

On polynomial systems arising from a Weil descent

Based on joint works

with JC Faugère, JJ Quisquater, L Perret, G Renault

Christophe Petit



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**
- ▶ Systems usually solved with **Gröbner basis algorithms**



Algebraic cryptanalysis

- ▶ Reduce some cryptanalytic problems to the resolution of some systems of **multivariate polynomial equations**
- ▶ Systems usually solved with **Gröbner basis algorithms**
- ▶ Success stories :
 - ▶ HFE and variants
 - ▶ Isomorphism of polynomials
 - ▶ MacEliece variants
 - ▶ Algebraic side-channel attacks



Structured systems

- ▶ Generic systems are hard to solve, but **“cryptanalysis” systems are far from generic**
- ▶ The special structure of these systems helps their resolution
- ▶ Sometimes, dedicated algorithms can be built



Structured systems

- ▶ Generic systems are hard to solve, but **“cryptanalysis” systems are far from generic**
- ▶ The special structure of these systems helps their resolution
- ▶ Sometimes, dedicated algorithms can be built
- ▶ This talk : a class of polynomial systems, their analysis, and some cryptographic applications (including ECDLP)



Outline

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Application to ECDLP

Further applications



Outline

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Application to ECDLP

Further applications



Polynomial systems

- ▶ Let K be a field and $R := K[x_1, \dots, x_n]$.

Let $f_1, \dots, f_m \in R$.

Solve

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



Polynomial systems

- ▶ Let K be a field and $R := K[x_1, \dots, x_n]$.

Let $f_1, \dots, f_m \in R$.

Solve

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

- ▶ 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** \mathcal{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 \mathcal{M}_d provides a new “triangular” system of equations

$$\begin{cases} g_1(x_1, \dots, x_{n-1}, x_n) = 0 \\ \dots \\ g_{m'-1}(x_{n-1}, x_n) = 0 \\ g_{m'}(x_n) = 0 \end{cases}$$



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 \mathcal{M}_d provides a new “triangular” system of equations

$$\begin{cases} g_1(x_1, \dots, x_{n-1}, x_n) = 0 \\ \dots \\ g_{m'-1}(x_{n-1}, x_n) = 0 \\ g_{m'}(x_n) = 0 \end{cases}$$

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



Gröbner bases

- ▶ Given an ideal $I(f_1, \dots, f_m)$ and a monomial ordering $>$, a *Gröbner basis* (GB) for this ordering is a basis $\{f'_1, \dots, f'_{\ell'}\}$ such that for any $f \in I(f_1, \dots, f_m)$, there exists $i \in \{1, \dots, \ell'\}$ 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 $I(f_1, \dots, f_m)$ and a monomial ordering $>$, a *Gröbner basis* (GB) for this ordering is a basis $\{f'_1, \dots, f'_{\ell'}\}$ such that for any $f \in I(f_1, \dots, f_m)$, there exists $i \in \{1, \dots, \ell'\}$ 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
- ▶ Ideal membership ($f \in I?$) trivial given GB



Gröbner basis algorithms

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



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
≈ cost of linear algebra on the largest Macaulay matrix
- ▶ Important parameter : **degree of regularity**
maximal degree D_{reg} of all polynomials computed



Complexity of Gröbner basis algorithms

- ▶ Complexity of GB algorithms
≈ cost of linear algebra on the largest Macaulay matrix
- ▶ Important parameter : **degree of regularity**
maximal degree D_{reg} of all polynomials computed
- ▶ # monomials at this degree bounded by $n^{D_{reg}}$



Complexity of Gröbner basis algorithms

- ▶ Complexity of GB algorithms
≈ cost of linear algebra on the largest Macaulay matrix
- ▶ Important parameter : **degree of regularity**
maximal degree D_{reg} of all polynomials computed
- ▶ # monomials at this degree bounded by $n^{D_{reg}}$
- ▶ Total cost (n variables) bounded in time and memory by

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

$\omega \leq 3$ linear algebra constant



“Random” systems

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

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



“Random” systems

- ▶ For a random system of n polynomial equations with degrees d_1, \dots, 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



