The end of paring-based cryptography using small characteristic finite fields

Gora Adj [1]

In joint work with

Isaac Canales-Martínez [1], Nareli Cruz-Cortés [2], Alfred Menezes [3], Thomaz Oliveira [1], Luis Rivera-Zamarripa [2] and Francisco Rodríguez-Henríquez [1]

[1] CINVESTAV-IPN, Mexico
Departmento de Computación

[2] Instituto Politécnico Nacional, Mexico
Centro de investigación en Computación

[3] University of Waterloo, Canada
Department of Combinatorics and Optimization

Seminario de Computación Científica, UNAM - November 3, 2016

Outline

1. Pairing-Based Cryptography
2. Small-Characteristic Pairings
3. Security of Small-Characteristic Pairings
4. 2013's Advances
5. First Contributions
6. More Improvements
7. The 509's Computations
8. DLP at The 192-bit Security Level

Pairing-Based Cryptography

Symmetric bilinear pairings

Symmetric bilinear pairings

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

Symmetric bilinear pairings

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

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

Symmetric bilinear pairings

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

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

  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 prime order $|\mathbb{G}| = |\mathbb{G}_T| = r$.
- A symmetric bilinear pairing on $(\mathbb{G}, \mathbb{G}_T)$ is a mapping

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

  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 $\hat{e}$ to be efficiently computable.

Symmetric bilinear pairings

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

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

  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 $\hat{e}$ to be efficiently computable.

- Immediate property: for any integer k,

$$\hat{e}(kQ, R) = \hat{e}(Q, R)^k = \hat{e}(Q, kR).$$

Examples of pairing-based protocols

- **Identity-based non-interactive key exchange**
  - Sakai-Oghishi-Kasahara, 2000.

Examples of pairing-based protocols

- **Identity-based non-interactive key exchange**
  - Sakai-Oghishi-Kasahara, 2000.
- **One-round three-party key agreement**
  - Joux, 2000.

Examples of pairing-based 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.

Examples of pairing-based 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.

Small-Characteristic Pairings

Small-characteristic pairing-based cryptography

In these parings, we have $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where:

Small-characteristic pairing-based cryptography

In these parings, we have $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, where:

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

Small-characteristic pairing-based cryptography

In these parings, we have $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where:

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

Small-characteristic pairing-based cryptography

In these parings, we have $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where:

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

Small-characteristic pairing-based cryptography

In these parings, we have $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where:

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

  [$p = 2, 3$ and n is a prime.]

Small-characteristic pairing-based cryptography

In these parings, we have $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where:

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

  [$p = 2, 3$ and n is a prime.]
- $\mathbb{G}_T$ is the subgroup of order r of $\mathbb{F}^*_{p^{kn}}$,
  - k is the embedding degree of $\mathbb{G}$, that is the smallest positive integer k such that $r | (p^{kn} - 1)$.

Small-characteristic pairing-based cryptography

In these parings, we have $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where:

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

  [$p = 2, 3$ and n is a prime.]
- $\mathbb{G}_T$ is the subgroup of order r of $\mathbb{F}^*_{p^{kn}}$,
  - k is the embedding degree of $\mathbb{G}$, that is the smallest positive integer k such that $r|(p^{kn} - 1)$.

Most common pairing maps:

- Weil pairings.
- Tate pairings and modifications (Eta, Ate, ...).

Primary small-characteristic pairings

The most interesting small-characteristic:

Primary small-characteristic pairings

The most interesting small-characteristic:

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

Primary small-characteristic pairings

The most interesting small-characteristic:

- 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
  - $Y^2 = X^3 - X - 1$.

Primary small-characteristic pairings

The most interesting small-characteristic:

- 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
  - $Y^2 + Y = X^5 + X^3 + 1$.

Security of Small-Characteristic Pairings (Prior to 2013)

Discrete logarithm problem (ECDLP, DLP)

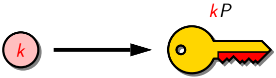
- Let $(\mathbb{G}, +)$ be a subgroup of prime order r of an elliptic or hyperelliptic curve and let $P \in \mathbb{G}$.

Discrete logarithm problem (ECDLP, DLP)

- Let $(\mathbb{G}, +)$ be a subgroup of prime order r of an elliptic or hyperelliptic curve and let $P \in \mathbb{G}$.

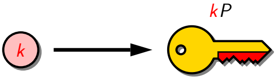
Discrete logarithm problem (ECDLP, DLP)

- Let $(\mathbb{G}, +)$ be a subgroup of prime order r of an elliptic or hyperelliptic curve and let $P \in \mathbb{G}$.



Discrete logarithm problem (ECDLP, DLP)

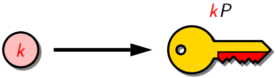
- Let $(\mathbb{G}, +)$ be a subgroup of prime order r of an elliptic or hyperelliptic curve and let $P \in \mathbb{G}$.



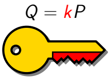
ECDLP: given $Q \in \mathbb{G}$, compute $0 \leq k < r$ such that $Q = kP$.

Discrete logarithm problem (ECDLP, DLP)

- Let $(\mathbb{G}, +)$ be a subgroup of prime order r of an elliptic or hyperelliptic curve and let $P \in \mathbb{G}$.

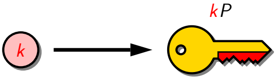


ECDLP: given $Q \in \mathbb{G}$, compute $0 \leq k < r$ such that $Q = kP$.

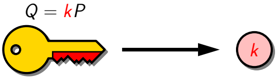


Discrete logarithm problem (ECDLP, DLP)

- Let $(\mathbb{G}, +)$ be a subgroup of prime order r of an elliptic or hyperelliptic curve and let $P \in \mathbb{G}$.

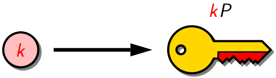


ECDLP: given $Q \in \mathbb{G}$, compute $0 \leq k < r$ such that $Q = kP$.

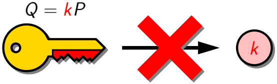


Discrete logarithm problem (ECDLP, DLP)

- Let $(\mathbb{G}, +)$ be a subgroup of prime order r of an elliptic or hyperelliptic curve and let $P \in \mathbb{G}$.

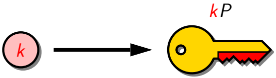


ECDLP: given $Q \in \mathbb{G}$, compute $0 \leq k < r$ such that $Q = kP$.

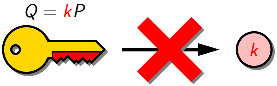


Discrete logarithm problem (ECDLP, DLP)

- Let $(\mathbb{G}, +)$ be a subgroup of prime order r of an elliptic or hyperelliptic curve and let $P \in \mathbb{G}$.



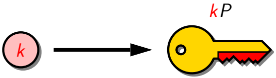
ECDLP: given $Q \in \mathbb{G}$, compute $0 \leq k < r$ such that $Q = kP$.



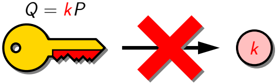
- Let $(\mathbb{G}_T, \cdot)$ be a subgroup of order r in a finite field. Let $g \in \mathbb{G}_T$.

