A variant of the F4 algorithm

Vanessa VITSE - Antoine JOUX

Université de Versailles Saint-Quentin, Laboratoire PRISM

CT-RSA, February 18, 2011

Motivation

An example of algebraic cryptanalysis

Discrete logarithm problem over elliptic curves (ECDLP)

E elliptic curve over a finite field
Given $P \in E$ and $Q \in \langle P \rangle$, find x such that $Q = [x]P$

Motivation

An example of algebraic cryptanalysis

Discrete logarithm problem over elliptic curves (ECDLP)

E elliptic curve over a finite field

Given $P \in E$ and $Q \in \langle P \rangle$, find x such that $Q = [x]P$

Basic outline of index calculus method for DLP

1. define a factor base: $\mathcal{F} = \{P_1, \ldots, P_N\}$
2. relation search: for random (a_i, b_i), try to decompose $[a_i]P + [b_i]Q$ as sum of points in $\mathcal{F}$
3. linear algebra step: once $k > N$ relations found, deduce with sparse algebra techniques the DL of Q

Motivation

Cryptanalysis of the DLP on $E(\mathbb{F}_{q^n})$

Relation search on $E(\mathbb{F}_{q^n})$ - [Gaudry,Diem]

- Factor base: $\mathcal{F} = \{(x,y) \in E(\mathbb{F}_{q^n}) : x \in \mathbb{F}_q\}$
- Goal: find a least $\#\mathcal{F}$ decompositions of random combinations $R = [a]P + [b]Q$ into m points of $\mathcal{F}$: $R = P_1 + \ldots + P_m$

Algebraic attack

- for each R, construct the corresponding polynomial system $\mathcal{S}_R$
  - Semaev's summation polynomials and symmetrization
  - Weil restriction: write $\mathbb{F}_{q^n}$ as $\mathbb{F}_q[t]/(f(t))$
- $\mathcal{S}_R = \{f_1, \ldots, f_n\} \subset \mathbb{F}_q[X_1, \ldots, X_m]$
  - coefficients depend polynomially on x_R

each decomposition trial $\leftrightarrow$ find the solutions of $\mathcal{S}_R$ over $\mathbb{F}_q$

Polynomial system solving over finite fields

Difficult pb: how to compute $V(I)$ where $I = \langle f_1, ..., f_r \rangle \subset \mathbb{F}_q[X_1, ..., X_m]$?

Gröbner bases: good representations for ideals

- Convenient generators $g_1, \ldots, g_s$ of I capturing the main features of I
- $G \subset I$ is a Gröbner basis of I if $\langle LT(G) \rangle = LT(I)$

Polynomial system solving over finite fields

Difficult pb: how to compute $V(I)$ where $I = \langle f_1, ..., f_r \rangle \subset \mathbb{F}_q[X_1, ..., X_m]$?

Gröbner bases: good representations for ideals

- Convenient generators $g_1, \ldots, g_s$ of I capturing the main features of I
- $G \subset I$ is a Gröbner basis of I if $\langle LT(G) \rangle = LT(I)$

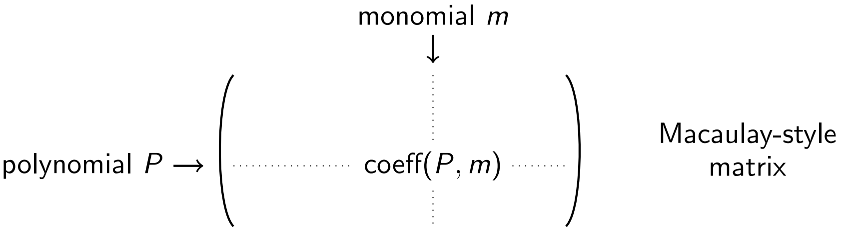
Gröbner basis computation

- Basic operation: computation and reduction of critical pair $S(p_1, p_2) = u_1 p_1 - u_2 p_2$ where $lcm = LM(p_1) \vee LM(p_2)$, $u_i = \frac{lcm}{LM(p_i)}$
- Buchberger's result: to compute a GB of I,
  1. start with $G = \{f_1, \ldots, f_r\}$
  2. iterate basic operation on all possible critical pairs of elements of G, add non-zero remainders to G

