Solving a 6120-bit DLP on a Desktop Computer

Faruk Göloğlu, **Robert Granger**, Gary McGuire, and Jens Zumbrägel

Claude Shannon Institute
Complex & Adaptive Systems Laboratory
School of Mathematical Sciences
University College Dublin, Ireland

15th August, SAC 2013

Our Contributions

Practical Results:

Our Contributions

Practical Results:

- Set a DLP record in $\mathbb{F}_{2^{6120}} = \mathbb{F}_{(2^{8\cdot 3})^{2^8-1}}$, in 750 core-hours:

Our Contributions

Practical Results:

- Set a DLP record in $\mathbb{F}_{2^{6120}} = \mathbb{F}_{(2^{8\cdot 3})^{2^8-1}}$, in 750 core-hours:
- Bitlength is 50% bigger than the previous record, set by Joux in $\mathbb{F}_{2^{4080}} = \mathbb{F}_{(2^{8\cdot 2})^{2^8-1}}$, but required only 5% of the core-hours

Our Contributions

Practical Results:

- Set a DLP record in $\mathbb{F}_{2^{6120}} = \mathbb{F}_{(2^{8\cdot 3})^{2^8-1}}$, in 750 core-hours:
- Bitlength is 50% bigger than the previous record, set by Joux in $\mathbb{F}_{2^{4080}} = \mathbb{F}_{(2^{8\cdot 2})^{2^8-1}}$, but required only 5% of the core-hours

Theoretical Results:

Our Contributions

Practical Results:

- Set a DLP record in $\mathbb{F}_{2^{6120}} = \mathbb{F}_{(2^{8\cdot 3})^{2^8-1}}$, in 750 core-hours:
- Bitlength is 50% bigger than the previous record, set by Joux in $\mathbb{F}_{2^{4080}} = \mathbb{F}_{(2^{8\cdot 2})^{2^8-1}}$, but required only 5% of the core-hours

Theoretical Results:

- Optimised Joux's $L_Q(1/4 + o(1))$ algorithm to give an $L_Q(1/4, (\omega/8)^{1/4})$ algorithm for $Q \approx (q^k)^q$, $k \geq 2$, $q \to \infty$

Overview

Big Field Hunting

Solving the DLP in $\mathbb{F}_{2^{6120}}$

Complexity Considerations

Polynomial Time Relation Generation [GGMZ13]

Setup for $\mathbb{F}_{(q^k)^n}$ with $k \geq 3$, $n \leq qd_1$ and $d_1 \geq 1$ (cf. [JL06]):

Polynomial Time Relation Generation [GGMZ13]

Setup for $\mathbb{F}_{(q^k)^n}$ with $k \geq 3$, $n \leq qd_1$ and $d_1 \geq 1$ (cf. [JL06]):

- Search for $g_1(X) \in \mathbb{F}_{q^k}[X]$ s.t. $X - g_1(X^q) \equiv 0 \pmod{f(X)}$ with $\deg(g_1) = d_1$, f irreducible and $\deg(f) = n$
- Let $\mathbb{F}_{(q^k)^n} = \mathbb{F}_{q^k}(x)$ with x a root of $f(X)$
- Let $y = x^q$, so that one has $x = g_1(y)$ in $\mathbb{F}_{(q^k)^n}$
- Factor base is $\{x - a \mid a \in \mathbb{F}_{q^k}\}$

Polynomial Time Relation Generation [GGMZ13]

Setup for $\mathbb{F}_{(q^k)^n}$ with $k \geq 3$, $n \leq qd_1$ and $d_1 \geq 1$ (cf. [JL06]):

- Search for $g_1(X) \in \mathbb{F}_{q^k}[X]$ s.t. $X - g_1(X^q) \equiv 0 \pmod{f(X)}$ with $\deg(g_1) = d_1$, f irreducible and $\deg(f) = n$
- Let $\mathbb{F}_{(q^k)^n} = \mathbb{F}_{q^k}(x)$ with x a root of $f(X)$
- Let $y = x^q$, so that one has $x = g_1(y)$ in $\mathbb{F}_{(q^k)^n}$
- Factor base is $\{x - a \mid a \in \mathbb{F}_{q^k}\}$

Relation generation:

Polynomial Time Relation Generation [GGMZ13]

Setup for $\mathbb{F}_{(q^k)^n}$ with $k \geq 3$, $n \leq qd_1$ and $d_1 \geq 1$ (cf. [JL06]):

