Discrete logarithms in small characteristic finite fields: Attacking Type 1 pairing-based cryptography

Gora Adj

CINVESTAV, Mexico

Joint work with:

Alfred Menezes	U. of Waterloo, Canada
Thomaz Oliveira	CINVESTAV, Mexico
Francisco Rodríguez-Henríquez	CINVESTAV, Mexico

CATREL Workshop, September 1-2 2015

DLP on general groups

DLP on general groups

Let $\mathbb{H}$ be a cyclic group of order N with a generator g then

$$\mathbb{H} = \{g^i : 0 \leq i < N\}.$$

DLP on general groups

Let $\mathbb{H}$ be a cyclic group of order N with a generator g then

$$\mathbb{H} = \{g^i : 0 \leq i < N\}.$$

The discrete logarithm problem (DLP) in $\mathbb{H}$ consists in:

- Given: $h \in \mathbb{H}$,
- Find: $0 \leq i < N$, such that $h = g^i$.

Notation: i is the discrete logarithm of h in base g, denoted $\log_g h$.

DLP on general groups

Let $\mathbb{H}$ be a cyclic group of order N with a generator g then

$$\mathbb{H} = \{g^i : 0 \leq i < N\}.$$

The discrete logarithm problem (DLP) in $\mathbb{H}$ consists in:

- Given: $h \in \mathbb{H}$,
- Find: $0 \leq i < N$, such that $h = g^i$.

Notation: i is the discrete logarithm of h in base g, denoted $\log_g h$.

For the general case where we don't know very specific structures on $\mathbb{H}$, this problem is believed to be hard (exponential run time in the size of N).

Symmetric bilinear pairings

Symmetric bilinear pairings

- $(\mathbb{G}, +)$, $(\mathbb{G}_T, \cdot)$, cyclic groups of order $|\mathbb{G}| = |\mathbb{G}_T| = r$.

Symmetric bilinear pairings

- $(\mathbb{G}, +)$, $(\mathbb{G}_T, \cdot)$, cyclic groups of order $|\mathbb{G}| = |\mathbb{G}_T| = r$.
- A symmetric bilinear pairing on $(\mathbb{G}, \mathbb{G}_T)$ is a map

$$\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T,$$

  such that

  - $\hat{e}(P, P) \neq 1$ for $P \neq 0_{\mathbb{G}}$,
  - $\hat{e}(Q_1 + Q_2, R) = \hat{e}(Q_1, R) \cdot \hat{e}(Q_2, R)$,
  - $\hat{e}(Q, R_1 + R_2) = \hat{e}(Q, R_1) \cdot \hat{e}(Q, R_2)$.

Symmetric bilinear pairings

- $(\mathbb{G}, +)$, $(\mathbb{G}_T, \cdot)$, cyclic groups of order $|\mathbb{G}| = |\mathbb{G}_T| = r$.
- A symmetric bilinear pairing on $(\mathbb{G}, \mathbb{G}_T)$ is a map

$$\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T,$$

  such that

  - $\hat{e}(P, P) \neq 1$ for $P \neq 0_{\mathbb{G}}$,
  - $\hat{e}(Q_1 + Q_2, R) = \hat{e}(Q_1, R) \cdot \hat{e}(Q_2, R)$,
  - $\hat{e}(Q, R_1 + R_2) = \hat{e}(Q, R_1) \cdot \hat{e}(Q, R_2)$.

  For cryptographic purpose, we want it to be efficiently computable.

Symmetric bilinear pairings

- $(\mathbb{G}, +)$, $(\mathbb{G}_T, \cdot)$, cyclic groups of order $|\mathbb{G}| = |\mathbb{G}_T| = r$.
- A symmetric bilinear pairing on $(\mathbb{G}, \mathbb{G}_T)$ is a map

$$\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T,$$

  such that

  - $\hat{e}(P, P) \neq 1$ for $P \neq 0_{\mathbb{G}}$,
  - $\hat{e}(Q_1 + Q_2, R) = \hat{e}(Q_1, R) \cdot \hat{e}(Q_2, R)$,
  - $\hat{e}(Q, R_1 + R_2) = \hat{e}(Q, R_1) \cdot \hat{e}(Q, R_2)$.

  For cryptographic purpose, we want it to be efficiently computable.

- Immediate property: for any two integers k_1 and k_2,

$$\hat{e}(k_1 Q, k_2 R) = \hat{e}(Q, R)^{k_1 k_2}.$$

Type 1 pairing-based cryptography

A Type 1 paring means that we have

Type 1 pairing-based cryptography

A Type 1 paring means that we have

- $\mathbb{G}$ is a subgroup of prime order r of either

Type 1 pairing-based cryptography

A Type 1 paring means that we have

- $\mathbb{G}$ is a subgroup of prime order r of either
  - $E(\mathbb{F}_q)$, the group of rational points of an elliptic curve E; or

Type 1 pairing-based cryptography

A Type 1 paring means that we have

- $\mathbb{G}$ is a subgroup of prime order r of either
  - $E(\mathbb{F}_q)$, the group of rational points of an elliptic curve E; or
  - $\text{Jac}_C(\mathbb{F}_q)$, the jacobian of a genus-2 hyperelliptic curve C.

Type 1 pairing-based cryptography

A Type 1 paring means that we have

- $\mathbb{G}$ is a subgroup of prime order r of either
  - $E(\mathbb{F}_q)$, the group of rational points of an elliptic curve E; or
  - $\mathrm{Jac}_C(\mathbb{F}_q)$, the jacobian of a genus-2 hyperelliptic curve C.
- $\mathbb{G}_T$ is the subgroup of order r of $\mathbb{F}_{q^k}^*$,
  - k is the embedding degree of $\mathbb{G}$, that is the smallest positive integer k such that $r|(q^k - 1)$.

Type 1 pairing-based cryptography

A Type 1 paring means that we have

- $\mathbb{G}$ is a subgroup of prime order r of either
  - $E(\mathbb{F}_q)$, the group of rational points of an elliptic curve E; or
  - $\mathrm{Jac}_C(\mathbb{F}_q)$, the jacobian of a genus-2 hyperelliptic curve C.
- $\mathbb{G}_T$ is the subgroup of order r of $\mathbb{F}_{q^k}^*$,
  - k is the embedding degree of $\mathbb{G}$, that is the smallest positive integer k such that $r|(q^k - 1)$.
- Used pairing maps:
  - Weil pairings.
  - Tate pairings and modifications (Eta, Ate ...).