Techniques for resolution of polynomial systems

F4: efficient implementation of Buchberger's algorithm

- linear algebra to process several pairs simultaneously
- selection strategy (e.g. lowest total degree lcm)
- at each step construct a Macaulay-style matrix containing
  - products $u_i p_i$ coming from the selected critical pairs
  - polynomials from preprocessing phase



Techniques for resolution of polynomial systems

Standard Gröbner basis algorithms

1. F4 algorithm (Faugère '99)
   - fast and complete reductions of critical pairs
   - drawback: many reductions to zero
2. F5 algorithm (Faugère '02)
   - elaborate criterion → skip unnecessary reductions
   - drawback: incomplete polynomial reductions

- multipurpose algorithms
- do not take advantage of the common shape of the systems
- knowledge of a prior computation

  → no more reduction to zero in F4 ?

Specifically devised algorithms

Outline of our F4 variant

1. `F4Precomp`: on the first system
   - at each step, store the list of all involved polynomial multiples
   - reduction to zero $\rightarrow$ remove well-chosen multiple from the list
2. `F4Remake`: for each subsequent system
   - no queue of untreated pairs
   - at each step, pick directly from the list the relevant multiples

Former works

- Gröbner basis over $\mathbb{Q}$ using CRT and modular computations
- Traverso '88: analysis of *Gröbner trace* for rational Gröbner basis computations with Buchberger's algorithm

Analysis of F4Remake

"Similar" systems

- parametric family of systems: $\{F_1(y), \ldots, F_r(y)\}_{y \in \mathbb{K}^\ell}$ where $F_1, \ldots, F_r \in \mathbb{K}[Y_1, \ldots, Y_\ell][X_1, \ldots, X_n]$
- $\{f_1, \ldots, f_r\} \subset \mathbb{K}[\underline{X}]$ random instance of this parametric family

Generic behaviour

1. "compute" the GB of $\langle F_1, \ldots, F_r \rangle$ in $\mathbb{K}(\underline{Y})[\underline{X}]$ with F4 algorithm
2. $f_1, \ldots, f_r$ behaves generically if during the GB computation with F4
   - same number of iterations
   - at each step, same new leading monomials $\rightarrow$ similar critical pairs

Analysis of F4Remake

"Similar" systems

- parametric family of systems: $\{F_1(y), \ldots, F_r(y)\}_{y \in \mathbb{K}^\ell}$ where $F_1, \ldots, F_r \in \mathbb{K}[Y_1, \ldots, Y_\ell][X_1, \ldots, X_n]$
- $\{f_1, \ldots, f_r\} \subset \mathbb{K}[\underline{X}]$ random instance of this parametric family

Generic behaviour

1. "compute" the GB of $\langle F_1, \ldots, F_r \rangle$ in $\mathbb{K}(\underline{Y})[\underline{X}]$ with F4 algorithm
2. $f_1, \ldots, f_r$ behaves generically if during the GB computation with F4
   - same number of iterations
   - at each step, same new leading monomials $\rightarrow$ similar critical pairs

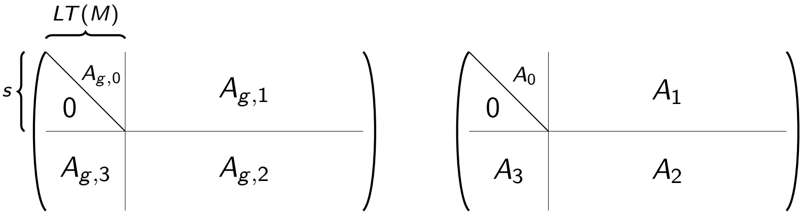
F4Remake computes successfully the GB of $f_1, \ldots, f_r$ if the system behaves generically

Algebraic condition for generic behaviour

1. Assume $f_1, \ldots, f_r$ behaves generically until the $(i-1)$-th step
2. At step i, F4 constructs
   - M_g =matrix of polynomial multiples at step i for the parametric system
   - M =matrix of polynomial multiples at step i for $f_1, \ldots, f_r$

