Cryptanalytic world records, summer 2014

World record! Discrete logarithms in GF(2^{9234})

Dr. Daniel Loebenberger

Recall: Index calculus

We have a generator g for the multiplicative group $\mathbb{Z}_p^\times = \langle g \rangle$ of units modulo p, of order $d = p - 1$, and want to compute $\operatorname{dlog}_g x$ for some given $x \in \mathbb{Z}_p^\times$. We choose a *factor base* $\{p_1, \ldots, p_h\}$ consisting of the primes up to some bound $B = p_h$.

In a preprocessing step, which does not depend on x, we choose random exponents $e \xleftarrow{⚄⚅} \mathbb{Z}_{p-1}$ and check if g^e in $\mathbb{Z}_p^\times = \{1, \ldots, p-1\}$ is B-smooth. If it is, we find nonnegative integers $\alpha_1, \ldots, \alpha_h$ with

$$\begin{aligned} g^e &= p_1^{\alpha_1} \cdots p_h^{\alpha_h} \text{ in } \mathbb{Z}_p^\times, \\ e &= \alpha_1 \operatorname{dlog}_g p_1 + \cdots + \alpha_h \operatorname{dlog}_g p_h \text{ in } \mathbb{Z}_{p-1}. \end{aligned} \qquad (\star)$$

We collect enough of such *relations* until we can solve these linear equations in $\mathbb{Z}_{p-1}$ for the $\operatorname{dlog}_g p_i$. Typically, a little more than h relations $(\star)$ will be enough, say $h + 10$.

Recall: Index calculus

To solve the system of linear equations, we may factor $p-1$, solve modulo each prime power factor of $p-1$, and piece the solutions together via the Chinese Remainder Theorem.

At this point, we know $\mathsf{dlog}_g\, p_1, \ldots, \mathsf{dlog}_g\, p_h$. Now on input x, we choose random exponents e until some xg^e in $\mathbb{Z}_p$ is B-smooth, say $xg^e = p_1^{\beta_1} \cdots p_h^{\beta_h}$ in $\mathbb{Z}_p$. Then

$$\mathsf{dlog}_g\, x \quad = \quad -e + \beta_1\, \mathsf{dlog}_g\, p_1 + \cdots + \beta_h\, \mathsf{dlog}_g\, p_h \text{ in } \mathbb{Z}_{p-1}.$$

B	h	average # of relations for unique solution	average # of attempts to find 1 rel.	average # of attempts until unique solution	expected # of attempts to find 1 rel. (exact)	expected # of attempts to find 1 rel. (approx.)
5	3	5.41	3.97	21.47	3.77	38.3
7	4	7.27	2.74	19.9	2.70	12.4
11	5	10.06	2.31	23.2	2.22	4.96
13	6	12.19	1.92	23.4	1.91	3.91
17	7	16.82	1.74	29.23	1.72	2.87
19	8	22.95	1.62	37.2	1.59	2.58

Finding suitable relations is one important bottle-neck in this approach!

There are finite fields in which we can speed-up the process of finding relations considerably!

Definition

A finite field K admits a *sparse medium subfield representation* if

- K is isomorphic to $\mathbb{F}_{q^{2k}}$ for some $k \geq 1$.
- there are two polynomials h_0 and h_1 over $\mathbb{F}_{q^2}$ of small degree, such that $h_1 X^q - h_0$ has a degree k irreducible factor.

Think of these fields for the moment as "nice" in some suitable sense :)

Theorem

Let $K = \mathbb{F}_{q^{2k}}$ be a “nice” finite field. Under certain heuristics, there exists an algorithm whose complexity is polynomial in q and k, which can be used for the following two tasks:

1. Given an element of K, represented as a polynomial $P \in \mathbb{F}_{q^2}[X]$ with $2 \leq \deg P < k$, find a representation of $\log P(X)$ as a linear combination of at most $O(kq^2)$ logarithms $\log P_i(X)$ with $\deg P_i \leq \lceil \frac{1}{2} \deg P \rceil$ and of $\log h_1(X)$.
2. Find the logarithm of $h_1(X)$ and the logarithm of all elements of K that are represented by linear polynomials $X + a$ for $a \in \mathbb{F}_{q^2}$.

