On the complexity of ECDLP for composite fields

Based on joint works with JC Faugère, JJ Quisquater, L Perret, G Renault

Christophe Petit
Discrete logarithm problem (DLP)

- **Discrete logarithm problem**
  Given $G$ a finite (multiplicative) cyclic group
  Given $g$ a generator of $G$ and given $h \in G$
  Find $k \in \mathbb{Z}$ such that $g^k = h$

- Diffie-Hellman key exchange, ElGamal encryption, Digital Signature algorithm,...
Discrete logarithm problem (DLP)

- Discrete logarithm problem
  Given $G$ a finite (multiplicative) cyclic group
  Given $g$ a generator of $G$ and given $h \in G$
  Find $k \in \mathbb{Z}$ such that $g^k = h$

- Diffie-Hellman key exchange, ElGamal encryption, Digital Signature algorithm,...

- Cryptographic assumption: DLP is “hard” for
  - Multiplicative groups of finite fields
  - Elliptic curves
  - Jacobians of hyperelliptic curves
How hard is DLP?

- Answer **depends on the group**
  - Subexponential algorithms exist for finite fields and hyperelliptic curves
  - Particular elliptic curve families are weaker
  - 160-bit ECDLP $\approx$ 2048-bit DLP or factoring
How hard is DLP?

- Answer **depends on the group**
  - Subexponential algorithms exist for finite fields and hyperelliptic curves
  - Particular elliptic curve families are weaker
  - 160-bit ECDLP $\approx$ 2048-bit DLP or factoring

- This talk: **elliptic curves over binary fields** $\mathbb{F}_{2^n}$
  - Includes 10/15 curves standardized by NIST
  - Complexity thought to be exponential in $n$
How hard is DLP?

- Answer **depends on the group**
  - Subexponential algorithms exist for finite fields and hyperelliptic curves
  - Particular elliptic curve families are weaker
  - 160-bit ECDLP $\approx$ 2048-bit DLP or factoring

- This talk: **elliptic curves over binary fields** $\mathbb{F}_{2^n}$
  - Includes 10/15 curves standardized by NIST
  - Complexity thought to be exponential in $n$
  - We argue it is
    \[
    \leq 2^{2n^{2/3} \log n}
    \]
Outline

From ECDLP to polynomial systems

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Back to ECDLP
Outline

From ECDLP to polynomial systems

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Back to ECDLP
ECDLP on binary curves

- **Elliptic curve discrete logarithm problem**
  Given $E$ over a finite field $K$,
  Given $P \in E(K)$, given $Q \in G := \langle P \rangle$,
  Find $k \in \mathbb{Z}$ such that $Q = kP$.

- **Binary curves** $K = \mathbb{F}_{2^n}$

  
  \[ y^2 + xy = x^3 + a_2x^2 + a_6 \quad \text{with} \quad a_6 \neq 0 \]

  Koblitz curve if $a_6 = 1$ and $a_2 \in \{0, 1\}$
ECDLP on binary curves

- Elliptic curve discrete logarithm problem
  Given $E$ over a finite field $K$,
  Given $P \in E(K)$, given $Q \in G := \langle P \rangle$,
  Find $k \in \mathbb{Z}$ such that $Q = kP$.

- Binary curves $K = \mathbb{F}_{2^n}$

  $$y^2 + xy = x^3 + a_2x^2 + a_6 \quad \text{with} \quad a_6 \neq 0$$

  Koblitz curve if $a_6 = 1$ and $a_2 \in \{0, 1\}$

- How hard is ECDLP on binary curves?
Generic DLP attacks

- Some attacks apply to DLP for any group $G$
  - Exhaustive search
  - Baby-step, giant step
  - Pollard’s rho
  - Pohlig-Hellman if $|G|$ is smooth
Generic DLP attacks

- Some attacks apply to DLP for any group $G$
  - Exhaustive search
  - Baby-step, giant step
  - Pollard’s rho
  - Pohlig-Hellman if $|G|$ is smooth

- In general, no better algorithm for elliptic curves
  160-bit ECDLP $\approx$ 2048-bit DLP or factoring
Reductions to simpler DLP

- Idea: transfer ECDLP to a “simpler” DLP problem through a group homomorphism
**Reductions to simpler DLP**

- Idea: transfer ECDLP to a “simpler” DLP problem through a group homomorphism

- MOV reduction if $|G| \text{ divides } q^m - 1$ [MOV93]
  
  Transfer ECDLP to DLP on $K^m$
Reductions to simpler DLP

- Idea: transfer ECDLP to a “simpler” DLP problem through a group homorphism

- **MOV reduction** if $|G|$ divides $q^m - 1$ [MOV93]
  - Transfer ECDLP to DLP on $K^m$

- Polynomial time for **anomalous curves** [SA98, S98, S99]
  - Transfer ECDLP to a $p$-adic elliptic logarithm if $|G| = |K|$
Reductions to simpler DLP

- Idea: transfer ECDLP to a “simpler” DLP problem through a group homorphism

- MOV reduction if $|G|$ divides $q^m - 1$ [MOV93]
  Transfer ECDLP to DLP on $K^m$

- Polynomial time for anomalous curves [SA98,S98,S99]
  Transfer ECDLP to a $p$-adic elliptic logarithm if $|G| = |K|$

- Weil descent for some curves over $\mathbb{F}_{p^n}$ [GS99,GHS00]
  Transfer ECDLP to the Jacobian of an hyperelliptic curve
Reductions to simpler DLP

- Idea: transfer ECDLP to a “simpler” DLP problem through a group homorphism
- MOV reduction if $|G|$ divides $q^m - 1$ \([MOV93]\)
  Transfer ECDLP to DLP on $K^m$
- Polynomial time for \textbf{anomalous curves} \([SA98,S98,S99]\)
  Transfer ECDLP to a $p$-adic elliptic logarithm if $|G| = |K|$
- \textbf{Weil descent} for some curves over $\mathbb{F}_{p^n}$ \([GS99,GHS00]\)
  Transfer ECDLP to the Jacobian of an hyperelliptic curve
- Only work for specific families
This talk: Index calculus

- General method to solve discrete logarithm problems
  1. Define a factor basis $\mathcal{F} \subset G$
  2. Relation search: find about $|\mathcal{F}|$ relations

$$a_i P + b_i Q = \sum_{P_j \in \mathcal{F}} e_{ij} P_j$$

3. Do linear algebra modulo $|G|$ on the relations to get

$$aP + bQ = 0$$
This talk: Index calculus

- General method to solve discrete logarithm problems
  1. Define a factor basis \( \mathcal{F} \subset G \)
  2. Relation search: find about \( |\mathcal{F}| \) relations

\[
a_i P + b_i Q = \sum_{P_j \in \mathcal{F}} e_{ij} P_j
\]

3. Do linear algebra modulo \( |G| \) on the relations to get

\[
aP + bQ = 0
\]

- Define \( \mathcal{F} \) s.t. there is an “efficient” algorithm for Step 2
- Balance relation search and linear algebra
Example: a naive index calculus for $\mathbb{F}_2^n$

- DLP: given $g, h \in \mathbb{F}_2^n$, find $k$ such that $h = g^k$
- Factor basis made of small “primes”

$$\mathcal{F}_B := \{\text{irreducible } f(X) \in \mathbb{F}_2[X] \mid \deg(f) \leq B\}$$
Example: a naive index calculus for $\mathbb{F}_{2^n}^*$

- DLP: given $g, h \in \mathbb{F}_{2^n}^*$, find $k$ such that $h = g^k$
- Factor basis made of small “primes”

$$\mathcal{F}_B := \{\text{irreducible } f(X) \in \mathbb{F}_2[X] \mid \deg(f) \leq B\}$$

- Relation search
  - Choose random $a, b \in \{1, \ldots, 2^n - 1\}$
  - Compute $r := g^a h^b$
  - Factor $r$ with Berlekamp’s algorithm
Example: a naive index calculus for $\mathbb{F}_{2^n}^*$

- DLP: given $g, h \in \mathbb{F}_{2^n}^*$, find $k$ such that $h = g^k$

- Factor basis made of small “primes”

$$\mathcal{F}_B := \{\text{irreducible } f(X) \in \mathbb{F}_2[X] | \deg(f) \leq B\}$$

- Relation search
  - Choose random $a, b \in \{1, \ldots, 2^n - 1\}$
  - Compute $r := g^ah^b$
  - Factor $r$ with Berlekamp’s algorithm
  - If all factors $\in \mathcal{F}_B$, we have a relation $g^ah^b = \prod_{f_i \in \mathcal{F}} f_i^{e_i}$
Example: a naive index calculus for $\mathbb{F}_2^{*n}$