Algebraic condition for generic behaviour

1. Assume $f_1, \dots, f_r$ behaves generically until the $(i-1)$-th step
2. At step i, F4 constructs
   - M_g =matrix of polynomial multiples at step i for the parametric system
   - M =matrix of polynomial multiples at step i for $f_1, \dots, f_r$
3. Reduced row echelon form of M_g and M



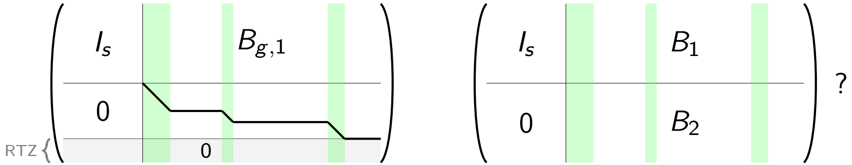
Algebraic condition for generic behaviour

1. Assume $f_1, \ldots, f_r$ behaves generically until the $(i-1)$-th step
2. At step i, F4 constructs
   - M_g =matrix of polynomial multiples at step i for the parametric system
   - M =matrix of polynomial multiples at step i for $f_1, \ldots, f_r$
3. Reduced row echelon form of M_g and M

$$\left(\begin{array}{c|c} I_s & B_{g,1} \\ \hline 0 & B_{g,2} \end{array}\right) \qquad \left(\begin{array}{c|c} I_s & B_1 \\ \hline 0 & B_2 \end{array}\right)$$

Algebraic condition for generic behaviour

1. Assume $f_1, \dots, f_r$ behaves generically until the $(i-1)$-th step
2. At step i, F4 constructs
   - M_g =matrix of polynomial multiples at step i for the parametric system
   - M =matrix of polynomial multiples at step i for $f_1, \dots, f_r$
3. Reduced row echelon form of M_g and M



Algebraic condition for generic behaviour

1. Assume $f_1, \ldots, f_r$ behaves generically until the $(i-1)$-th step
2. At step i, F4 constructs
   - M_g =matrix of polynomial multiples at step i for the parametric system
   - M =matrix of polynomial multiples at step i for $f_1, \ldots, f_r$
3. Reduced row echelon form of M_g and M

$$\begin{pmatrix} I_s & 0 & C_{g,1} \\ 0 & I_\ell & C_{g,2} \\ 0 & 0 & 0 \end{pmatrix} \qquad \begin{pmatrix} I_s & & B'_1 \\ 0 & B & B'_2 \end{pmatrix} ?$$

Algebraic condition for generic behaviour

1. Assume $f_1, \dots, f_r$ behaves generically until the $(i-1)$-th step
2. At step i, F4 constructs
   - M_g =matrix of polynomial multiples at step i for the parametric system
   - M =matrix of polynomial multiples at step i for $f_1, \dots, f_r$
3. Reduced row echelon form of M_g and M

$$\begin{pmatrix} I_s & 0 & C_{g,1} \\ \hline 0 & I_\ell & C_{g,2} \\ \hline 0 & 0 & 0 \end{pmatrix} \qquad \begin{pmatrix} I_s & & B'_1 \\ \hline 0 & B & B'_2 \end{pmatrix}$$

$f_1, \dots, f_r$ behaves generically at step i $\Leftrightarrow$ B has full rank

Probability of success

Heuristic assumption

- The B matrices are uniformly random over $\mathcal{M}_{n,\ell}(\mathbb{F}_q)$
- The probabilities that the B matrices have full rank are independent

Probability estimates over $\mathbb{F}_q$

The probability that a system $f_1, \ldots, f_r$ behaves generically is heuristically greater than $c(q)^{n_{step}}$ where

- n_{step} is the number of steps during the F4 computation of the parametric system $F_1, \ldots, F_r \in \mathbb{K}(\underline{Y})[\underline{X}]$
- $c(q) = \prod_{i=1}^{\infty}(1 - q^{-i}) = 1 - 1/q + \underset{q\to\infty}{O}(1/q^2)$

Application to index calculus method for ECDLP

Joux-V. approach

ECDLP: $P \in E(\mathbb{F}_{q^n}), Q \in \langle P \rangle$, find x such that $Q = [x]P$

- find $\simeq q$ decompositions of random combination $R = [a]P + [b]Q$ into $n-1$ points of $\mathcal{F} = \{P \in E(\mathbb{F}_{q^n}) : x_P \in \mathbb{F}_q\}$
- solve $\simeq q^2$ overdetermined systems of n eq. and $n-1$ var. over $\mathbb{F}_q$
- heuristic assumption makes sense

Experimental results on $E(\mathbb{F}_{p^5})$, p odd (Joux-V.)