Polynomial systems over finite fields

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

$$\left\{ \begin{array}{l} f_1(x_1, \dots, x_n) = 0 \\ \dots \\ f_m(x_1, \dots, x_n) = 0 \\ x_1^q - x_1 = 0 \\ \dots \\ x_n^q - x_n = 0 \end{array} \right.$$



Polynomial systems over finite fields

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

$$\left\{ \begin{array}{l} f_1(x_1, \dots, x_n) = 0 \\ \dots \\ f_m(x_1, \dots, x_n) = 0 \\ x_1^q - x_1 = 0 \\ \dots \\ x_n^q - x_n = 0 \end{array} \right.$$

- ▶ 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

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



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 :

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



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

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Application to ECDLP

Further applications



Polynomial systems arising from a Weil descent

- ▶ Parameters : n, n', m, t
 $f \in \mathbb{F}_{2^n}[x_1, \dots, x_m]$ 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 $x_i \in V, i = 1, \dots, m$ such that

$$f(x_1, \dots, x_m) = 0.$$



Polynomial systems arising from a Weil descent

- ▶ Parameters : n, n', m, t
 $f \in \mathbb{F}_{2^n}[x_1, \dots, x_m]$ 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 $x_i \in V, i = 1, \dots, m$ such that

$$f(x_1, \dots, x_m) = 0.$$

- ▶ If $V := \mathbb{F}_{2^n}$, we can use Berlekamp [B70]



Polynomial systems arising from a Weil descent

- ▶ Parameters : n, n', m, t
 $f \in \mathbb{F}_{2^n}[x_1, \dots, x_m]$ 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 $x_i \in V, i = 1, \dots, m$ such that

$$f(x_1, \dots, x_m) = 0.$$

- ▶ If $V := \mathbb{F}_{2^n}$, we can use Berlekamp [B70]
- ▶ If $mn' \approx n$, we expect ≈ 1 solution



Polynomial systems arising from a Weil descent

- ▶ **Weil descent** : if $\{v_1, \dots, v_{n'}\}$ is a basis of V and $\{\theta_1, \dots, \theta_n\}$ is a basis of \mathbb{F}_{2^n} over \mathbb{F}_2 , define **binary variables** x_{ij} such that $\mathbf{x}_i = \sum_j x_{ij} \mathbf{v}_j$



Polynomial systems arising from a Weil descent

- ▶ **Weil descent** : if $\{v_1, \dots, v_{n'}\}$ is a basis of V and $\{\theta_1, \dots, \theta_n\}$ is a basis of \mathbb{F}_{2^n} over \mathbb{F}_2 , define **binary variables** x_{ij} such that $\mathbf{x}_i = \sum_j x_{ij} v_j$ substitute in f and “**reduce modulo** $\mathbf{x}_{ij}^2 - \mathbf{x}_{ij} = \mathbf{0}$ ”
decompose in the basis $\{\theta_1, \dots, \theta_n\}$

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



Polynomial systems arising from a Weil descent

- ▶ **Weil descent** : if $\{v_1, \dots, v_{n'}\}$ is a basis of V and $\{\theta_1, \dots, \theta_n\}$ is a basis of \mathbb{F}_{2^n} over \mathbb{F}_2 , define **binary variables** x_{ij} such that $\mathbf{x}_i = \sum_j x_{ij} \mathbf{v}_j$ substitute in f and “**reduce modulo** $\mathbf{x}_{ij}^2 - \mathbf{x}_{ij} = \mathbf{0}$ ” **decompose in the basis** $\{\theta_1, \dots, \theta_n\}$

$$\begin{aligned} 0 &= f(x_1, \dots, x_m) = f\left(\sum_{j=1}^{n'} x_{1j} v_j, \dots, \sum_{j=1}^{n'} x_{mj} v_j\right) \\ &= [f]_1^\downarrow \theta_1 + \dots + [f]_n^\downarrow \theta_n \end{aligned}$$

- ▶ We get n equations $[f]_k^\downarrow = 0$ in mn' variables x_{ij}



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 $2^t - 1$ but 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 ($t = 1$)