Discrete logarithm problem (ECDLP, DLP)

- Let $(\mathbb{G}, +)$ be a subgroup of prime order r of an elliptic or hyperelliptic curve and let $P \in \mathbb{G}$.



ECDLP: given $Q \in \mathbb{G}$, compute $0 \leq k < r$ such that $Q = kP$.

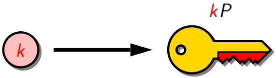


- Let $(\mathbb{G}_T, \cdot)$ be a subgroup of order r in a finite field. Let $g \in \mathbb{G}_T$.

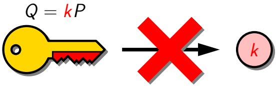
$$\mathbb{G}_T = \{g^i : 0 \leq i < r\}.$$

Discrete logarithm problem (ECDLP, DLP)

- Let $(\mathbb{G}, +)$ be a subgroup of prime order r of an elliptic or hyperelliptic curve and let $P \in \mathbb{G}$.



ECDLP: given $Q \in \mathbb{G}$, compute $0 \leq k < r$ such that $Q = kP$.



- Let $(\mathbb{G}_T, \cdot)$ be a subgroup of order r in a finite field. Let $g \in \mathbb{G}_T$.

$$\mathbb{G}_T = \{g^i : 0 \leq i < r\}.$$

DLP: given $h \in \mathbb{G}_T$, find $0 \leq i < r$ such that $h = g^i$.

The MOV attack

The ECDLP is known to be hard.

- The best general-purpose algorithm is of complexity exponential.

The MOV attack

The ECDLP is known to be hard.

- The best general-purpose algorithm is of complexity exponential.

However, efficient problem reductions exist:

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

$$\mathrm{ECDLP}_{\mathbb{G}} \quad <_{\text{Polynomial time}} \quad \mathrm{DLP}_{\mathbb{G}_T}$$

$$dP \quad \longrightarrow \quad \hat{e}(dP, P) = \hat{e}(P, P)^d.$$

The MOV attack

The ECDLP is known to be hard.

- The best general-purpose algorithm is of complexity exponential.

However, efficient problem reductions exist:

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

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

- Then the DLP in $\mathbb{F}_{q^k}$ is also required to be hard.

The MOV attack

The ECDLP is known to be hard.

- The best general-purpose algorithm is of complexity exponential.

However, efficient problem reductions exist:

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

$$\text{ECDLP}_{\mathbb{G}} \quad <_{\text{Polynomial time}} \quad \text{DLP}_{\mathbb{G}_T}$$

$$dP \quad \longrightarrow \quad \hat{e}(dP, P) = \hat{e}(P, P)^d.$$

- Then the DLP in $\mathbb{F}_{q^k}$ is also required to be hard.
- For pairing-based cryptography over supersingular curves:

The MOV attack

The ECDLP is known to be hard.

- The best general-purpose algorithm is of complexity exponential.

However, efficient problem reductions exist:

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

$$\text{ECDLP}_{\mathbb{G}} \quad <_{\text{Polynomial time}} \quad \text{DLP}_{\mathbb{G}_T}$$

$$dP \quad \longrightarrow \quad \hat{e}(dP, P) = \hat{e}(P, P)^d.$$

- Then the DLP in $\mathbb{F}_{q^k}$ is also required to be hard.
- For pairing-based cryptography over supersingular curves:
  - The embedding degree is relatively small ($k = 4$, 6, or 12).

The MOV attack

The ECDLP is known to be hard.

- The best general-purpose algorithm is of complexity exponential.

However, efficient problem reductions exist:

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

$$\text{ECDLP}_{\mathbb{G}} \quad <_{\text{Polynomial time}} \quad \text{DLP}_{\mathbb{G}_T}$$

$$dP \quad \longrightarrow \quad \hat{e}(dP, P) = \hat{e}(P, P)^d.$$

- Then the DLP in $\mathbb{F}_{q^k}$ is also required to be hard.
- For pairing-based cryptography over supersingular curves:
  - The embedding degree is relatively small ($k = 4$, 6, or 12).
  - So, the finite field $\mathbb{F}_{q^k}$ (containing $\mathbb{G}_T$) is not very large.

DLP algorithms for small-characteristic fields $\mathbb{F}_Q$

- Subexponential running time, for $0 < \alpha < 1$ and $c > 0$, at input Q:

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

DLP algorithms for small-characteristic fields $\mathbb{F}_Q$

- Subexponential running time, for $0 < \alpha < 1$ and $c > 0$, at input Q:

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

- Coppersmith's algorithm [Coppersmith84] of complexity $L_Q[\frac{1}{3}, 1.526]$ is the fastest general-purpose algorithm for solving the DLP in $\mathbb{F}_Q$:

DLP algorithms for small-characteristic fields $\mathbb{F}_Q$

- Subexponential running time, for $0 < \alpha < 1$ and $c > 0$, at input Q:

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

- Coppersmith's algorithm [Coppersmith84] of complexity $L_Q[\frac{1}{3}, 1.526]$ is the fastest general-purpose algorithm for solving the DLP in $\mathbb{F}_Q$:

Table: Security of small-characteristic parings as in 2012 (DLP in $\mathbb{F}_{p^{kn}}$)

Underlying field ($\mathbb{F}p^n$)	$\mathbb{F}_{2^n}$	$\mathbb{F}_{3^n}$	$\mathbb{F}_{2^n}$
Embedding degree (k)	4	6	12
Lower security ($\approx 2^{64}$)	$n = 239$	$n = 97$	$n = 127$
Medium security ($\approx 2^{80}$)	$n = 373$	$n = 163$	$n = 163$
Standard security ($\approx 2^{128}$)	$n = 1223$	$n = 509$	$n = 367$
Higher security ($\approx 2^{192}$)	$n = 3041$	$n = 1429$	$n \approx 983$

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

- In 2006, Joux and Lercier [JL06] 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 [JL06] 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 et al. [SSHT12]
  - analyzed the [JL06] algorithm to estimate:

Extension Field $\mathbb{F}_{3^{6 \cdot n}}$	$n = 97$	$n = 163$	$n = 509$
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 [JL06] 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 et al. [SSHT12]
  - analyzed the [JL06] algorithm to estimate:

Extension Field $\mathbb{F}_{3^{6\cdot n}}$	$n = 97$	$n = 163$	$n = 509$
Security level	$2^{52.79}$	$2^{68.17}$	$2^{111.35}$

  - solved the DLP in the field $\mathbb{F}_{3^{6\cdot 97}}$ in 103.74 CPU years.

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

- In 2006, Joux and Lercier [JL06] 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 et al. [SSHT12]
  - analyzed the [JL06] algorithm to estimate:

Extension Field $\mathbb{F}_{3^{6 \cdot n}}$	$n = 97$	$n = 163$	$n = 509$
Security level	$2^{52.79}$	$2^{68.17}$	$2^{111.35}$

  - solved the DLP in the field $\mathbb{F}_{3^{6 \cdot 97}}$ in 103.74 CPU years.

- Later in 2012, Joux [Joux12] introduced a technique that improved the [JL06] algorithm to $L_Q[\frac{1}{3}, 0.961]$.

2013's Advances

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, May 2013** - Joux [Joux13]:
  - presented an algorithm of complexity $L_Q[\frac{1}{4} + o(1), c]$.