Theorem

Let $K = \mathbb{F}_{q^{2k}}$ be a “nice” finite field. Under certain heuristics, any discrete logarithm in K can be computed in time bounded by $\max(q, k)^{O(\log k)}$.

Corollary

For finite fields of size $Q = q^{2k}$ with $q \approx k$, there exists a heuristic quasi-polynomial time algorithm for computing discrete logarithms, which runs in time $2^{O((\log\log Q)^2)}$.

Corollary

For finite fields of size Q and characteristic bounded by $(\log Q)^{O(1)}$, there exists a heuristic quasi-polynomial time algorithm for computing discrete logarithms, which runs in time $2^{O((\log\log Q)^2)}$.

Corollary

For finite fields of size $Q = q^{2k}$ with $q \leq L_Q(\alpha)$, where $L_Q(\alpha) = \exp(\mathrm{O}((\log Q)^{\alpha}(\log\log Q)^{1-\alpha}))$, there exists a heuristic sub-exponential time algorithm for computing discrete logarithms, which runs in time $L_Q(\alpha)^{\mathrm{O}(\log\log Q)}$.

Tool: Projective geometry

Let k be any field. Think of it as the *affine line*. We want to define what is called the projective line $\mathbb{P}^1(k)$. The idea is to embed k in k^2 with second coordinate equal to 1:

$$\begin{array}{ccc} k & \longrightarrow & k^2, \\ a & \longmapsto & (a, 1) \end{array}.$$

Next, a point a in the affine line k corresponds to the point $(a, 1)$ which in turn defines and is given by a line through the origin of k^2 and this point $(a, 1)$. Observe that one line does not correspond to a point, actually exactly the one that is parallel to the line $b = 1$. Now, the *projective line* $\mathbb{P}^1(k)$ is the set of all pairs $a : b$, where not both a and b are zero. Two such pairs $a_1 : b_1$ and $a_2 : b_2$ are equal if there is a nonzero constant $\alpha \in k$ with $a_1 = \alpha a_2$ and $b_1 = \alpha b_2$.

Goal:

Given any polynomial $P \in F_{q^2}[X]$ of degree $1 \leq D < k$. Find a relation between $P(X)$ and its translates.

Use the systematic equation

$$X^q - X = \prod_{a \in F_q} (X - a).$$

Choose a set $S = \{(\alpha, \beta)\}$ of representatives of the $q+1$ points $(\alpha : \beta) \in \mathbb{P}^1(\mathbb{F}_q)$, such that the following systematic projective equation holds:

$$X^q Y - X Y^q = \prod_{(\alpha,\beta) \in S} (\beta X - \alpha Y).$$

Consider the transformations $m = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathbb{F}_{q^2}$ and define $m \cdot P = \frac{aP+b}{cP+d}$. Substitute $X = aP + b$ and $Y = cP + d$ in the above equation.

We obtain

$$(aP+b)^q(cP+d)-(aP+b)(cP+d)^q = \lambda \prod_{(\alpha,\beta)\in S} P-\mathsf{x}(m^{-1}\cdot(\alpha:\beta))$$

for some suitable constant $\lambda \in \mathbb{F}_{q^2}$ and

$$P - \mathsf{x}(m^{-1}\cdot(\alpha:\beta)) = \begin{cases} P-u & \text{, when } m^{-1}\cdot(\alpha:\beta) = (u:1) \\ 1 & \text{, when } m^{-1}\cdot(\alpha:\beta) = (1:0) \end{cases}.$$

Using the field-equation $X^q = \frac{h_0(X)}{h_1(X)}$, we can rewrite the left-hand side with smaller degree polynomials, writing $\tilde{P}$ for the polynomial, where all coefficients are raised to their q-th power:

$$(a^q\tilde{P}(\frac{h_0}{h_1}) + b^q)(cP(X)+d) - (aP(X)+b)(c^q\tilde{P}(\frac{h_0}{h_1}) + d^q).$$

If its numerator is $\lceil\frac{D}{2}\rceil$-smooth, we say that the transformation m yields a *relation*.