Degrees and block structure

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

$$\begin{aligned}x_i^e &= \left(\sum x_{ij} v_j\right)^{e_0} \left(\sum x_{ij}^2 v_j^2\right)^{e_1} \dots \left(\sum x_{ij}^{2^{t-1}} v_j^{2^{t-1}}\right)^{e_{t-1}} \\ &= \left(\sum x_{ij} v_j\right)^{e_0} \left(\sum x_{ij} v_j^2\right)^{e_1} \dots \left(\sum x_{ij} v_j^{2^{t-1}}\right)^{e_{t-1}}\end{aligned}$$

degree = Hamming weight of (e_0, \dots, e_{t-1})



Degrees and block structure

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

$$\begin{aligned}x_i^e &= \left(\sum x_{ij} v_j\right)^{e_0} \left(\sum x_{ij}^2 v_j^2\right)^{e_1} \dots \left(\sum x_{ij}^{2^{t-1}} v_j^{2^{t-1}}\right)^{e_{t-1}} \\ &= \left(\sum x_{ij} v_j\right)^{e_0} \left(\sum x_{ij} v_j^2\right)^{e_1} \dots \left(\sum x_{ij} v_j^{2^{t-1}}\right)^{e_{t-1}}\end{aligned}$$

degree = Hamming weight of (e_0, \dots, e_{t-1})

- ▶ $f(x_1, \dots, x_m) = [f]_1^\downarrow \theta_1 + \dots + [f]_n^\downarrow \theta_n$

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* $X_i := \{x_{i1}, \dots, x_{i,n'}\}$



New equations ?

- ▶ Already n equations in mn' variables x_{ij} , given by

$$0 = f(x_1, \dots, x_m) = [f]_1^\downarrow \theta_1 + \dots + [f]_n^\downarrow \theta_n$$

- ▶ Adding new (low degree) equations may accelerate the resolution



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
 $\Rightarrow [f]_i^\downarrow$ and $[f^2]_i^\downarrow$ 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
 $\Rightarrow [f]_i^\downarrow$ and $[f^2]_i^\downarrow$ have the same degrees
- ▶ But

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



Frobenius transforms are useless

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

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



Frobenius transforms are useless

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

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



Frobenius transforms are useless

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

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

same equations! (linear combinations)



New equations

▶ $0 = f \Rightarrow 0 = x_1 f$



New equations

- ▶ $0 = f \Rightarrow 0 = x_1 f$
 $0 = x_1 f(x_1, \dots, x_m) = [x_1 f]_1^\downarrow \theta_1 + \dots + [x_1 f]_n^\downarrow \theta_n$



New equations

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



New equations

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



New equations

- ▶ $0 = f \Rightarrow 0 = x_1 f$
 $0 = x_1 f(x_1, \dots, x_m) = [x_1 f]_1^\downarrow \theta_1 + \dots + [x_1 f]_n^\downarrow \theta_n$
 - ▶ $x_1 f$ has degree $\leq (2^t)$ in x_1 and $\leq (2^t - 1)$ in x_2, \dots, x_m
 - ▶ $[x_1 f]_k^\downarrow$ has degree at most t in each block X_i
 - ▶ Not the same equations!
In particular, all terms have degree ≥ 1 in block X_1
 $f(x_1, \dots, x_m) = f_0(x_2, \dots, x_m) + x_1 f_1(x_1, x_2, \dots, x_m)$
 $\Rightarrow x_1 f(x_1, \dots, x_m) = x_1 f_0(x_2, \dots, x_m) + x_1^2 f_1(x_1, x_2, \dots, x_m)$
 - ▶ Similar equations with other monomials instead of x_1
- Many new equations**



New equations, revisited

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

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



New equations, revisited

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

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

- ▶ Hence

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

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



New equations, revisited

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

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

- ▶ Hence

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

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

- ▶ The “new” equations $[x_1 f]_k^\downarrow = 0$ are algebraic combinations of the original ones $[f]_j^\downarrow = 0$



New equations, revisited

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

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

- ▶ Hence

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

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

- ▶ The “new” equations $[x_1 f]_k^\downarrow = 0$ are algebraic combinations of the original ones $[f]_j^\downarrow = 0$
- ▶ Will be recovered “blindly” by GB algorithms



First fall degree

- ▶ We have

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

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



First fall degree

- ▶ We have

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

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

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



First fall degree

- ▶ We have

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

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

- ▶ 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 $D_{\text{reg}} \approx D_{\text{ff}}$**
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)