- DLP: given $g, h \in \mathbb{F}_2^{*n}$, find $k$ such that $h = g^k$
- Factor basis made of small “primes”

$$\mathcal{F}_B := \{\text{irreducible } f(X) \in \mathbb{F}_2[X] \mid \deg(f) \leq B\}$$

- Relation search
  - Choose random $a, b \in \{1, \ldots, 2^n - 1\}$
  - Compute $r := g^a h^b$
  - Factor $r$ with Berlekamp’s algorithm
  - If all factors $\in \mathcal{F}_B$, we have a relation $g^a h^b = \prod_{f_i \in \mathcal{F}} f_i^{e_i}$

- For $B \approx n^{1/2}$, we get subexponential complexity
Index calculus : success stories

- **Finite fields**: Adleman [A79,A94], Coppersmith [C84], Adleman and Huang [AH99]
  Subexponential complexity

  \[ \exp\left(\log^{1/3} |K| \log^{2/3} \log |K|\right) \]
Index calculus : success stories

- **Finite fields**: Adleman [A79,A94], Coppersmith [C84], Adleman and Huang [AH99]
  Subexponential complexity
  \[ \exp(\log^{1/3}|K| \log^{2/3} \log |K|) \]

- **Hyperelliptic curves**: Adleman-DeMarrais-Huang [ADH94], Gaudry [G00], Gaudry-Thomé-Thériault-Diem [GTTD07]
  Subexponential for large genus; beat BSGS if \( g \geq 3 \)
Index calculus : success stories

- **Finite fields** : Adleman [A79,A94], Coppersmith [C84], Adleman and Huang [AH99]
  Subexponential complexity

\[
\exp\left(\log^{1/3} |K| \log^{2/3} \log |K|\right)
\]

- **Hyperelliptic curves** :
  Adleman-DeMarrais-Huang [ADH94], Gaudry [G00], Gaudry-Thomé-Thériault-Diem [GTTD07]
  Subexponential for large genus ; beat BSGS if \( g \geq 3 \)

- **Elliptic curves** : no algorithm at all until 2005
Index calculus for elliptic curves

- For finite fields, **small “primes”** are a natural factor basis
  - Every element factors uniquely as a product of primes
  - “Good” probability that random elements are smooth
Index calculus for elliptic curves

- For finite fields, **small “primes”** are a natural factor basis
  - Every element factors uniquely as a product of primes
  - “Good” probability that random elements are smooth

- Similarly for elliptic curves, we will need
  1. A definition of “small” elements
  2. An algorithm to decompose general elements into (potentially) small elements
Index calculus for elliptic curves

- For finite fields, **small “primes”** are a natural factor basis
  - Every element factors uniquely as a product of primes
  - “Good” probability that random elements are smooth

- Similarly for elliptic curves, we will need
  1. A definition of “small” elements
  2. An algorithm to decompose general elements into (potentially) small elements

- First partial solutions given by Semaev [S04]
Summation polynomials \([S04]\)

- Relate the \(x\)-coordinates of points that sum to \(O\)
- \(S_r(x_1, \ldots, x_r) = 0\)
  \(\iff\) \(\exists (x_i, y_i) \in E\) s.t. \((x_1, y_1) + \cdots + (x_r, y_r) = O\)
Summation polynomials \([S04]\)

- Relate the \(x\)-coordinates of points that sum to \(O\)

- \(S_r(x_1, \ldots, x_r) = 0\)

  \[\iff \exists (x_i, y_i) \in E \text{ s.t. } (x_1, y_1) + \cdots + (x_r, y_r) = O\]

- Recursive formulae:
  \(S_2(x_1, x_2) = x_1 - x_2\)
  \(S_3(x_1, x_2, x_3) = \ldots \quad \text{(depends on } E\text{)}\)
  \(S_r(x_1, \ldots, x_r) = \)
  \(\text{Res}_X(S_{r-k}(x_1, \ldots, x_{m-k-1}, X), S_{k+2}(x_{r-k}, \ldots, x_r, X))\)
Summation polynomials \([S04]\)

- Relate the \(x\)-coordinates of points that sum to \(O\)
  \[ S_r(x_1, \ldots, x_r) = 0 \]
  \[ \Leftrightarrow \exists (x_i, y_i) \in E \text{ s.t. } (x_1, y_1) + \cdots + (x_r, y_r) = O \]

- Recursive formulae:
  \[ S_2(x_1, x_2) = x_1 - x_2 \]
  \[ S_3(x_1, x_2, x_3) = \ldots \quad \text{(depends on } E) \]
  \[ S_r(x_1, \ldots, x_r) = \]
  \[ \text{Res}_X (S_{r-k}(x_1, \ldots, x_{m-k-1}, X), S_{k+2}(x_{r-k}, \ldots, x_r, X)) \]

- \(S_r\) has degree \(2^{r-2}\) in each variable
  Symmetric set of solutions
Semaev’s variant of index calculus

- Semaev’s variant of index calculus:
  - **Factor basis**: define $\mathcal{F}_V := \{(x, y) \in E | x \in V\}$ where $V \subset K$
  - **Relation search**: for each relation,
    Compute $(X_i, Y_i) := a_iP + b_iQ$ for random $a_i, b_i$
    Find $x_j \in V$ with $S_{m+1}(x_1, \ldots, x_m, X_i) = 0$
    Find the corresponding $y_j$
Semaev’s variant of index calculus

- Semaev’s variant of index calculus:
  - **Factor basis**:
    define \( \mathcal{F}_V := \{(x, y) \in E | x \in V\} \) where \( V \subset K \)
  - **Relation search**:
    for each relation,
    Compute \((X_i, Y_i) := a_iP + b_iQ\) for random \(a_i, b_i\)
    Find \(x_j \in V\) with \(S_{m+1}(x_1, \ldots, x_m, X_i) = 0\)
    Find the corresponding \(y_j\)

- **Semaev’s observation**:
  ECDLP reduced to solving summation’s polynomial with constraints \(x_i \in V\)
Semaev’s variant of index calculus

- Semaev’s variant of index calculus:
  - **Factor basis**: define \( \mathcal{F}_V := \{(x, y) \in E | x \in V\} \) where \( V \subset K \)
  - **Relation search**: for each relation,
    Compute \( (X_i, Y_i) := a_iP + b_iQ \) for random \( a_i, b_i \)
  Find \( x_j \in V \) with \( S_{m+1}(x_1, \ldots, x_m, X_i) = 0 \)
  Find the corresponding \( y_j \)

- **Semaev’s observation**: ECDLP reduced to solving summation’s polynomial with constraints \( x_i \in V \)

- Remains to define \( V \) such that relation search is feasible
Focus on composite fields

- For $K := \mathbb{F}_p$, Semaev proposed $V := \{x < B\}$
  But could not solve summation polynomials
Focus on composite fields

- For $K := \mathbb{F}_p$, Semaev proposed $V := \{x < B\}$
  But could not solve summation polynomials

- For $K := \mathbb{F}_{q^n}$, Gaudry and Diem proposed $V := \mathbb{F}_q$
  - Gaudry [G09]: algorithm faster than generic ones for any $q, n \geq 3$ (but still exponential)
  - Diem [D11]: subexponential algorithm when $q$ and $n$ increase in an appropriate way
Focus on composite fields

- For $K := \mathbb{F}_p$, Semaev proposed $V := \{x < B\}$
  But could not solve summation polynomials

- For $K := \mathbb{F}_{q^n}$, Gaudry and Diem proposed $V := \mathbb{F}_q$
  - Gaudry [G09]: algorithm faster than generic ones for any $q, n \geq 3$ (but still exponential)
  - Diem [D11]: subexponential algorithm when $q$ and $n$ increase in an appropriate way

- Idea in both cases: **Weil descent** on Semaev polynomial
  Reduction to a polynomial system of equations
Finding relations : Weil descent

- Finding relations amounts to
  
  Finding \( x_j \in \mathbb{F}_q \) with \( S_{n+1}(x_1, \ldots, x_n, X_i) = 0 \)
Finding relations : Weil descent

- Finding relations amounts to
  Finding \( x_j \in \mathbb{F}_q \) with \( S_{n+1}(x_1, \ldots, x_n, X_i) = 0 \)

- See \( \mathbb{F}_{q^n} \) as a vector space over \( \mathbb{F}_q \)
- See polynomial equation \( S_{n+1} = 0 \) over \( \mathbb{F}_{q^n} \) as a **system** of polynomial equations over \( \mathbb{F}_q \)
- Solve the system
Finding relations: Weil descent

Finding relations amounts to

Finding \( x_j \in \mathbb{F}_q \) with \( S_{n+1}(x_1, \ldots, x_n, X_i) = 0 \)

See \( \mathbb{F}_{q^n} \) as a vector space over \( \mathbb{F}_q \)