- Search for $g_1(X) \in \mathbb{F}_{q^k}[X]$ s.t. $X - g_1(X^q) \equiv 0 \pmod{f(X)}$ with $\deg(g_1) = d_1$, f irreducible and $\deg(f) = n$
- Let $\mathbb{F}_{(q^k)^n} = \mathbb{F}_{q^k}(x)$ with x a root of $f(X)$
- Let $y = x^q$, so that one has $x = g_1(y)$ in $\mathbb{F}_{(q^k)^n}$
- Factor base is $\{x - a \mid a \in \mathbb{F}_{q^k}\}$

Relation generation:

- Considering elements $xy + ay + bx + c$ with $a, b, c \in \mathbb{F}_{q^k}$, one obtains the $\mathbb{F}_{(q^k)^n}$-equality

$$x^{q+1} + ax^q + bx + c = yg_1(y) + ay + bg_1(y) + c$$

- When both sides split over $\mathbb{F}_{q^k}$ one obtains a relation

Bluher Polynomials

Consider the l.h.s. polynomial $x^{q+1} + ax^q + bx + c$.

Bluher Polynomials

Consider the l.h.s. polynomial $x^{q+1} + ax^q + bx + c$.

If $ab \neq c$ and $a^q \neq b$, this may be transformed into

$$F_B(\overline{x}) = \overline{x}^{q+1} + B\overline{x} + B\,, \quad \text{with} \quad B = \frac{(b - a^q)^{q+1}}{(c - ab)^q}\,,$$

via $x = \frac{c-ab}{b-a^q}\,\overline{x} - a$.

Bluher Polynomials

Consider the l.h.s. polynomial $x^{q+1} + ax^q + bx + c$.

If $ab \neq c$ and $a^q \neq b$, this may be transformed into

$$F_B(\overline{x}) = \overline{x}^{q+1} + B\overline{x} + B\,, \quad \text{with} \quad B = \frac{(b-a^q)^{q+1}}{(c-ab)^q}\,,$$

via $x = \frac{c-ab}{b-a^q}\,\overline{x} - a$.

Theorem (*Bluher 2004, Helleseth-Kholosha 2010*)

The number of elements $B \in \mathbb{F}_{q^k}^\times$ such that the polynomial $F_B(X) \in \mathbb{F}_{q^k}[X]$ splits completely over $\mathbb{F}_{q^k}$ equals

$$\frac{q^{k-1}-1}{q^2-1} \quad \text{if } k \text{ is odd}\,, \qquad \frac{q^{k-1}-q}{q^2-1} \quad \text{if } k \text{ is even}\,.$$

Polynomial Time Relation Generation [GGMZ13]

- Let $S_B = \{B \in \mathbb{F}_{q^k}^{\times} \mid X^{q+1} + BX + B \text{ splits over } \mathbb{F}_{q^k}\}$

Polynomial Time Relation Generation [GGMZ13]

- Let $S_B = \{B \in \mathbb{F}_{q^k}^{\times} \mid X^{q+1} + BX + B \text{ splits over } \mathbb{F}_{q^k}\}$
- Since $B = (b - a^q)^{q+1}/(c - ab)^q$, for any $a, b \in \mathbb{F}_{q^k}$ s.t. $b \neq a^q$, and $B \in S_B$, there exists a unique $c \in \mathbb{F}_{q^k}$ s.t. $x^{q+1} + ax^q + bx + c$ splits over $\mathbb{F}_{q^k}$

Polynomial Time Relation Generation [GGMZ13]

- Let $S_B = \{B \in \mathbb{F}_{q^k}^{\times} \mid X^{q+1} + BX + B$ splits over $\mathbb{F}_{q^k}\}$
- Since $B = (b - a^q)^{q+1}/(c - ab)^q$, for any $a, b \in \mathbb{F}_{q^k}$ s.t. $b \neq a^q$, and $B \in S_B$, there exists a unique $c \in \mathbb{F}_{q^k}$ s.t. $x^{q+1} + ax^q + bx + c$ splits over $\mathbb{F}_{q^k}$
- For each such (a, b, c), test if r.h.s. $yg_1(y) + ay + bg_1(y) + c$ splits; if so then have a relation

Polynomial Time Relation Generation [GGMZ13]

