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:

• Set a DLP record in $\mathbb{F}_{2^{6120}} = \mathbb{F}_{2^{28}}$, in 750 core-hours:
  - Bitlength is 50% bigger than the previous record, set by Joux in $\mathbb{F}_{2^{4080}} = \mathbb{F}_{2^{28}}$, 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 $Q \approx (q^k)^q$, $k \geq 2$, $q \to \infty$. 
Our Contributions

Practical Results:

- Set a DLP record in $\mathbb{F}_{2^{6120}} = \mathbb{F}_{((2^8 \cdot 3)^{28} - 1)}$, in 750 core-hours:
Our Contributions

Practical Results:

- Set a DLP record in $\mathbb{F}_{26120} = \mathbb{F}_{(2^8 \cdot 3)^{2^8 - 1}}$, in 750 core-hours:
- Bitlength is 50% bigger than the previous record, set by Joux in $\mathbb{F}_{24080} = \mathbb{F}_{(2^8 \cdot 2)^{2^8 - 1}}$, but required only 5% of the core-hours
Our Contributions

Practical Results:

- Set a DLP record in \( \mathbb{F}_{26120} = \mathbb{F}_{(2^8 \cdot 3)^{28} - 1} \), in 750 core-hours:
- Bitlength is 50\% bigger than the previous record, set by Joux in \( \mathbb{F}_{24080} = \mathbb{F}_{(2^8 \cdot 2)^{28} - 1} \), but required only 5\% of the core-hours

Theoretical Results:
Our Contributions

Practical Results:

- Set a DLP record in $\mathbb{F}_{26120} = \mathbb{F}_{(2^8 \cdot 3)^{2^8 - 1}}$, in 750 core-hours:
  - Bitlength is 50% bigger than the previous record, set by Joux in $\mathbb{F}_{24080} = \mathbb{F}_{(2^8 \cdot 2)^{2^8 - 1}}$, 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]):

• Search for $g_1(X) \in \mathbb{F}_{q^k}[X]$ s.t.

\[ X - g_1(X) \equiv 0 \pmod{f(X)} \]

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 \{\(x - a\) | $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

\[ xq + 1 + axq + bx + c = yg_1(y) + ay + bg_1(y) + c \]

• When both sides split over $\mathbb{F}_{q^k}$ one obtains a relation.
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 $g_1(X) \in \mathbb{F}_q[X]$ 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

$$F_B(x) = x^{q+1} + Bx + B,$$

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

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

$$F_B(x) = x^{q+1} + Bx + B,$$

with

$$B = \frac{(b - a^q)^{q+1}}{(c - ab)^q},$$

via

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

Theorem (Bluher 2004, Helleseth-Kholosha 2010)

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

$$\frac{q^{k-1} - 1}{q^2 - 1} \quad \text{if } k \text{ is odd}, \quad \frac{q^{k-1} - q}{q^2 - 1} \quad \text{if } k \text{ is even}.$$
Polynomial Time Relation Generation [GGMZ13]

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

- 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
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 $\tilde{O}(q^{2k+1})$
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 $\implies$ 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.
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:
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 $\tilde{O}(q^3)$ for K.E., or $\tilde{O}(q^5)$ otherwise
- Degree 2 logs cost $\tilde{O}(q^6)$ for K.E., or $\tilde{O}(q^7)$ otherwise
Kummer Extensions $\rightarrow$ 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 $\rightarrow$ 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 $\tilde{O}(q^3)$ for K.E., or $\tilde{O}(q^5)$ otherwise
- Degree 2 logs cost $\tilde{O}(q^6)$ for K.E., or $\tilde{O}(q^7)$ 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 $\mathbb{F}_{(q^k)q^{-1}}/\mathbb{F}_{q^k}$,

$$q(x) = x^2 + q_1 x + q_0 = x^{q+1} + ax^q + bx + c$$

where r.h.s splits over $\mathbb{F}_{q^k}^\times$.
New Degree 2 elimination for K.E.’s and $k \geq 3$

Let $q(x) := x^2 + q_1 x + q_0 \in \mathbb{F}_{q^k}^{(q^k)_{q-1}}$ 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 $\mathbb{F}_{q^k}^{(q^k)_{q-1}}/\mathbb{F}_{q^k}^\times$, $q(x) = x^2 + q_1 x + q_0 = x^{q+1} + ax^q + bx + c$