- system of 5 eq / 4 var over $\mathbb{F}_p$, total degree 8
- Precomputation done in 8.963 sec, 29 steps, $d_{reg} = 19$

size of p	est. failure proba.	F4Remake[1]	F4[1]	F4/F4Remake	F4 Magma[2]
8 bits	0.11	2.844	5.903	2.1	9.660
16 bits	4.4×10^{-4}	3.990	9.758	2.4	9.870
25 bits	2.4×10^{-6}	4.942	16.77	3.4	118.8
32 bits	5.8×10^{-9}	8.444	24.56	2.9	1046

Step	degree	F4Remake matrix sizes	F4 matrix sizes	ratio
14	17	1062×3072	1597×3207	1.6
15	16	1048×2798	1853×2999	1.9
16	15	992×2462	2001×2711	2.2
17	14	903×2093	2019×2369	2.5
18	13	794×1720	1930×2000	2.8

[1] 2.93 GHz Intel Xeon processor
[2] V2.15-15

Results in characteristic 2

The IPSEC Oakley key determination protocol 'Well Known Group' 3 curve

The Oakley curve: an interesting target

$\mathbb{F}_{2^{155}} = \mathbb{F}_2[u]/_{(u^{155}+u^{62}+1)}$
$E : y^2+xy = x^3+(u^{18}+u^{17}+u^{16}+u^{13}+u^{12}+u^9+u^8+u^7+u^3+u^2+u+1)$
$G = E(\mathbb{F}_{2^{155}})$,
$\#G = 12 * 3805993847215893016155463826195386266397436443$

Remarks

- this curve is known to be theoretically weaker than curves over comparable size prime fields (GHS)
- we show that an actual attack on this curve is feasible.

Attack of Oracle-assisted Static Diffie-Hellman Problem

Granger-Joux-V.

Oracle-assisted SDHP

G finite group and d secret integer

- Initial learning phase: the attacker has access to an oracle which outputs $[d]Y$ for any $Y \in G$
- After a number of oracle queries, the attacker has to compute $[d]X$ for a previously unseen challenge X

Attack of Oracle-assisted Static Diffie-Hellman Problem

Granger-Joux-V.

Oracle-assisted SDHP

G finite group and d secret integer

- Initial learning phase: the attacker has access to an oracle which outputs $[d]Y$ for any $Y \in G$
- After a number of oracle queries, the attacker has to compute $[d]X$ for a previously unseen challenge X

Attack on the Oakley curve

- learning phase: ask the oracle $Q = [d]P$ for each $P \in \mathcal{F}$ where $\mathcal{F} = \{P \in E(\mathbb{F}_{2^{155}}) : P = (x_P, y_P), x_P \in \mathbb{F}_{2^{31}}\}$
- find a decomposition of $[r]X$ (r random) in a sum of 4 points in $\mathcal{F}$ $\leftrightarrow$ solve $\simeq 5.10^{10}$ systems of 5 eq / 4 var over $\mathbb{F}_{2^{31}}$, total deg 8

Results for the 'Well Known Group' 3 Oakley curve

Timings

- Magma (V2.15-15): each decomposition trial takes about 1 sec
- F4Variant + dedicated optimizations of arithmetic and linear algebra
  $\rightarrow$ only 22.95 ms per test on a 2.93 GHz Intel Xeon processor
  $\rightarrow \simeq 400\times$ faster than results in odd characteristic

Feasible attack : oracle-assisted SDHP solvable in $\leq$ 2 weeks with 1000 processors after a learning phase of 2^{30} oracle queries

Limits of the heuristic assumption

Specific case

Parametric polynomials with highest degree homogeneous part in $\mathbb{K}[\underline{X}]$

- heuristic assumption not valid
- but generic behaviour until the first fall of degree occurs

Limits of the heuristic assumption

Specific case

Parametric polynomials with highest degree homogeneous part in $\mathbb{K}[\underline{X}]$