2013's advances

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

- **Feb, May 2013** - Joux [Joux13]:
  - presented an algorithm of complexity $L_Q[\frac{1}{4} + o(1), c]$.
  - solved the 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, May 2013** - Joux [Joux13]:
  - presented an algorithm of complexity $L_Q[\frac{1}{4} + o(1), c]$.
  - solved the DLP in $\mathbb{F}^*_{2^{6168}} = \mathbb{F}^*_{(2^8)^{3\cdot 257}}$ in 550 CPU hours.
- **Feb, Apr 2013** - Göloğlu-Granger-McGuire-Zumbrägel:
  - presented 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, May 2013** - Joux [Joux13]:
  - presented an algorithm of complexity $L_Q[\frac{1}{4} + o(1), c]$.
  - solved the DLP in $\mathbb{F}^*_{2^{6168}} = \mathbb{F}^*_{(2^8)^{3 \cdot 257}}$ in 550 CPU hours.
- **Feb, Apr 2013** - Göloğlu-Granger-McGuire-Zumbrägel:
  - presented ideas somewhat similar to Joux's.
  - solved DLP in $\mathbb{F}^*_{2^{6120}} = \mathbb{F}^*_{(2^8)^{3 \cdot 255}}$ in 750 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, May 2013** - Joux [Joux13]:
  - presented an algorithm of complexity $L_Q[\frac{1}{4} + o(1), c]$.
  - solved the DLP in $\mathbb{F}^*_{2^{6168}} = \mathbb{F}^*_{(2^8)^{3\cdot 257}}$ in 550 CPU hours.
- **Feb, Apr 2013** - Göloğlu-Granger-McGuire-Zumbrägel:
  - presented ideas somewhat similar to Joux's.
  - solved DLP in $\mathbb{F}^*_{2^{6120}} = \mathbb{F}^*_{(2^8)^{3\cdot 255}}$ in 750 CPU hours.
- **Jun 2013** - Barbulescu-Gaudry-Joux-Thomé:
  - Quasi-polynomial time algorithm (QPA) when $d = 2$:

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

2013's advances

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

- **Feb, May 2013** - Joux [Joux13]:
  - presented an algorithm of complexity $L_Q[\frac{1}{4} + o(1), c]$.
  - solved the DLP in $\mathbb{F}^*_{2^{6168}} = \mathbb{F}^*_{(2^8)^{3\cdot 257}}$ in 550 CPU hours.
- **Feb, Apr 2013** - Göloğlu-Granger-McGuire-Zumbrägel:
  - presented ideas somewhat similar to Joux's.
  - solved DLP in $\mathbb{F}^*_{2^{6120}} = \mathbb{F}^*_{(2^8)^{3\cdot 255}}$ in 750 CPU hours.
- **Jun 2013** - Barbulescu-Gaudry-Joux-Thomé:
  - Quasi-polynomial time algorithm (QPA) when $d = 2$:

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

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

First Contributions

Cryptographic impact

2013-2014, A.-Menezes-Oliveira-Rodríguez

- We combined Joux's algorithm and the QPA to show that the DLP in the cryptographic field $\mathbb{F}_{3^{6\cdot 509}}$ can be computed much faster than previously:
  - 2^{75} operations vs. 2^{128} for Coppersmith.

Cryptographic impact

2013-2014, A.-Menezes-Oliveira-Rodríguez

- We combined Joux's algorithm and the QPA to show that the DLP in the cryptographic field $\mathbb{F}_{3^{6\cdot 509}}$ can be computed much faster than previously:
  - 2^{75} operations vs. 2^{128} for Coppersmith.
- We also analyzed the cryptographic DLP in the field $\mathbb{F}_{2^{12\cdot 367}}$ and found the new algorithms more effective than previously:
  - 2^{95} operations effectively parallelizable vs. 2^{92} for Joux 2012.

Cryptographic impact

2013-2014, A.-Menezes-Oliveira-Rodríguez

- We combined Joux's algorithm and the QPA to show that the DLP in the cryptographic field $\mathbb{F}_{3^{6\cdot 509}}$ can be computed much faster than previously:
  - 2^{75} operations vs. 2^{128} for Coppersmith.
- We also analyzed the cryptographic DLP in the field $\mathbb{F}_{2^{12\cdot 367}}$ and found the new algorithms more effective than previously:
  - 2^{95} operations effectively parallelizable vs. 2^{92} for Joux 2012.
- We used Granger-Zumbrägel's field representation (ECC 2013) to:
  - show that the DLP in cryptographic fields $\mathbb{F}_{3^{6\cdot 1429}}$ and $\mathbb{F}_{2^{4\cdot 3041}}$ can be solved in time 2^{96} and 2^{129}, respectively, vs. 2^{192} for Coppersmith.

Cryptographic impact

2013-2014, A.-Menezes-Oliveira-Rodríguez

- We combined Joux's algorithm and the QPA to show that the DLP in the cryptographic field $\mathbb{F}_{3^{6\cdot 509}}$ can be computed much faster than previously:
  - 2^{75} operations vs. 2^{128} for Coppersmith.
- We also analyzed the cryptographic DLP in the field $\mathbb{F}_{2^{12\cdot 367}}$ and found the new algorithms more effective than previously:
  - 2^{95} operations effectively parallelizable vs. 2^{92} for Joux 2012.
- We used Granger-Zumbrägel's field representation (ECC 2013) to:
  - show that the DLP in cryptographic fields $\mathbb{F}_{3^{6\cdot 1429}}$ and $\mathbb{F}_{2^{4\cdot 3041}}$ can be solved in time 2^{96} and 2^{129}, respectively, vs. 2^{192} for Coppersmith.
  - solve the DLP in the 155 and 259-bit prime subgroups of $\mathbb{F}^*_{3^{6\cdot 137}}$ and $\mathbb{F}^*_{3^{6\cdot 137}}$ within 888 and 1201 CPU hours, respectively.

Overview on the Joux' algorithm: DLP in $\mathbb{F}_{q^{dn}}$

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

Overview on the Joux' algorithm: DLP in $\mathbb{F}_{q^{dn}}$

- Select polynomials $h_0, h_1 \in \mathbb{F}_{q^d}[X]$ such that
  - degree of h_0 and h_1 is at most 2, a small positive integer.

Overview on the Joux' algorithm: DLP in $\mathbb{F}_{q^{dn}}$

- Select polynomials $h_0, h_1 \in \mathbb{F}_{q^d}[X]$ such that
  - degree of h_0 and h_1 is at most 2, a small positive integer.
  - $X^q \cdot h_1 - h_0$ has a degree-n irreducible factor I_X in $\mathbb{F}_{q^d}[X]$.

Overview on the Joux' algorithm: DLP in $\mathbb{F}_{q^{dn}}$

- Select polynomials $h_0, h_1 \in \mathbb{F}_{q^d}[X]$ such that
  - degree of h_0 and h_1 is at most 2, a small positive integer.
  - $X^q \cdot h_1 - h_0$ has a degree-n irreducible factor I_X in $\mathbb{F}_{q^d}[X]$.