See polynomial equation \( S_{n+1} = 0 \) over \( \mathbb{F}_{q^n} \) as a \textbf{system} of polynomial equations over \( \mathbb{F}_q \)

Solve the system

System harder to solve for larger \( n \)

\textbf{Attack does not work for} \( \mathbb{F}_{2^n} \) \textbf{when} \( n \) prime
Diem’s variant of index calculus [D11b]

Let $K := \mathbb{F}_{2^n}$. Fix $n' < n$ and $m \approx n/n'$

- **Factor basis**:
  Choose a vector subspace $V$ of $\mathbb{F}_{2^n}$ with dimension $n'$
  Define $\mathcal{F}_V := \{(x, y) \in E | x \in V\}$
Diem’s variant of index calculus \([D11b]\)

Let \(K := \mathbb{F}_{2^n}\). Fix \(n' < n\) and \(m \approx n/n'\)

- **Factor basis**: Choose a **vector subspace** \(V\) of \(\mathbb{F}_{2^n}\) with dimension \(n'\)
  
  Define \(\mathcal{F}_V := \{(x, y) \in E | x \in V\}\)

- **Relation search**: find about \(2^{n'}\) relations. For each one,
  
  Compute \((X_i, Y_i) := a_i P + b_i Q\) for random \(a_i, b_i\)
  
  Find \(x_j \in V\) with \(S_{m+1}(x_1, \ldots, x_m, X_i) = 0\)
  
  Find the corresponding \(y_j\)
Diem’s variant of index calculus \([D11b]\)

Let \( K := \mathbb{F}_{2^n} \). Fix \( n' < n \) and \( m \approx n/n' \)

- **Factor basis**:
  Choose a vector subspace \( V \) of \( \mathbb{F}_{2^n} \) with dimension \( n' \)
  Define \( \mathcal{F}_V := \{ (x, y) \in E | x \in V \} \)

- **Relation search**: find about \( 2^{n'} \) relations. For each one,
  Compute \( (X_i, Y_i) := a_iP + b_iQ \) for random \( a_i, b_i \)
  Find \( x_j \in V \) with \( S_{m+1}(x_1, \ldots, x_m, X_i) = 0 \)
  Find the corresponding \( y_j \)

- **Linear algebra** between the relations
Finding relations: Weil descent

- Finding relations amounts to
  Finding \( x_i \in V \) with \( S_{m+1}(x_1, \ldots, x_m, X) = 0 \)
Finding relations: Weil descent

Finding relations amounts to:

Finding \( x_i \in V \) with \( S_{m+1}(x_1, \ldots, x_m, X) = 0 \)

Let \( \{v_1, \ldots, v_{n'}\} \) be a basis of \( V \).
Define \( x_{ij} \in \mathbb{F}_2 \) such that
\[
    x_i = \sum_{j=1}^{n'} x_{ij} v_j
\]

\[
    S_{m+1} \left( \sum_{j=1}^{n'} x_{1j} v_j, \ldots, \sum_{j=1}^{n'} x_{n'j} v_j, X \right) = 0
\]
Finding relations : Weil descent

- Finding relations amounts to finding \( x_i \in V \) with \( S_{m+1}(x_1, \ldots, x_m, X) = 0 \)

- Let \( \{v_1, \ldots, v_{n'}\} \) be a basis of \( V \)
  
  Define \( x_{ij} \in \mathbb{F}_2 \) such that \( x_i = \sum_{j=1}^{n'} x_{ij} v_j \)

  \[
  S_{m+1} \left( \sum_{j=1}^{n'} x_{1j} v_j, \ldots, \sum_{j=1}^{n'} x_{n'j} v_j, X \right) = 0
  \]

- See \( \mathbb{F}_{2^n} \) as a vector space over \( \mathbb{F}_2 \)

- The polynomial equation over \( \mathbb{F}_{2^n} \) corresponds to a system of polynomial equations over \( \mathbb{F}_2 \)
Complexity of Diem’s algorithm

- Computing $S_{m+1}$ with resultants: cost $2^{t_1}$ where

  $$t_1 \approx m(m + 1)$$
Complexity of Diem’s algorithm

- Computing $S_{m+1}$ with resultants: cost $2^{t_1}$ where
  \[ t_1 \approx m(m + 1) \]
- Finding $2^{n'}$ relations: total cost $2^{t_2}$ where
  \[ t_2 \approx n' + \log TR \]
  where $TR(m, n', n)$ is **time to compute one relation**
Complexity of Diem’s algorithm

- Computing $S_{m+1}$ with resultants: cost $2^{t_1}$ where
  
  $$t_1 \approx m(m + 1)$$

- Finding $2^{n'}$ relations: total cost $2^{t_2}$ where
  
  $$t_2 \approx n' + \log T_R$$

  where $T_R(m, n', n)$ is time to compute one relation

- (Sparse) linear algebra on relations: cost $2^{\omega't_3}$ where
  
  $$t_3 \approx \log m + \log n + \omega'n'$$
Our result

- When $p$ is small, systems arising from a Weil descent are much easier to solve than random systems.
Our result

- When $p$ is small, systems arising from a Weil descent are much easier to solve than random systems.
- Under a common heuristic assumption validated by experiments for small parameters, we can choose $m$ and $n'$ such that Diem’s algorithm for ECDLP over $\mathbb{F}_{2^n}$ has subexponential complexity:
  \[ \leq 2^{2n^{2/3} \log n} \]
Outline

From ECDLP to polynomial systems

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Back to ECDLP
Algebraic cryptanalysis

- Reduce some cryptanalytic problems to the resolution of some systems of multivariate polynomial equations
Algebraic cryptanalysis

- Reduce some cryptanalytic problems to the resolution of some systems of multivariate polynomial equations
- Generic polynomial systems are hard to solve, but “cryptanalysis” systems are far from generic
Algebraic cryptanalysis

- Reduce some cryptanalytic problems to the resolution of some systems of multivariate polynomial equations
- Generic polynomial systems are hard to solve, but “cryptanalysis” systems are far from generic
- Systems usually solved with Gröbner basis algorithms
Algebraic cryptanalysis

- Reduce some cryptanalytic problems to the resolution of some systems of **multivariate polynomial equations**
- Generic polynomial systems are hard to solve, but “cryptanalysis” systems are far from generic
- Systems usually solved with **Gröbner basis algorithms**
- Success stories:
  - HFE and variants
  - Isomorphism of polynomials
  - MacEliece variants
  - Algebraic side-channel attacks
Polynomial systems

- Let $K$ be a field and $R := K[x_1, \ldots, x_n]$. Let $f_1, \ldots, f_m \in R$. Solve

$$\begin{align*}
  f_1(x_1, \ldots, x_n) &= 0 \\
  \vdots \\
  f_m(x_1, \ldots, x_n) &= 0
\end{align*}$$
Polynomial systems

Let $K$ be a field and $R := K[x_1, \ldots, x_n]$. Let $f_1, \ldots, f_m \in R$. Solve

$$\begin{cases}
f_1(x_1, \ldots, x_n) = 0 \\
\ldots \\
f_m(x_1, \ldots, x_n) = 0
\end{cases}$$

Linear systems can be solved by triangulation with Gaussian elimination. What about polynomial systems?
Linearization

- Construct all products

\[ g_{i,j} = t_j f_i \]

where \( t_j \) is a monomial and \( \deg(g_{i,j}) \leq d \)
Linearization

- Construct all products

\[ g_{i,j} = t_j f_i \]

where \( t_j \) is a monomial and \( \deg(g_{i,j}) \leq d \)

- Decompose each product in monomial terms

\[ g_{i,j} = \sum_k c_{i,j}^k m_k \]
Linearization

- Construct all products

\[ g_{i,j} = t_j f_i \]

where \( t_j \) is a monomial and \( \deg(g_{i,j}) \leq d \)

- Decompose each product in monomial terms

\[ g_{i,j} = \sum_k c_{i,j}^k m_k \]

- Write all coefficients in a Macaulay matrix \( M_d \), each row corresponding to one polynomial \( g_{i,j} \) and each column corresponding to one monomial term \( m_k \)
Linearization

- If $d$ large enough, some linear combinations of the rows lead to new polynomials with lower degrees
Linearization

- If $d$ large enough, some linear combinations of the rows lead to new polynomials with lower degrees
- If $d$ large enough, linear algebra on $M_d$ provides a new “triangular” system of equations

\[
\begin{align*}
g_1(x_1, \ldots, x_{n-1}, x_n) &= 0 \\
\quad \vdots \\
g_{m'-1}(x_{n-1}, x_n) &= 0 \\
g_{m'}(x_n) &= 0
\end{align*}
\]
Linearization