Main Type 1 pairings

Most interesting small characteristic Type 1 pairings:

Main Type 1 pairings

Most interesting small characteristic Type 1 pairings:

- The $k = 4$ pairings derived from supersingular elliptic curves over $\mathbb{F}_{2^n}$:
  - $Y^2 + Y = X^3 + X$; and
  - $Y^2 + Y = X^3 + X + 1$.

Main Type 1 pairings

Most interesting small characteristic Type 1 pairings:

- The $k = 4$ pairings derived from supersingular elliptic curves over $\mathbb{F}_{2^n}$:
  - $Y^2 + Y = X^3 + X$; and
  - $Y^2 + Y = X^3 + X + 1$.
- The $k = 6$ pairings derived from supersingular elliptic curves over $\mathbb{F}_{3^n}$:
  - $Y^2 = X^3 - X + 1$; and
  - $Y^2 = X^3 - X - 1$.

Main Type 1 pairings

Most interesting small characteristic Type 1 pairings:

- The $k = 4$ pairings derived from supersingular elliptic curves over $\mathbb{F}_{2^n}$:
  - $Y^2 + Y = X^3 + X$; and
  - $Y^2 + Y = X^3 + X + 1$.
- The $k = 6$ pairings derived from supersingular elliptic curves over $\mathbb{F}_{3^n}$:
  - $Y^2 = X^3 - X + 1$; and
  - $Y^2 = X^3 - X - 1$.
- The $k = 12$ pairing derived from supersingular gen.-2 curves over $\mathbb{F}_{2^n}$:
  - $Y^2 + Y = X^5 + X^3$; and
  - $Y^2 + Y = X^5 + X^3 + 1$.

Example of protocols

- **Identity-based non-interactive key exchange**
  - Sakai-Oghishi-Kasahara, 2000.
- **One-round three-party** key agreement
  - Joux, 2000.
- **Identity-based encryption**
  - Boneh–Franklin, 2001.
  - Sakai–Kasahara, 2001.
- **Short digital signatures**
  - Boneh–Lynn–Shacham, 2001.
  - Zang–Safavi-Naini–Susilo, 2004.
- ...

The MOV attack

The elliptic (hyperelliptic) curve discrete logarithm problem (ECDLP) is believed to be hard in genus-1 and 2 curves (exponential complexity).

The MOV attack

The elliptic (hyperelliptic) curve discrete logarithm problem (ECDLP) is believed to be hard in genus-1 and 2 curves (exponential complexity).

Reduction attack on supersingular elliptic curves:

- Menezes-Okamoto-Vanstone (1993), Frey-Rück (1994)

$$\begin{array}{ccc} \mathrm{DLP}_{\mathbb{G}} & <_{\mathrm{P}} & \mathrm{DLP}_{\mathbb{G}_T} \\ dP & \longrightarrow & \hat{e}(dP, P) = \hat{e}(P, P)^d. \end{array}$$

The MOV attack

The elliptic (hyperelliptic) curve discrete logarithm problem (ECDLP) is believed to be hard in genus-1 and 2 curves (exponential complexity).

Reduction attack on supersingular elliptic curves:

- Menezes-Okamoto-Vanstone (1993), Frey-Rück (1994)

$$\begin{array}{ccc} \mathrm{DLP}_{\mathbb{G}} & <_{\mathrm{P}} & \mathrm{DLP}_{\mathbb{G}_T} \\ dP & \longrightarrow & \hat{e}(dP, P) = \hat{e}(P, P)^{d}. \end{array}$$

- For cryptographic applications on pairings over supersingular curves:
  - The embedding degree is relatively small.
  - Require the DLP in $\mathbb{G}_T$ to be hard.

Algorithm for small characteristic DLP in $\mathbb{F}_Q$

Fastest general-purpose algorithm: Coppersmith (1984) of subexponential run time $L_Q[\frac{1}{3}, 1.526]$, where $L_Q[\alpha, c]$ with $0 < \alpha < 1$ and $c > 0$ denotes

$$L_Q[\alpha, c] = e^{[c+o(1)](\log Q)^{\alpha}(\log\log Q)^{1-\alpha}} = (\log Q)^{[c+o(1)]\left(\frac{\log Q}{\log\log Q}\right)^{\alpha}}.$$

Algorithm for small characteristic DLP in $\mathbb{F}_Q$

Fastest general-purpose algorithm: Coppersmith (1984) of subexponential run time $L_Q[\frac{1}{3}, 1.526]$, where $L_Q[\alpha, c]$ with $0 < \alpha < 1$ and $c > 0$ denotes

$$L_Q[\alpha, c] = e^{[c+o(1)](\log Q)^{\alpha}(\log\log Q)^{1-\alpha}} = (\log Q)^{[c+o(1)]\left(\frac{\log Q}{\log\log Q}\right)^{\alpha}}.$$

Table: Believed security for supersingular curves till 2012

Base field ($\mathbb{F}q$)	$\mathbb{F}_{2^n}$	$\mathbb{F}_{3^n}$	$\mathbb{F}_{2^n}$
Embedding degree (k)	4	6	12
Lower security ($\approx 2^{64}$)	$n = 239$	$n = 97$	$n = 71$
Medium security ($\approx 2^{80}$)	$n = 373$	$n = 163$	$n = 127$
Higher security ($\approx 2^{128}$)	$n = 1223$	$n = 509$	$n = 367$