- Let $S_B = \{B \in \mathbb{F}_{q^k}^{\times} \mid X^{q+1} + BX + B \text{ splits over } \mathbb{F}_{q^k}\}$
- Since $B = (b - a^q)^{q+1}/(c - ab)^q$, for any $a, b \in \mathbb{F}_{q^k}$ s.t. $b \neq a^q$, and $B \in S_B$, there exists a unique $c \in \mathbb{F}_{q^k}$ s.t. $x^{q+1} + ax^q + bx + c$ splits over $\mathbb{F}_{q^k}$
- For each such (a, b, c), test if r.h.s. $yg_1(y) + ay + bg_1(y) + c$ splits; if so then have a relation
- If $q^{3k-3} > q^k(d_1 + 1)!$ then expect to compute logs of degree 1 elements in time

$$\widetilde{O}(q^{2k+1})$$

Kummer Extensions $\implies$ More Efficient Attacks

The solution of DLPs in $\mathbb{F}_{p^{47}}$, $\mathbb{F}_{p^{57}}$, $\mathbb{F}_{2^{1778}}$, $\mathbb{F}_{2^{1971}}$, $\mathbb{F}_{2^{3164}}$ and $\mathbb{F}_{2^{4080}}$ all used Kummer extensions.

Kummer Extensions $\Longrightarrow$ More Efficient Attacks

The solution of DLPs in $\mathbb{F}_{p^{47}}$, $\mathbb{F}_{p^{57}}$, $\mathbb{F}_{2^{1778}}$, $\mathbb{F}_{2^{1971}}$, $\mathbb{F}_{2^{3164}}$ and $\mathbb{F}_{2^{4080}}$ all used Kummer extensions.

Why? Factor base-preserving automorphisms reduce effective size of factor base $\Longrightarrow$ relation finding & linear algebra become faster.

Kummer Extensions $\Longrightarrow$ More Efficient Attacks

The solution of DLPs in $\mathbb{F}_{p^{47}}$, $\mathbb{F}_{p^{57}}$, $\mathbb{F}_{2^{1778}}$, $\mathbb{F}_{2^{1971}}$, $\mathbb{F}_{2^{3164}}$ and $\mathbb{F}_{2^{4080}}$ all used Kummer extensions.

Why? Factor base-preserving automorphisms reduce effective size of factor base $\Longrightarrow$ relation finding & linear algebra become faster.

Observe that $\mathbb{F}_{2^{1778}}$ and $\mathbb{F}_{2^{4080}}$ are of the form $\mathbb{F}_{(q^2)^{q-1}}$, for which:

Kummer Extensions $\Longrightarrow$ More Efficient Attacks

The solution of DLPs in $\mathbb{F}_{p^{47}}$, $\mathbb{F}_{p^{57}}$, $\mathbb{F}_{2^{1778}}$, $\mathbb{F}_{2^{1971}}$, $\mathbb{F}_{2^{3164}}$ and $\mathbb{F}_{2^{4080}}$ all used Kummer extensions.

Why? Factor base-preserving automorphisms reduce effective size of factor base $\Longrightarrow$ relation finding & linear algebra become faster.

Observe that $\mathbb{F}_{2^{1778}}$ and $\mathbb{F}_{2^{4080}}$ are of the form $\mathbb{F}_{(q^2)^{q-1}}$, for which:

- Degree 1 logs cost $\widetilde{O}(q^3)$ for K.E., or $\widetilde{O}(q^5)$ otherwise
- Degree 2 logs cost $\widetilde{O}(q^6)$ for K.E., or $\widetilde{O}(q^7)$ otherwise

Kummer Extensions $\implies$ More Efficient Attacks

The solution of DLPs in $\mathbb{F}_{p^{47}}$, $\mathbb{F}_{p^{57}}$, $\mathbb{F}_{2^{1778}}$, $\mathbb{F}_{2^{1971}}$, $\mathbb{F}_{2^{3164}}$ and $\mathbb{F}_{2^{4080}}$ all used Kummer extensions.

Why? Factor base-preserving automorphisms reduce effective size of factor base $\implies$ relation finding & linear algebra become faster.

Observe that $\mathbb{F}_{2^{1778}}$ and $\mathbb{F}_{2^{4080}}$ are of the form $\mathbb{F}_{(q^2)^{q-1}}$, for which:

- Degree 1 logs cost $\widetilde{O}(q^3)$ for K.E., or $\widetilde{O}(q^5)$ otherwise
- Degree 2 logs cost $\widetilde{O}(q^6)$ for K.E., or $\widetilde{O}(q^7)$ otherwise

However, for $\mathbb{F}_{(q^k)^{q\pm1}}$ with $k \geq 4$ one can compute logs of degree two elements *on the fly* [GGMZ13].

