Elliptic and Hyperelliptic Curves
a Practical Security Analysis

Joppe W. Bos

Conference on the
Theoretical and Practical Aspects of the Discrete Logarithm Problem

Elliptic and Hyperelliptic Curves
a Practical Security Analysis

Joppe W. Bos

Conference on the
Theoretical and Practical Aspects of the Discrete Logarithm Problem

Outline

- Practical issues when using Pollard rho and the negation map to solve the ECDLP (genus 1).

  J. W. Bos, T. Kleinjung, A. K. Lenstra. *On the Use of the Negation Map in the Pollard Rho Method*, ANTS-IX, LNCS 6197, Springer, 2010.

- How does this look for genus 2 curves?

  J. W. Bos, C. Costello, A. Miele. *Elliptic and Hyperelliptic Curves: a Practical Security Analysis*, PKC 2014, LNCS 8383, pp. 203-220, Springer, 2014.

Outline

- Practical issues when using Pollard rho and the negation map to solve the ECDLP (genus 1).

  J. W. Bos, T. Kleinjung, A. K. Lenstra. *On the Use of the Negation Map in the Pollard Rho Method*, ANTS-IX, LNCS 6197, Springer, 2010.

- How does this look for genus 2 curves?

  J. W. Bos, C. Costello, A. Miele. *Elliptic and Hyperelliptic Curves: a Practical Security Analysis*, PKC 2014, LNCS 8383, pp. 203-220, Springer, 2014.

Motivation – Common belief

When computing the DLP on curves with an automorphism group of cardinality m, one can obtain a constant-factor speedup of $\sqrt{m}$.

How accurate is this factor $\sqrt{m}$ for various cryptographic genus 1 and genus 2 curves in practice?

Preliminaries

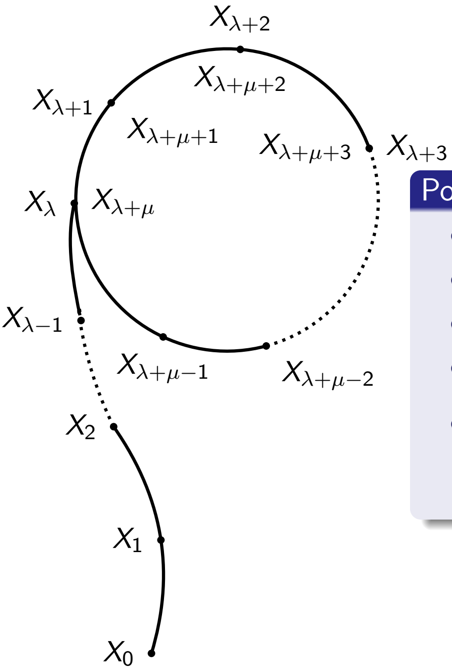
The Elliptic Curve Discrete Logarithm Problem

Let p be an odd prime and $E(\mathbf{F}_p)$ an elliptic curve over $\mathbf{F}_p$. Given $\mathfrak{g} \in E(\mathbf{F}_p)$ of prime order q and $\mathfrak{h} \in \langle \mathfrak{g} \rangle$ find $m \in \mathbf{Z}$ such that $m\mathfrak{g} = \mathfrak{h}$.

Believed to be a hard problem ($\mathcal{O}(\sqrt{q})$).
Algorithms to solve the ECDLP:
Baby-step Giant-step, Pollard ρ, Pollard Kangaroo

Basic Idea

Pick random objects: $u\mathfrak{g} + v\mathfrak{h} \in \langle \mathfrak{g} \rangle$ $(u, v \in \mathbf{Z})$
Find duplicate / collision: $u\mathfrak{g} + v\mathfrak{h} = \bar{u}\mathfrak{g} + \bar{v}\mathfrak{h}$.
If $\bar{v} \not\equiv v \bmod q$, $m = \frac{u-\bar{u}}{\bar{v}-v} \bmod q$ solves the discrete logarithm problem.
Expected number of random objects: $\sqrt{\pi q/2}$



Pollard Rho

- "Walk" through the set $\langle \mathfrak{g} \rangle$.
- $X_i = u \cdot \mathfrak{g} + v \cdot \mathfrak{h}$
- Iteration function $f : \langle \mathfrak{g} \rangle \to \langle \mathfrak{g} \rangle$
- This sequence eventually collides.
- Expected number of steps (iterations): $\sqrt{\frac{\pi \cdot |\langle \mathfrak{g} \rangle|}{2}}$

Pollard ρ

Approximate random walk in $\langle \mathfrak{g} \rangle$

Index function $\ell : \langle \mathfrak{g} \rangle = \mathfrak{G}_0 \cup \ldots \cup \mathfrak{G}_{t-1} \mapsto [0, t-1]$

$$\mathfrak{G}_i = \{\mathfrak{x} : \mathfrak{x} \in \langle \mathfrak{g} \rangle, \ell(\mathfrak{x}) = i\}, \qquad |\mathfrak{G}_i| \approx \frac{q}{t}$$