- If $d$ large enough, some linear combinations of the rows lead to new polynomials with lower degrees
- If $d$ large enough, linear algebra on $M_d$ provides a new “triangular” system of equations

\[
\begin{align*}
&g_1(x_1, \ldots, x_{n-1}, x_n) = 0 \\
&\quad \vdots \\
&g_{m'-1}(x_{n-1}, x_n) = 0 \\
&g_{m'}(x_n) = 0
\end{align*}
\]

- The new system is in fact a Gröbner basis for the lexicographic ordering
Gröbner bases

- Given an ideal \( I(f_1, \ldots, f_m) \) and a monomial ordering \( > \), a Gröbner basis (GB) for this ordering is a basis \( \{ f'_1, \ldots, f'_{\ell'} \} \) such that for any \( f \in I(f_1, \ldots f_\ell) \), there exists \( i \in \{1, \ldots, \ell'\} \) such that \( \text{LT}(f'_i) \mid \text{LT}(f) \) (LT = leading term for the ordering)

- Any \( f \in I \) can be (uniquely) reduced by the GB
Gröbner bases

- Given an ideal $I(f_1, \ldots, f_m)$ and a monomial ordering $>$, a Gröbner basis (GB) for this ordering is a basis $\{f'_1, \ldots, f'_{\ell'}\}$ such that for any $f \in I(f_1, \ldots f_{\ell})$, there exists $i \in \{1, \ldots, \ell'\}$ such that $\text{LT}(f'_i) | \text{LT}(f)$ (LT = leading term for the ordering)

- Any $f \in I$ can be (uniquely) reduced by the GB

- Ideal membership ($f \in I$?) trivial given GB
Gröbner basis algorithms

- First algorithm by Buchberger [B65]
- Connection with linear algebra by Lazard [L83]

In F4 and F5, Macaulay matrices of increasing size are successively computed and linearly dependent rows are removed with linear algebra until a Gröbner basis is found.

In F5, some rows of the Macaulay matrices are omitted to avoid trivial relations like $0 = f_1 f_2 - f_2 f_1$.

In F4, the reductions are parallelized.
Gröbner basis algorithms

- First algorithm by Buchberger [B65]
- Connection with linear algebra by Lazard [L83]
- Best algorithms today are Faugère’s F4 and F5 [F99,F02]
- In F4 and F5, **Macaulay matrices** of increasing size are successively computed and linearly dependent rows are removed with linear algebra until a Gröbner basis is found
Gröbner basis algorithms

- First algorithm by Buchberger [B65]
- Connection with linear algebra by Lazard [L83]
- Best algorithms today are Faugère’s F4 and F5 [F99,F02]
- In F4 and F5, **Macaulay matrices** of increasing size are successively computed and linearly dependent rows are removed with linear algebra until a Gröbner basis is found
- In F5, some rows of the Macaulay matrices are omitted to avoid trivial relations like $0 = f_1 f_2 - f_2 f_1$
- In F4, the reductions are parallelized
Complexity of Gröbner basis algorithms

- Complexity of GB algorithms
  \( \approx \) cost of linear algebra on the largest Macaulay matrix
Complexity of Gröbner basis algorithms

- Complexity of GB algorithms
  \( \approx \) cost of linear algebra on the largest Macaulay matrix

- Important parameter: **degree of regularity**
  maximal degree \( D_{\text{reg}} \) of all polynomials computed
Complexity of Gröbner basis algorithms

- Complexity of GB algorithms
  \( \approx \) cost of linear algebra on the largest Macaulay matrix

- Important parameter: degree of regularity
  maximal degree \( D_{\text{reg}} \) of all polynomials computed

- \# monomials at this degree \( \approx n^{D_{\text{reg}}} \)
Complexity of Gröbner basis algorithms

- Complexity of GB algorithms
  \( \approx \) cost of linear algebra on the largest Macaulay matrix

- Important parameter: **degree of regularity**
  maximal degree \( D_{\text{reg}} \) of all polynomials computed

- \# monomials at this degree \( \approx n^{D_{\text{reg}}} \)

- Total cost (\( n \) variables) bounded in time and memory by
  \[ n^{\omega D_{\text{reg}}} \quad \text{and} \quad n^{2D_{\text{reg}}} \]

\( \omega \leq 3 \) linear algebra constant
“Random” systems

- For a random system of $n$ polynomial equations with degrees $d_1, \ldots, d_n$ in $n$ variables,

$$D_{\text{reg}} = 1 + \sum_{i=1}^{n} (d_i - 1)$$
“Random” systems

- For a random system of $n$ polynomial equations with degrees $d_1, \ldots, d_n$ in $n$ variables,

$$D_{reg} = 1 + \sum_{i=1}^{n} (d_i - 1)$$

- Overdetermined systems have lower degrees of regularity. Adding new equations helps
If $K := \mathbb{F}_q$, add the field equations $x_i^q - x_i = 0$ to the system

$$
\begin{align*}
  f_1(x_1, \ldots, x_n) &= 0 \\
  \ldots \\
  f_m(x_1, \ldots, x_n) &= 0 \\
  x_1^q - x_1 &= 0 \\
  \ldots \\
  x_n^q - x_n &= 0
\end{align*}
$$

 Degrees of regularity known for "generic" binary systems\cite{BFS04, BFS05}.
Polynomial systems over finite fields

- If $K := \mathbb{F}_q$, add the field equations $x_i^q - x_i = 0$ to the system

$$
\begin{cases}
  f_1(x_1, \ldots, x_n) = 0 \\
  \quad \vdots \\
  f_m(x_1, \ldots, x_n) = 0 \\
  x_1^q - x_1 = 0 \\
  \quad \vdots \\
  x_n^q - x_n = 0
\end{cases}
$$

- Degrees of regularity known for “generic” binary systems [BFS04,BFS05]
First fall degree

Other important parameter: **first fall degree** $D_{ff}$

Lowest degree $d$ such that there exist non trivial $g_i \in R$ with

\[
\max \deg(g_if_i) = d, \quad \deg \left( \sum g_if_i \right) < d
\]
First fall degree

- Other important parameter: **first fall degree** $D_{ff}$
  Lowest degree $d$ such that there exist *non trivial* $g_i \in R$ with

  \[
  \max \deg(g_i f_i) = d, \quad \deg \left( \sum g_i f_i \right) < d
  \]

- Trivial degree fall relations

  \[
  \sum g_i f_i = 0, \quad \text{or} \quad (f_i^{q-1} - 1)f_i = 0
  \]
First fall degree

- Other important parameter: **first fall degree** $D_{ff}$
  Lowest degree $d$ such that there exist *non trivial* $g_i \in R$ with

  $$\max \deg (g_i f_i) = d, \quad \deg \left( \sum g_i f_i \right) < d$$

- Trivial degree fall relations

  $$\sum g_i f_i = 0, \quad \text{or} \quad (f_i^{q-1} - 1)f_i = 0$$

- Sometimes called *degree of regularity* in the literature [DG10, DH11]
Degree of regularity vs. first fall degree

- For many classes of systems:
  
  \[ D_{ff} \approx D_{reg} \]

- Not true in general but experimental evidence for “random” systems and many “crypto” systems, including HFE and some variants.
Degree of regularity vs. first fall degree

- For many classes of systems:
  
  first fall degree $D_{ff} \approx$ degree of regularity $D_{reg}$

- Not true in general but experimental evidence for “random” systems and many “crypto” systems, including HFE and some variants.

- Intuition: for these systems, there are in fact many degree fall relations at $D_{ff}$ or $D_{ff} + 1$, that in turn produce many further lower degree relations, etc.
Degree of regularity vs. first fall degree

- For many classes of systems:
  
  first fall degree $D_{ff} \approx$ degree of regularity $D_{reg}$

- Not true in general but experimental evidence for “random” systems and many “crypto” systems, including HFE and some variants

- Intuition: for these systems, there are in fact many degree fall relations at $D_{ff}$ or $D_{ff} + 1$, that in turn produce many further lower degree relations, etc

- Assumption $D_{ff} \approx D_{reg}$ used in our analysis
Outline

From ECDLP to polynomial systems

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Back to ECDLP
Polynomial systems arising from a Weil descent

- Parameters: $n, n', m, t$
  $f \in \mathbb{F}_2^n[x_1, \ldots x_m]$ with degrees $\leq 2^t - 1$ in all variables
  $V$ a vector subspace of $\mathbb{F}_2^n/\mathbb{F}_2$ with dimension $n'$

- Problem: find $x_i \in V, i = 1, \ldots, m$ such that

  \[ f(x_1, \ldots, x_m) = 0. \]
Polynomial systems arising from a Weil descent

- Parameters: \( n, n', m, t \)
  \( f \in \mathbb{F}_{2^n}[x_1, \ldots, x_m] \) with degrees \( \leq 2^t - 1 \) in all variables
  \( V \) a vector subspace of \( \mathbb{F}_{2^n}/\mathbb{F}_2 \) with dimension \( n' \)

