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 background

Polynomial systems arising from a Weil descent

Applications to HFE and ECDLP

Outline

Algebraic cryptanalysis background

Polynomial systems arising from a Weil descent

Applications to HFE and ECDLP

Polynomial systems

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

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

Polynomial systems

Problem : *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 with Gaussian elimination
- What about polynomial systems ?

Reduction to linear systems

- Create **all products**

$$g_{i,j} = t_j f_i$$

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

Reduction to linear systems

- Create **all products**

$$g_{i,j} = t_j f_i$$

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

- **Write all coefficients in a matrix**,
  each row corresponding to one polynomial $g_{i,j}$,
  each column to one monomial term of the polynomials

Reduction to linear systems

- Create **all products**

$$g_{i,j} = t_j f_i$$

  where t_j is a *monomial* and $\deg(g_{i,j}) \leq d$
- **Write all coefficients in a matrix**,
  each row corresponding to one polynomial $g_{i,j}$,
  each column to one monomial term of the polynomials
- **If d is large enough**, **linear algebra** on the rows
  gives new polynomials with **lower degrees**

Reduction to linear systems

- Create **all products**

$$g_{i,j} = t_j f_i$$

  where t_j is a *monomial* and $\deg(g_{i,j}) \leq d$
- **Write all coefficients in a matrix**,
  each row corresponding to one polynomial $g_{i,j}$,
  each column to one monomial term of the polynomials
- **If d is large enough**, **linear algebra** on the rows
  gives new polynomials with **lower degrees**
- Eventually gives linear polynomials when solution unique

Gröbner basis algorithms

- Gröbner basis algorithms like F4 or F5 succesively **increase** d **until linearization is possible**

Gröbner basis algorithms

- Gröbner basis algorithms like F4 or F5 succesively **increase** d **until linearization is possible**
- Cost $\approx$ linear algebra on the largest matrix

Gröbner basis algorithms

- Gröbner basis algorithms like F4 or F5 succesively **increase** d **until linearization is possible**
- Cost $\approx$ linear algebra on the largest matrix
- Matrix size at degree d is bounded by n^d

Gröbner basis algorithms

- Gröbner basis algorithms like F4 or F5 succesively **increase** d **until linearization is possible**
- Cost $\approx$ linear algebra on the largest matrix
- Matrix size at degree d is bounded by n^d
- If D_{reg} is the largest degree occuring in the computation, time and memory costs bounded by

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

  $\omega \leq 3$ linear algebra constant

Degree of regularity D_{reg}

- **Largest degree occuring during computation** with grevlex ordering
- Very important complexity parameter

Degree of regularity D_{reg}

- **Largest degree occuring during computation** with grevlex ordering
- Very important complexity parameter
- For "random" systems, $D_{reg} \approx$ sum of degrees

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

Degree of regularity D_{reg}

- **Largest degree occuring during computation** with grevlex ordering
- Very important complexity parameter
- For "random" systems, $D_{reg} \approx$ sum of degrees

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

- Otherwise, D_{reg} usually hard to guess

Degree of regularity D_{reg}

- **Largest degree occuring during computation** with grevlex ordering
- Very important complexity parameter
- For "random" systems, $D_{reg} \approx$ sum of degrees

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

- Otherwise, D_{reg} usually hard to guess
- **Usually much smaller for overdetermined and structured systems**

First fall degree D_{ff}

- Lowest degree d such that there exist *non trivial* $t_i \in R$ with

$$\max \deg(t_i f_i) = d, \qquad \deg\left(\sum t_i f_i\right) < d$$

First fall degree D_{ff}

- Lowest degree d such that there exist *non trivial* $t_i \in R$ with

$$\max \deg(t_i f_i) = d, \qquad \deg\left(\sum t_i f_i\right) < d$$

- Other important complexity parameter

First fall degree D_{ff}

- Lowest degree d such that there exist *non trivial* $t_i \in R$ with

$$\max \deg(t_i f_i) = d, \qquad \deg\left(\sum t_i f_i\right) < d$$

- Other important complexity parameter
- Sometimes called *degree of regularity* in the literature [DG10,DH11]

Degree of regularity vs. first fall degree

- For many classes of systems :

$$\text{degree of regularity } D_{reg} \quad \approx \quad \text{first fall degree } D_{ff}$$