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 $D_{reg} \approx mt + 1$

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



Experimental evidence that $D_{reg} \approx mt + 1$

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



Complexity analysis

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

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

$\omega \leq 3$: linear algebra constant



Complexity analysis

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

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

$\omega \leq 3$: 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
- ▶ Assumption must be adapted (and checked) in particular cases



Remarks

- ▶ Heuristic assumption
- ▶ Assumption must be adapted (and checked) in particular cases
- ▶ Similar analysis for other “small characteristic” fields

$$D_{reg} \approx (p - 1)mt + 1$$



Outline

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Application to ECDLP

Further applications



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 $h \in G$

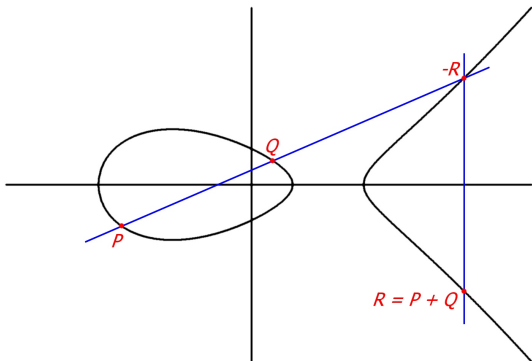
Find $k \in \mathbb{Z}$ such that $g^k = h$

- ▶ 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



Elliptic curves

- ▶ For binary fields : $y^2 + xy = x^3 + a_2x^2 + a_6$ with $a_6 \neq 0$
- ▶ Group structure given by chord and tangent rule



Elliptic curve discrete logarithm problem

► **Elliptic curve discrete logarithm problem (ECDLP) over binary curves :**

Given E over \mathbb{F}_{2^n} ,

Given $P \in E(\mathbb{F}_{2^n})$, given $Q \in \langle P \rangle$,

Find $k \in \mathbb{Z}$ such that $Q = kP$.



Elliptic curve discrete logarithm problem

- ▶ **Elliptic curve discrete logarithm problem (ECDLP) over binary curves :**

Given E over \mathbb{F}_{2^n} ,

Given $P \in E(\mathbb{F}_{2^n})$, given $Q \in \langle P \rangle$,

Find $k \in \mathbb{Z}$ such that $Q = kP$.

- ▶ Includes 10/15 curves standardized by NIST (FIPS 186-3)
- ▶ Complexity thought to be exponential in n

We argue it is

$$\leq 2^{2n^{2/3} \log n}$$



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

$$aP + bQ = 0$$



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

$$aP + bQ = 0$$

- ▶ Need “efficient” algorithm to find relations
Choose $|\mathcal{F}|$ to balance sieving 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 $a, b \in \{1, \dots, 2^n - 1\}$
 - ▶ 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”**

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

- ▶ **Relation search**

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



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 $a, b \in \{1, \dots, 2^n - 1\}$

- ▶ 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

$$\exp(\log^{1/3} |K| \log^{2/3} \log |K|)$$



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-DeMarras-Huang [ADH94], Gaudry [G00],
Gaudry-Thomé-Thériault-Diem [GTDD07]
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

$$\exp(\log^{1/3} |K| \log^{2/3} \log |K|)$$

- ▶ **Hyperelliptic curves** :
Adleman-DeMarras-Huang [ADH94], Gaudry [G00],
Gaudry-Thomé-Thériault-Diem [GTDD07]
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
- ▶ $S_r(x_1, \dots, x_r) = 0$
 $\Leftrightarrow \exists (x_i, y_i) \in E \quad \text{s.t.} \quad (x_1, y_1) + \dots + (x_r, y_r) = O$



Summation polynomials [S04]

- ▶ Relate the x -coordinates of points that sum to O
- ▶ $S_r(x_1, \dots, x_r) = 0$
 $\Leftrightarrow \exists (x_i, y_i) \in E \quad \text{s.t.} \quad (x_1, y_1) + \dots + (x_r, y_r) = O$
- ▶ Recursive formulae :
 $S_2(x_1, x_2) = x_1 - x_2$
 $S_3(x_1, x_2, x_3) = \dots \quad (\text{depends on } E)$
 $S_r(x_1, \dots, x_r) =$
 $\text{Res}_X (S_{r-k}(x_1, \dots, x_{m-k-1}, X), S_{k+2}(x_{r-k}, \dots, x_r, X))$