- Problem: find \( x_i \in V, i = 1, \ldots, m \) such that
  \[
  f(x_1, \ldots, x_m) = 0.
  \]

- If \( V := \mathbb{F}_{2^n} \), we can use Berlekamp [B70]
Polynomial systems arising from a Weil descent

- Parameters: \( n, n', m, t \)
  \[ f \in \mathbb{F}_{2^n}[x_1, \ldots, x_m] \] with degrees \( \leq 2^t - 1 \) in all variables
  \( V \) a vector subspace of \( \mathbb{F}_{2^n}/\mathbb{F}_2 \) with dimension \( n' \)

- Problem: find \( x_i \in V, i = 1, \ldots, m \) such that
  \[ f(x_1, \ldots, x_m) = 0. \]

- If \( V := \mathbb{F}_{2^n} \), we can use Berlekamp [B70]
- If \( mn' \approx n \), we expect \( \approx 1 \) solution
Polynomial systems arising from a Weil descent

- **Weil descent**: if \( \{v_1, \ldots, v_{n'}\} \) is a basis of \( V \) and \( \{\theta_1, \ldots, \theta_n\} \) is a basis of \( \mathbb{F}_{2^n} \) over \( \mathbb{F}_2 \), define **binary variables** \( x_{ij} \) such that \( x_i = \sum_j x_{ij}v_j \)
Polynomial systems arising from a Weil descent

- **Weil descent**: if \( \{v_1, \ldots, v_{n'}\} \) is a basis of \( V \) and \( \{\theta_1, \ldots, \theta_n\} \) is a basis of \( \mathbb{F}_{2^n} \) over \( \mathbb{F}_2 \), define **binary variables** \( x_{ij} \) such that \( x_i = \sum_j x_{ij}v_j \)

  substitute in \( f \) and “reduce modulo \( x_{ij}^2 - x_{ij} = 0 \)”

  decompose in the basis \( \{\theta_1, \ldots, \theta_n\} \)

\[
0 = f(x_1, \ldots, x_m) = f \left( \sum_{j=1}^{n'} x_{1j}v_j, \ldots, \sum_{j=1}^{n'} x_{mj}v_j \right)
\]

\[
= \left[ f \right]_1^\perp \theta_1 + \ldots + \left[ f \right]_n^\perp \theta_n
\]

- We get \( n \) equations \( \left[ f \right]_k^\perp = 0 \) in \( mn' \) variables \( x_{ij} \)
Degrees and block structure

If $e = e_0 + e_1 2 + e_2 4 + \ldots + e_{t-1} 2^{t-1}$ then

$$x_i^e = \left( \sum x_{ij} v_j \right)^{e_0} \left( \sum x_{ij}^2 v_j^2 \right)^{e_1} \ldots \left( \sum x_{ij}^{2^{t-1}} v_j^{2^{t-1}} \right)^{e_{t-1}}$$

$$= \left( \sum x_{ij} v_j \right)^{e_0} \left( \sum x_{ij} v_j^2 \right)^{e_1} \ldots \left( \sum x_{ij} v_j^{2^{t-1}} \right)^{e_{t-1}}$$

degree = Hamming weight of $(e_0, \ldots, e_{t-1})$
Degrees and block structure

- If \( e = e_0 + e_12 + e_24 + \ldots + e_{t-1}2^{t-1} \) then
  \[
  x_i^e = \left( \sum x_{ij}v_j \right)^{e_0} \left( \sum x_{ij}^2v_j^2 \right)^{e_1} \ldots \left( \sum x_{ij}^{2^{t-1}}v_j^{2^{t-1}} \right)^{e_{t-1}}
  \]
  degree = Hamming weight of \((e_0, \ldots, e_{t-1})\)

- \( f(x_1, \ldots, x_m) = [f]_1^{\downarrow} \theta_1 + \ldots + [f]_n^{\downarrow} \theta_n \)
  Since \( f \) has degree at most \( 2^t - 1 \) in each variable \( x_i \),
  Each \([f]_k^{\downarrow}\) has degree at most \( t \)
  in each block of variables \( X_i := \{x_{i1}, \ldots, x_{i,n'}\} \)
Applications

- Index calculus for binary elliptic curves
  Semaev’s polynomials: degree $2^{m-1}$ in each variable

- Hidden Field Equation (HFE) polynomial
  degree bounded by $D$; quadratic system over $\mathbb{F}_2$

- Index calculus for $\mathbb{F}_2^{*n}$
  degree 1 in each variable ($t = 1$)

- Factorization problem in $SL(2, \mathbb{F}_{2^n})$
  degree 1 in each variable ($t = 1$)
Example : HFE

- Public Key Cryptosystem proposed by Patarin [P96]
- Private key is a polynomial $f \in \mathbb{F}_{2^n}[x]$
  - Public key is a disguised version of its Weil descent
  - Attacker only knows the disguised system
Example : HFE

- Public Key Cryptosystem proposed by Patarin [P96]
- Private key is a polynomial \( f \in \mathbb{F}_{2^n}[x] \)
  Public key is a disguised version of its Weil descent
  Attacker only knows the disguised system
- Particularities
  - “Disguised” . . . but no impact on GB complexity
Example: HFE

- Public Key Cryptosystem proposed by Patarin [P96]
- Private key is a polynomial $f \in \mathbb{F}_{2^n}[x]$
  Public key is a disguised version of its Weil descent
  Attacker only knows the disguised system
- Particularities
  - “Disguised” . . . but no impact on GB complexity
  - Monovariate ($m = 1$)
  - $f$ has a particular shape

$$f(x) := \sum_{2^i + 2j < D} a_{ij}x^{2^i + 2j} + \sum_{2i < D} b_i x^{2^i} + c$$

Weil descent on $f$ leads to a quadratic system
Back to the general case

We have $n$ equations in $mn'$ variables $x_{ij}$, given by

$$0 = f(x_1, \ldots, x_m) = [f]_1 \theta_1 + \ldots + [f]_n \theta_n$$
Back to the general case

- We have $n$ equations in $mn'$ variables $x_{ij}$, given by

$$0 = f(x_1, \ldots, x_m) = \left[f\right]^1_{1} \theta_1 + \ldots + \left[f\right]^1_{n} \theta_n$$

- Adding new (low degree) equations may accelerate the resolution
- Can we find more equations?
Frobenius transforms are useless

- Frobenius transforms $f = 0 \Rightarrow f^2 = 0$
Frobenius transforms are useless

- Frobenius transforms $f = 0 \Rightarrow f^2 = 0$
- HW of exponents in $f$ and $f^2$ are equal
  $\Rightarrow [f]_i$ and $[f^2]_i$ have the same degrees
Frobenius transforms are useless

- Frobenius transforms $f = 0 \Rightarrow f^2 = 0$
- HW of exponents in $f$ and $f^2$ are equal
  $\Rightarrow [f]_i$ and $[f^2]_i$ have the same degrees
- But

$$f^2 = \left( \sum_{i=1}^{n} [f]_i \theta_i \right)^2 = \sum_{i=1}^{n} [f]_i \theta_i^2 =$$
Frobenius transforms are useless

- Frobenius transforms $f = 0 \Rightarrow f^2 = 0$
- HW of exponents in $f$ and $f^2$ are equal
  $\Rightarrow [f]^\dagger_i$ and $[f^2]^\dagger_i$ have the same degrees
- But

$$f^2 = \left( \sum_{i=1}^{n} [f]^\dagger_i \theta_i \right)^2 = \sum_{i=1}^{n} [f]^\dagger_i \theta_i^2 = \sum_{i=1}^{n} [f]^\dagger_i \left( \sum_{j=1}^{n} a_{ij} \theta_j \right)$$
Frobenius transforms are useless

- Frobenius transforms $f = 0 \Rightarrow f^2 = 0$
- HW of exponents in $f$ and $f^2$ are equal
  $\Rightarrow [f]^\dagger_i$ and $[f^2]^\dagger_i$ have the same degrees
- But

$$f^2 = \left(\sum_{i=1}^{n} [f]^\dagger_i \theta_i\right)^2 = \sum_{i=1}^{n} [f]^\dagger_i \theta_i^2 = \sum_{i=1}^{n} [f]^\dagger_i \left(\sum_{j=1}^{n} a_{ij} \theta_j\right)$$

$$= \sum_{j=1}^{n} \left(\sum_{i=1}^{n} a_{ij} [f]^\dagger_i\right) \theta_j$$
Frobenius transforms are useless