Remark: $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}[X]/(I_X)$ and elements are seen as polynomials in $\mathbb{F}_{q^d}[X]$ of degree at most $n-1$.

Overview on the Joux' algorithm: DLP in $\mathbb{F}_{q^{dn}}$

- Select polynomials $h_0, h_1 \in \mathbb{F}_{q^d}[X]$ such that
  - degree of h_0 and h_1 is at most 2, a small positive integer.
  - $X^q \cdot h_1 - h_0$ has a degree-n irreducible factor I_X in $\mathbb{F}_{q^d}[X]$.

  Remark: $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}[X]/(I_X)$ and elements are seen as polynomials in $\mathbb{F}_{q^d}[X]$ of degree at most $n-1$.

- Let $g \in \mathbb{F}^*_{q^{dn}}$ (a linear) be a generator, and $h \in \mathbb{F}^*_{q^{dn}}$ a target element.

Overview on the Joux' algorithm: DLP in $\mathbb{F}_{q^{dn}}$

- Select polynomials $h_0, h_1 \in \mathbb{F}_{q^d}[X]$ such that
  - degree of h_0 and h_1 is at most 2, a small positive integer.
  - $X^q \cdot h_1 - h_0$ has a degree-n irreducible factor I_X in $\mathbb{F}_{q^d}[X]$.

  Remark: $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}[X]/(I_X)$ and elements are seen as polynomials in $\mathbb{F}_{q^d}[X]$ of degree at most $n-1$.

- Let $g \in \mathbb{F}^*_{q^{dn}}$ (a linear) be a generator, and $h \in \mathbb{F}^*_{q^{dn}}$ a target element.
- 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' algorithm: DLP in $\mathbb{F}_{q^{dn}}$

- Select polynomials $h_0, h_1 \in \mathbb{F}_{q^d}[X]$ such that
  - degree of h_0 and h_1 is at most 2, a small positive integer.
  - $X^q \cdot h_1 - h_0$ has a degree-n irreducible factor I_X in $\mathbb{F}_{q^d}[X]$.

Remark: $\mathbb{F}_{q^{dn}} = \mathbb{F}_{q^d}[X]/(I_X)$ and elements are seen as polynomials in $\mathbb{F}_{q^d}[X]$ of degree at most $n-1$.

- Let $g \in \mathbb{F}^*_{q^{dn}}$ (a linear) be a generator, and $h \in \mathbb{F}^*_{q^{dn}}$ a target element.
- 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).

The idea behind the descent stage

- Let $f \in \mathbb{F}_{q^d}[X]$ irreducible of degree D.

The idea behind the descent stage

- Let $f \in \mathbb{F}_{q^d}[X]$ irreducible of degree D.
- Let $(f_i)_{i \in I}$ and $(h_i)_{j \in J}$ be two families of polynomials in $\mathbb{F}_{q^d}[X]$ of degree at most $m < D$

The idea behind the descent stage

- Let $f \in \mathbb{F}_{q^d}[X]$ irreducible of degree D.
- Let $(f_i)_{i \in I}$ and $(h_i)_{j \in J}$ be two families of polynomials in $\mathbb{F}_{q^d}[X]$ of degree at most $m < D$ and $(\alpha_i)_{i \in I}$ and $(\beta_i)_{j \in J}$ two families of positive integers

The idea behind the descent stage

- Let $f \in \mathbb{F}_{q^d}[X]$ irreducible of degree D.
- Let $(f_i)_{i \in I}$ and $(h_i)_{j \in J}$ be two families of polynomials in $\mathbb{F}_{q^d}[X]$ of degree at most $m < D$ and $(\alpha_i)_{i \in I}$ and $(\beta_i)_{j \in J}$ two families of positive integers such that

$$f \cdot \prod_{i \in I} f_i^{\alpha_i} = \prod_{j \in J} h_i^{\beta_i}.$$

The idea behind the descent stage

- Let $f \in \mathbb{F}_{q^d}[X]$ irreducible of degree D.
- Let $(f_i)_{i\in I}$ and $(h_i)_{j\in J}$ be two families of polynomials in $\mathbb{F}_{q^d}[X]$ of degree at most $m < D$ and $(\alpha_i)_{i\in I}$ and $(\beta_i)_{j\in J}$ two families of positive integers such that

$$f \cdot \prod_{i\in I} f_i^{\alpha_i} = \prod_{j\in J} h_i^{\beta_i}.$$

- Then

$$\log_g f = \left(\sum_{j\in J} \alpha_i \cdot \log_g h_i\right) - \left(\sum_{i\in I} \beta_i \cdot \log_g f_i\right).$$

The idea behind the descent stage

- Let $f \in \mathbb{F}_{q^d}[X]$ irreducible of degree D.
- Let $(f_i)_{i \in I}$ and $(h_i)_{j \in J}$ be two families of polynomials in $\mathbb{F}_{q^d}[X]$ of degree at most $m < D$ and $(\alpha_i)_{i \in I}$ and $(\beta_i)_{j \in J}$ two families of positive integers such that

$$f \cdot \prod_{i \in I} f_i^{\alpha_i} = \prod_{j \in J} h_i^{\beta_i}.$$

- Then

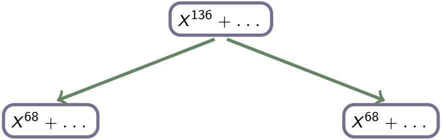
$$\log_g f = \left(\sum_{j \in J} \alpha_i \cdot \log_g h_i \right) - \left(\sum_{i \in I} \beta_i \cdot \log_g f_i \right).$$

- In this case, we say that we expressed $\log_g f$ as a linear combination of logarithms of polynomials of degree at most m.

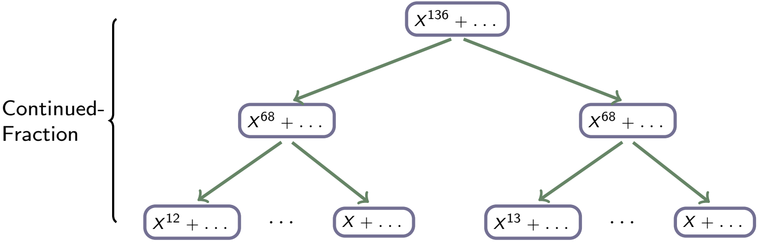
DLP in $\mathbb{F}_{3^{6\cdot 137}}$: $q = 3^4$, $d = 3$ and $n = 137$

$X^{136} + \ldots$

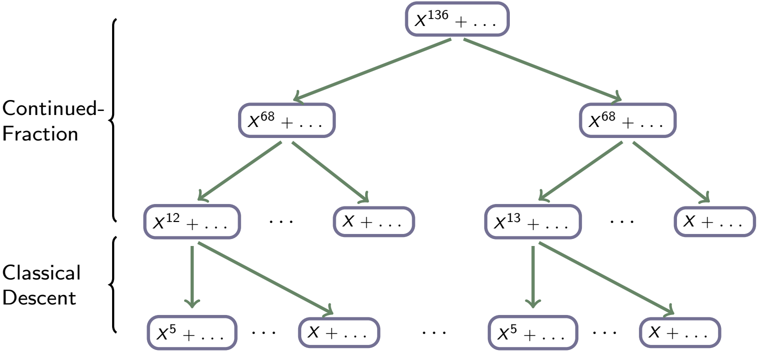
DLP in $\mathbb{F}_{3^{6\cdot 137}}$: $q = 3^4$, $d = 3$ and $n = 137$