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

- As $x^{q-1} = \gamma$, we have r.h.s. $= \gamma(x^2 + (a + \frac{b}{\gamma})x + \frac{c}{\gamma})$:
  $\implies \gamma q_0 = c, \gamma q_1 = \gamma a + b$
New Degree 2 elimination for K.E.’s and \( k \geq 3 \)

Let \( q(x) := x^2 + q_1x + q_0 \in \mathbb{F}_{(q^k)q^{-1}} \) 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 \( \mathbb{F}_{(q^k)q^{-1}}/\mathbb{F}_{q^k} \),

\[
q(x) = x^2 + q_1x + q_0 = x^{q+1} + ax^q + bx + c
\]

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

- 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

\[
(a^q + \gamma a + \gamma q_1)^{q+1} + B(\gamma a^2 + \gamma q_1 a + \gamma q_0)^q = 0
\]
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 $\mathbb{F}_{(q^k)^{q-1}} / \mathbb{F}_{q^k}$,

$$q(x) = x^2 + q_1 x + q_0 = x^{q+1} + ax^q + bx + c$$

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

• 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:

<table>
<thead>
<tr>
<th>( l ) ( k )</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>756</td>
<td>1134</td>
<td>1512</td>
<td>1890</td>
<td>2268</td>
</tr>
<tr>
<td>7</td>
<td>1778</td>
<td>2667</td>
<td>3556</td>
<td>4445</td>
<td>5334</td>
</tr>
<tr>
<td>8</td>
<td>4080</td>
<td>6120</td>
<td>8160</td>
<td>10200</td>
<td>12240</td>
</tr>
<tr>
<td>9</td>
<td>9198</td>
<td>13797</td>
<td>18396</td>
<td>22995</td>
<td>27594</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>( l ) ( k )</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>756</td>
<td>1134</td>
<td>1512</td>
<td>1890</td>
<td>2268</td>
</tr>
<tr>
<td>7</td>
<td>1778</td>
<td>2667</td>
<td>3556</td>
<td>4445</td>
<td>5334</td>
</tr>
<tr>
<td>8</td>
<td>4080</td>
<td>6120</td>
<td>8160</td>
<td>10200</td>
<td>12240</td>
</tr>
<tr>
<td>9</td>
<td>9198</td>
<td>13797</td>
<td>18396</td>
<td>22995</td>
<td>27594</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$l \ \backslash \ k$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>756</td>
<td>1134</td>
<td>1512</td>
<td>1890</td>
<td>2268</td>
</tr>
<tr>
<td>7</td>
<td>1778</td>
<td>2667</td>
<td>3556</td>
<td>4445</td>
<td>5334</td>
</tr>
<tr>
<td>8</td>
<td>4080</td>
<td>6120</td>
<td>8160</td>
<td>10200</td>
<td>12240</td>
</tr>
<tr>
<td>9</td>
<td>9198</td>
<td>13797</td>
<td>18396</td>
<td>22995</td>
<td>27594</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$l \setminus k$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>756</td>
<td>1134</td>
<td>1512</td>
<td>1890</td>
<td>2268</td>
</tr>
<tr>
<td>7</td>
<td>1778</td>
<td>2667</td>
<td>3556</td>
<td>4445</td>
<td>5334</td>
</tr>
<tr>
<td>8</td>
<td>4080</td>
<td>6120</td>
<td>8160</td>
<td>10200</td>
<td>12240</td>
</tr>
<tr>
<td>9</td>
<td>9198</td>
<td>13797</td>
<td>18396</td>
<td>22995</td>
<td>27594</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$k$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>$\tilde{O}(q^6)$</td>
<td>$\tilde{O}(q^8)$</td>
<td>$\tilde{O}(q^7)$</td>
<td>$\tilde{O}(q^9)$</td>
<td>$\tilde{O}(q^{11})$</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>$l \setminus k$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>756</td>
<td>1134</td>
<td>1512</td>
<td>1890</td>
<td>2268</td>
</tr>
<tr>
<td>7</td>
<td>1778</td>
<td>2667</td>
<td>3556</td>
<td>4445</td>
<td>5334</td>
</tr>
<tr>
<td>8</td>
<td>4080</td>
<td>6120</td>
<td>8160</td>
<td>10200</td>
<td>12240</td>
</tr>
<tr>
<td>9</td>
<td>9198</td>
<td>13797</td>
<td>18396</td>
<td>22995</td>
<td>27594</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$k$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>$\tilde{O}(q^6)$</td>
<td>$\tilde{O}(q^5)$</td>
<td>$\tilde{O}(q^7)$</td>
<td>$\tilde{O}(q^9)$</td>
<td>$\tilde{O}(q^{11})$</td>
</tr>
</tbody>
</table>
Field Setup and Target Element