Joux-Lercier Algorithm for $\mathbb{F}_Q = \mathbb{F}_{q^n}$

In 2006, Joux and Lercier presented an algorithm with running time $L_Q[\frac{1}{3}, 1.442]$ when q and n are 'balanced'

$$q = L_Q[1/3, 3^{-2/3}], \quad n = 3^{2/3} \cdot (\log Q/(\log\log Q))^{2/3}.$$

Joux-Lercier Algorithm for $\mathbb{F}_Q = \mathbb{F}_{q^n}$

In 2006, Joux and Lercier presented an algorithm with running time $L_Q[\frac{1}{3}, 1.442]$ when q and n are 'balanced'

$$q = L_Q[1/3, 3^{-2/3}], \quad n = 3^{2/3} \cdot (\log Q/(\log\log Q))^{2/3}.$$

In 2012, Shinohara-Shimoyama-Hayashi-Takagi analyzed the Joux-Lercier algorithm applied to base fields $\mathbb{F}_{3^n}$ for the case $k = 6$:

Base Field $\mathbb{F}_{3^n}$	$n = 97$	$n = 163$	$n = 509$
Security level	$2^{52.79}$	$2^{68.17}$	$2^{111.35}$

Joux-Lercier Algorithm for $\mathbb{F}_Q = \mathbb{F}_{q^n}$

In 2006, Joux and Lercier presented an algorithm with running time $L_Q[\frac{1}{3}, 1.442]$ when q and n are 'balanced'

$$q = L_Q[1/3, 3^{-2/3}], \quad n = 3^{2/3} \cdot (\log Q/(\log\log Q))^{2/3}.$$

In 2012, Shinohara-Shimoyama-Hayashi-Takagi analyzed the Joux-Lercier algorithm applied to base fields $\mathbb{F}_{3^n}$ for the case $k = 6$:

Base Field $\mathbb{F}_{3^n}$	$n = 97$	$n = 163$	$n = 509$
Security level	$2^{52.79}$	$2^{68.17}$	$2^{111.35}$

That same year, they solved the DLP in the order 923-bit subgroup of $\mathbb{F}_{3^{97}}$ in 103.74 CPU years (using 252 CPU cores).

Joux-Lercier Algorithm for $\mathbb{F}_Q = \mathbb{F}_{q^n}$

In 2006, Joux and Lercier presented an algorithm with running time $L_Q[\frac{1}{3}, 1.442]$ when q and n are 'balanced'

$$q = L_Q[1/3, 3^{-2/3}], \quad n = 3^{2/3} \cdot (\log Q/(\log\log Q))^{2/3}.$$

In 2012, Shinohara-Shimoyama-Hayashi-Takagi analyzed the Joux-Lercier algorithm applied to base fields $\mathbb{F}_{3^n}$ for the case $k = 6$:

Base Field $\mathbb{F}_{3^n}$	$n = 97$	$n = 163$	$n = 509$
Security level	$2^{52.79}$	$2^{68.17}$	$2^{111.35}$

That same year, they solved the DLP in the order 923-bit subgroup of $\mathbb{F}_{3^{97}}$ in 103.74 CPU years (using 252 CPU cores).

Later in 2012, Joux introduced a "pinpointing" technique that improved the Joux-Lercier algorithm to $L_Q[\frac{1}{3}, 0.961]$.

2013's advances

Let $Q = q^{dn}$, with q a power of 2 or 3, $n \approx q$ and d a small integer

2013's advances

Let $Q = q^{dn}$, with q a power of 2 or 3, $n \approx q$ and d a small integer

- **Feb 2013** - Joux:

$$L_Q[\frac{1}{4} + o(1), c].$$

2013's advances

Let $Q = q^{dn}$, with q a power of 2 or 3, $n \approx q$ and d a small integer

- **Feb 2013** - Joux:
$$L_Q[\frac{1}{4} + o(1), c].$$
- **Feb, May 2013** - Göloğlu, Granger, McGuire and Zumbrägel: ideas somewhat similar to Joux's.

2013's advances

Let $Q = q^{dn}$, with q a power of 2 or 3, $n \approx q$ and d a small integer

- **Feb 2013** - Joux:

$$L_Q[\frac{1}{4} + o(1), c].$$

- **Feb, May 2013** - Göloğlu, Granger, McGuire and Zumbrägel: ideas somewhat similar to Joux's.

Subsequent records

- Apr 2013 - Göloğlu et al. solve DLP in $\mathbb{F}^*_{2^{6120}} = \mathbb{F}^*_{(2^8)^{3\cdot 255}}$ in 750 CPU hours.

- May 2013 - Joux solves DLP in $\mathbb{F}^*_{2^{6168}} = \mathbb{F}^*_{(2^8)^{3\cdot 257}}$ in 550 CPU hours.

2013's advances

Let $Q = q^{dn}$, with q a power of 2 or 3, $n \approx q$ and d a small integer

- **Feb 2013** - Joux:
$$L_Q[\frac{1}{4} + o(1), c].$$
- **Feb, May 2013** - Göloğlu, Granger, McGuire and Zumbrägel: ideas somewhat similar to Joux's.

Subsequent records

- Apr 2013 - Göloğlu et al. solve DLP in $\mathbb{F}^*_{2^{6120}} = \mathbb{F}^*_{(2^8)^{3\cdot 255}}$ in 750 CPU hours.
- May 2013 - Joux solves DLP in $\mathbb{F}^*_{2^{6168}} = \mathbb{F}^*_{(2^8)^{3\cdot 257}}$ in 550 CPU hours.

Kummer/twisted Kummer extensions: $\mathbb{F}_{q^n}$ with $n \mid q \mp 1$.

Overview on the Joux 2013 algorithm

Select polynomials $h_0, h_1 \in \mathbb{F}_{q^d}[X]$ such that

- degree of h_0 and h_1 is at most δ, a small positive integer.
- $X^q \cdot h_1 - h_0$ has a degree-n irreducible factor I_X in $\mathbb{F}_{q^d}[X]$.

Then $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}(x) = \mathbb{F}_{q^d}[X]/(I_X)$ and $x^q = \frac{h_0(x)}{h_1(x)}$.

Overview on the Joux 2013 algorithm

Select polynomials $h_0, h_1 \in \mathbb{F}_{q^d}[X]$ such that

- degree of h_0 and h_1 is at most δ, a small positive integer.
- $X^q \cdot h_1 - h_0$ has a degree-n irreducible factor I_X in $\mathbb{F}_{q^d}[X]$.

Then $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}(x) = \mathbb{F}_{q^d}[X]/(I_X)$ and $x^q = \frac{h_0(x)}{h_1(x)}$.