Precomputed partition constants: $\mathfrak{f}_0, \ldots, \mathfrak{f}_{t-1}$

J. M. Pollard, Monte Carlo methods for index computation $(\bmod\, p)$, Math. Comp. 32 (1978)

Pollard ρ

Approximate random walk in $\langle \mathfrak{g} \rangle$

Index function $\ell : \langle \mathfrak{g} \rangle = \mathfrak{G}_0 \cup \ldots \cup \mathfrak{G}_{t-1} \mapsto [0, t-1]$

$$\mathfrak{G}_i = \{\mathfrak{x} : \mathfrak{x} \in \langle \mathfrak{g} \rangle, \ell(\mathfrak{x}) = i\}, \qquad |\mathfrak{G}_i| \approx \frac{q}{t}$$

Precomputed partition constants: $\mathfrak{f}_0, \ldots, \mathfrak{f}_{t-1}$

r-**adding walk**	$r + s$-**mixed walk**
$t = r$	$t = r + s$
$\mathfrak{p}_{i+1} = \mathfrak{p}_i + \mathfrak{f}_{\ell(\mathfrak{p}_i)}$	$\mathfrak{p}_{i+1} = \begin{cases} \mathfrak{p}_i + \mathfrak{f}_{\ell(\mathfrak{p}_i)}, & \text{if } 0 \leq \ell(\mathfrak{p}_i) < r \\ 2\mathfrak{p}_i, & \text{if } \ell(\mathfrak{p}_i) \geq r \end{cases}$

$r \geq 20$ performance close to a random walk

J. M. Pollard, Monte Carlo methods for index computation $(\bmod p)$, Math. Comp. 32 (1978)
E. Teske: *On random walks for Pollard's rho method*, Math. Comp. 70 (2001)

The Negation Map

M. J. Wiener, R. J. Zuccherato: *Faster attacks on elliptic curve cryptosystems.* SAC, LNCS 1556, Springer, 1999

Equivalence relation $\sim$ on $\langle \mathfrak{g} \rangle$ by $\mathfrak{p} \sim -\mathfrak{p}$ for $\mathfrak{p} \in \langle \mathfrak{g} \rangle$.

$\langle \mathfrak{g} \rangle$ of size q versus $\langle \mathfrak{g} \rangle / \sim$ of size about $\frac{q}{2}$.

Advantage: Reduces the number of steps by a factor of $\sqrt{2}$.
Efficient to compute: Given $(x, y) \in \langle \mathfrak{g} \rangle \to -(x, y) = (x, -y)$

Observation

Certicom challenges

79-bit	exercise	December 1997
89-bit	exercise	February 1998
97-bit	exercise	September 1999
109-bit	level 1	November 2002

- 112-bit, standard curve, July 2009

Observation

Certicom challenges

79-bit	exercise	December 1997
89-bit	exercise	February 1998
97-bit	exercise	September 1999
109-bit	level 1	November 2002

- 112-bit, standard curve, July 2009

Textbook optimization

Negation map ($\sqrt{2}$ speed-up for EC) published in 1999.
Not used in any of the above mentioned ECDLP records.

Observation

Certicom challenges

79-bit	exercise	December 1997
89-bit	exercise	February 1998
97-bit	exercise	September 1999
109-bit	level 1	November 2002

- 112-bit, standard curve, July 2009
- No theoretical improvement: complexity still $\mathcal{O}(\sqrt{q})$
- Practical implications: e.g. 112-bit ECDLP took us $\approx$ 58.3 PS3 years. The negation map could have saved up to 17.1 PS3 years

Negation Map, Side-Effects

Well-known disadvantage: as presented no solution to large ECDLPs

Negation Map, Side-Effects

Well-known disadvantage: fruitless cycles

$$\mathfrak{p} \xrightarrow{(i,-)} -(\mathfrak{p} + \mathfrak{f}_i) \xrightarrow{(i,-)} \mathfrak{p}.$$

Fruitless 2-cycle starts from a random point with probability $\frac{1}{2r}$

I. M. Duursma, P. Gaudry, F. Morain. *Speeding up the discrete log computation on curves with automorphisms*. ASIACRYPT, LNCS 1716, Springer, 1999.

Negation Map, Side-Effects

Well-known disadvantage: fruitless cycles

$$\mathfrak{p} \xrightarrow{(i,-)} -(\mathfrak{p}+\mathfrak{f}_i) \xrightarrow{(i,-)} \mathfrak{p}.$$

Fruitless 2-cycle starts from a random point with probability $\frac{1}{2r}$

I. M. Duursma, P. Gaudry, F. Morain. *Speeding up the discrete log computation on curves with automorphisms*. ASIACRYPT, LNCS 1716, Springer, 1999.