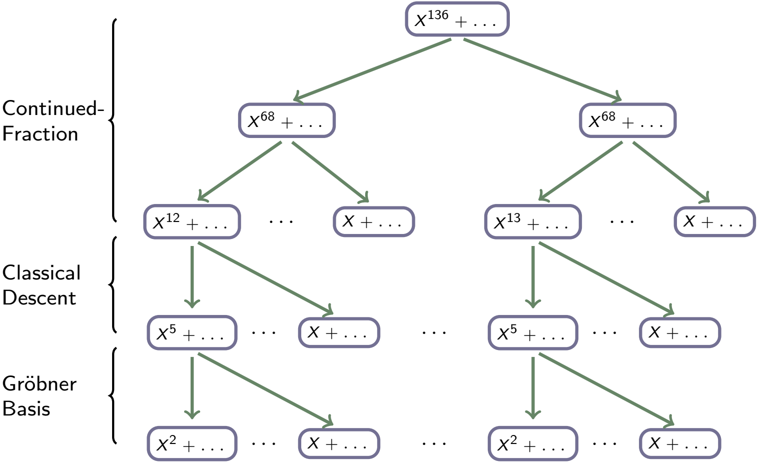
DLP in $\mathbb{F}_{3^{6\cdot 137}}$: $q = 3^4$, $d = 3$ and $n = 137$



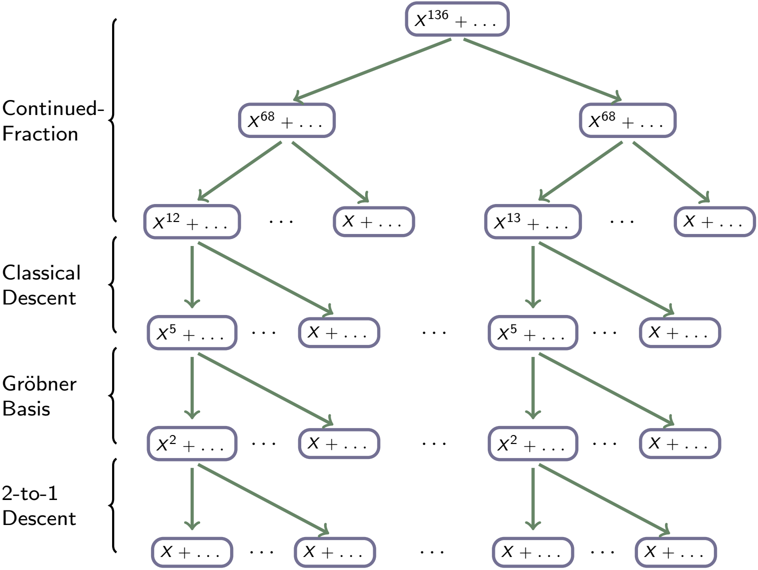
DLP in $\mathbb{F}_{3^{6\cdot 137}}$: $q = 3^4$, $d = 3$ and $n = 137$



DLP in $\mathbb{F}_{3^{6\cdot 137}}$: $q = 3^4$, $d = 3$ and $n = 137$



DLP in $\mathbb{F}_{3^{6\cdot 137}}$: $q = 3^4$, $d = 3$ and $n = 137$



More Improvements

Practical improvements

January 30 2014, Granger-Kleinjung-Zumbrägel [GKZ14]: $\mathbb{F}_{2^{12 \cdot 367}}$

Practical improvements

January 30 2014, Granger-Kleinjung-Zumbrägel [GKZ14]: $\mathbb{F}_{2^{12\cdot 367}}$

- 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 [GKZ14]: $\mathbb{F}_{2^{12\cdot 367}}$

- 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.

Practical improvements

January 30 2014, Granger-Kleinjung-Zumbrägel [GKZ14]: $\mathbb{F}_{2^{12 \cdot 367}}$

- 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.

September 15 2014, Joux and Pierrot [JP14]: $\mathbb{F}_{3^{5 \cdot 479}}$.

Practical improvements

January 30 2014, Granger-Kleinjung-Zumbrägel [GKZ14]: $\mathbb{F}_{2^{12\cdot 367}}$

- 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.

September 15 2014, Joux and Pierrot [JP14]: $\mathbb{F}_{3^{5\cdot 479}}$.

- Compute the logarithms of degree-1 and degree-2 elements by solving one linear algebra in time $O(q^5)$.

Practical improvements

January 30 2014, Granger-Kleinjung-Zumbrägel [GKZ14]: $\mathbb{F}_{2^{12\cdot 367}}$

- 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.

September 15 2014, Joux and Pierrot [JP14]: $\mathbb{F}_{3^{5\cdot 479}}$.

- Compute the logarithms of degree-1 and degree-2 elements by solving one linear algebra in time $O(q^5)$.
- Compute the logarithms of degree-3 elements and a degree-4 family elements by solving q linear algebras for each in time $O(q^6)$ and the logarithms of some other degree-4 families of smaller size.

Practical improvements

January 30 2014, Granger-Kleinjung-Zumbrägel [GKZ14]: $\mathbb{F}_{2^{12\cdot 367}}$

- 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.

September 15 2014, Joux and Pierrot [JP14]: $\mathbb{F}_{3^{5\cdot 479}}$.

- Compute the logarithms of degree-1 and degree-2 elements by solving one linear algebra in time $O(q^5)$.
- Compute the logarithms of degree-3 elements and a degree-4 family elements by solving q linear algebras for each in time $O(q^6)$ and the logarithms of some other degree-4 families of smaller size.
- Discrete logarithm computation in $\mathbb{F}_{3^{5\cdot 479}}$ within 8,600 CPU hours.

The 509's Computations

Latest computations on $\mathbb{F}_{3^{6\cdot 509}}$

July 18 2016, A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez

Latest computations on $\mathbb{F}_{3^{6\cdot 509}}$

July 18 2016, A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez

- Let $E : y^2 = x^3 - x + 1$ be the supersingular elliptic curve over $\mathbb{F}_{3^{509}}$ with $|E(\mathbb{F}_{3^{509}})| = 7r$, where $r = (3^{509} - 3^{255} + 1)/7$ is a 804-bit prime.

Latest computations on $\mathbb{F}_{3^{6\cdot 509}}$

July 18 2016, A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez

- Let $E : y^2 = x^3 - x + 1$ be the supersingular elliptic curve over $\mathbb{F}_{3^{509}}$ with $|E(\mathbb{F}_{3^{509}})| = 7r$, where $r = (3^{509} - 3^{255} + 1)/7$ is a 804-bit prime.
- We solved the DLP in the order-r subgroup of the 4404-bit field $\mathbb{F}^*_{3^{6\cdot 509}}$, initially proposed for 128-bit security.