- heuristic assumption not valid
- but generic behaviour until the first fall of degree occurs

Unbalanced Oil and Vinegar scheme

Security based on problem of solving multivariate quadratic systems
Recommended parameters: 16 eq., 32 (or 48) variables over $\mathbb{K} = \mathbb{F}_{2^4}$

$$P_k = \sum_{i,j=1}^{48} a_{ij}^k x_i x_j + \sum_{i=1}^{48} b_i^k x_i + c^k, \quad k = 1 \dots 16$$

Limits of the heuristic assumption

Specific case

Parametric polynomials with highest degree homogeneous part in $\mathbb{K}[\underline{X}]$

- heuristic assumption not valid
- but generic behaviour until the first fall of degree occurs

Unbalanced Oil and Vinegar scheme

Recommended parameters : $m = 16$ eq, $n = 32$ (or 48) var over $\mathbb{K} = \mathbb{F}_{2^4}$

Hybrid approach [Bettale, Faugère, Perret]:

- fix $m - n$ variables and find a solution of the system with 16 eq / var
- exhaustive search over 3 more variables (overdetermined system)

$$P_k = \sum_{i,j=1}^{13} a_{ij}^k x_i x_j + \sum_{i=1}^{13} \left(b_i^k + \sum_{j=14}^{16} a_{ij}^k x_j \right) x_i + \left(\sum_{i,j=14}^{16} a_{ij}^k x_i x_j + \sum_{i=14}^{16} b_i^k x_i + c^k \right)$$

UOV and Hybrid approach example

Goal : compute GB of systems $S_{x_{14},x_{15},x_{16}} = \{P_1, \ldots, P_{16}\}$ for all $(x_{14}, x_{15}, x_{16}) \in \mathbb{F}_{2^4}^3$ where

$$P_k = \sum_{i,j=1}^{13} a_{ij}^k x_i x_j + \sum_{i=1}^{13} \left(b_i^k + \sum_{j=14}^{16} a_{ij}^k x_j \right) x_i + \left(\sum_{i,j=14}^{16} a_{ij}^k x_i x_j + \sum_{i=14}^{16} b_i^k x_i + c^k \right)$$

Resolution with F4Remake

- 6 steps, first fall of degree observed at step 5

$$\text{Proba}(S_{x_{14},x_{15},x_{16}} \text{ behaves generically}) \geq c(16)^2 \simeq 0.87$$

- exhaustive search: the probability observed on different examples is about 90%

UOV and Hybrid approach example

	F4Remake[1]	F4[1]	F4 Magma[2]	F4/F4Remake
Timing (sec)	5.04	16.77	120.6	3.3
Largest matrix	5913×7005	10022×8329	10245×8552	2.0

- precomputation done in 32.3 sec
- to be compared to the 9.41 sec of F5[3] mentioned by Faugère et al.
- generically the GB is $\langle 1 \rangle$
  $\rightarrow$ solutions to be found among the non generic systems

[1] 2.6 GHz Intel Core 2 duo
[2] V2.16-12
[3] 2.4 GHz Bi-pro Xeon

A variant of the F4 algorithm

Vanessa VITSE - Antoine JOUX

Université de Versailles Saint-Quentin, Laboratoire PRISM

CT-RSA, February 18, 2011

Addendum: What about non genericity?

1. When the precomputation is correct:
   - correctness of `F4Remake` easy to detect: non generic behaviour as soon as we encounter a reduction to zero or a polynomial with smaller LT than excepted
   - when `F4Remake` fails, continue the computation with classical `F4`

2. The precomputation is incorrect if:
   - `F4Remake` produces a leading monomial greater than the one obtained by `F4Precomp` during the same step
   - other possibility: execute `F4Precomp` on several systems and compare the lists of leading monomials

Addendum: Comparison with F5

Common features:

- elimination of the reductions to zero
- same upper bound for the theoretical complexity:

$$\tilde{O}\left(\binom{d_{reg}+n}{n}^{\omega}\right)$$

In practice, for the system on $E(\mathbb{F}_{p^5})$:

- F5 generates many redundant polynomials (F5 criterion) : 17249 polynomials in the GB before minimization
- F4 creates only 2789 polynomials
  $\rightarrow$ better behavior, independent of the implementation