2-cycle reduction technique: look ahead

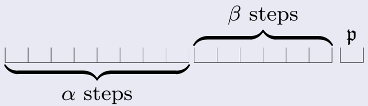
$$f(\mathfrak{p}) = \begin{cases} E(\mathfrak{p}) & \text{if } j = \ell(\sim(\mathfrak{p}+\mathfrak{f}_j)) \text{ for } 0 \le j < r \\ \sim(\mathfrak{p}+\mathfrak{f}_i) & \text{with } i \ge \ell(\mathfrak{p}) \text{ minimal s.t. } \ell(\sim(\mathfrak{p}+\mathfrak{f}_i)) \ne i \bmod r. \end{cases}$$

once every r^r steps: $E : \langle\mathfrak{g}\rangle \to \langle\mathfrak{g}\rangle$ may restart the walk

Cost increase $c = \sum_{i=0}^{r} \frac{1}{r^i}$ with $1 + \frac{1}{r} \le c \le 1 + \frac{1}{r-1}$.

Dealing with Fruitless Cycles in General

Cycle detection

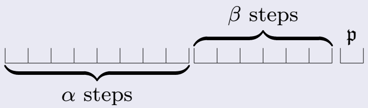


Compare $\mathfrak{p}$ to all β points. Detect cycles of length $\leq \beta$.

α too small: frequent cycle-checking (expensive)
α too large: higher probability of trapped walks (useless steps)

Dealing with Fruitless Cycles in General

Cycle detection



Compare $\mathfrak{p}$ to all β points. Detect cycles of length $\leq \beta$.

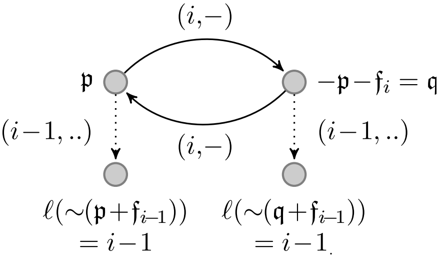
Cycle Escaping

Add

- $\mathfrak{f}_{\ell(\mathfrak{p})+c}$ for a fixed $c \in \mathbf{Z}$
- a precomputed value $\mathfrak{f}'$
- $\mathfrak{f}''_{\ell(\mathfrak{p})}$ from a distinct list of r precomputed values $\mathfrak{f}''_0, \mathfrak{f}''_1, \ldots, \mathfrak{f}''_{r-1}$.

to a representative element of this cycle.

2-cycles when using the 2-cycle reduction technique



Lemma

The probability to enter a fruitless 2-cycle when looking ahead to reduce 2-cycles while using an r-adding walk is

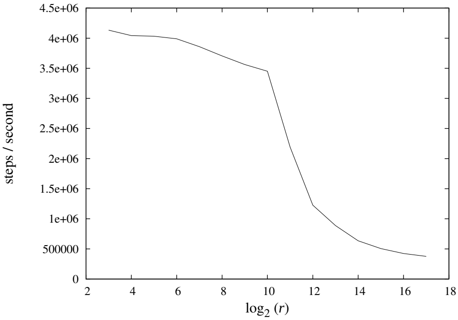
$$\frac{1}{2r}\left(\sum_{i=1}^{r-1}\frac{1}{r^i}\right)^2 = \frac{(r^{r-1}-1)^2}{2r^{2r-1}(r-1)^2} = \frac{1}{2r^3} + \mathcal{O}\left(\frac{1}{r^4}\right).$$

Size of the Random Walk

- Probability to enter cycle depends on the number of partitions r
- Why not simply increase r?

Size of the Random Walk

- Probability to enter cycle depends on the number of partitions r
- Why not simply increase r?



- Practical performance penalty (cache-misses)
- Fruitless cycles still occur

Recurring Cycles

Using

- *r*-adding walk with a medium sized *r* **and**
- { 2, 4 }-reduction technique **and**
- cycle escaping techniques

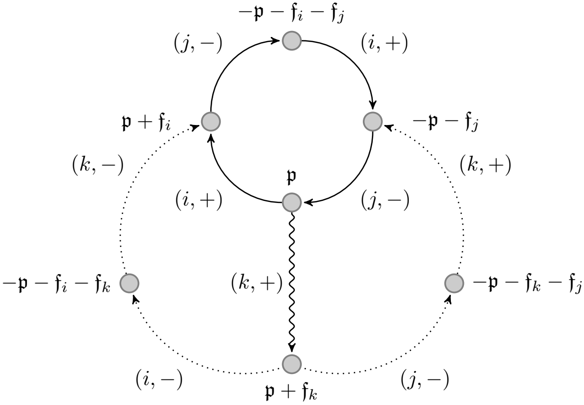
it is expected that many walks will never find a DTP.