Degree of regularity vs. first fall degree

- For many classes of systems :

  degree of regularity D_{reg} $\approx$ first fall degree D_{ff}

- 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 :

$$\text{degree of regularity } D_{reg} \quad \approx \quad \text{first fall degree } D_{ff}$$

- 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 :

$$\text{degree of regularity } D_{reg} \quad \approx \quad \text{first fall degree } D_{ff}$$

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

Outline

Algebraic cryptanalysis background

Polynomial systems arising from a Weil descent

Applications to HFE and ECDLP

A polynomial problem with linear constraints

Problem *: let* $n, n', m, t \in \mathbb{Z}$*. Let* $f \in \mathbb{F}_{2^n}[x_1, \ldots x_m]$ *with degrees* $\leq 2^t - 1$ *in all variables. Let* V *be a vector subspace of* $\mathbb{F}_{2^n}/\mathbb{F}_2$ *with dimension* n'*.*
Solve $\mathbf{f(x_1, \ldots, x_m) = 0}$ **such that** $\mathbf{x_i \in V}$*.*

A polynomial problem with linear constraints

Problem *: let* $n, n', m, t \in \mathbb{Z}$*. Let* $f \in \mathbb{F}_{2^n}[x_1, \ldots x_m]$ *with degrees* $\leq 2^t - 1$ *in all variables. Let* V *be a vector subspace of* $\mathbb{F}_{2^n}/\mathbb{F}_2$ *with dimension* n'*.*
Solve $\mathbf{f(x_1, \ldots, x_m) = 0}$ **such that** $\mathbf{x_i \in V}$*.*

- If $mn' \approx n$, we expect ≈ 1 solution

A polynomial problem with linear constraints

Problem *: let* $n, n', m, t \in \mathbb{Z}$*. Let* $f \in \mathbb{F}_{2^n}[x_1, \ldots x_m]$ *with degrees* $\leq 2^t - 1$ *in all variables. Let* V *be a vector subspace of* $\mathbb{F}_{2^n}/\mathbb{F}_2$ *with dimension* n'*.*
Solve $\mathbf{f(x_1, \ldots, x_m) = 0}$ **such that** $\mathbf{x_i \in V}$*.*

- If $mn' \approx n$, we expect ≈ 1 solution
- Applying a **Weil descent**, this problem is reduced to a **polynomial system**

Weil descent

- We want to solve $f(x_1, \ldots, x_m) = 0$ such that $x_i \in V$

Weil descent

- We want to solve $f(x_1, \ldots, x_m) = 0$ such that $x_i \in V$
- Let $\{\theta_1, \ldots, \theta_n\}$ be a basis of $\mathbb{F}_{2^n}/\mathbb{F}_2$ and let $\{v_1, \ldots, v_{n'}\}$ be a basis of V

Weil descent

- We want to solve $f(x_1, \ldots, x_m) = 0$ such that $x_i \in V$
- Let $\{\theta_1, \ldots, \theta_n\}$ be a basis of $\mathbb{F}_{2^n}/\mathbb{F}_2$ and let $\{v_1, \ldots, v_{n'}\}$ be a basis of V
- **Define binary variables x_{ij}** such that $x_i = \sum_{j=1}^{n'} x_{ij} v_j$

Weil descent

- We want to solve $f(x_1, \ldots, x_m) = 0$ such that $x_i \in V$
- Let $\{\theta_1, \ldots, \theta_n\}$ be a basis of $\mathbb{F}_{2^n}/\mathbb{F}_2$ and let $\{v_1, \ldots, v_{n'}\}$ be a basis of V
- **Define binary variables x_{ij}** such that $x_i = \sum_{j=1}^{n'} x_{ij} v_j$
- **Substitute x_i in f**

$$f\left(\sum x_{1j} v_j, \ldots, \sum x_{mj} v_j\right) = 0$$

Weil descent

- We want to solve $f(x_1, \ldots, x_m) = 0$ such that $x_i \in V$
- Let $\{\theta_1, \ldots, \theta_n\}$ be a basis of $\mathbb{F}_{2^n}/\mathbb{F}_2$ and let $\{v_1, \ldots, v_{n'}\}$ be a basis of V
- **Define binary variables x_{ij}** such that $x_i = \sum_{j=1}^{n'} x_{ij} v_j$
- **Substitute x_i in f**