Let $g \in \mathbb{F}_{q^{dn}}^*$ be a generator, and let $h \in \mathbb{F}_{q^{dn}}^*$.

Compute $\log_g h$:

Overview on the Joux 2013 algorithm

Select polynomials $h_0, h_1 \in \mathbb{F}_{q^d}[X]$ such that

- degree of h_0 and h_1 is at most δ, a small positive integer.
- $X^q \cdot h_1 - h_0$ has a degree-n irreducible factor I_X in $\mathbb{F}_{q^d}[X]$.

Then $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}(x) = \mathbb{F}_{q^d}[X]/(I_X)$ and $x^q = \frac{h_0(x)}{h_1(x)}$.

Let $g \in \mathbb{F}^*_{q^{dn}}$ be a generator, and let $h \in \mathbb{F}^*_{q^{dn}}$.

Compute $\log_g h$:

- Factor base computation: find logarithms of all degree-1 elements (and degree-2 if $d = 2$) in $\mathbb{F}_{q^{dn}}$ in polynomial time.

Overview on the Joux 2013 algorithm

Select polynomials $h_0, h_1 \in \mathbb{F}_{q^d}[X]$ such that

- degree of h_0 and h_1 is at most δ, a small positive integer.
- $X^q \cdot h_1 - h_0$ has a degree-n irreducible factor I_X in $\mathbb{F}_{q^d}[X]$.

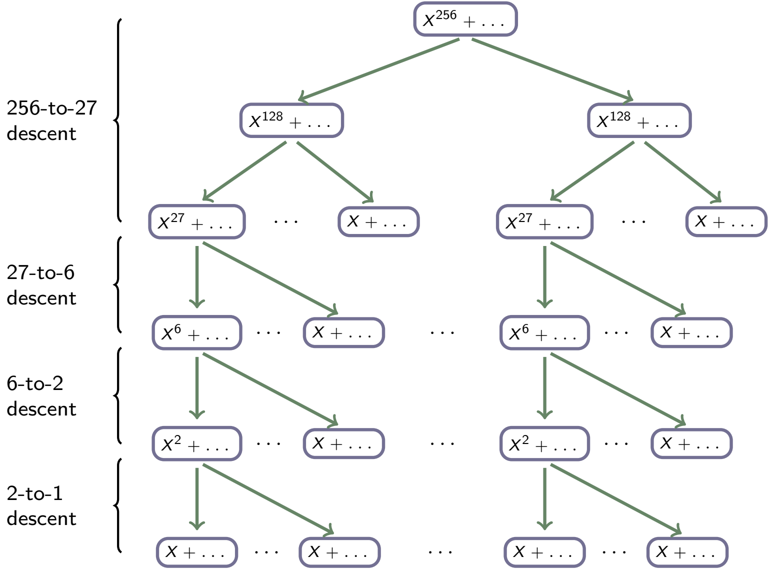
Then $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}(x) = \mathbb{F}_{q^d}[X]/(I_X)$ and $x^q = \frac{h_0(x)}{h_1(x)}$.

Let $g \in \mathbb{F}^*_{q^{dn}}$ be a generator, and let $h \in \mathbb{F}^*_{q^{dn}}$.

Compute $\log_g h$:

- Factor base computation: find logarithms of all degree-1 elements (and degree-2 if $d = 2$) in $\mathbb{F}_{q^{dn}}$ in polynomial time.
- Descent stage: $\log_g h$ is expressed as a linear combination of logs of elements in the factor base using classical methods and a new descent method (based on solving multivariate bilinear equations).

Descent Steps in $\mathbb{F}_{2^{8\cdot 3\cdot 257}}$



QPA: a much faster algorithm

Let $Q = q^{2n}$, with q a power of 2 or 3 and $n \leq q + 2$.

Jun 2013 - Barbulescu, Gaudry, Joux and Thomé:

- quasi-polynomial time algorithm (QPA):

$$(\log Q)^{O(\log \log Q)} \approx (L_Q[-1, c])^{(\log Q)}.$$

QPA: a much faster algorithm

Let $Q = q^{2n}$, with q a power of 2 or 3 and $n \leq q + 2$.

Jun 2013 - Barbulescu, Gaudry, Joux and Thomé:

- quasi-polynomial time algorithm (QPA):

$$(\log Q)^{O(\log \log Q)} \approx (L_Q[-1, c])^{(\log Q)}.$$

- asymptotically smaller than $L_Q[\alpha, c]$, for any $\alpha > 0$ and $c > 0$.

QPA: a much faster algorithm

Let $Q = q^{2n}$, with q a power of 2 or 3 and $n \leq q + 2$.

Jun 2013 - Barbulescu, Gaudry, Joux and Thomé:

- quasi-polynomial time algorithm (QPA):

$$(\log Q)^{O(\log \log Q)} \approx (L_Q[-1, c])^{(\log Q)}.$$

- asymptotically smaller than $L_Q[\alpha, c]$, for any $\alpha > 0$ and $c > 0$.
- same setup as in Joux's algorithm:

QPA: a much faster algorithm

Let $Q = q^{2n}$, with q a power of 2 or 3 and $n \leq q + 2$.

Jun 2013 - Barbulescu, Gaudry, Joux and Thomé:

- quasi-polynomial time algorithm (QPA):

$$(\log Q)^{O(\log \log Q)} \approx (L_Q[-1, c])^{(\log Q)}.$$

- asymptotically smaller than $L_Q[\alpha, c]$, for any $\alpha > 0$ and $c > 0$.
- same setup as in Joux's algorithm:
    - Factor base computation: find logarithms of linears in polynomial time.

QPA: a much faster algorithm

Let $Q = q^{2n}$, with q a power of 2 or 3 and $n \leq q + 2$.

Jun 2013 - Barbulescu, Gaudry, Joux and Thomé:

- quasi-polynomial time algorithm (QPA):

$$(\log Q)^{O(\log \log Q)} \approx (L_Q[-1, c])^{(\log Q)}.$$

- asymptotically smaller than $L_Q[\alpha, c]$, for any $\alpha > 0$ and $c > 0$.
- same setup as in Joux's algorithm:
    - Factor base computation: find logarithms of linears in polynomial time.
    - Descent stage: $\log_g h$ is expressed as a linear combination of logs of elements in the factor base using a descent strategy quite similar to Joux's method for computing logarithms of degree-2 elements.