Recurring Cycles

Using

- *r*-adding walk with a medium sized *r* **and**
- { 2, 4 }-reduction technique **and**
- cycle escaping techniques

it is expected that many walks will never find a DTP.



Probabilities Overview

		Cycle reduction method:	none	2-cycle	4-cycle
Probability to enter	2-cycle		$\mathbf{\frac{1}{2r}}$	$\frac{1}{2r^3}$	$\frac{2(r-2)^2}{(r-1)r^4}$
	4-cycle		$\mathbf{\frac{r-1}{4r^3}}$	$\frac{r-1}{4r^3}$	$\frac{4(r-2)^4(r-1)}{r^{11}}$
Probability to recur to escape point using	$\mathfrak{f}_{\ell(\mathfrak{p})+c}$		$\frac{1}{2r}$	$\frac{1}{2r^2}$	$\frac{(r-2)^2}{r^4}$
	$\mathfrak{f}'$		$\frac{1}{8r}$	$\frac{1}{8r^3}$	$\frac{(r-2)^2}{2r^5}$
	$\mathfrak{f}''_{\ell(\mathfrak{p})}$		$\frac{1}{8r^2}$	$\frac{1}{8r^4}$	$\frac{(r-2)^2}{2r^6}$
Slowdown factor of iteration function			n/a	$\frac{r+1}{r}$	$\frac{r+4}{r}$

Dealing with Recurring Cycles

Heuristic

A cycle with at least one doubling is most likely not fruitless.

Dealing with Recurring Cycles

Heuristic

A cycle with at least one doubling is most likely not fruitless.

$$\bar{f}(\mathfrak{p}) = \begin{cases} \sim(\mathfrak{p} + \mathfrak{f}_{\ell(\mathfrak{p})}) & \text{if } \ell(\mathfrak{p}) \neq \ell(\sim(\mathfrak{p} + \mathfrak{f}_{\ell(\mathfrak{p})})), \\ \sim(2\mathfrak{p}) & \text{otherwise} \end{cases}$$

Dealing with Recurring Cycles

Heuristic

A cycle with at least one doubling is most likely not fruitless.

$$\bar{f}(\mathfrak{p}) = \begin{cases} \sim(\mathfrak{p} + \mathfrak{f}_{\ell(\mathfrak{p})}) & \text{if } \ell(\mathfrak{p}) \neq \ell(\sim(\mathfrak{p} + \mathfrak{f}_{\ell(\mathfrak{p})})), \\ \sim(2\mathfrak{p}) & \text{otherwise} \end{cases}$$

This completely eliminates 2-cycles.
EC doubling is more expensive compared to a EC addition (in affine Weierstrass).

What is used in practice?

- perform $w > 0$ steps before checking for a cycle
- enter a cycle with probability p at each step
- once we enter a cycle at step $0 \le i \le w$ all subsequent $w - i$ steps are fruitless

After w steps we expect to have computed

$$W(w,p) = \sum_{i=0}^{w-1} p(1-p)^i (w-i)$$

fruitless steps.

Estimate the maximum speedup possible
(ignoring various implementation overheads)

What is used in practice?

Cost EC-addition: 6 multiplications
Cost EC-doubling: 7 multiplications

Setting I

$\alpha = 3000$, $\beta = 10$, $r = 128$
2-cycle reduction, cycle escape by doubling
$Pr[\text{enter 2-cycle}] \approx 1/(2r^3)$
$Pr[\text{enter 4-cycle}] \approx (r-1)/(4r^3)$

$$\frac{\text{fruitful}}{\text{total}}\sqrt{2} = \frac{6(\alpha - W(\alpha, \frac{r-1}{4r^3}))}{\frac{r+1}{r}6\alpha + 7}\sqrt{2} = 0.97\sqrt{2}$$

J. W. Bos, T. Kleinjung, A. K. Lenstra: *On the Use of the Negation Map in the Pollard Rho Method*, ANTS-IX, LNCS 6197, pp. 66-82, Springer, 2010.

What is used in practice?

Cost EC-addition: 6 multiplications
Cost EC-doubling: 7 multiplications

Setting I

$\alpha = 3000$, $\beta = 10$, $r = 128$
2-cycle reduction, cycle escape by doubling
$Pr[\text{enter 2-cycle}] \approx 1/(2r^3)$
$Pr[\text{enter 4-cycle}] \approx (r-1)/(4r^3)$

$$\frac{\text{fruitful}}{\text{total}}\sqrt{2} = \frac{6(\alpha - W(\alpha, \frac{r-1}{4r^3}))}{\frac{r+1}{r}6\alpha + 7}\sqrt{2} = 0.97\sqrt{2}$$

In practice the observed speedup was $0.91\sqrt{2}$

J. W. Bos, T. Kleinjung, A. K. Lenstra: *On the Use of the Negation Map in the Pollard Rho Method*, ANTS-IX, LNCS 6197, pp. 66-82, Springer, 2010.