$$f\left(\sum x_{1j} v_j, \ldots, \sum x_{mj} v_j\right) = 0$$

- **Decompose f** in the basis $\{\theta_j\}$

$$f = [f]_1^{\downarrow} \theta_1 + \ldots + [f]_n^{\downarrow} \theta_n = 0$$

Weil descent

- We want to solve $f(x_1, \ldots, x_m) = 0$ such that $x_i \in V$
- Let $\{\theta_1, \ldots, \theta_n\}$ be a basis of $\mathbb{F}_{2^n}/\mathbb{F}_2$ and let $\{v_1, \ldots, v_{n'}\}$ be a basis of V
- **Define binary variables x_{ij}** such that $x_i = \sum_{j=1}^{n'} x_{ij} v_j$
- **Substitute x_i in f**

$$f\left(\sum x_{1j} v_j, \ldots, \sum x_{mj} v_j\right) = 0$$

- **Decompose** f in the basis $\{\theta_j\}$

$$f = [f]_1^{\downarrow} \theta_1 + \ldots + [f]_n^{\downarrow} \theta_n = 0$$

- Deduce n equations $[f]_k^{\downarrow} = 0$

A polynomial system

- We have n equations $[f]_i^{\downarrow}$ in mn' variables
- Degrees depend on Hamming weights of exponents in f

A polynomial system

- We have n equations $[f]_i^{\downarrow}$ in mn' variables
- Degrees depend on Hamming weights of exponents in f
- Degree f bounded by $2^t - 1$ in each variable implies that degree $[f]_i^{\downarrow}$ bounded by t in each **block of variables** $X_i := \{x_{ij}, j \in \{1, \ldots, n'\}\}$

A polynomial system

- We have n equations $[f]_i^{\downarrow}$ in mn' variables
- Degrees depend on Hamming weights of exponents in f
- Degree f bounded by $2^t - 1$ in each variable implies that degree $[f]_i^{\downarrow}$ bounded by t in each **block of variables** $X_i := \{x_{ij}, j \in \{1, \ldots, n'\}\}$
- Total degree $[f]_i^{\downarrow}$ bounded by mt

Finding new equations

- Adding equations helps in general

Finding new equations

- Adding equations helps in general
- $f = 0 \Rightarrow f^2 = 0$
  - Add n equations $\left[f^2\right]_j^{\downarrow}$ to the system

Finding new equations

- Adding equations helps in general
- $f = 0 \Rightarrow f^2 = 0$
  - Add n equations $\left[f^2\right]_j^{\downarrow}$ to the system
  - Same degrees

Finding new equations

- Adding equations helps in general
- $f = 0 \Rightarrow f^2 = 0$
  - Add n equations $\left[f^2\right]_j^{\downarrow}$ to the system
  - Same degrees
  - **Useless :** $\left[f^2\right]_j^{\downarrow}$ are in fact **linear combinations** of $[f]_i^{\downarrow}$

Finding new equations

- Adding equations helps in general
- $f = 0 \Rightarrow f^2 = 0$
  - Add n equations $\left[f^2\right]_j^{\downarrow}$ to the system
  - Same degrees
  - **Useless :** $\left[f^2\right]_j^{\downarrow}$ are in fact **linear combinations** of $[f]_i^{\downarrow}$
- $f = 0 \Rightarrow x_1 f = 0$
  - Add n equations $[x_1 f]_j^{\downarrow}$ to the system
  - Same degrees
  - **Useful :** $[x_1 f]_j^{\downarrow}$ are **not linear combinations** of $[f]_i^{\downarrow}$

New equations and First fall degree

- Similar equations for $x_2 f, x_3 f, \ldots, x_1 x_2 f \ldots$
- Only exist for these systems

New equations and First fall degree

- Similar equations for $x_2 f, x_3 f, \ldots, x_1 x_2 f \ldots$
- Only exist for these systems
- In fact, **algebraic combinations** of the original ones