Associate to every transformation m for which we obtained a relation a row-vector $v(m)$, indexed by all elements $\mu \in \mathbb{P}^1(\mathbb{F}_{q^2})$. Its coordinates are defined in the following way:

$$v(m)_{\mu \in \mathbb{P}^1(\mathbb{F}_{q^2})} = \begin{cases} 1 & \text{, if } \mu = m^{-1} \cdot (\alpha : \beta) \in \mathbb{P}^1(\mathbb{F}_q), \\ 0 & \text{, otherwise.} \end{cases}$$

Heuristic

For any $P(X)$, the set of rows $v(m)$ for which m yields a relation form a matrix that has full rank $q^2 + 1$.

If the heuristic is true, we can express the vector $(0, \ldots, 0, 1, 0, \ldots, 0)$ corresponding to the polynomial $P(X)$ as a sum of row-vectors and trace the computation using the smooth representation we found for each row and solve the above stated task of representing $\log P(X)$ as a linear combination of at most $\mathrm{O}(kq^2)$ logarithms $\log P_i(X)$ with $\deg P_i \leq \lceil \frac{1}{2} \deg P \rceil$ and of $\log h_1(X)$.

For the other task of computing the logarithm of $h_1(X)$ and the logarithm of all elements of K that are represented by linear polynomials $X + a$ for $a \in \mathbb{F}_{q^2}$, we perform exactly the same computation as above while setting $P(X) = X$.
Then only linear polynomials are involved and we can solve a linear system whose unknowns are $\log(X + a)$.

Heuristic

For $P(X) = X$, the linear system from all collected equations form a matrix that has full rank.

Theorem

Let $K = \mathbb{F}_{q^{2k}}$ be a finite field that admits sparse medium subfield representation. Under the above heuristics, there exists an algorithm whose complexity is polynomial in q and k, which can be used for the following two tasks:

1. Given an element of K, represented as a polynomial $P \in \mathbb{F}_{q^2}[X]$ with $2 \leq \deg P < k$, find a representation of $\log P(X)$ as a linear combination of at most $\mathrm{O}(kq^2)$ logarithms $\log P_i(X)$ with $\deg P_i \leq \lceil \frac{1}{2} \deg P \rceil$ and of $\log h_1(X)$.
2. Find the logarithm of $h_1(X)$ and the logarithm of all elements of K that are represented by linear polynomials $X + a$ for $a \in \mathbb{F}_{q^2}$.

Theorem

Let ℓ be a prime not dividing $q^3 - q$. Then the matrix $\mathcal{H}$ over $\mathbb{F}_\ell$, consisting of *all* rows corresponding to *any* transformation m has full rank $q^2 + 1$.

At present, the quasi-polynomial time algorithm for discrete logarithms was *not* successfully implemented, yet. However, a predecessor of the algorithm lead to the current world record.

Subject: Discrete Logarithms in GF(2^9234)
From: Jens Zumbrägel <[log in to unmask]>
Reply-To: Number Theory List <[log in to unmask]>
Date: Fri, 31 Jan 2014 07:59:39 -0600
Content-Type: text/plain
Parts/Attachments: text/plain (240 lines)

Dear Number Theorists,

We are pleased to announce a new record for the computation of discrete logarithms in finite fields. In particular, we were able to compute discrete logarithms in GF(2^9234) using about 400’000 core hours. To our knowledge the previous record was announced on 21 May 2013 in (a multiplicative subgroup of) the field GF((2^24)^257) of 6168 bits [8].

[...]

The running time (in core hours) is as follows:

```
 - relation generation         640 h  (AMD: 6128 Opteron 2.0 GHz)
 - linear algebra          258’048 h  (Intel: Ivy Bridge 2.4 GHz)
 - classical descent       134’889 h  (Intel)
 - Grobner basis descent     3’832 h  (AMD)
 - Pollard’s rho                13 h  (AMD)
```

totalling in 397’422 core hours.

Robert Granger*, Thorsten Kleinjung*, Jens Zumbragel^