What is used in practice?

- No cycle reduction, check frequently for cycles.
- Large r value: look for special precomputed points (x, y) such that $x \equiv y \equiv 0 \pmod{2^c}$

Setting II

Check for 2-cycles every $\alpha = 48$ steps, escape by doubling, $r = 2048$

$$\frac{\text{fruitful}}{\text{total}}\sqrt{2} = \frac{6(\alpha - W(\alpha, \frac{1}{2r}))}{6\alpha + 7}\sqrt{2} = 0.97\sqrt{2}$$

D. J. Bernstein, T. Lange, and P. Schwabe. *On the correct use of the negation map in the Pollard rho method*. PKC, LNCS 6571, Springer, 2011.

Extending this approach

The idea behind the negation map applies to larger efficiently computable cyclic automorphism groups

- Assume the target curve comes equipped with such an automorphism group of cardinality m and generator ψ
- Define an equivalence relation $\sim$ on $\langle \mathfrak{g} \rangle$ by $R \sim R'$ iff $R = \psi^i(R')$ for some $0 \leq i < m$
- Modify Pollard rho:
  Find the unique representative $\tilde{R}$ of the class containing R (i.e. $\tilde{R}_1 = \tilde{R}_2$ iff $R_1 \sim R_2$), then call $f(\tilde{R})$ as usual
- We always have the identity and the negation map

Extending this approach

The idea behind the negation map applies to larger **efficiently computable** cyclic automorphism groups

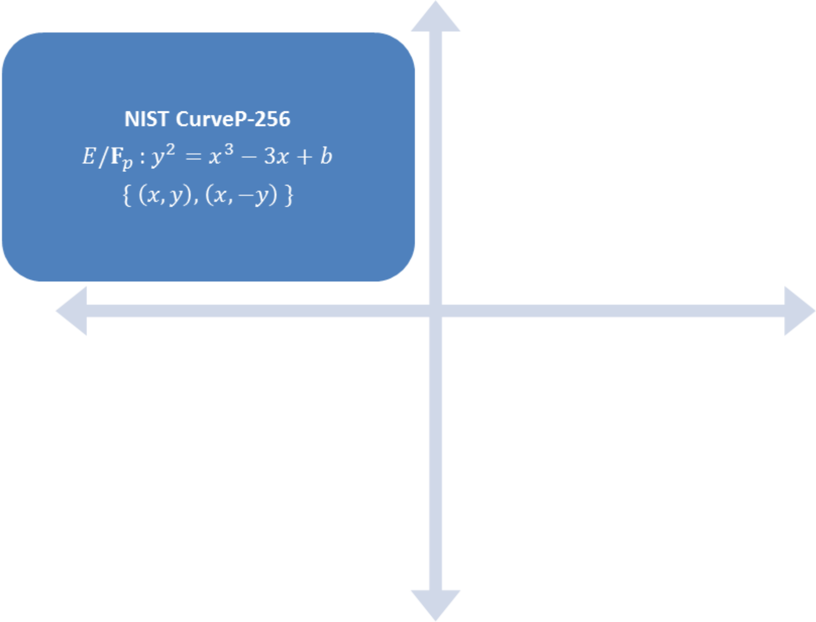
- Assume the target curve comes equipped with such an automorphism group of cardinality m and generator ψ
- Define an equivalence relation $\sim$ on $\langle \mathfrak{g} \rangle$ by $R \sim R'$ iff $R = \psi^i(R')$ for some $0 \leq i < m$
- Modify Pollard rho:
  Find the unique representative $\tilde{R}$ of the class containing R (i.e. $\tilde{R}_1 = \tilde{R}_2$ iff $R_1 \sim R_2$), then call $f(\tilde{R})$ as usual
- We always have the identity and the negation map

Extending this approach

The idea behind the negation map applies to larger **efficiently computable** cyclic automorphism groups

- Assume the target curve comes equipped with such an automorphism group of cardinality m and generator ψ
- Define an equivalence relation $\sim$ on $\langle \mathfrak{g} \rangle$ by $R \sim R'$ iff $R = \psi^i(R')$ for some $0 \leq i < m$
- Modify Pollard rho:
  Find the unique representative $\tilde{R}$ of the class containing R (i.e. $\tilde{R}_1 = \tilde{R}_2$ iff $R_1 \sim R_2$), then call $f(\tilde{R})$ as usual
- We always have the identity and the negation map

speedup	slowdown
$\sqrt{m}$ fewer iterations	find the unique representative $\rightarrow$ more costly iteration
	overhead to deal with fruitless cycles