QPA: a much faster algorithm

Let $Q = q^{2n}$, with q a power of 2 or 3 and $n \leq q + 2$.

Jun 2013 - Barbulescu, Gaudry, Joux and Thomé:

- quasi-polynomial time algorithm (QPA):

$$(\log Q)^{O(\log \log Q)} \approx (L_Q[-1, c])^{(\log Q)}.$$

- asymptotically smaller than $L_Q[\alpha, c]$, for any $\alpha > 0$ and $c > 0$.
- same setup as in Joux's algorithm:
  - Factor base computation: find logarithms of linears in polynomial time.
  - Descent stage: $\log_g h$ is expressed as a linear combination of logs of elements in the factor base using a descent strategy quite similar to Joux's method for computing logarithms of degree-2 elements.

[We have another QPA by Granger-Kleinjung-Zumbrägel from April 2014.]

First contributions

We combined the Joux 2013 algorithm and QPA to show in a concrete analysis that the cryptographic DLP in the field $\mathbb{F}_{3^{6\cdot 509}}$ can be computed much faster than previously believed:

First contributions

We combined the Joux 2013 algorithm and QPA to show in a concrete analysis that the cryptographic DLP in the field $\mathbb{F}_{3^{6\cdot 509}}$ can be computed much faster than previously believed:

DLP algorithm	Coppersmith04	JL06	Joux12	Joux13-QPA13
Run time	2^{128}	2^{111}	2^{103}	2^{75}

First contributions

We combined the Joux 2013 algorithm and QPA to show in a concrete analysis that the cryptographic DLP in the field $\mathbb{F}_{3^{6\cdot 509}}$ can be computed much faster than previously believed:

DLP algorithm	Coppersmith04	JL06	Joux12	Joux13-QPA13
Run time	2^{128}	2^{111}	2^{103}	2^{75}

We also analyzed the cryptographic DLP in the field $\mathbb{F}_{2^{12\cdot 367}}$ and found the new algorithms more effective (much more parallelizable) than the Joux 2012 algorithm:

DLP algorithm	Coppersmith04	Joux12	Joux13-QPA13
Run time	2^{128}	2^{92}	2^{95}

First contributions

We combined the Joux 2013 algorithm and QPA to show in a concrete analysis that the cryptographic DLP in the field $\mathbb{F}_{3^{6\cdot 509}}$ can be computed much faster than previously believed:

DLP algorithm	Coppersmith04	JL06	Joux12	Joux13-QPA13
Run time	2^{128}	2^{111}	2^{103}	2^{75}

We also analyzed the cryptographic DLP in the field $\mathbb{F}_{2^{12\cdot 367}}$ and found the new algorithms more effective (much more parallelizable) than the Joux 2012 algorithm:

DLP algorithm	Coppersmith04	Joux12	Joux13-QPA13
Run time	2^{128}	2^{92}	2^{95}

Our preliminary analysis suggested that the new algorithms have no effect in computing discrete logs in $\mathbb{F}_{2^{4\cdot 1223}}$.

First contributions

We combined the Joux 2013 algorithm and QPA to show in a concrete analysis that the cryptographic DLP in the field $\mathbb{F}_{3^{6\cdot 509}}$ can be computed much faster than previously believed:

DLP algorithm	Coppersmith04	JL06	Joux12	Joux13-QPA13
Run time	2^{128}	2^{111}	2^{103}	2^{75}

We also analyzed the cryptographic DLP in the field $\mathbb{F}_{2^{12\cdot 367}}$ and found the new algorithms more effective (much more parallelizable) than the Joux 2012 algorithm:

DLP algorithm	Coppersmith04	Joux12	Joux13-QPA13
Run time	2^{128}	2^{92}	2^{95}

Our preliminary analysis suggested that the new algorithms have no effect in computing discrete logs in $\mathbb{F}_{2^{4\cdot 1223}}$. [Incredibly optimistic!]

A new polynomial representation

In ECC 2013, Granger presented (joint work with Zumbrägel) a modification of Joux's field representation:

- $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}[X]/(I_X)$ with I_X dividing $X \cdot h_1(X^q) - h_0(X^q)$.
- $h_0(X), h_1(X)$ polynomials over $\mathbb{F}_q[X]$ of small degree δ.

A new polynomial representation

In ECC 2013, Granger presented (joint work with Zumbrägel) a modification of Joux's field representation:

- $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}[X]/(I_X)$ with I_X dividing $X \cdot h_1(X^q) - h_0(X^q)$.
- $h_0(X), h_1(X)$ polynomials over $\mathbb{F}_q[X]$ of small degree δ.

Very useful: allows to have $n \leq \delta \cdot q$ instead of necessarily $n \leq q + \delta$.

A new polynomial representation

In ECC 2013, Granger presented (joint work with Zumbrägel) a modification of Joux's field representation:

- $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}[X]/(I_X)$ with I_X dividing $X \cdot h_1(X^q) - h_0(X^q)$.
- $h_0(X), h_1(X)$ polynomials over $\mathbb{F}_q[X]$ of small degree δ.

Very useful: allows to have $n \le \delta \cdot q$ instead of necessarily $n \le q + \delta$.

Resulting analysis:

DLP algorithm	Coppersmith04	Joux13-QPA13 (as analyzed by A. et al)	GZ13
Run time on $\mathbb{F}_{2^{12\cdot 367}}$	2^{128}	2^{95}	2^{76}
Run time on $\mathbb{F}_{2^{4\cdot 1223}}$	2^{128}	$\geq 2^{128}$	2^{95}

A new polynomial representation

In ECC 2013, Granger presented (joint work with Zumbrägel) a modification of Joux's field representation:

- $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}[X]/(I_X)$ with I_X dividing $X \cdot h_1(X^q) - h_0(X^q)$.
- $h_0(X), h_1(X)$ polynomials over $\mathbb{F}_q[X]$ of small degree δ.