Latest computations on $\mathbb{F}_{3^{6\cdot 509}}$

July 18 2016, A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez

- Let $E : y^2 = x^3 - x + 1$ be the supersingular elliptic curve over $\mathbb{F}_{3^{509}}$ with $|E(\mathbb{F}_{3^{509}})| = 7r$, where $r = (3^{509} - 3^{255} + 1)/7$ is a 804-bit prime.
- We solved the DLP in the order-r subgroup of the 4404-bit field $\mathbb{F}^*_{3^{6\cdot 509}}$, initially proposed for 128-bit security.
- We used the Joux-Pierrot method to compute the logarithms of elements of the factor base, i.e., the elements of degree at most 3 and a portion of 29/728 of the quartic elements.

Latest computations on $\mathbb{F}_{3^{6\cdot 509}}$

July 18 2016, A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez

- Let $E : y^2 = x^3 - x + 1$ be the supersingular elliptic curve over $\mathbb{F}_{3^{509}}$ with $|E(\mathbb{F}_{3^{509}})| = 7r$, where $r = (3^{509} - 3^{255} + 1)/7$ is a 804-bit prime.
- We solved the DLP in the order-r subgroup of the 4404-bit field $\mathbb{F}^*_{3^{6\cdot 509}}$, initially proposed for 128-bit security.
- We used the Joux-Pierrot method to compute the logarithms of elements of the factor base, i.e., the elements of degree at most 3 and a portion of $29/728$ of the quartic elements.
- To write the logarithm of a 508-degree target in terms of elements of degree ≤ 15, we employed the continued-fractions and classical descent methods.

Latest computations on $\mathbb{F}_{3^{6 \cdot 509}}$

July 18 2016, A.-Canales-Cruz-Menezes-Oliveira-Rivera-Rodríguez

- Let $E : y^2 = x^3 - x + 1$ be the supersingular elliptic curve over $\mathbb{F}_{3^{509}}$ with $|E(\mathbb{F}_{3^{509}})| = 7r$, where $r = (3^{509} - 3^{255} + 1)/7$ is a 804-bit prime.
- We solved the DLP in the order-r subgroup of the 4404-bit field $\mathbb{F}^*_{3^{6 \cdot 509}}$, initially proposed for 128-bit security.
- We used the Joux-Pierrot method to compute the logarithms of elements of the factor base, i.e., the elements of degree at most 3 and a portion of 29/728 of the quartic elements.
- To write the logarithm of a 508-degree target in terms of elements of degree ≤ 15, we employed the continued-fractions and classical descent methods.
- We used Granger-Kleinjung-Zumbrägel's techniques to have the logarithms of elements of degree ≤ 15 written in terms of logarithms of elements in the factor base.

Running-time

Computation stage	CPU time (years)		CPU frequency (GHz)
Finding logarithms of quadratic polynomials			
Relation generation	0.01	(CS Dept.)	3.20
Linear algebra	0.50	(CS Dept.)	2.40
Finding logarithms of cubic polynomials			
Relation generation	0.15	(CS Dept.)	3.20
Linear algebra	43.88	(ABACUS)	2.60
Finding logarithms of quartic polynomials			
Relation generation	4.07	(CS Dept.)	2.60
Linear algebra	96.02	(ABACUS)	2.60
Descent			
Continued-fractions (254 to 40)	51.71	(CS Dept.)	2.87
Classical (40 to 21)	9.99	(CS Dept., U Wat.)	2.66
Classical (21 to 15)	10.24	(CS Dept., U Wat.)	2.66
Gröbner bases (15 to 4)	6.27	(CS Dept., U Wat.)	3.00
Total CPU time (years)	222.81		

Table: CPU times of each stage of the discrete logarithm computation in $\mathbb{F}_{3^{6\cdot 509}}$.

Experienced problems

- Main issues:
  - We used Magma's implementation of Faugère's F4 algorithm.

Experienced problems

- Main issues:
  - We used Magma's implementation of Faugère's F4 algorithm.
  - Our Magma script for the descent phase very frequently needs to read the logarithms of quartic polynomials, of total size 618 GB.

Experienced problems

- Main issues:
  - We used Magma's implementation of Faugère's F4 algorithm.
  - Our Magma script for the descent phase very frequently needs to read the logarithms of quartic polynomials, of total size 618 GB.
  - But our machines have only 256 GB of RAM. Thus, most of the logarithms must be stored in the hard drive HD (much slower than the virtual memory VM).

Experienced problems

- Main issues:
  - We used Magma's implementation of Faugère's F4 algorithm.
  - Our Magma script for the descent phase very frequently needs to read the logarithms of quartic polynomials, of total size 618 GB.
  - But our machines have only 256 GB of RAM. Thus, most of the logarithms must be stored in the hard drive HD (much slower than the virtual memory VM). In addition, binary encoding occupies less space but Magma does not fully handle binary files.

Experienced problems

- Main issues:
  - We used Magma's implementation of Faugère's F4 algorithm.
  - Our Magma script for the descent phase very frequently needs to read the logarithms of quartic polynomials, of total size 618 GB.
  - But our machines have only 256 GB of RAM. Thus, most of the logarithms must be stored in the hard drive HD (much slower than the virtual memory VM). In addition, binary encoding occupies less space but Magma does not fully handle binary files.
  - Moreover, since many copies of the Magma code should be executed in parallel, the memory accesses quickly cause traffic congestions.

Experienced problems

- Main issues:
  - We used Magma's implementation of Faugère's F4 algorithm.
  - Our Magma script for the descent phase very frequently needs to read the logarithms of quartic polynomials, of total size 618 GB.
  - But our machines have only 256 GB of RAM. Thus, most of the logarithms must be stored in the hard drive HD (much slower than the virtual memory VM). In addition, binary encoding occupies less space but Magma does not fully handle binary files.
  - Moreover, since many copies of the Magma code should be executed in parallel, the memory accesses quickly cause traffic congestions.
- Our fixes:
  - We filled the VM with logarithms in hexadecimal encoding and the rest of the logarithms are stored in the HD in binary encoding.

Experienced problems

- Main issues:
  - We used Magma's implementation of Faugère's F4 algorithm.
  - Our Magma script for the descent phase very frequently needs to read the logarithms of quartic polynomials, of total size 618 GB.
  - But our machines have only 256 GB of RAM. Thus, most of the logarithms must be stored in the hard drive HD (much slower than the virtual memory VM). In addition, binary encoding occupies less space but Magma does not fully handle binary files.
  - Moreover, since many copies of the Magma code should be executed in parallel, the memory accesses quickly cause traffic congestions.
- Our fixes:
  - We filled the VM with logarithms in hexadecimal encoding and the rest of the logarithms are stored in the HD in binary encoding.
  - Logarithms in the VM are read by Magma and those in the HD from some C-codes called from Magma. The System V shared memory is used for the Magma-C for the interprocess communication.

DLP at The 192-bit Security Level

Guillevic's descent method (July 2016)

- Let $q = 3^6$, n a prime number and r a prime divisor of $\Phi_6(3^n)$, where $\Phi_6(X)$ is 6^{th} cyclotomic polynomial.

Guillevic's descent method (July 2016)

- Let $q = 3^6$, n a prime number and r a prime divisor of $\Phi_6(3^n)$, where $\Phi_6(X)$ is 6^{th} cyclotomic polynomial.