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

- Will be recovered "blindly" by GB algorithms

New equations and First fall degree

- Similar equations for $x_2 f, x_3 f, \ldots, x_1 x_2 f \ldots$
- Only exist for these systems
- In fact, **algebraic combinations** of the original ones

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

- Will be recovered "blindly" by GB algorithms
- $\deg([x_1 f]_k^{\downarrow}) = mt, \quad \deg(p_{ik}) = 1, \quad \deg([f]_j^{\downarrow}) = mt$

New equations and First fall degree

- Similar equations for $x_2 f, x_3 f, \ldots, x_1 x_2 f \ldots$
- Only exist for these systems
- In fact, **algebraic combinations** of the original ones

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

- Will be recovered "blindly" by GB algorithms
- $\deg([x_1 f]_k^{\downarrow}) = mt, \quad \deg(p_{ik}) = 1, \quad \deg([f]_j^{\downarrow}) = mt$
- **First fall degree $\leq$ mt + 1**

Assumption $D_{reg} \approx D_{ff}$

- Implies $\mathbf{D_{reg} \approx mt + 1}$
  instead of $D_{reg} \approx nmt$ for "generic" systems
- D_{reg} essentially as small as it could be
- **Experimentally verified** for "small" parameters

Assumption $D_{reg} \approx D_{ff}$

- Implies $\mathbf{D_{reg} \approx mt + 1}$
  instead of $D_{reg} \approx nmt$ for "generic" systems
- D_{reg} essentially as small as it could be
- **Experimentally verified** for "small" parameters
- Time and memory bounded by

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

Assumption $D_{reg} \approx D_{ff}$

- Implies $\mathbf{D_{reg} \approx mt + 1}$
  instead of $D_{reg} \approx nmt$ for "generic" systems
- D_{reg} essentially as small as it could be
- **Experimentally verified** for "small" parameters
- Time and memory bounded by

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

- Block structure $\Rightarrow$ **time and memory** bounded by

$$(\mathbf{n'})^{\omega \mathbf{D_{reg}}} \quad \text{and} \quad (\mathbf{n'})^{\mathbf{2D_{reg}}}$$

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

Outline

Algebraic cryptanalysis background

Polynomial systems arising from a Weil descent

Applications to HFE and ECDLP

HFE cryptosystem

- HFE = Hidden Field Equation
- Public key cryptosystem proposed by Patarin [P96]

HFE cryptosystem

- HFE = Hidden Field Equation
- Public key cryptosystem proposed by Patarin [P96]
- Private key is a polynomial $f \in \mathbb{F}_{2^n}[x]$

HFE cryptosystem

- HFE = Hidden Field Equation
- 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**
  Linear transformations on the variables and the equations

HFE cryptosystem

- HFE = Hidden Field Equation
- 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**
  Linear transformations on the variables and the equations
- Cryptanalysis leads to **solving the disguised system**

HFE particularities

- "Disguised" by linear transformations, but this has no impact on GB complexity
- Monovariate ($m = 1$)
- $V = \mathbb{F}_{2^n}$

HFE particularities

- "Disguised" by linear transformations, but this has no impact on GB complexity
- Monovariate ($m = 1$)
- $V = \mathbb{F}_{2^n}$
- 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

- $m = 1, t = \lceil \log_2 D \rceil$

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

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

HFE as a particular case

- $m = 1, t = \lceil \log_2 D \rceil$

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

- We recover [KS99,FJ03,GJS06,DG10,DH11,...]
- No impact of HFE special shape
  Other restrictions may have a (positive) impact [DH11]

HFE as a particular case

- $m = 1, t = \lceil \log_2 D \rceil$

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

- We recover [KS99,FJ03,GJS06,DG10,DH11,...]
- No impact of HFE special shape
  Other restrictions may have a (positive) impact [DH11]
- Assumption $D_{reg} \approx D_{ff}$
  widely verified for HFE polynomials [FJ03,GJS06,...]

HFE as a particular case

- $m = 1, t = \lceil \log_2 D \rceil$

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

- We recover [KS99,FJ03,GJS06,DG10,DH11,...]
- No impact of HFE special shape
  Other restrictions may have a (positive) impact [DH11]
- Assumption $D_{reg} \approx D_{ff}$
  widely verified for HFE polynomials [FJ03,GJS06,...]
- Can build a subsystem with less variables
  generalizing [GJS06] (not discussed here)

Elliptic curve discrete logarithm problem

ECDLP over binary curves :

Let E be an elliptic curve defined over a binary field.
Let $P \in E(\mathbb{F}_{2^n})$ and $Q \in < P >$.
Find $k \in \mathbb{Z}$ such that $Q = kP$.