New Degree 2 elimination for K.E.'s and $k \geq 3$

Let $q(x) := x^2 + q_1 x + q_0 \in \mathbb{F}_{(q^k)^{q-1}}$ be an element to be written as a product of linear elements.

New Degree 2 elimination for K.E.'s and $k \geq 3$

Let $q(x) := x^2 + q_1 x + q_0 \in \mathbb{F}_{(q^k)^{q-1}}$ be an element to be written as a product of linear elements.

- When possible, compute $a, b, c \in \mathbb{F}_{q^k}$ s.t. in $\mathbb{F}^{\times}_{(q^k)^{q-1}} / \mathbb{F}^{\times}_{q^k}$,

$$q(x) = x^2 + q_1 x + q_0 = x^{q+1} + a x^q + b x + c$$

  where r.h.s splits over $\mathbb{F}^{\times}_{q^k}$

New Degree 2 elimination for K.E.'s and $k \geq 3$

Let $q(x) := x^2 + q_1 x + q_0 \in \mathbb{F}_{(q^k)^{q-1}}$ be an element to be written as a product of linear elements.

- When possible, compute $a, b, c \in \mathbb{F}_{q^k}$ s.t. in $\mathbb{F}^{\times}_{(q^k)^{q-1}} / \mathbb{F}^{\times}_{q^k}$,

$$q(x) = x^2 + q_1 x + q_0 = x^{q+1} + a x^q + b x + c$$

  where r.h.s splits over $\mathbb{F}^{\times}_{q^k}$

- As $x^{q-1} = \gamma$, we have r.h.s. $= \gamma(x^2 + (a + \frac{b}{\gamma})x + \frac{c}{\gamma})$:
  $\Longrightarrow \gamma q_0 = c, \gamma q_1 = \gamma a + b$

New Degree 2 elimination for K.E.'s and $k \geq 3$

Let $q(x) := x^2 + q_1 x + q_0 \in \mathbb{F}_{(q^k)^{q-1}}$ be an element to be written as a product of linear elements.

- When possible, compute $a, b, c \in \mathbb{F}_{q^k}$ s.t. in $\mathbb{F}^{\times}_{(q^k)^{q-1}} / \mathbb{F}^{\times}_{q^k}$,

$$q(x) = x^2 + q_1 x + q_0 = x^{q+1} + a x^q + b x + c$$

  where r.h.s splits over $\mathbb{F}^{\times}_{q^k}$
- As $x^{q-1} = \gamma$, we have r.h.s. $= \gamma(x^2 + (a + \frac{b}{\gamma})x + \frac{c}{\gamma})$:
  $\Longrightarrow \gamma q_0 = c, \gamma q_1 = \gamma a + b$
- For any $B \in S_B$, using $(a^q + b)^{q+1} = B(ab + c)^q$ we arrive at the condition

$$(a^q + \gamma a + \gamma q_1)^{q+1} + B(\gamma a^2 + \gamma q_1 a + \gamma q_0)^q = 0$$

New Degree 2 elimination for K.E.'s and $k \geq 3$

Let $q(x) := x^2 + q_1 x + q_0 \in \mathbb{F}_{(q^k)^{q-1}}$ be an element to be written as a product of linear elements.

- When possible, compute $a, b, c \in \mathbb{F}_{q^k}$ s.t. in $\mathbb{F}^{\times}_{(q^k)^{q-1}} / \mathbb{F}^{\times}_{q^k}$,

  $$q(x) = x^2 + q_1 x + q_0 = x^{q+1} + a x^q + b x + c$$

  where r.h.s splits over $\mathbb{F}^{\times}_{q^k}$
- As $x^{q-1} = \gamma$, we have r.h.s. $= \gamma(x^2 + (a + \frac{b}{\gamma})x + \frac{c}{\gamma})$:
  $\implies \gamma q_0 = c, \gamma q_1 = \gamma a + b$
- For any $B \in S_B$, using $(a^q + b)^{q+1} = B(ab + c)^q$ we arrive at the condition

  $$(a^q + \gamma a + \gamma q_1)^{q+1} + B(\gamma a^2 + \gamma q_1 a + \gamma q_0)^q = 0$$