Very useful: allows to have $n \leq \delta \cdot q$ instead of necessarily $n \leq q + \delta$.

Resulting analysis:

DLP algorithm	Coppersmith04	Joux13-QPA13 (as analyzed by A. et al)	GZ13
Run time on $\mathbb{F}_{2^{12 \cdot 367}}$	2^{128}	2^{95}	2^{76}
Run time on $\mathbb{F}_{2^{4 \cdot 1223}}$	2^{128}	$\geq 2^{128}$	2^{95}

In December 2013, we used the Granger-Zumbrägel representation to show that the cryptographic DLP in $\mathbb{F}_{3^{6 \cdot 1429}}$ and $\mathbb{F}_{2^{4 \cdot 3041}}$ can be solved in time 2^{96} and 2^{129}, respectively. [Initially believed to enjoy a 2^{192} security level.]

Solving cryptographic DLP using Magma

- January 27 2014, A.-Menezes-Oliveira-Rodríguez: $\mathbb{F}_{3^{6\cdot 137}}$.

Solving cryptographic DLP using Magma

- January 27 2014, A.-Menezes-Oliveira-Rodríguez: $\mathbb{F}_{3^{6\cdot 137}}$.
  - We used Joux's algorithm with the Granger-Zumbrägel representation to break the supersingular curve $E : y^2 = x^3 - x + 1$ defined over $\mathbb{F}_{3^{137}}$.

Solving cryptographic DLP using Magma

- January 27 2014, A.-Menezes-Oliveira-Rodríguez: $\mathbb{F}_{3^{6\cdot 137}}$.
  - We used Joux's algorithm with the Granger-Zumbrägel representation to break the supersingular curve $E : y^2 = x^3 - x + 1$ defined over $\mathbb{F}_{3^{137}}$.
  - $\mathbb{F}_{3^{6\cdot 137}}$: 2^{76} against Coppersmith. [we actually embed $\mathbb{F}_{3^{6\cdot 137}}$ in $\mathbb{F}_{3^{4\cdot 3\cdot 137}}$.]

Solving cryptographic DLP using Magma

- January 27 2014, A.-Menezes-Oliveira-Rodríguez: $\mathbb{F}_{3^{6\cdot 137}}$.
  - We used Joux's algorithm with the Granger-Zumbrägel representation to break the supersingular curve $E : y^2 = x^3 - x + 1$ defined over $\mathbb{F}_{3^{137}}$.
  - $\mathbb{F}_{3^{6\cdot 137}}$: 2^{76} against Coppersmith. [we actually embed $\mathbb{F}_{3^{6\cdot 137}}$ in $\mathbb{F}_{3^{4\cdot 3\cdot 137}}$.]
  - Field of size of 1303-bit and we worked in a 155-bit prime order subgroup. [previous record on characteristic 3: the 923-bit field $\mathbb{F}_{3^{6\cdot 97}}$.]

Solving cryptographic DLP using Magma

- January 27 2014, A.-Menezes-Oliveira-Rodríguez: $\mathbb{F}_{3^{6\cdot 137}}$.
  - We used Joux's algorithm with the Granger-Zumbrägel representation to break the supersingular curve $E : y^2 = x^3 - x + 1$ defined over $\mathbb{F}_{3^{137}}$.
  - $\mathbb{F}_{3^{6\cdot 137}}$: 2^{76} against Coppersmith. [we actually embed $\mathbb{F}_{3^{6\cdot 137}}$ in $\mathbb{F}_{3^{4\cdot 3\cdot 137}}$.]
  - Field of size of 1303-bit and we worked in a 155-bit prime order subgroup. [previous record on characteristic 3: the 923-bit field $\mathbb{F}_{3^{6\cdot 97}}$.]
  - Main issue: only 50% of the degree-2 polynomial descended.
    Fix: adapt an idea of Coppersmith and employ a Joux and Göloğlu et al. strategy to get 97% of the quadratics descending while avoiding the remainder in the descent phase.

Solving cryptographic DLP using Magma

- January 27 2014, A.-Menezes-Oliveira-Rodríguez: $\mathbb{F}_{3^{6\cdot 137}}$.
  - We used Joux's algorithm with the Granger-Zumbrägel representation to break the supersingular curve $E : y^2 = x^3 - x + 1$ defined over $\mathbb{F}_{3^{137}}$.
  - $\mathbb{F}_{3^{6\cdot 137}}$: 2^{76} against Coppersmith. [we actually embed $\mathbb{F}_{3^{6\cdot 137}}$ in $\mathbb{F}_{3^{4\cdot 3\cdot 137}}$.]
  - Field of size of 1303-bit and we worked in a 155-bit prime order subgroup. [previous record on characteristic 3: the 923-bit field $\mathbb{F}_{3^{6\cdot 97}}$.]
  - Main issue: only 50% of the degree-2 polynomial descended. Fix: adapt an idea of Coppersmith and employ a Joux and Göloğlu et al. strategy to get 97% of the quadratics descending while avoiding the remainder in the descent phase.
  - Run time: 888 CPU hours. [previous record: 896313 CPU hours]

Solving cryptographic DLP using Magma

- January 27 2014, A.-Menezes-Oliveira-Rodríguez: $\mathbb{F}_{3^{6\cdot 137}}$.
  - We used Joux's algorithm with the Granger-Zumbrägel representation to break the supersingular curve $E : y^2 = x^3 - x + 1$ defined over $\mathbb{F}_{3^{137}}$.
  - $\mathbb{F}_{3^{6\cdot 137}}$: 2^{76} against Coppersmith. [we actually embed $\mathbb{F}_{3^{6\cdot 137}}$ in $\mathbb{F}_{3^{4\cdot 3\cdot 137}}$.]
  - Field of size of 1303-bit and we worked in a 155-bit prime order subgroup. [previous record on characteristic 3: the 923-bit field $\mathbb{F}_{3^{6\cdot 97}}$.]
  - Main issue: only 50% of the degree-2 polynomial descended.
    Fix: adapt an idea of Coppersmith and employ a Joux and Göloğlu et al. strategy to get 97% of the quadratics descending while avoiding the remainder in the descent phase.
  - Run time: 888 CPU hours. [previous record: 896313 CPU hours]
  - First computations of discrete logarithms in a cryptographic finite field using the new algorithms.