Elliptic curve discrete logarithm problem

ECDLP over binary curves :
Let E be an elliptic curve defined over a binary field.
Let $P \in E(\mathbb{F}_{2^n})$ and $Q \in < P >$.
Find $k \in \mathbb{Z}$ such that $Q = kP$.

- Includes 10/15 curves standardized by NIST (FIPS 186-3)

Elliptic curve discrete logarithm problem

ECDLP over binary curves :
Let E be an elliptic curve defined over a binary field.
Let $P \in E(\mathbb{F}_{2^n})$ and $Q \in < P >$.
Find $k \in \mathbb{Z}$ such that $Q = kP$.

- Includes 10/15 curves standardized by NIST (FIPS 186-3)
- When n prime, complexity thought to be **exponential** in n, in contrast with finite fields and hyperelliptic curves

Elliptic curve discrete logarithm problem

ECDLP over binary curves :
Let E be an elliptic curve defined over a binary field.
Let $P \in E(\mathbb{F}_{2^n})$ and $Q \in < P >$.
Find $k \in \mathbb{Z}$ such that $Q = kP$.

- Includes 10/15 curves standardized by NIST (FIPS 186-3)
- When n prime, complexity thought to be **exponential** in n, in contrast with finite fields and hyperelliptic curves
- Our analysis implies that Diem's variant of index calculus has **subexponential** complexity

Diem's index calculus for ECDLP [*D11b*]

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

Diem's index calculus for ECDLP [D11b]

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

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

Diem's index calculus for ECDLP [D11b]

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

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

$$a_i P + b_i Q + \sum_{j=1}^{m} P_{ij} = O$$

for $P_{ij} \in \mathcal{F}_V$ and random a_i, b_i,

Diem's index calculus for ECDLP [D11b]

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

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

$$a_i P + b_i Q + \sum_{j=1}^{m} P_{ij} = O$$

  for $P_{ij} \in \mathcal{F}_V$ and random a_i, b_i,
- **Linear algebra** to get $aP + bQ = O$

Summation polynomials [S04]

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

Summation polynomials [S04]

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

Summation polynomials [S04]

- Relate the x-coordinates of points that sum to O
- $S_r(x_1, \ldots, x_r) = 0$
  $\Leftrightarrow \quad \exists (x_i, y_i) \in E \quad$ s.t. $\quad (x_1, y_1) + \cdots + (x_r, y_r) = O$
- Recursive formulae :
  $S_2(x_1, x_2) = x_1 - x_2$
  $S_3(x_1, x_2, x_3) = \ldots$ (depends on E)
  $S_r(x_1, \ldots, x_r) =$
  $Res_X\left(S_{r-k}(x_1, \ldots, x_{m-k-1}, X), S_{k+2}(x_{r-k}, \ldots, x_r, X)\right)$
- S_r has degree 2^{r-2} in each variable
  Symmetric set of solutions

Subexponential complexity

- Relation search $\sim$ solving Semaev's polynomials [S04]

$$\textbf{Solve } S_{m+1}(x_1, \ldots, x_m, X) = 0 \textbf{ with } x_j \in V$$

Subexponential complexity

- Relation search $\sim$ solving Semaev's polynomials [S04]

$$\textbf{Solve } S_{m+1}(x_1, \ldots, x_m, X) = 0 \textbf{ with } x_j \in V$$

- Particular instance of our problem with $t = m$

Subexponential complexity

- Relation search $\sim$ solving Semaev's polynomials [S04]

$$\textbf{Solve } S_{m+1}(x_1, \ldots, x_m, X) = 0 \textbf{ with } x_j \in V$$

- Particular instance of our problem with $t = m$
- We experimentally checked **assumption $D_{reg} \approx D_{ff}$** in that case for "small" parameters