- Let g be a generator of $\mathbb{F}^*_{3^{6 \cdot n}}$ and h a target element of degree $n-1$.

Guillevic's descent method (July 2016)

- Let $q = 3^6$, n a prime number and r a prime divisor of $\Phi_6(3^n)$, where $\Phi_6(X)$ is 6^{th} cyclotomic polynomial.

- Let g be a generator of $\mathbb{F}^*_{3^{6 \cdot n}}$ and h a target element of degree $n-1$.

- Guillevic showed that one can expect to find two elements $h' \in \mathbb{F}^*_{3^{6 \cdot n}}$ and $v \in F^*_{3^{3n}}$, with h' of degree $\frac{n-1}{2} \leq n' \leq n-1$, such that $h' = hv$. Then

$$\log_g h' \equiv \log_g h \pmod{r}.$$

Guillevic's descent method (July 2016)

- Let $q = 3^6$, n a prime number and r a prime divisor of $\Phi_6(3^n)$, where $\Phi_6(X)$ is 6^{th} cyclotomic polynomial.

- Let g be a generator of $\mathbb{F}^*_{3^{6\cdot n}}$ and h a target element of degree $n-1$.

- Guillevic showed that one can expect to find two elements $h' \in \mathbb{F}^*_{3^{6\cdot n}}$ and $v \in F^*_{3^{3n}}$, with h' of degree $\frac{n-1}{2} \le n' \le n-1$, such that $h' = hv$. Then

$$\log_g h' \equiv \log_g h \pmod{r}.$$

- Elements h' and v are found by solving a $6(n-n') \times 3n$ linear algebra problem at a small cost.

Guillevic's descent method (July 2016)

- Let $q = 3^6$, n a prime number and r a prime divisor of $\Phi_6(3^n)$, where $\Phi_6(X)$ is 6^{th} cyclotomic polynomial.
- Let g be a generator of $\mathbb{F}^*_{3^{6 \cdot n}}$ and h a target element of degree $n-1$.
- Guillevic showed that one can expect to find two elements $h' \in \mathbb{F}^*_{3^{6 \cdot n}}$ and $v \in F^*_{3^{3n}}$, with h' of degree $\frac{n-1}{2} \leq n' \leq n-1$, such that $h' = hv$. Then
$$\log_g h' \equiv \log_g h \pmod{r}.$$
- Elements h' and v are found by solving a $6(n-n') \times 3n$ linear algebra problem at a small cost.
- For a successful $(n-1)$-to-m descent, several h' should be obtained and tested for m-smoothness.

Guillevic's descent method (July 2016)

- Let $q = 3^6$, n a prime number and r a prime divisor of $\Phi_6(3^n)$, where $\Phi_6(X)$ is 6^{th} cyclotomic polynomial.
- Let g be a generator of $\mathbb{F}^*_{3^{6\cdot n}}$ and h a target element of degree $n-1$.
- Guillevic showed that one can expect to find two elements $h' \in \mathbb{F}^*_{3^{6\cdot n}}$ and $v \in F^*_{3^{3n}}$, with h' of degree $\frac{n-1}{2} \leq n' \leq n-1$, such that $h' = hv$. Then
$$\log_g h' \equiv \log_g h \pmod{r}.$$
- Elements h' and v are found by solving a $6(n-n') \times 3n$ linear algebra problem at a small cost.
- For a successful $(n-1)$-to-m descent, several h' should be obtained and tested for m-smoothness.
- In our case, we choose n' so that $3^{6n'-3n} \gg q^{n'}/N_q(m,n')$, where $N_q(m,n')$ denotes the number of monic m-smooth degree-n' polynomials in $\mathbb{F}_q[X]$.

Discrete logarithms in $\mathbb{F}_{3^{6\cdot 1429}}$

- $E : Y^2 = X^3 - X - 1$ a supersingular elliptic curve over $\mathbb{F}_3$. $|E(\mathbb{F}_{3^{1429}})| = cr$, where $c = 7622150170693$ and $r = (3^{1429} - 3^{715} + 1)/c$, a 2223-bit prime.

Discrete logarithms in $\mathbb{F}_{3^{6\cdot 1429}}$

- $E : Y^2 = X^3 - X - 1$ a supersingular elliptic curve over $\mathbb{F}_3$. $|E(\mathbb{F}_{3^{1429}})| = cr$, where $c = 7622150170693$ and $r = (3^{1429} - 3^{715} + 1)/c$, a 2223-bit prime.
- The Weil and Tate pairings can be used to embed the order-r subgroup of $E(\mathbb{F}_{3^{1429}})$ in the multiplicative group of the 13590-bit field $\mathbb{F}_{3^{6\cdot 1429}}$.

Discrete logarithms in $\mathbb{F}_{3^{6\cdot 1429}}$

- $E : Y^2 = X^3 - X - 1$ a supersingular elliptic curve over $\mathbb{F}_3$. $|E(\mathbb{F}_{3^{1429}})| = cr$, where $c = 7622150170693$ and $r = (3^{1429} - 3^{715} + 1)/c$, a 2223-bit prime.
- The Weil and Tate pairings can be used to embed the order-r subgroup of $E(\mathbb{F}_{3^{1429}})$ in the multiplicative group of the 13590-bit field $\mathbb{F}_{3^{6\cdot 1429}}$.
- For g a generator of $\mathbb{F}^*_{3^{6\cdot 1429}}$ and h a target of degree 1428, we estimated the cost of finding $x = \log_g h \bmod r$ at $2^{63.4} M_q$.

Discrete logarithms in $\mathbb{F}_{3^{6\cdot 1429}}$

- $E : Y^2 = X^3 - X - 1$ a supersingular elliptic curve over $\mathbb{F}_3$. $|E(\mathbb{F}_{3^{1429}})| = cr$, where $c = 7622150170693$ and $r = (3^{1429} - 3^{715} + 1)/c$, a 2223-bit prime.
- The Weil and Tate pairings can be used to embed the order-r subgroup of $E(\mathbb{F}_{3^{1429}})$ in the multiplicative group of the 13590-bit field $\mathbb{F}_{3^{6\cdot 1429}}$.
- For g a generator of $\mathbb{F}^*_{3^{6\cdot 1429}}$ and h a target of degree 1428, we estimated the cost of finding $x = \log_g h \bmod r$ at $2^{63.4} M_q$.

Finding logarithms of quadratic polynomials	
Degree 1 and 2	$2^{50.7}$
Degree 3	$2^{56.9}$
Degree 4 (36/728)	$2^{56.3}$
Descent	
Guillevic (1428 to 71)	$2^{62.4}$
Classical (71 to 32)	$2^{61.8}$
Classical (31 to $\{1, \ldots, 16, 18, 20, 22, 24, 28, 32\}$)	$2^{59.2}$
Small degree ($\{5, \ldots, 16, 18, 20, 22, 24, 28, 32\}$ to 4)	$2^{60.0}$
Total cost	$2^{63.4}$

Feasibility of computing discrete logarithms in $\mathbb{F}_{3^{6\cdot 1429}}$

- We assume that we have access to a 9000-core cluster $\mathcal{A}$, where each core has access to 16 gigabytes of shared RAM, such as ABACUS-Cinvestav.