- Frobenius transforms $f = 0 \Rightarrow f^2 = 0$
- HW of exponents in $f$ and $f^2$ are equal
  $\Rightarrow [f]^\uparrow_i$ and $[f^2]^\uparrow_i$ have the same degrees
- But

$$f^2 = \left( \sum_{i=1}^{n} [f]^\uparrow_i \theta_i \right)^2 = \sum_{i=1}^{n} [f]^\uparrow_i \theta_i^2 = \sum_{i=1}^{n} [f]^\uparrow_i \left( \sum_{j=1}^{n} a_{ij} \theta_j \right)$$

$$= \sum_{j=1}^{n} \left( \sum_{i=1}^{n} a_{ij} [f]^\uparrow_i \right) \theta_j = \sum_{i=1}^{n} [f^2]^\uparrow_j \theta_j$$
Frobenius transforms are useless

- Frobenius transforms $f = 0 \Rightarrow f^2 = 0$
- HW of exponents in $f$ and $f^2$ are equal
  $\Rightarrow [f]_i$ and $[f^2]_i$ have the same degrees
- But

$$f^2 = \left( \sum_{i=1}^{n} [f]_i^\dagger \theta_i \right)^2 = \sum_{i=1}^{n} [f]_i^\dagger \theta_i^2 = \sum_{i=1}^{n} [f]_i^\dagger \left( \sum_{j=1}^{n} a_{ij} \theta_j \right)$$

$$= \sum_{j=1}^{n} \left( \sum_{i=1}^{n} a_{ij} [f]_i^\dagger \right) \theta_j = \sum_{i=1}^{n} [f^2]_j^\dagger \theta_j$$

same equations! (linear combinations)
New equations

- \( 0 = f \Rightarrow 0 = x_1 f \)
New equations

- $0 = f \Rightarrow 0 = x_1 f$
  
  $0 = x_1 f(x_1, \ldots, x_m) = [x_1 f]^\frac{1}{1} \theta_1 + \ldots + [x_1 f]^\frac{1}{n} \theta_n$

- Not the same equations!

In particular, homogeneous in block $X_1$
New equations

- $0 = f \Rightarrow 0 = x_1 f$
  \[ 0 = x_1 f(x_1, \ldots, x_m) = [x_1 f]_1 \theta_1 + \ldots + [x_1 f]_n \theta_n \]
- $x_1 f$ has degree $\leq (2^t)$ in $x_1$ and $\leq (2^t - 1)$ in $x_2, \ldots, x_m$
- $[x_1 f]_k$ has degree at most $t$ in each block $X_i$
New equations

- $0 = f \Rightarrow 0 = x_1 f$
- $0 = x_1 f(x_1, \ldots, x_m) = [x_1 f]^1_1 \theta_1 + \ldots + [x_1 f]^1_n \theta_n$
- $x_1 f$ has degree $\leq (2^t)$ in $x_1$ and $\leq (2^t - 1)$ in $x_2, \ldots, x_m$
- $[x_1 f]^1_k$ has degree at most $t$ in each block $X_i$
- Not the same equations!
  In particular, homogeneous in block $X_1$
  $f(x_1, \ldots, x_m) = f_0(x_2, \ldots, x_m) + x_1 f_1(x_2, \ldots, x_m)$
  $\Rightarrow x_1 f(x_1, \ldots, x_m) = x_1 f_0(x_2, \ldots, x_m) + x_1^2 f_1(x_2, \ldots, x_m)$
**New equations**

- \(0 = f \Rightarrow 0 = x_1 f\)
  
  \[0 = x_1 f(x_1, \ldots, x_m) = [x_1 f]_1 \theta_1 + \ldots + [x_1 f]_n \theta_n\]

- \(x_1 f\) has degree \(\leq (2^t)\) in \(x_1\) and \(\leq (2^t - 1)\) in \(x_2, \ldots, x_m\)

- \([x_1 f]_k\) has degree at most \(t\) in each block \(X_i\)

- Not the same equations!
  
  In particular, homogeneous in block \(X_1\)
  
  \[f(x_1, \ldots, x_m) = f_0(x_2, \ldots, x_m) + x_1 f_1(x_2, \ldots, x_m)\]
  
  \(\Rightarrow x_1 f(x_1, \ldots, x_m) = x_1 f_0(x_2, \ldots, x_m) + x_1^2 f_1(x_2, \ldots, x_m)\)

- Similar equations with other monomials instead of \(x_1\)
  
  Many new low degree equations
New equations, revisited

- Let $a_{ijk} \in \mathbb{F}_2$ such that $\theta_i \theta_j = \sum_k a_{ijk} \theta_k$

$$x_1 f = \left( \sum_{i=1}^n [x_1]_i \theta_i \right) \left( \sum_{j=1}^n [f]_j \theta_j \right) = \sum_{i,j,k=1}^n a_{ijk} [x_1]_i [f]_j \theta_k.$$
New equations, revisited

- Let $a_{ijk} \in \mathbb{F}_2$ such that $\theta_i \theta_j = \sum_k a_{ijk} \theta_k$

\[ x_1 f = \left( \sum_{i=1}^{n} [x_1]_i \theta_i \right) \left( \sum_{j=1}^{n} [f]_j \theta_j \right) = \sum_{i,j,k=1}^{n} a_{ijk} [x_1]_i [f]_j \theta_k. \]

- Hence

\[ [x_1 f]_k^\dagger = \sum_{i,j=1}^{n} a_{ijk} [x_1]_i^\dagger [f]_j^\dagger = \sum_{j=1}^{n} p_{ik}(x_{11}, \ldots, x_{1,n'}) [f]_j^\dagger \]

with $\deg(p_{ik}) = 1$
New equations, revisited

- Let \( a_{ijk} \in \mathbb{F}_2 \) such that \( \theta_i \theta_j = \sum_k a_{ijk} \theta_k \)

\[
x_1 f = \left( \sum_{i=1}^{n} [x_1]_i^\dagger \theta_i \right) \left( \sum_{j=1}^{n} [f]_j^\dagger \theta_j \right) = \sum_{i,j,k=1}^{n} a_{ijk} [x_1]_i^\dagger [f]_j^\dagger \theta_k.
\]

- Hence

\[
[x_1 f]_k^\dagger = \sum_{i,j=1}^{n} a_{ijk} [x_1]_i^\dagger [f]_j^\dagger = \sum_{j=1}^{n} p_{ik}(x_{11}, \ldots, x_{1,n'}) [f]_j^\dagger
\]

with \( \text{deg}(p_{ik}) = 1 \)

- The “new” equations \( [x_1 f]_k^\dagger = 0 \) are algebraic combinations of the original ones \( [f]_j^\dagger = 0 \)
New equations, revisited

Let \( a_{ijk} \in \mathbb{F}_2 \) such that

\[
\theta_i \theta_j = \sum_k a_{ijk} \theta_k
\]

\[
x_1 f = \left( \sum_{i=1}^{n} [x_1]_i \theta_i \right) \left( \sum_{j=1}^{n} [f]_j \theta_j \right) = \sum_{i,j,k=1}^{n} a_{ijk} [x_1]_i [f]_j \theta_k.
\]

Hence

\[
[x_1 f]_k^\perp = \sum_{i,j=1}^{n} a_{ijk} [x_1]_i [f]_j^\perp = \sum_{j=1}^{n} p_{ik}(x_{11}, \ldots, x_{1,n'}) [f]_j^\perp
\]

with \( \deg(p_{ik}) = 1 \)

The “new” equations \( [x_1 f]_k^\perp = 0 \) are algebraic combinations of the original ones \( [f]_j^\perp = 0 \)

Will be recovered “blindly” by GB algorithms
First fall degree

We have

\[ [x_1 f]_k^\dagger = \sum_{j=1}^{n} p_{ik}(x_{11}, \ldots, x_{1,n'}) [f]_j^\dagger \]

\[ \deg([x_1 f]_k^\dagger) = mt, \quad \deg(p_{ik}) = 1, \quad \deg([f]_j^\dagger) = mt \]
First fall degree

- We have

\[ [x_1 f]_k = \sum_{j=1}^{n} p_{ik}(x_{11}, \ldots, x_{1, n'}) [f]_j \]

\[ \deg([x_1 f]_k) = mt, \quad \deg(p_{ik}) = 1, \quad \deg([f]_j) = mt \]

- Non trivial low degree relation!
- First fall degree \( D_{ff} \leq mt + 1 \)
First fall degree

- We have

\[ [x_1 f]^\uparrow_k = \sum_{j=1}^{n} p_{ik}(x_{11}, \ldots, x_{1,n'}) [f]^\uparrow_j \]

\[ \deg([x_1 f]^\uparrow_k) = mt, \quad \deg(p_{ik}) = 1, \quad \deg([f]^\uparrow_j) = mt \]

- Non trivial low degree relation!
- First fall degree \( D_{ff} \leq mt + 1 \)
- Essentially as small as it could be (unless \( f \) degenerate)
Heuristic assumption