NIST CurveP-256
$E/\mathbf{F}_p : y^2 = x^3 - 3x + b$
$\{ (x, y), (x, -y) \}$

Target Curves

NIST CurveP-256

$E/\mathbf{F}_p : y^2 = x^3 - 3x + b$

$\{ (x, y), (x, -y) \}$

Generic1271

$C/\mathbf{F}_p$:

$y^2 = x^5 + a_3x^3 + a_2x^2 + a_1x + a_0$

Mumford representation:

$\{ (x^2 + u_1x + u_0, v_1x + v_0),$

$(x^2 + u_1x + u_0, -(v_1x + v_0)) \}$

Target Curves

NIST CurveP-256

$E/\mathbf{F}_p : y^2 = x^3 - 3x + b$

$\{ (x, y), (x, -y) \}$

Generic1271

$C/\mathbf{F}_p :$

$y^2 = x^5 + a_3x^3 + a_2x^2 + a_1x + a_0$

Mumford representation:

$\{ (x^2 + u_1x + u_0, v_1x + v_0),$

$(x^2 + u_1x + u_0, -(v_1x + v_0)) \}$

BN254

$E/\mathbf{F}_p : y^2 = x^3 + 2$

Since $p \equiv 1 \bmod 3 \rightarrow \zeta \neq 1 \in \mathbf{F}_p$ s.t. $\zeta^3 = 1$

$\{ (x, y), (x, -y),$

$(\zeta x, y), (\zeta x, -y),$

$(\zeta^2 x, y), (\zeta^2 x, -y) \}$

Target Curves

NIST CurveP-256

$E/\mathbf{F}_p : y^2 = x^3 - 3x + b$

$\{ (x, y), (x, -y) \}$

Generic1271

$C/\mathbf{F}_p :$

$y^2 = x^5 + a_3 x^3 + a_2 x^2 + a_1 x + a_0$

Mumford representation:

$\{ (x^2 + u_1 x + u_0, v_1 x + v_0),$

$(x^2 + u_1 x + u_0, -(v_1 x + v_0)) \}$

BN254

$E/\mathbf{F}_p : y^2 = x^3 + 2$

Since $\mathrm{p} \equiv 1 \bmod 3 \rightarrow \zeta \neq 1 \in \mathbf{F}_p$

s.t. $\zeta^3 = 1$

$\{ (x, y), (x, -y),$

$(\zeta x, y), (\zeta x, -y),$

$(\zeta^2 x, y), (\zeta^2 x, -y) \}$

4GLV127-BK

$C/\mathbf{F}_p : y^2 = x^5 + 17$

Since $\mathrm{p} \equiv 1 \bmod 5 \rightarrow \zeta \neq 1 \in \mathbf{F}_p$

s.t. $\zeta^5 = 1$

$\phi : (x^2 + u_1 x + u_0,\ v_1 x + v_0) \mapsto$
$(x^2 + \zeta u_1 x + \zeta^2 u_0,\ \zeta^4 v_1 x + v_0)$

Updated Speedup Estimate

curve	(g, m)	cost of one step		
		divisor addition	compute representative	
			worst	average
CurveP-256	$(1, 2)$	$5\mathbf{M} + \mathbf{S} + 6\mathbf{a}$	$1\mathbf{a}$	$\frac{1}{2}\mathbf{a}$

	original	$\to$	updated estimate
CurveP-256	$\sqrt{2}$	$\to$	$\sqrt{2}$

Representative point with odd y-coordinate (when $0 \leq y < p$)

Updated Speedup Estimate

curve	(g, m)	cost of one step		
		divisor addition	compute representative	
			worst	average
CurveP-256	$(1,2)$	$5\mathbf{M} + \mathbf{S} + 6\mathbf{a}$	$1\mathbf{a}$	$\frac{1}{2}\mathbf{a}$
BN254	$(1,6)$	$5\mathbf{M} + \mathbf{S} + 6\mathbf{a}$	$1\mathbf{M} + 3\mathbf{a}$	$1\mathbf{M} + 2\frac{1}{2}\mathbf{a}$

	original	$\rightarrow$	updated estimate
CurveP-256	$\sqrt{2}$	$\rightarrow$	$\sqrt{2}$
BN254	$\sqrt{6}$	$\rightarrow$	$\frac{6}{7}\sqrt{6} \approx 0.857\sqrt{6}$