Feasibility of computing discrete logarithms in $\mathbb{F}_{3^{6 \cdot 1429}}$

- We assume that we have access to a 9000-core cluster $\mathcal{A}$, where each core has access to 16 gigabytes of shared RAM, such as ABACUS-Cinvestav.
- In addition, we assume that we have access to a 1500-core cluster $\mathcal{B}$, where each core has access to 1 terabytes of shared RAM.

Feasibility of computing discrete logarithms in $\mathbb{F}_{3^{6\cdot 1429}}$

- We assume that we have access to a 9000-core cluster $\mathcal{A}$, where each core has access to 16 gigabytes of shared RAM, such as ABACUS-Cinvestav.

- In addition, we assume that we have access to a 1500-core cluster $\mathcal{B}$, where each core has access to 1 terabytes of shared RAM.

- We further assume that each core can execute $2^{27} M_q$ per second.

Feasibility of computing discrete logarithms in $\mathbb{F}_{3^{6\cdot 1429}}$

- We assume that we have access to a 9000-core cluster $\mathcal{A}$, where each core has access to 16 gigabytes of shared RAM, such as ABACUS-Cinvestav.
- In addition, we assume that we have access to a 1500-core cluster $\mathcal{B}$, where each core has access to 1 terabytes of shared RAM.
- We further assume that each core can execute $2^{27} M_q$ per second.

Computation	Cluster	# cores	# days
Degree-3	$\mathcal{A}$	5824	2
Degree-4	$\mathcal{A}$	9000	1
Guillevic descent	$\mathcal{A}$	9000	59
First classical descent	$\mathcal{A}$	9000	39
Second classical descent	$\mathcal{A}$	9000	7
Small degree descent	$\mathcal{B}$	1500	65
Total time			173

Table: Estimated calendar time for computing discrete logarithms in $\mathbb{F}_{3^{6\cdot 1429}}$ using clusters $\mathcal{A}$ and $\mathcal{B}$.

Open problem

Since the effort in the previous slide is still beyond the reach of the computer resources available to us, it would be worthwhile to improve the descent strategy.

Open problem

Since the effort in the previous slide is still beyond the reach of the computer resources available to us, it would be worthwhile to improve the descent strategy.

Muchas Gracias

References

R. Barbulescu, P. Gaudry, A. Joux and E. Thomé, "A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic: Improvements over FFS in small to medium characteristic", Advances in Cryptology – EUROCRYPT 2014, LNCS 8441 (2014), 1–16.

R. Granger, T. Kleinjung and J. Zumbrägel, "Breaking '128-bit secure' supersingular binary curves (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$)", Advances in Cryptology – CRYPTO 2014, LNCS 8617 (2014), 126–145.

A. Guillevic, "Faster individual discrete logarithms in non-prime finite fields with the NFS and FFS algorithms", available at `http://eprint.iacr.org/2016/684`, (2016).

T. Hayashi, T. Shimoyama, N. Shinohara and T. Takagi, "Breaking pairing-based cryptosystems using η_T pairing over $GF(3^{97})$", Advances in Cryptology – ASIACRYPT 2012, LNCS 7658 (2012), 43–60.

A. Joux, "A new index-calculus algorithm with complexity $L(1/4 + o(1))$ in very small characteristic", Selected Areas in Cryptography – SAC 2013, LNCS 8282 (2014), 355–379.

A. Joux and C. Pierrot, "Improving the polynomial time precomputation of Frobenius representation discrete logarithm algorithms", Advances in Cryptology – ASIACRYPT 2014, LNCS 8873 (2014), 378–397.

References

G. Adj, I. Canales-Martínez, N. Cruz-Cortés, A. Menezes, T. Oliveira, L. Rivera-Zamarripa and F. Rodríguez-Henríquez, "Computing discrete logarithms in cryptographically-interesting characteristic-three finite fields", available at http://eprint.iacr.org/2016/.

G. Adj, A. Menezes, T. Oliveira and F. Rodríguez-Henríquez, "Computing discrete logarithms in $\mathbb{F}_{3^{6\cdot 137}}$ and $\mathbb{F}_{3^{6\cdot 163}}$ using Magma", Arithmetic of Finite Fields – WAIFI 2014, LNCS 9061 (2014), 3–22.

G. Adj, A. Menezes, T. Oliveira and F. Rodríguez-Henríquez, "Weakness of $\mathbb{F}_{3^{6\cdot 1429}}$ and $\mathbb{F}_{2^{4\cdot 3041}}$ for discrete logarithm cryptography", Finite Fields and Their Applications, 32 (2015), 148–170.

G. Adj, A. Menezes, T. Oliveira and F. Rodríguez-Henríquez, "Weakness of $\mathbb{F}_{3^{6\cdot 509}}$ for discrete logarithm cryptography", Pairing-Based Cryptography – Pairing 2013, LNCS 8365 (2014), 20–44.

The continued-fraction descent

Recall that we want to compute $\log_g h$ and assume that n is odd and $\deg h = n - 1$.

For a chosen $m < n - 1$, we want to express $\log_g h$ as a linear combination of logarithms of polynomials of degree at most m.

- (1) Multiply h by a random power of g to get $h' = g^i * h$.
- (2) Use the extended Euclidean algorithm to express h' in the form

$$w_2 \cdot h' + v \cdot I_X = w_1 \quad \text{where} \quad \deg w_1 = \deg w_2 = \frac{n-1}{2}.$$

Repeat (1)-(2) until w_1 and w_2 are m-smooth.

In $\mathbb{F}_{3^{6 \cdot 137}}$, for $m = 13$, the total running time of the continued-fraction step is 22 CPU hours.

The Gröbner bases descent

Let $f \in \mathbb{F}_{q^d}[X]$ of degree D, and let $m = \lceil (D+1)/2 \rceil$.

We want 2 polynomials k_1 and $k_2 \in \mathbb{F}_{q^d}[X]$ of degree d such that $f \mid G$,

$$\text{where} \quad G = k_1 \widetilde{k}_2 - \widetilde{k}_1 k_2 \pmod{I_X},$$

with $\widetilde{k}_i(X) = \overline{h}_1^m \cdot \overline{k}_i \left(\frac{\overline{h}_0}{\overline{h}_1}\right)$ and $\widetilde{k}_i(X) = \overline{h}_1^m \cdot \overline{k}_i \left(\frac{\overline{h}_0}{\overline{h}_1}\right)$.
We then have

$$G^q \equiv \overline{h}_1^{mq} \cdot k_2 \cdot \prod_{\lambda \in \mathbb{F}_q} (k_1 - \lambda k_2) \pmod{I_X}.$$

as can be seen by making the substitution $Y \mapsto k_1/k_2$ into the systematic equation $Y^q - Y = \prod_{\lambda \in \mathbb{F}_q} (Y - \lambda)$ and clearing denominators.

If $3m < n$, then $G = k_1 \widetilde{k}_2 - \widetilde{k}_1 k_2$, since $k1\widetilde{k}_2 - \widetilde{k}_1 k2$ has degree $3m$ and so $G(X) = f(X)R(X)$ for some $R \in \mathbb{F}_{q^d}[X]$ with $\deg R = 3m - D$.