- We will heuristically assume $D_{\text{reg}} \approx D_{\text{ff}}$ in most cases, for $f$ chosen randomly with degrees $\leq 2^{t-1}$ for $V$ chosen randomly with dimension $n'$
Heuristic assumption

- We will heuristically assume $D_{reg} \approx D_{ff}$
  in most cases,
  for $f$ chosen randomly with degrees $\leq 2^{t-1}$
  for $V$ chosen randomly with dimension $n'$

- “Classical” assumption in algebraic cryptanalysis
  - Experimental evidence for “random” and many “crypto”
    systems including HFE
  - (Confusion in literature between the two notions)
Heuristic assumption

- We will heuristically assume $D_{\text{reg}} \approx D_{\text{ff}}$ in most cases,
  for $f$ chosen randomly with degrees $\leq 2^{t-1}$
  for $V$ chosen randomly with dimension $n'$

- “Classical” assumption in algebraic cryptanalysis
  - Experimental evidence for “random” and many “crypto” systems including HFE
  - (Confusion in literature between the two notions)

- Leads to $D_{\text{reg}} \approx mt + 1$
  (instead of $D_{\text{reg}} = n(mt - 1) + 1$ for generic systems)
Experimental evidence that $D_{reg} \approx mt + 1$

<table>
<thead>
<tr>
<th>$t$</th>
<th>$n$</th>
<th>$n'$</th>
<th>$m$</th>
<th>$mt + 1$</th>
<th>$D_{av}$</th>
<th>Av. time (s)</th>
<th>Mem (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3.1</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3.8</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3.0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>3.6</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4.2</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5.3</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>7.4</td>
<td>1</td>
<td>23</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4.1</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>6.3</td>
<td>7</td>
<td>114</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>3.0</td>
<td>14</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5.3</td>
<td>16</td>
<td>98</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>9.6</td>
<td>69</td>
<td>3388</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>9</td>
<td>2</td>
<td>3</td>
<td>3.0</td>
<td>85</td>
<td>74</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>4.1</td>
<td>86</td>
<td>89</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>7.4</td>
<td>233</td>
<td>5398</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>3.0</td>
<td>487</td>
<td>291</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>6.2</td>
<td>515</td>
<td>733</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>6.2</td>
<td>669</td>
<td>3226</td>
</tr>
</tbody>
</table>
Experimental evidence that $D_{\text{reg}} \approx mt + 1$

<table>
<thead>
<tr>
<th>$t$</th>
<th>$n$</th>
<th>$n'$</th>
<th>$m$</th>
<th>$mt + 1$</th>
<th>$D_{\text{av}}$</th>
<th>Av. time (s)</th>
<th>Mem (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>5.1</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>6.7</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>5.1</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>3</td>
<td>3</td>
<td>7</td>
<td>7.2</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>7.1</td>
<td>1</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>13</td>
<td>3</td>
<td>4</td>
<td>9</td>
<td>9.3</td>
<td>2</td>
<td>95</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>7.0</td>
<td>12</td>
<td>263</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>5</td>
<td>5.1</td>
<td>13</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>6.6</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>7.0</td>
<td>1</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>4</td>
<td>3</td>
<td>10</td>
<td>10.1</td>
<td>9</td>
<td>70</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>13</td>
<td>12.6</td>
<td>70</td>
<td>113</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>5</td>
<td>3</td>
<td>10</td>
<td>10.0</td>
<td>118</td>
<td>2371</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>7</td>
<td>7.0</td>
<td>23</td>
<td>253</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>4</td>
<td>4</td>
<td>13</td>
<td>13.2</td>
<td>1891</td>
<td>20135</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>8.7</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>4</td>
<td>3</td>
<td>13</td>
<td>12.6</td>
<td>199</td>
<td>116</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>5</td>
<td>3</td>
<td>13</td>
<td>13.1</td>
<td>2904</td>
<td>6696</td>
</tr>
</tbody>
</table>
Complexity analysis

- Assuming $D_{reg} \approx D_{ff}$, we have $D_{reg} \approx mt + 1$
- Time and memory bounded by

$$n^{\omega D_{reg}} \text{ and } n^{2D_{reg}}$$

$\omega \leq 3$: linear algebra constant
Complexity analysis

- Assuming $D_{reg} \approx D_{ff}$, we have $D_{reg} \approx mt + 1$
- Time and memory bounded by
  \[ n^{\omega D_{reg}} \quad \text{and} \quad n^{2D_{reg}} \]
  \[
  \omega \leq 3 : \text{linear algebra constant}
  \]
- Block structure $\Rightarrow$ time and memory bounded by
  \[ (n')^{\omega D_{reg}} \quad \text{and} \quad (n')^{2D_{reg}} \]
Remarks

- Heuristic assumption $D_{\text{reg}} \approx D_{\text{ff}}$
- Assumption must be adapted (and checked) in particular cases
Remarks

- Heuristic assumption $D_{reg} \approx D_{ff}$
- Assumption must be adapted (and checked) in particular cases
- Similar analysis for other “small characteristic” fields

$$D_{reg} \approx (p - 1)mt + 1$$
HFE as a particular case

- Cryptanalysis leads to a particular case of our systems with $m = 1$, $t = \lceil \log_2 D \rceil$, $V = \mathbb{F}_{2^n}$

$$D_{\text{reg}} \approx D_{\text{ff}} \geq mt + 1 = \lceil \log_2 D \rceil + 1$$
HFE as a particular case

- Cryptanalysis leads to a particular case of our systems with $m = 1$, $t = \lceil \log_2 D \rceil$, $V = \mathbb{F}_{2^n}$

\[
D_{reg} \approx D_{ff} \geq mt + 1 = \lceil \log_2 D \rceil + 1
\]

We recover [KS99,FJ03,GJS06,DG10,DH11,...]
HFE as a particular case

- Cryptanalysis leads to a particular case of our systems with $m = 1$, $t = \lceil \log_2 D \rceil$, $V = \mathbb{F}_{2^n}$

\[ D_{\text{reg}} \approx D_{\text{ff}} \geq mt + 1 = \lceil \log_2 D \rceil + 1 \]

We recover [KS99,FJ03,GJS06,DG10,DH11,...]

- No impact of HFE special shape
  Other restrictions may have a (positive) impact [DH11]
Similarities with HFE

- Polynomial system arising from a Weil descent
- Many low degree relations [C01,...]
- First fall degree [DG10,DH11,...]
Similarities with HFE

- Polynomial system arising from a Weil descent
- Many low degree relations [C01,...]
- First fall degree [DG10, DH11,...]
- Subsystem with smaller number of variables [GJS06,...] (not discussed here)
Similarities with HFE

- Polynomial system arising from a Weil descent
- Many low degree relations [C01,...]
- First fall degree [DG10,DH11,...]
- Subsystem with smaller number of variables [GJS06,...] (not discussed here)

- Assumption $D_{reg} \approx D_{ff}$ widely verified for HFE polynomials [FJ03,GJS06,...]
Outline

From ECDLP to polynomial systems

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Back to ECDLP
Diem’s variant of index calculus [D11b]

Fix $n' < n$ and $m \approx n/n'$

- **Factor basis**: Choose a vector subspace $V$ of $\mathbb{F}_{2^n}$ with dimension $n'$
  Define $\mathcal{F}_V := \{(x, y) \in E | x \in V\}$

- **Relation search**: find about $2^{n'}$ relations. For each one,
  Compute $(X_i, Y_i) := a_i P + b_i Q$ for random $a_i, b_i$
  Find $x_j \in V$ with $S_{m+1}(x_1, \ldots, x_m, X_i) = 0$
  Find the corresponding $y_j$

- **Linear algebra** between the relations
Finding relations

- Find $x_j \in V$ with $S_{m+1}(x_1, \ldots, x_m, X_i) = 0$
Finding relations

- Find \( x_j \in V \) with \( S_{m+1}(x_1, \ldots, x_m, X_i) = 0 \)
- Weil descent \( \rightarrow \) polynomial system
  - finite field \( \mathbb{F}_{2^n} \), vector subspace \( V \) dimension \( n' \)
  - \( m \) variables
  - degree \( 2^{m-1} \) in each variable \( \Rightarrow t = m \)
- Our analysis leads to \( D_{ff} \leq mt + 1 = m^2 + 1 \) (not tight)
Finding relations

- Find \( x_j \in V \) with \( S_{m+1}(x_1, \ldots, x_m, X_i) = 0 \)
- Weil descent \( \rightarrow \) polynomial system
  - finite field \( \mathbb{F}_{2^n} \), vector subspace \( V \) dimension \( n' \)
  - \( m \) variables
  - degree \( 2^{m-1} \) in each variable \( \Rightarrow t = m \)
- Our analysis leads to \( D_{ff} \leq mt + 1 = m^2 + 1 \) (not tight)
- ! Summation polynomials not “random”! (symmetric, . . . )
Heuristic assumption

- Let $n, n', m, E$ be fixed.
  Let $R_i = (X_i, Y_i)$ be a random point of $E$.
  Let $V$ be a random vector space of dimension $n'$.