Representative point whose x-coordinate has least absolute value *and* whose y-coordinate is odd

$$\begin{array}{ll} 1 \text{ neg} & (x, y) \rightarrow (x, -y) \\ 1 \text{ mul} & (\zeta x, y) \leftrightarrow (\zeta x, -y) \\ 1 \text{ mul} & (\zeta^2 x, y) \leftrightarrow (\zeta^2 x, -y) \end{array}$$

Updated Speedup Estimate

curve	(g, m)	cost of one step		
		divisor addition	compute representative: worst	compute representative: average
CurveP-256	$(1, 2)$	$5\mathbf{M} + \mathbf{S} + 6\mathbf{a}$	$1\mathbf{a}$	$\frac{1}{2}\mathbf{a}$
BN254	$(1, 6)$	$5\mathbf{M} + \mathbf{S} + 6\mathbf{a}$	$1\mathbf{M} + 3\mathbf{a}$	$1\mathbf{M} + 2\frac{1}{2}\mathbf{a}$

	original	$\rightarrow$	updated estimate
CurveP-256	$\sqrt{2}$	$\rightarrow$	$\sqrt{2}$
BN254	$\sqrt{6}$	$\rightarrow$	$\frac{6}{7}\sqrt{6} \approx 0.857\sqrt{6}$

Representative point whose x-coordinate has least absolute value *and* whose y-coordinate is odd. $\zeta^2 x = -(\zeta + 1)x$

1 neg	$(x, y) \rightarrow (x, -y)$
1 mul	$(\zeta x, y) \leftrightarrow (\zeta x, -y)$
1 neg, 1 add	$(-(\zeta x + x), y) \leftrightarrow (-(\zeta x + x), -y)$

Updated Speedup Estimate

curve	(g, m)	cost of one step		
		divisor addition	compute representative worst	compute representative average
CurveP-256	$(1, 2)$	$5\mathbf{M} + \mathbf{S} + 6\mathbf{a}$	$1\mathbf{a}$	$\frac{1}{2}\mathbf{a}$
BN254	$(1, 6)$	$5\mathbf{M} + \mathbf{S} + 6\mathbf{a}$	$1\mathbf{M} + 3\mathbf{a}$	$1\mathbf{M} + 2\frac{1}{2}\mathbf{a}$
Generic1271	$(2, 2)$	$20\mathbf{M} + 4\mathbf{S} + 48\mathbf{a}$	$2\mathbf{a}$	$1\mathbf{a}$

	original	$\rightarrow$	updated estimate
CurveP-256	$\sqrt{2}$	$\rightarrow$	$\sqrt{2}$
BN254	$\sqrt{6}$	$\rightarrow$	$\frac{6}{7}\sqrt{6} \approx 0.857\sqrt{6}$
Generic1271	$\sqrt{2}$	$\rightarrow$	$\sqrt{2}$

Representative divisor with odd v_0-coordinate

Updated Speedup Estimate

curve	(g, m)	cost of one step		
		divisor addition	compute representative worst	compute representative average
CurveP-256	$(1, 2)$	$5\mathbf{M} + \mathbf{S} + 6\mathbf{a}$	$1\mathbf{a}$	$\frac{1}{2}\mathbf{a}$
BN254	$(1, 6)$	$5\mathbf{M} + \mathbf{S} + 6\mathbf{a}$	$1\mathbf{M} + 3\mathbf{a}$	$1\mathbf{M} + 2\frac{1}{2}\mathbf{a}$
Generic1271	$(2, 2)$	$20\mathbf{M} + 4\mathbf{S} + 48\mathbf{a}$	$2\mathbf{a}$	$1\mathbf{a}$
4GLV127-BK	$(2, 10)$	$20\mathbf{M} + 4\mathbf{S} + 48\mathbf{a}$	$6\mathbf{M} + 1\mathbf{S} + 5\mathbf{a}$	$5\frac{2}{5}\mathbf{M} + \frac{4}{5}\mathbf{S} + \frac{3}{5}\mathbf{a}$

	original	→	updated estimate
CurveP-256	$\sqrt{2}$	→	$\sqrt{2}$
BN254	$\sqrt{6}$	→	$\frac{6}{7}\sqrt{6} \approx 0.857\sqrt{6}$
Generic1271	$\sqrt{2}$	→	$\sqrt{2}$
4GLV127-BK	$\sqrt{10}$	→	$\frac{120}{151}\sqrt{10} \approx 0.784\sqrt{10}$

Representative divisor whose u_1-coordinate has least absolute value *and* whose v_0-coordinate is odd. Use: $\zeta^4 = -(\zeta^3 + \zeta^2 + \zeta + 1)$

$u_1 u_0$ and u_1^2 are required for the efficient formulas

Other (popular) cryptographic curves?