Summation polynomials [S04]

- ▶ Relate the x -coordinates of points that sum to O
- ▶ $S_r(x_1, \dots, x_r) = 0$
 $\Leftrightarrow \exists (x_i, y_i) \in E \quad \text{s.t.} \quad (x_1, y_1) + \dots + (x_r, y_r) = O$
- ▶ Recursive formulae :
 $S_2(x_1, x_2) = x_1 - x_2$
 $S_3(x_1, x_2, x_3) = \dots \quad (\text{depends on } E)$
 $S_r(x_1, \dots, x_r) =$
 $\text{Res}_X (S_{r-k}(x_1, \dots, x_{m-k-1}, X), S_{k+2}(x_{r-k}, \dots, x_r, X))$
- ▶ 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 $\mathcal{F}_V := \{(x, y) \in E \mid \mathbf{x} \in \mathbf{V}\}$ where $V \subset K$
 - ▶ **Relation search** : for each relation,
Compute $(X_i, Y_i) := a_i P + b_i Q$ for random a_i, b_i
Find $\mathbf{x}_j \in \mathbf{V}$ with $\mathbf{S}_{m+1}(\mathbf{x}_1, \dots, \mathbf{x}_m, \mathbf{X}_i) = \mathbf{0}$
Find the corresponding y_j



Semaev's variant of index calculus

- ▶ Semaev's variant of index calculus :
 - ▶ **Factor basis** :
define $\mathcal{F}_V := \{(x, y) \in E \mid \mathbf{x} \in \mathbf{V}\}$ where $V \subset K$
 - ▶ **Relation search** : for each relation,
Compute $(X_i, Y_i) := a_i P + b_i Q$ for random a_i, b_i
Find $\mathbf{x}_j \in \mathbf{V}$ with $\mathbf{S}_{m+1}(\mathbf{x}_1, \dots, \mathbf{x}_m, \mathbf{x}_j) = \mathbf{0}$
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 $\mathcal{F}_V := \{(x, y) \in E \mid \mathbf{x} \in \mathbf{V}\}$ where $V \subset K$
 - ▶ **Relation search** : for each relation,
Compute $(X_i, Y_i) := a_i P + b_i Q$ for random a_i, b_i
Find $\mathbf{x}_j \in \mathbf{V}$ with $\mathbf{S}_{m+1}(\mathbf{x}_1, \dots, \mathbf{x}_m, \mathbf{x}_j) = \mathbf{0}$
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**



Focus on composite fields

- ▶ Finding relations amounts to
Finding $\mathbf{x}_j \in \mathbb{F}_q$ with $\mathbf{S}_{n+1}(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{X}_i) = \mathbf{0}$



Focus on composite fields

- ▶ Finding relations amounts to
Finding $\mathbf{x}_j \in \mathbb{F}_q$ with $\mathbf{S}_{n+1}(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{X}_i) = \mathbf{0}$
- ▶ 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



Focus on composite fields

- ▶ Finding relations amounts to
Finding $\mathbf{x}_j \in \mathbb{F}_q$ with $S_{n+1}(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{X}_i) = 0$
- ▶ 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

- ▶ System harder to solve for larger n
Attack does not work for \mathbb{F}_{2^n} 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 $\mathcal{F}_V := \{(x, y) \in E \mid x \in V\}$



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 $\mathcal{F}_V := \{(x, y) \in E \mid x \in V\}$

- ▶ **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 $S_{m+1}(x_1, \dots, x_m, X_i) = 0$
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 $\mathcal{F}_V := \{(x, y) \in E \mid x \in V\}$

- ▶ **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 $S_{m+1}(x_1, \dots, x_m, X_i) = 0$

Find the corresponding y_j

- ▶ **Linear algebra** between the relations



Finding relations : Weil descent

- ▶ Find $x_j \in V$ with $S_{m+1}(x_1, \dots, x_m, X_j) = 0$



Finding relations : Weil descent

- ▶ Find $x_j \in V$ with $S_{m+1}(x_1, \dots, x_m, X_i) = 0$
- ▶ 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 $\Rightarrow t = m$