- Considering $\mathbb{F}_{q^k}/\mathbb{F}_q$ gives a quadratic system in the $\mathbb{F}_q$-components of a, solvable with a Gröbner basis computation

Cost of Computing Factor base Logs for K.E.'s

For $q = 2^l$ and $n = q - 1$, $\mathbb{F}_{(q^k)^n}$ has bitlength:

$l \setminus k$	2	3	4	5	6
6	756	1134	1512	1890	2268
7	1778	2667	3556	4445	5334
8	4080	6120	8160	10200	12240
9	9198	13797	18396	22995	27594

Cost of Computing Factor base Logs for K.E.'s

For $q = 2^l$ and $n = q - 1$, $\mathbb{F}_{(q^k)^n}$ has bitlength:

$l \setminus k$	2	3	4	5	6
6	756	1134	1512	1890	2268
7	1778	2667	3556	4445	5334
8	4080	6120	8160	10200	12240
9	9198	13797	18396	22995	27594

- Degree 1: #variables $\approx q^{k-1}$ so for $k \geq 2$, cost is $\widetilde{O}(q^{2k-1})$

Cost of Computing Factor base Logs for K.E.'s

For $q = 2^l$ and $n = q - 1$, $\mathbb{F}_{(q^k)^n}$ has bitlength:

$l \setminus k$	2	3	4	5	6
6	756	1134	1512	1890	2268
7	1778	2667	3556	4445	5334
8	4080	6120	8160	10200	12240
9	9198	13797	18396	22995	27594

- Degree 1: #variables $\approx q^{k-1}$ so for $k \geq 2$, cost is $\widetilde{O}(q^{2k-1})$
- Degree 2: For $k = 2, 3$ cost is $\widetilde{O}(q^{2k+2})$, and free for $k \geq 4$

Cost of Computing Factor base Logs for K.E.'s

For $q = 2^l$ and $n = q - 1$, $\mathbb{F}_{(q^k)^n}$ has bitlength:

$l \setminus k$	2	3	4	5	6
6	756	1134	1512	1890	2268
7	1778	2667	3556	4445	5334
8	4080	6120	8160	10200	12240
9	9198	13797	18396	22995	27594

- Degree 1: #variables $\approx q^{k-1}$ so for $k \geq 2$, cost is $\widetilde{O}(q^{2k-1})$
- Degree 2: For $k = 2, 3$ cost is $\widetilde{O}(q^{2k+2})$, and free for $k \geq 4$

k	2	3	4	5	6
Cost	$\widetilde{O}(q^6)$	$\widetilde{O}(q^8)$	$\widetilde{O}(q^7)$	$\widetilde{O}(q^9)$	$\widetilde{O}(q^{11})$

Cost of Computing Factor base Logs for K.E.'s

For $q = 2^l$ and $n = q - 1$, $\mathbb{F}_{(q^k)^n}$ has bitlength:

$l \setminus k$	2	3	4	5	6
6	756	1134	1512	1890	2268
7	1778	2667	3556	4445	5334
8	4080	6120	8160	10200	12240
9	9198	13797	18396	22995	27594

- Degree 1: #variables $\approx q^{k-1}$ so for $k \geq 2$, cost is $\widetilde{O}(q^{2k-1})$
- Degree 2: For $k = 2, 3$ cost is $\widetilde{O}(q^{2k+2})$, and free for $k \geq 4$

k	2	3	4	5	6
Cost	$\widetilde{O}(q^6)$	$\widetilde{O}(q^5)$	$\widetilde{O}(q^7)$	$\widetilde{O}(q^9)$	$\widetilde{O}(q^{11})$

Field Setup and Target Element

- Let $\mathbb{F}_{2^8} = \mathbb{F}_2[T]/((T^8 + T^4 + T^3 + T + 1)\mathbb{F}_2[T]) = \mathbb{F}_2(t)$
- Let $\mathbb{F}_{2^{24}} = \mathbb{F}_{2^8}[W]/((W^3 + t)\mathbb{F}_{2^8}[W]) = \mathbb{F}_{2^8}(w)$
- Let $\mathbb{F}_{2^{6120}} = \mathbb{F}_{2^{24}}[X]/((X^{255} + w + 1)\mathbb{F}_{2^{24}}[X]) = \mathbb{F}_{2^{24}}(x)$
- Our generator is $g = x + w$, which has proven order $2^{6120} - 1$