- **Assumption**: after applying a Weil descent to
  
  $$S_{m+1}(x_1, \ldots, x_m, X_i) = 0,$$

  the resulting system satisfies $D_{\text{reg}} \approx D_{\text{ff}}$
Experimental verification $D_{\text{reg}} \approx D_{\text{ff}}$

- Random curves $E : y^2 + xy = x^3 + a_4x^2 + a_6$ for random $a_4, a_6$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n'$</th>
<th>$m$</th>
<th>$t$</th>
<th>$mt + 1 \geq D_{\text{ff}}$</th>
<th>$D_{\text{av}}$</th>
<th>Time</th>
<th>Mem.</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>3.0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>10</td>
<td>7.1</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>4.0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>17</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>10</td>
<td>7.1</td>
<td>130</td>
<td>2136</td>
</tr>
</tbody>
</table>

$D_{\text{reg}}$ even lower than expected
Experimental verification $D_{\text{reg}} \approx D_{\text{ff}}$

- Koblitz curves $E : y^2 + xy = x^3 + x^2 + 1$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n'$</th>
<th>$m$</th>
<th>$t$</th>
<th>$mt + 1 \geq D_{\text{ff}}$</th>
<th>$D_{\text{av}}$</th>
<th>Time</th>
<th>Mem.</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>3.0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>10</td>
<td>7.1</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>4.0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>17</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>10</td>
<td>7.2</td>
<td>132</td>
<td>2133</td>
</tr>
</tbody>
</table>

$D_{\text{reg}}$ even lower than expected
Complexity of Diem’s algorithm

- Computing $S_{m+1}$ with resultants: cost $2^{t_1}$ where
  \[ t_1 \approx m(m + 1) \]
Complexity of Diem’s algorithm

- Computing $S_{m+1}$ with resultants: cost $2^{t_1}$ where
  \[ t_1 \approx m(m + 1) \]

- Finding $2^{n'}$ relations: total cost $2^{t_2}$ where
  \[ t_2 \approx n' + m \log m + \omega(m^2 + 1) \log n' \]
  - Each one costs $(n')^{\omega(mt+1)} = (n')^{\omega(m^2+1)}$
  - Additional factor $m!$ lost due to symmetry

(Sparse) linear algebra on relations: cost $2^{\omega' t_3}$ where
\[ t_3 \approx \log m + \log n' + \omega'(n') \]
Complexity of Diem’s algorithm

- Computing $S_{m+1}$ with resultants: cost $2^{t_1}$ where
  \[ t_1 \approx m(m + 1) \]

- Finding $2^{n'}$ relations: total cost $2^{t_2}$ where
  \[ t_2 \approx n' + m \log m + \omega (m^2 + 1) \log n' \]
  - Each one costs $(n')^\omega (mt+1) = (n')^{\omega (m^2 + 1)}$
  - Additional factor $m!$ lost due to symmetry

- (Sparse) linear algebra on relations: cost $2^{\omega't_3}$ where
  \[ t_3 \approx \log m + \log n + \omega' n' \]
Estimations for “small” parameters

<table>
<thead>
<tr>
<th>$n$</th>
<th>$m$</th>
<th>$n'$</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>2</td>
<td>25</td>
<td>6</td>
<td>97</td>
<td>57</td>
<td>97</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
<td>50</td>
<td>6</td>
<td>137</td>
<td>108</td>
<td>137</td>
</tr>
<tr>
<td>160</td>
<td>2</td>
<td>80</td>
<td>6</td>
<td>177</td>
<td>168</td>
<td>177</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
<td>100</td>
<td>6</td>
<td>202</td>
<td>209</td>
<td>209</td>
</tr>
<tr>
<td>500</td>
<td>3</td>
<td>167</td>
<td>12</td>
<td>393</td>
<td>344</td>
<td>393</td>
</tr>
<tr>
<td>1000</td>
<td>4</td>
<td>250</td>
<td>20</td>
<td>664</td>
<td>512</td>
<td>664</td>
</tr>
<tr>
<td>2000</td>
<td>4</td>
<td>500</td>
<td>20</td>
<td>965</td>
<td>1013</td>
<td>1013</td>
</tr>
<tr>
<td>5000</td>
<td>6</td>
<td>833</td>
<td>42</td>
<td>1926</td>
<td>1682</td>
<td>1926</td>
</tr>
<tr>
<td>10000</td>
<td>7</td>
<td>1429</td>
<td>56</td>
<td>3020</td>
<td>2873</td>
<td>3020</td>
</tr>
<tr>
<td>20000</td>
<td>9</td>
<td>2222</td>
<td>90</td>
<td>4986</td>
<td>4462</td>
<td>4986</td>
</tr>
<tr>
<td>50000</td>
<td>11</td>
<td>4545</td>
<td>132</td>
<td>9030</td>
<td>9110</td>
<td>9110</td>
</tr>
<tr>
<td>100000</td>
<td>14</td>
<td>7143</td>
<td>210</td>
<td>14762</td>
<td>14306</td>
<td>14762</td>
</tr>
</tbody>
</table>
Asymptotic estimates

- Fix $n' := n^\alpha$ and $m := n^{1-\alpha}$ for $\alpha := 2/3$
  
  $t_1 \approx n^{2/3}$,
  
  $t_2 \approx (1/3)n^{1/3} \log n + n^{2/3} + (2/3)\omega n^{2/3} \log n$,
  
  $t_3 \approx (4/3) \log n + \omega' n^{2/3}$
Asymptotic estimates

- Fix \( n' := n^\alpha \) and \( m := n^{1-\alpha} \) for \( \alpha := 2/3 \)
  
  \[
  \begin{align*}
  t_1 & \approx n^{2/3}, \\
  t_2 & \approx (1/3)n^{1/3}\log n + n^{2/3} + (2/3)\omega n^{2/3}\log n, \\
  t_3 & \approx (4/3)\log n + \omega' n^{2/3}
  \end{align*}
  \]

- Overall complexity

  \[2^T \text{ with } T \approx cn^{2/3}\log n \text{ and } c := \frac{2}{3}\omega \leq 2\]
Outline

From ECDLP to polynomial systems

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Back to ECDLP
Conclusion

- ECDLP subexponential for binary curves?
  - Reasonable evidence under heuristic assumption
  - Diem’s algorithm would beat BSGS for $n \geq 2000$
  - NIST curves remain safe so far
  - Extension to any “small” characteristic field
Conclusion

- ECDLP subexponential for binary curves?
  - Reasonable evidence under heuristic assumption
  - Diem’s algorithm would beat BSGS for $n \geq 2000$
  - NIST curves remain safe so far
  - Extension to any “small” characteristic field

- Polynomial systems arising from a Weil descent
  - Very important class of systems for cryptography
  - ECDLP, HFE, DLP, factoring in $SL(2, \mathbb{F}_{2^n})$, . . .
Conclusion

- ECDLP subexponential for binary curves?
  - Reasonable evidence under heuristic assumption
  - Diem’s algorithm would beat BSGS for \( n \geq 2000 \)
  - NIST curves remain safe so far
  - Extension to any “small” characteristic field

- Polynomial systems arising from a Weil descent
  - Very important class of systems for cryptography
  - ECDLP, HFE, DLP, factoring in \( SL(2, \mathbb{F}_{2^n}) \), . . .

- Future work
  - Better algorithms, remove heuristic assumptions
  - Extension to prime fields?
References

References

- [B70] E Berlekamp. *Factoring polynomials over large finite fields.*
References

References

- [F99] JC Faugère. *A new efficient algorithm for computing Gröbner bases (F4).*
- [F02] JC Faugère. *A new efficient algorithm for computing Gröbner bases without reduction to zero (F5).*
References

- [GS99] S Galbraith, N Smart. *A cryptographic application of the Weil descent.*
References

- [G09] P Gaudry. *Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem.*
- [GHS00] P Gaudry, F Hess, N Smart. *Constructive and destructive facets of Weil descent on elliptic curves.*
- [GJS06] L Granboulan and A Joux and J Stern. *Inverting HFE Is Quasipolynomial.*
References

- [MOV93] A Menezes, T Okamoto, S Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field
- [MQ01] A Menezes, M Qu. Analysis of the Weil descent attack of Gaudry, Hess and Smart
References

- [SA21] T Satoh, K Araki. *Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves*
- [S98] I Semaev. *Evaluation of a discrete logarithm in a group of p-torsion points of an elliptic curve in characteristic p*
- [S99] N Smart. *The discrete logarithm problem on elliptic curves of trace one*