Finding relations : Weil descent

- ▶ Find $x_j \in V$ with $S_{m+1}(x_1, \dots, x_m, X_j) = 0$
- ▶ 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 polynomial 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

$$S_{m+1}(x_1, \dots, x_m, X_i) = 0,$$

the resulting system satisfies $\mathbf{D}_{\text{reg}} \approx \mathbf{D}_{\text{ff}}$



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

n	n'	m	t	$mt + 1 (\geq D_{\text{ff}})$	D_{av}	Time	Mem.
11	6	2	2	5	3.0	0	11
11	4	3	3	10	7.1	1	15
17	9	2	2	5	4.0	0	16
17	6	3	3	10	7.1	130	2136

D_{reg} even *lower* than expected



Experimental verification $D_{\text{reg}} \approx D_{\text{ff}}$

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

n	n'	m	t	$mt + 1 (\geq D_{\text{ff}})$	D_{av}	Time	Mem.
11	6	2	2	5	3.0	0	11
11	4	3	3	10	7.1	1	15
17	9	2	2	5	4.0	0	15
17	6	3	3	10	7.2	132	2133

D_{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



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

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



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 $\alpha := 2/3$

$$t_1 \approx n^{2/3},$$

$$t_2 \approx (1/3)n^{1/3} \log n + n^{2/3} + (2/3)\omega n^{2/3} \log n,$$

$$t_3 \approx (4/3) \log n + \omega' n^{2/3}$$

- ▶ Overall complexity

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



Outline

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Application to ECDLP

Further applications



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 $2^t - 1$ but 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 ($t = 1$)



HFE cryptosystem

- ▶ 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



HFE cryptosystem

- ▶ 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



HFE cryptosystem

- ▶ 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

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

Weil descent on f leads to a *quadratic* system



HFE as a particular case

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

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



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 $m = 1$, $t = \lceil \log_2 D \rceil$, $V = \mathbb{F}_{2^n}$

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

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,...]



Index calculus in $\mathbb{F}_{2^n}^$*

► **Discrete logarithm problem :**

Given a generator $g \in \mathbb{F}_{2^n}^*$,

Given h an element of $\mathbb{F}_{2^n}^*$,

Find $k \in \mathbb{Z}$ such that $h = g^k$



Index calculus in $\mathbb{F}_{2^n}^*$

- ▶ **Discrete logarithm problem :**

Given a generator $g \in \mathbb{F}_{2^n}^*$,

Given h an element of $\mathbb{F}_{2^n}^*$,

Find $k \in \mathbb{Z}$ such that $h = g^k$

- ▶ **Index calculus**

- ▶ **Factor basis :**

a vector subspace $V \subset \mathbb{F}_{2^n}$, $\dim(V) = n'$

- ▶ **Relation search :** for each relation,

Compute $r_i := g^{a_i} h^{b_i}$ for random a_i, b_i

Find $x_j \in V$ with $\prod_{j=1}^m x_j = r_i$

- ▶ **Linear algebra** on the relations



Link with our analysis

- ▶ For each relation, find $x_j \in V$ with $\prod_{i=1}^m x_j = r_i$



Link with our analysis

- ▶ For each relation, find $x_j \in V$ with $\prod_{i=1}^m x_j = r_i$
- ▶ Weil descent \rightarrow polynomial system
 n equations, mn' variables, *multilinear* case ($t = 1$)



Link with our analysis

- ▶ For each relation, find $x_j \in V$ with $\prod_{i=1}^m x_j = r_i$
- ▶ Weil descent \rightarrow polynomial system
 n equations, mn' variables, *multilinear* case ($t = 1$)
- ▶ Total time 2^T where

$$T \approx cn^{1/2} \log n \quad \text{and} \quad c = \frac{\omega + 1}{2} \leq 2$$



Link with our analysis

- ▶ For each relation, find $x_j \in V$ with $\prod_{i=1}^m x_j = r_i$
- ▶ Weil descent \rightarrow polynomial system
 n equations, mn' variables, *multilinear* case ($t = 1$)
- ▶ Total time 2^T where

$$T \approx cn^{1/2} \log n \quad \text{and} \quad c = \frac{\omega + 1}{2} \leq 2$$

- ▶ Comparison with Coppersmith's algorithm [C84]
 - ▶ Other heuristic assumption
 - ▶ Coppersmith much faster : $2^{n^{1/3} \log^{2/3} n}$Ours is similar to Adleman's first index calculus [A79]
... Improvements?