Field Setup and Target Element

- Let $\mathbb{F}_{2^8} = \mathbb{F}_2[T]/((T^8 + T^4 + T^3 + T + 1)\mathbb{F}_2[T]) = \mathbb{F}_2(t)$
- Let $\mathbb{F}_{2^{24}} = \mathbb{F}_{2^8}[W]/((W^3 + t)\mathbb{F}_{2^8}[W]) = \mathbb{F}_{2^8}(w)$
- Let $\mathbb{F}_{2^{6120}} = \mathbb{F}_{2^{24}}[X]/((X^{255} + w + 1)\mathbb{F}_{2^{24}}[X]) = \mathbb{F}_{2^{24}}(x)$
- Our generator is $g = x + w$, which has proven order $2^{6120} - 1$

Our target element β_π was derived as usual from the 2^{24}-ary expansion of π.

Degree 1 Logarithms

- Used the only Bluher polynomial for $k = 3$, namely $X^{257} + X + 1$ and our relation generation method
- Via automorphisms, reduced the #variables to $21,932$ and obtained $22,932$ relations *in 15 seconds* using C++/NTL on a 2.0 GHz AMD Opteron 6128
- For linear algebra, took as modulus the product of the largest 35 prime factors of $2^{6120} - 1$, which has bitlength 5121
- Ran a parallelised C/GMP implementation of Lanczos' algorithm on four of the Intel (Westmere) Xeon E5650 hex-core processors of ICHEC's SGI Altix ICE 8200EX Stokes cluster, completed *in 60.5 core-hours* (2.5 hours wall time)

Degree 2 Logarithms

Since there is only one Bluher polynomial for $k = 3$, elimination probability is $1/2$.

Degree 2 Logarithms

Since there is only one Bluher polynomial for $k = 3$, elimination probability is $1/2$.

- When it fails, exploit the fact that $6 \mid 24$ and $(8-6) \mid 24$ and the 64 Bluher polynomials of the form $X^{65} + BX + B \ /\mathbb{F}_{2^{24}}$

Degree 2 Logarithms

Since there is only one Bluher polynomial for $k = 3$, elimination probability is $1/2$.

- When it fails, exploit the fact that $6 \mid 24$ and $(8-6) \mid 24$ and the 64 Bluher polynomials of the form $X^{65} + BX + B \ /\mathbb{F}_{2^{24}}$
- Results in a probabilistic method to eliminate any given degree 2 element with probability $p = 1 - 6.3 \times 10^{-15}$
- $\implies$ probability that at least one degree 2 irreducible is not eliminable is $1 - p^{2^{22}} = 2.7 \times 10^{-8}$
- Implemented in MAGMA V2.16-12 on a 2.0GHz AMD Opteron 6128: *each took on average 0.03 seconds*

Eliminating Degrees 3,4,5 and 6

We used an analogue of Joux's method [J13], but with the Bluher polynomial $X^{257} + X + 1$ rather than $X^{256} + X$.

Eliminating Degrees 3,4,5 and 6

We used an analogue of Joux's method [J13], but with the Bluher polynomial $X^{257} + X + 1$ rather than $X^{256} + X$.

- Let $f(X), g(X) \in \mathbb{F}_{2^{24}}[X]$ have degrees δ_f and δ_g
- Substitute $\frac{f(X)}{g(X)}$ into Bluher polynomial, giving the numerator

$$P(X) := f(X)^{257} + Bf(X)\,g(X)^{256} + Bg(X)^{257}$$

- $P(X)$ is δ-smooth with $\delta = \max\{\delta_f, \delta_g\}$
- Since $x^{256} = (w+1)x$ holds in $\mathbb{F}_{(2^{24})^{255}}$, the element $P(x)$ can also be represented by a polynomial of degree 2δ
- For $Q(x)$ of degree 2δ or $2\delta - 1$ set $P(x) = Q(x)$ or $(x + a)Q(x)$ and solve resulting quadratic system over $\mathbb{F}_{2^8}$