Practical improvements

- January 30 2014, Granger-Kleinjung-Zumbrägel: $\mathbb{F}_{2^{12\cdot 367}}$, $\mathbb{F}_{2^{4\cdot 1223}}$

Practical improvements

- January 30 2014, Granger-Kleinjung-Zumbrägel: $\mathbb{F}_{2^{12\cdot 367}}$, $\mathbb{F}_{2^{4\cdot 1223}}$

DLP algorithm	Copp.04	Joux13-QPA13 (as analyzed by A. et al)	GZ13	GKZ14a
Run time on $\mathbb{F}_{2^{12\cdot 367}}$	2^{128}	2^{95}	2^{76}	2^{48}
Run time on $\mathbb{F}_{2^{4\cdot 1223}}$	2^{128}	$\geq 2^{128}$	2^{95}	2^{59}

Practical improvements

- January 30 2014, Granger-Kleinjung-Zumbrägel: $\mathbb{F}_{2^{12\cdot 367}}$, $\mathbb{F}_{2^{4\cdot 1223}}$

DLP algorithm	Copp.04	Joux13-QPA13 (as analyzed by A. et al)	GZ13	GKZ14a
Run time on $\mathbb{F}_{2^{12\cdot 367}}$	2^{128}	2^{95}	2^{76}	2^{48}
Run time on $\mathbb{F}_{2^{4\cdot 1223}}$	2^{128}	$\geq 2^{128}$	2^{95}	2^{59}

- Always start trying to descend an element down to elements of smaller degree staying in the same field.

Practical improvements

- January 30 2014, Granger-Kleinjung-Zumbrägel: $\mathbb{F}_{2^{12\cdot 367}}$, $\mathbb{F}_{2^{4\cdot 1223}}$

DLP algorithm	Copp.04	Joux13-QPA13 (as analyzed by A. et al)	GZ13	GKZ14a
Run time on $\mathbb{F}_{2^{12\cdot 367}}$	2^{128}	2^{95}	2^{76}	2^{48}
Run time on $\mathbb{F}_{2^{4\cdot 1223}}$	2^{128}	$\geq 2^{128}$	2^{95}	2^{59}

  - Always start trying to descend an element down to elements of smaller degree staying in the same field.
  - Only when this is not possible, promote it into an extension field where it may split into smaller elements.

Practical improvements

- January 30 2014, Granger-Kleinjung-Zumbrägel: $\mathbb{F}_{2^{12\cdot 367}}$, $\mathbb{F}_{2^{4\cdot 1223}}$

DLP algorithm	Copp.04	Joux13-QPA13 (as analyzed by A. et al)	GZ13	GKZ14a
Run time on $\mathbb{F}_{2^{12\cdot 367}}$	2^{128}	2^{95}	2^{76}	2^{48}
Run time on $\mathbb{F}_{2^{4\cdot 1223}}$	2^{128}	$\geq\ 2^{128}$	2^{95}	2^{59}

- Always start trying to descend an element down to elements of smaller degree staying in the same field.
- Only when this is not possible, promote it into an extension field where it may split into smaller elements.
- Not necessary to embed $\mathbb{F}_{q^n}$ into larger extensions whenever $q \approx \delta \cdot n$, for some small integer δ.

Practical improvements

- January 30 2014, Granger-Kleinjung-Zumbrägel: $\mathbb{F}_{2^{12\cdot 367}}$, $\mathbb{F}_{2^{4\cdot 1223}}$

DLP algorithm	Copp.04	Joux13-QPA13 (as analyzed by A. et al)	GZ13	GKZ14a
Run time on $\mathbb{F}_{2^{12\cdot 367}}$	2^{128}	2^{95}	2^{76}	2^{48}
Run time on $\mathbb{F}_{2^{4\cdot 1223}}$	2^{128}	$\geq\ 2^{128}$	2^{95}	2^{59}

- Always start trying to descend an element down to elements of smaller degree staying in the same field.
- Only when this is not possible, promote it into an extension field where it may split into smaller elements.
- Not necessary to embed $\mathbb{F}_{q^n}$ into larger extensions whenever $q \approx \delta \cdot n$, for some small integer δ.
- Discrete logarithm computation in the cryptographic subgroup of $\mathbb{F}_{2^{12\cdot 367}}$ in 52,240 CPU hours.

More improvements

- September 15 2014, Joux and Pierrot: $\mathbb{F}_{3^{5\cdot 479}}$.

  Solving DLP in $\mathbb{F}_{q^n}$ when $q \approx \delta \cdot n$ for some small integer δ

More improvements

- September 15 2014, Joux and Pierrot: $\mathbb{F}_{3^{5 \cdot 479}}$.

  Solving DLP in $\mathbb{F}_{q^n}$ when $q \approx \delta \cdot n$ for some small integer δ

  - The descent phase works exactly as in [GKZ14a].

More improvements

- September 15 2014, Joux and Pierrot: $\mathbb{F}_{3^{5\cdot479}}$.

  Solving DLP in $\mathbb{F}_{q^n}$ when $q \approx \delta \cdot n$ for some small integer δ

  - The descent phase works exactly as in [GKZ14a].
  - Compute the logarithms of degree-1 and degree-2 by solving one linear algebra in time $O(q^5)$.

More improvements

- September 15 2014, Joux and Pierrot: $\mathbb{F}_{3^{5\cdot 479}}$.

  Solving DLP in $\mathbb{F}_{q^n}$ when $q \approx \delta \cdot n$ for some small integer δ

  - The descent phase works exactly as in [GKZ14a].
  - Compute the logarithms of degree-1 and degree-2 by solving one linear algebra in time $O(q^5)$.
  - Compute the logarithms of degree-3 elements solving q linear algebras in time $O(q^6)$.

More improvements

- September 15 2014, Joux and Pierrot: $\mathbb{F}_{3^{5\cdot 479}}$.