• Let $F_{2^8} = F_2[T] / ((T^8 + T^4 + T^3 + T + 1)F_2[T]) = F_2(t)$
• Let $F_{2^{24}} = F_{2^8}[W] / ((W^3 + t)F_{2^8}[W]) = F_{2^8}(w)$
• Let $F_{2^{6120}} = F_{2^{24}}[X] / ((X^{255} + w + 1)F_{2^{24}}[X]) = F_{2^{24}}(x)$
• Our generator is $g = x + w$, which has proven order $2^{6120} - 1$
Field Setup and Target Element

- Let $F_{28} = F_2[T] / ((T^8 + T^4 + T^3 + T + 1)F_2[T]) = F_2(t)$
- Let $F_{24} = F_{28}[W] / ((W^3 + t)F_{28}[W]) = F_{28}(w)$
- Let $F_{26120} = F_{24}[X] / ((X^{255} + w + 1)F_{24}[X]) = F_{24}(x)$
- Our generator is $g = x + w$, which has proven order $2^{6120} - 1$

Our target element $\beta_{\pi}$ was derived as usual from the $2^{24}$-ary expansion of $\pi$. 
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 number of variables to 21,932 and obtained 22,932 relations in 15 seconds using C++/NTL on a 2.0GHz 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 \( X^{65} + BX + B \mod F_{24}^2 \)
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 $X^{65} + BX + B / \mathbb{F}_{24}^2$
- 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 $1 - p^{2^2} = 2.7 \times 10^{-8}$
- 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 $\delta_f$ and $\delta_g$
- 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 $\delta$-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 $2\delta$
- For $Q(x)$ of degree $2\delta$ 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 $log =$

1385875983639786926254757112831231710092363615038969923664959317045177002801271780222234894098617
581360131441835074256363730624426814293233474272521598166126957928116825443110965404253837938808
59450411103523802710772178229328187304334199731815140073481766513715358449279314556797352446
24686031794675012447568947440627494235603593650167405093344890920120983452226732247771897083223
217282051573645013603613042367782716361877817938374393824313019073624786387618414037541681120284
044659383192907436852526392087724304775451631271825250968111451400502733404381769675255289127346
639350098221570844400380788516332496583882522436381918008200167032186350245107751346979596314696
153666716168951481948091060066730184766758137773944303875429830867205463918144256843911730747265
146154193438041627833661739775057161236346096236566875251277843062329973044475486561062204356908
56847147127938378103853881888463796989906076079843248127252020839705886436071213650575186707456
948584072378916942925369140868417196479573481032711481021729162865973588174096389913305607677858
033996361734905537150362024720515772660781208855505434331055766570014211875602940633575763850457
503079087074376585304470520411320246292255375711457573555286060236699317039454479326718281128961
42327514278756942569053283328334404963552130259600897192512036695298807294032964530959691377087
204546348960132760095544105980198255245493202412831593891984788152417957691939817112366182063687
529915365150361180214451234387656883256149355994405051149585969163075307026647956035683671589546
448539955132726112034938655961291856203422247680387029078473520951160334472525475071680672623661
587292720329606182512044312194357516139201340952037872975243254476081554937002122953415949407262
137232099852298394838422907643191397673290238344183046040975859915928536530445697145317668044973
7096483324156185041
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.
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.

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

- **Stage 1**: Perform classical descent (with degree balancing) until elements are \( \alpha_1 q^{1/2} \)-smooth. For \( 0 < \mu < 1 \), this costs
  \[
  C_1 := L_{q^{kq}} \left( \frac{1}{4}, \frac{1}{\mu k^{1/4} \sqrt{8\alpha_1}} \right)
  \]
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 $\alpha_1$ as $1/\left(\mu \sqrt{8k\omega}\right)$
The Descent

- Balancing Stages 1 and 2 gives the optimal $\alpha_1$ 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 $\alpha_1$ 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

$$L_{q^k \lambda q}(1/4, (\omega/8)^{1/4})$$
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 $PGL_2(F_{q^k})/PGL_2(F_q)$:
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 $\tilde{O}(q^k)$