DLP Solution

On 11/4/13 we announced that $\beta_\pi = g^{\log}$, with $\log =$

13858759836397869262547571128312317100923636150389699236649593170451770028012717802223489409861758136013144183507425636373062442681429323347427252159816612695792811682544311096540425383793880859540411103523802710777217882293928187340345199973181514007348176651371535844927931455679735244624686031794675012447568947440627494235603593650167405093344890920102983452222673224777189708322321728205157364501360361304236778271636187781793837439382431301907362478638761841403754168112028404465938319290743685252639208772430477545163127182525096811145140050273340438176967525528912734663935009822157084440038078851633249658388252243638191800820016703218635024510775134697959631469615366671616895148194809106000667301847667581377739443038754298308672054639181442568439117307472651461541934380416278336617397750571612363460962365668752512778430623299730444754865610622043569085684714712793837810385388188844637969899060760798432481272520208397058864360712136505751867074569485840723789169429253691408684171964795734810327114810217291628659735881740963899133056076778580339963617349055371503620247205157726607812088555054343310557665700142118756029406335757638504575030790870743765853044705204113202462922553757114575735552860602366993170394544793267182811289614232751427875694256905328332833440496355213025960008971925120366952988072940329645309596913770872045463489601327600955441059801982552454932024128315938919847881524179576919398171123661820636875299153651503611802144512343876568832561493559944050511495859691630753070266479560356836715895464485399551327261120349386559612918562034222476803870290784735209511603344725254750716806726236615872927203296061825120443121943571561392013409520378729752432544760815549370021229534159494072621372320998522983948384229076431913976732902383441830460409758599159285365304456971453176680449737096483324156185041

Complexity Considerations

The quadratic systems we obtain using $X^{q+1} + BX + B$ are not bilinear $\implies$ we can't argue for the same $L_Q(1/4 + o(1))$ complexity that arises when using $X^q - X$.

Complexity Considerations

The quadratic systems we obtain using $X^{q+1} + BX + B$ are not bilinear $\implies$ we can't argue for the same $L_Q(1/4 + o(1))$ complexity that arises when using $X^q - X$.

However, when using $X^q - X$, with judiciously chosen parameters, the complexity can be improved.

Complexity Considerations

The quadratic systems we obtain using $X^{q+1} + BX + B$ are not bilinear $\Longrightarrow$ we can't argue for the same $L_Q(1/4 + o(1))$ complexity that arises when using $X^q - X$.

However, when using $X^q - X$, with judiciously chosen parameters, the complexity can be improved.

- Consider $\mathbb{F}_{(q^k)^n}$ with $k \geq 2$ fixed, $n \approx q$ and $q \to \infty$
- Assume degree 1 logs are known and degree 2 logs are either known or are efficiently computable (on the fly)

The Descent

Want to compute $\log_g h$. The descent consists of 3 parts:

The Descent

Want to compute $\log_g h$. The descent consists of 3 parts:

- Stage 0: Choose random i until hg^i is $\alpha_0 q^{3/4}$-smooth. This costs

$$C_0 := L_{q^{kq}}\left(1/4, \frac{1}{4\alpha_0 k^{1/4}}\right)$$

The Descent

Want to compute $\log_g h$. The descent consists of 3 parts:

- Stage 0: Choose random i until hg^i is $\alpha_0 q^{3/4}$-smooth. This costs

$$C_0 := L_{q^{kq}}\left(1/4, \frac{1}{4\alpha_0 k^{1/4}}\right)$$

- Stage 1: Perform classical descent (with degree balancing) until elements are $\alpha_1 q^{1/2}$-smooth. For $0 < \mu < 1$, this costs

$$C_1 := L_{q^{kq}}\left(1/4, \frac{1}{\mu k^{1/4}\sqrt{8\alpha_1}}\right)$$

The Descent

Want to compute $\log_g h$. The descent consists of 3 parts:

- Stage 0: Choose random i until hg^i is $\alpha_0 q^{3/4}$-smooth. This costs
$$C_0 := L_{q^{kq}}\left(1/4, \frac{1}{4\alpha_0 k^{1/4}}\right)$$
- Stage 1: Perform classical descent (with degree balancing) until elements are $\alpha_1 q^{1/2}$-smooth. For $0 < \mu < 1$, this costs
$$C_1 := L_{q^{kq}}\left(1/4, \frac{1}{\mu k^{1/4}\sqrt{8\alpha_1}}\right)$$
- Stage 2: Perform Joux's descent until elements are 2-smooth. This costs
$$C_2 := L_{q^{kq}}\left(1/4, k^{1/4}\sqrt{\omega\alpha_1}\right)$$