Outline

Algebraic cryptanalysis

Polynomial systems arising from a Weil descent

Application to ECDLP

Further applications



Conclusion

- ▶ 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

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

- ▶ Polynomial systems arising from a Weil descent
 - ▶ Very important class of systems for cryptography
 - ▶ ECDLP, HFE, DLP, factoring in $SL(2, \mathbb{F}_{2^n})$, ...
- ▶ 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
- ▶ Future work
 - ▶ Better algorithms, remove heuristic assumptions
 - ▶ Extension to prime fields?



References

- ▶ JC Faugère, L Perret, C Petit, G Renault, *Improving the complexity of index calculus for elliptic curves over binary fields.*
- ▶ C Petit, JJ Quisquater. *On polynomial systems arising from a Weil descent.*



References

- ▶ [B65] B Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.*
- ▶ [L83] D Lazard. *Gröbner-bases, Gaussian elimination and resolution of systems of algebraic equations.*
- ▶ [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).*
- ▶ [FGLM93] JC Faugère, P Gianni, D Lazard, T Mora. *Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering*



References

- ▶ [BFS04] M Bardet, JC Faugère, B Salvy. *On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations.*
- ▶ [BFS05] M Bardet, JC Faugère, B Salvy. *Asymptotic Expansion of the Degree of Regularity for Semi-Regular Systems of Equations.*
- ▶ [DG10] V Dubois and N Gama. *The Degree of Regularity of HFE Systems.*
- ▶ [DH11] J Ding and T Hodges. *Inverting HFE Systems Is Quasi-Polynomial for All Fields.*
- ▶ [B70] E Berlekamp. *Factoring polynomials over large finite fields.*



References

- ▶ [A79] L Adleman. *A Subexponential Algorithm for the Discrete Logarithm Problem with Applications to Cryptography.*
- ▶ [C84] D Coppersmith. *Fast evaluation of logarithms in fields of characteristic two.*
- ▶ [A94] L Adleman. *The function field sieve.*
- ▶ [AH99] L Adleman and MD Huang. *Function Field Sieve Method for Discrete Logarithms over Finite Fields.*
- ▶ [ADH94] L Adleman, J DeMarrais, MD Huang. *A subexponential algorithm for discrete logarithms over the rational subgroup of the jacobians of large genus hyperelliptic curves over finite fields.*



References

- ▶ [G00] P Gaudry. *An algorithm for solving the discrete log problem on hyperelliptic curves.*
- ▶ [G07] P Gaudry, E Thomé, N Thériault, C Diem. *A double large prime variation for small genus hyperelliptic index calculus.*
- ▶ [S04] I Semaev. *Summation polynomials and the discrete logarithm problem on elliptic curves.*
- ▶ [G09] P Gaudry. *Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem.*
- ▶ [D11] C Diem. *On the discrete logarithm problem in elliptic curves.*



References

- ▶ [D11b] C Diem. *On the discrete logarithm problem in elliptic curves (II)*.
- ▶ [P11] C Petit. *Towards factoring in $SL(2, \mathbb{F}_{2^n})$* .
- ▶ [FPPR11] JC Faugère, C Petit, L Perret, G Renault, *New subexponential algorithms for factoring in $SL(2, \mathbb{F}_{2^n})$*
- ▶ [B92] L Babai, A Seress, *On the diameter of permutation groups*
- ▶ [H08] H Helfgott, *Growth and generation in $SL_2(\mathbb{Z}/p\mathbb{Z})$*
- ▶ [Z91] G Zémor. *Hash functions and Cayley graphs*
- ▶ [TZ94] JP Tillich, G Zémor. *Group-theoretic hash functions*.



References

- ▶ [P09] C Petit. *On graph-based cryptographic hash functions.*
- ▶ [P96] J Patarin. *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP) : Two New Families of Asymmetric Algorithms.*
- ▶ [KS99] A Kipnis, A Shamir. *Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization.*
- ▶ [FJ03] JC Faugère and A Joux. *Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases.*
- ▶ [GJS06] L Granboulan and A Joux and J Stern. *Inverting HFE Is Quasipolynomial.*

