put: Take $C$ and apply $T^{-1}$, $p^{-1}$ and $S^{-1}$.

- Computing $S^{-1}$ and $T^{-1}$ is easy.
- Finding solutions for $p(x') = y'$ is crucial:
  - $\deg(p) = d \implies$ at most $d$ different solutions for one $y'$ ($p$ not nec. one-to-one).
  - Redundancy $r$ is added to message $M$ to get a unique solution.

How to break the system?

Solve a system of multivariate quadratic polynomials over $\mathbb{F}_q$:

\[
\begin{align*}
p_1(x_1, \ldots, x_n) &= y_1 \\
\vdots & \quad \vdots \\
p_n(x_1, \ldots, x_n) &= y_n
\end{align*}
\]
Patarin defined the so-called **HFE Challenge 1** by

- $d = 96$,
- $q = 2$,
- $n = 80$. 

Patarin defined the so-called **HFE Challenge 1** by

- $d = 96$,
- $q = 2$,
- $n = 80$.

Faugère broke this system computing a Gröbner basis of the corresponding system of quadratic multivariate polynomials over $F_2$ in 2002 using a specialized $F5$ Algorithm:
Patarin defined the so-called **HFE Challenge 1** by

- $d = 96$,
- $q = 2$,
- $n = 80$.

Faugère broke this system computing a Gröbner basis of the corresponding system of quadratic multivariate polynomials over $F_2$ in 2002 using a specialized $F5$ Algorithm:

96 hours of CPU time on an HP workstation with an alpha EV68 processor at 1 GHz and 4 GB RAM
(Whole computation approx. 7.65 GB.)
[FL10] J.-C. Faugère and S. Lachartre. Parallel Gaussian Elimination for Gröbner bases computations in finite fields