The Descent

- Balancing Stages 1 and 2 gives the optimal α_1 as $1/(\mu\sqrt{8k\omega})$

The Descent

- Balancing Stages 1 and 2 gives the optimal α_1 as $1/(\mu\sqrt{8k\omega})$
- Choosing $\alpha_0 > 1/(32k\omega)^{1/4}$ means Stage 0 is ignorable

The Descent

- Balancing Stages 1 and 2 gives the optimal α_1 as $1/(\mu\sqrt{8k\omega})$
- Choosing $\alpha_0 > 1/(32k\omega)^{1/4}$ means Stage 0 is ignorable
- In the limit as $\mu \to 1^-$, we obtain an overall complexity of

$$L_{q^{kq}}(1/4, (\omega/8)^{1/4})$$

A Final Remark

- Barbulescu, Gaudry, Joux and Thomé have proposed a quasi-polynomial algorithm for the DLP in finite fields of small characteristic (eprint.iacr.org/2013/400)

A Final Remark

- Barbulescu, Gaudry, Joux and Thomé have proposed a quasi-polynomial algorithm for the DLP in finite fields of small characteristic (eprint.iacr.org/2013/400)
- Our relation generation method gives an analogous quasi-polynomial algorithm; in fact ours and Joux's method based on Möbius transforms of $X^q - X$ are equivalent

A Final Remark

- Barbulescu, Gaudry, Joux and Thomé have proposed a quasi-polynomial algorithm for the DLP in finite fields of small characteristic (eprint.iacr.org/2013/400)
- Our relation generation method gives an analogous quasi-polynomial algorithm; in fact ours and Joux's method based on Möbius transforms of $X^q - X$ are equivalent

For BGJT algorithm, one setup issue is to find a set of coset representatives for $PGL_2(\mathbb{F}_{q^k})/PGL_2(\mathbb{F}_q)$:

A Final Remark

- Barbulescu, Gaudry, Joux and Thomé have proposed a quasi-polynomial algorithm for the DLP in finite fields of small characteristic (eprint.iacr.org/2013/400)
- Our relation generation method gives an analogous quasi-polynomial algorithm; in fact ours and Joux's method based on Möbius transforms of $X^q - X$ are equivalent

For BGJT algorithm, one setup issue is to find a set of coset representatives for $PGL_2(\mathbb{F}_{q^k})/PGL_2(\mathbb{F}_q)$:

- $|PGL_2(\mathbb{F}_{q^k})/PGL_2(\mathbb{F}_q)| = (q^{3k} - q^k)/(q^3 - q) \approx q^{3k-3}$

A Final Remark

- Barbulescu, Gaudry, Joux and Thomé have proposed a quasi-polynomial algorithm for the DLP in finite fields of small characteristic (eprint.iacr.org/2013/400)
- Our relation generation method gives an analogous quasi-polynomial algorithm; in fact ours and Joux's method based on Möbius transforms of $X^q - X$ are equivalent

For BGJT algorithm, one setup issue is to find a set of coset representatives for $PGL_2(\mathbb{F}_{q^k})/PGL_2(\mathbb{F}_q)$:

- $|PGL_2(\mathbb{F}_{q^k})/PGL_2(\mathbb{F}_q)| = (q^{3k} - q^k)/(q^3 - q) \approx q^{3k-3}$
- For $k \geq 3$ our search space has cardinality

$$q^k(q^k - 1)(q^k - \{q, q^2\})/(q^3 - q) \approx q^{3k-3}$$

A Final Remark

- Barbulescu, Gaudry, Joux and Thomé have proposed a quasi-polynomial algorithm for the DLP in finite fields of small characteristic (eprint.iacr.org/2013/400)
- Our relation generation method gives an analogous quasi-polynomial algorithm; in fact ours and Joux's method based on Möbius transforms of $X^q - X$ are equivalent

For BGJT algorithm, one setup issue is to find a set of coset representatives for $PGL_2(\mathbb{F}_{q^k})/PGL_2(\mathbb{F}_q)$:

- $|PGL_2(\mathbb{F}_{q^k})/PGL_2(\mathbb{F}_q)| = (q^{3k} - q^k)/(q^3 - q) \approx q^{3k-3}$
- For $k \geq 3$ our search space has cardinality

$$q^k(q^k - 1)(q^k - \{q, q^2\})/(q^3 - q) \approx q^{3k-3}$$

- Cost of finding all Bluher polynomials is only $\widetilde{O}(q^k)$