Subexponential complexity

- Relation search $\sim$ solving Semaev's polynomials [S04]

$$\textbf{Solve } S_{m+1}(x_1, \ldots, x_m, X) = 0 \textbf{ with } x_j \in V$$

- Particular instance of our problem with $t = m$
- We experimentally checked **assumption $D_{reg} \approx D_{ff}$** in that case for "small" parameters
- Assumption implies **ECDLP subexponential**

$$2^T \quad \text{with} \quad T \approx cn^{2/3} \log n$$

Subexponential complexity

- Relation search $\sim$ solving Semaev's polynomials [S04]

$$\textbf{Solve } S_{m+1}(x_1, \ldots, x_m, X) = 0 \textbf{ with } x_j \in V$$

- Particular instance of our problem with $t = m$
- We experimentally checked **assumption $D_{reg} \approx D_{ff}$** in that case for "small" parameters
- Assumption implies **ECDLP subexponential**

$$2^T \quad \text{with} \quad T \approx cn^{2/3} \log n$$

- **NIST curves remain safe so far**
  Pollard's rho beaten for $n \geq N$, where $N \approx 2000$

Outline

Algebraic cryptanalysis background

Polynomial systems arising from a Weil descent

Applications to HFE and ECDLP

Conclusion

- Polynomial systems arising from a Weil descent
  - Very important class of systems for cryptography
  - HFE, ECDLP, but also DLP, factoring in $SL(2, \mathbb{F}_{2^n})$, ...
  - Extension to any "small" characteristic field

Conclusion

- Polynomial systems arising from a Weil descent
  - Very important class of systems for cryptography
  - HFE, ECDLP, but also DLP, factoring in $SL(2, \mathbb{F}_{2^n})$, ...
  - Extension to any "small" characteristic field
- 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

Conclusion

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

References

- [A79] L Adleman. *A Subexponential Algorithm for the Discrete Logarithm Problem with Applications to Cryptography.*
- [A94] L Adleman. *The function field sieve.*
- [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.*
- [AH99] L Adleman and MD Huang. *Function Field Sieve Method for Discrete Logarithms over Finite Fields.*

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.*
- [B70] E Berlekamp. *Factoring polynomials over large finite fields.*
- [B65] B Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.*

References

- [C84] D Coppersmith. *Fast evaluation of logarithms in fields of characteristic two.*
- [C03] N Courtois. *The Security of Hidden Field Equations (HFE).*
- [D11] C Diem. *On the discrete logarithm problem in elliptic curves.*
- [D11b] C Diem. *On the discrete logarithm problem in elliptic curves (II).*
- [DH11] J Ding and T Hodges. *Inverting HFE Systems Is Quasi-Polynomial for All Fields.*

References

- [DG10] V Dubois and N Gama. *The Degree of Regularity of HFE Systems.*
- [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

- [FJ03] JC Faugère and A Joux. *Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases*.
- [FPPR11] JC Faugère, L Perret, C Petit, G Renault, *New subexponential algorithms for factoring in* $SL(2, \mathbb{F}_{2^n})$.
- [FPPR12] JC Faugère, L Perret, C Petit, G Renault, *Improving the complexity of index calculus for elliptic curves over binary fields.*
- [GS99] S Galbraith, N Smart. *A cryptographic application of the Weil descent.*
- [G00] P Gaudry. *An algorithm for solving the discrete log problem on hyperelliptic curves.*

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 desctructive facets of Weil descent on elliptic curves.*
- [G07] P Gaudry, E Thomé, N Thériault, C Diem. *A double large prime variation for small genus hyperelliptic index calculus.*
- [GJS06] L Granboulan and A Joux and J Stern. *Inverting HFE Is Quasipolynomial*.
- [HPS12] T Hodges, C Petit, J Schlaffer. *First fall degree of polynomials systems arising from a Weil descent*.

References

- [KS99] A Kipnis, A Shamir. *Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization*.
- [L83] D Lazard. *Gröbner-bases, Gaussian elimination and resolution of systems of algebraic equations.*
- [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*
- [P96]J Patarin. *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP) : Two New Families of Asymmetric Algorithms*.

References

- C Petit, JJ Quisquater. *On polynomial systems arising from a Weil descent.*
- [SA21] T Satoh, K Araki. *Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves*
- [S04] I Semaev. *Summation polynomials and the discrete logarithm problem on 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*