  Solving DLP in $\mathbb{F}_{q^n}$ when $q \approx \delta \cdot n$ for some small integer δ

  - The descent phase works exactly as in [GKZ14a].
  - Compute the logarithms of degree-1 and degree-2 by solving one linear algebra in time $O(q^5)$.
  - Compute the logarithms of degree-3 elements solving q linear algebras in time $O(q^6)$.
  - Compute the logarithms of elements in a degree-4 family solving q linear algebras in time $O(q^6)$ and the logarithms of some other degree-4 families of smaller size.

More improvements

- September 15 2014, Joux and Pierrot: $\mathbb{F}_{3^{5 \cdot 479}}$.

  Solving DLP in $\mathbb{F}_{q^n}$ when $q \approx \delta \cdot n$ for some small integer δ

  - The descent phase works exactly as in [GKZ14a].
  - Compute the logarithms of degree-1 and degree-2 by solving one linear algebra in time $O(q^5)$.
  - Compute the logarithms of degree-3 elements solving q linear algebras in time $O(q^6)$.
  - Compute the logarithms of elements in a degree-4 family solving q linear algebras in time $O(q^6)$ and the logarithms of some other degree-4 families of smaller size.
  - Discrete logarithm computation in the cryptographic subgroup of $\mathbb{F}_{3^{5 \cdot 479}}$ in time 8,600 CPU hours.

More improvements

- September 15 2014, Joux and Pierrot: $\mathbb{F}_{3^{5\cdot 479}}$.

  Solving DLP in $\mathbb{F}_{q^n}$ when $q \approx \delta \cdot n$ for some small integer δ

  - The descent phase works exactly as in [GKZ14a].
  - Compute the logarithms of degree-1 and degree-2 by solving one linear algebra in time $O(q^5)$.
  - Compute the logarithms of degree-3 elements solving q linear algebras in time $O(q^6)$.
  - Compute the logarithms of elements in a degree-4 family solving q linear algebras in time $O(q^6)$ and the logarithms of some other degree-4 families of smaller size.
  - Discrete logarithm computation in the cryptographic subgroup of $\mathbb{F}_{3^{5\cdot 479}}$ in time 8,600 CPU hours.
  - Current record in characteristic three.

Current computations

A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez: $\mathbb{F}_{3^{6\cdot 509}}$

- Want to break the field $\mathbb{F}_{3^{6\cdot 509}}$ of initial proposed security 2^{128}.

Current computations

A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez: $\mathbb{F}_{3^{6\cdot 509}}$

- Want to break the field $\mathbb{F}_{3^{6\cdot 509}}$ of initial proposed security 2^{128}.
- Use JP14 factor base comput. method + GKZ14a descent strategy.

Current computations

A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez: $\mathbb{F}_{3^{6\cdot 509}}$

- Want to break the field $\mathbb{F}_{3^{6\cdot 509}}$ of initial proposed security 2^{128}.
- Use JP14 factor base comput. method + GKZ14a descent strategy.

DLP algorithm	Copp.04	JL06	Joux12	Joux13-QPA13	JP14-GKZ14.
Run time	2^{128}	2^{111}	2^{103}	2^{75}	2^{49}.

Current computations

A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez: $\mathbb{F}_{3^{6\cdot 509}}$

- Want to break the field $\mathbb{F}_{3^{6\cdot 509}}$ of initial proposed security 2^{128}.
- Use JP14 factor base comput. method + GKZ14a descent strategy.

DLP algorithm	Copp.04	JL06	Joux12	Joux13-QPA13	JP14-GKZ14.
Run time	2^{128}	2^{111}	2^{103}	2^{75}	2^{49}.

- Computation of logarithms of degree-1, 2, 3 elements: already done 378163 CPU hours in a cluster of 5096 cores (Abacus Cinvestav).

Current computations

A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez: $\mathbb{F}_{3^{6\cdot 509}}$

- Want to break the field $\mathbb{F}_{3^{6\cdot 509}}$ of initial proposed security 2^{128}.
- Use JP14 factor base comput. method + GKZ14a descent strategy.

DLP algorithm	Copp.04	JL06	Joux12	Joux13-QPA13	JP14-GKZ14.
Run time	2^{128}	2^{111}	2^{103}	2^{75}	2^{49}.

- Computation of logarithms of degree-1, 2, 3 elements: already done 378163 CPU hours in a cluster of 5096 cores (Abacus Cinvestav).
- Descent of a challenge element 508-to-15: already done 619413 CPU hours using about 300 cores.

Current computations

A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez: $\mathbb{F}_{3^{6\cdot 509}}$

- Want to break the field $\mathbb{F}_{3^{6\cdot 509}}$ of initial proposed security 2^{128}.
- Use JP14 factor base comput. method + GKZ14a descent strategy.

DLP algorithm	Copp.04	JL06	Joux12	Joux13-QPA13	JP14-GKZ14.
Run time	2^{128}	2^{111}	2^{103}	2^{75}	2^{49}.

- Computation of logarithms of degree-1, 2, 3 elements: already done 378163 CPU hours in a cluster of 5096 cores (Abacus Cinvestav).
- Descent of a challenge element 508-to-15: already done 619413 CPU hours using about 300 cores.
- Main issue: management of small degree elements during the descent (billions of nodes expected).

Future work

Future work

- Get our own C implementation of Faugère's F4 or F5 algorithm.

Future work

- Get our own C implementation of Faugère's F4 or F5 algorithm.
- Improve our C implementation for solving linear algebra systems.

Future work

- Get our own C implementation of Faugère's F4 or F5 algorithm.
- Improve our C implementation for solving linear algebra systems.
- Find a polynomial-time algorithm for DLP in $\mathbb{F}_{2^n}$ or $\mathbb{F}_{3^n}$.

Future work

- Get our own C implementation of Faugère's F4 or F5 algorithm.
- Improve our C implementation for solving linear algebra systems.
- Find a polynomial-time algorithm for DLP in $\mathbb{F}_{2^n}$ or $\mathbb{F}_{3^n}$.
  [It's just a dream!]

Future work

- Get our own C implementation of Faugère's F4 or F5 algorithm.
- Improve our C implementation for solving linear algebra systems.
- Find a polynomial-time algorithm for DLP in $\mathbb{F}_{2^n}$ or $\mathbb{F}_{3^n}$.
  [It's just a dream!]

Thanks For Your Attention!