Genus 1

- Curve25519 (Bernstein)
  no additional automorphisms $\rightarrow$ identical analysis as CurveP-256
- j-invariant zero curves (not pairing-friendly) using GLV techniques
  - $E/F_p : y^2 = x^3 + 2$ with $p = 2^{256} - 11733$ (Longa and Sica)
  - $E/F_p : y^2 = x^3 + 7$ with $p = 2^{256} - 2^{32} - 977$ standard curve used in Bitcoin

Automorphism group the same as BN254 so identical analysis.

Other (popular) cryptographic curves?

Genus 1

- Curve25519 (Bernstein)
  no additional automorphisms $\rightarrow$ identical analysis as CurveP-256
- j-invariant zero curves (not pairing-friendly) using GLV techniques
  - $E/F_p : y^2 = x^3 + 2$ with $p = 2^{256} - 11733$ (Longa and Sica)
  - $E/F_p : y^2 = x^3 + 7$ with $p = 2^{256} - 2^{32} - 977$ standard curve used in Bitcoin

Automorphism group the same as BN254 so identical analysis.

Genus 2

Kummer surface over F_p with $p = 2^{127} - 1$ (Gaudry and Schost)

- No known way how to exploit the fast arithmetic on the Kummer surface (only pseudo-additions exist)
- Map DLP back to the Jacobian group $\rightarrow$ same situation as Generic1271 (except cofactor of 16)

Results

Curve	Performance (10^6 it/sec)		speedup	
	without	with	expected	real
NIST CurveP-256	2.569	2.447	$\sqrt{2}$	$0.947\sqrt{2}$
BN254	2.816	2.238	$0.857\sqrt{6}$	$0.790\sqrt{6}$
Generic1271	2.941	2.780	$\sqrt{2}$	$0.940\sqrt{2}$
4GLV127-BK	2.074	1.643	$0.795\sqrt{10}$	$0.784\sqrt{10}$

Results

Curve	Performance (10^6 it/sec)		speedup	
	without	with	expected	real
NIST CurveP-256	2.569	2.447	$\sqrt{2}$	$0.947\sqrt{2}$
BN254	2.816	2.238	$0.857\sqrt{6}$	$0.790\sqrt{6}$
Generic1271	**2.941**	2.780	$\sqrt{2}$	$0.940\sqrt{2}$
4GLV127-BK	2.074	1.643	$0.795\sqrt{10}$	$0.784\sqrt{10}$

Results

Curve	Performance (10^6 it/sec)		speedup	
	without	with	expected	real
NIST CurveP-256	2.569	2.447	$\sqrt{2}$	$0.947\sqrt{2}$
BN254	2.816	2.238	$0.857\sqrt{6}$	$0.790\sqrt{6}$
Generic1271	2.941	2.780	$\sqrt{2}$	$0.940\sqrt{2}$
4GLV127-BK	2.074	1.643	$0.795\sqrt{10}$	$0.784\sqrt{10}$

Results

Curve	Performance (10^6 it/sec)		speedup	
	without	with	expected	real
NIST CurveP-256	2.569	2.447	$\sqrt{2}$	$0.947\sqrt{2}$
BN254	2.816	2.238	$0.857\sqrt{6}$	$0.790\sqrt{6}$
Generic1271	2.941	2.780	$\sqrt{2}$	$0.940\sqrt{2}$
4GLV127-BK	2.074	1.643	$0.795\sqrt{10}$	$0.784\sqrt{10}$

real / expected $=$ 0.922 to 0.986

Results

Curve	Performance (10^6 it/sec)		speedup	
	without	with	expected	real
NIST CurveP-256	2.569	2.447	$\sqrt{2}$	$0.947\sqrt{2}$
BN254	2.816	2.238	$0.857\sqrt{6}$	$0.790\sqrt{6}$
Generic1271	2.941	2.780	$\sqrt{2}$	$0.940\sqrt{2}$
4GLV127-BK	2.074	1.643	$0.795\sqrt{10}$	$0.784\sqrt{10}$

	CurveP-256	BN254	Generic1271	4GLV127-BK
core years	$3.946 \cdot 10^{24}$	$9.486 \cdot 10^{23}$	$1.736 \cdot 10^{24}$	$1.309 \cdot 10^{24}$
security	128.0	125.9	126.8	126.4

Security

	CurveP-256	BN254	Generic1271	4GLV127-BK
core years	$3.946 \cdot 10^{24}$	$9.486 \cdot 10^{23}$	$1.736 \cdot 10^{24}$	$1.309 \cdot 10^{24}$
security	128.0	125.9	126.8	126.4

Interesting

- Certicom challenge over a 239-bit prime field: estimate $1.4 \cdot 10^{27}$ Pentium-100 core years

Security

	CurveP-256	BN254	Generic1271	4GLV127-BK
core years	$3.946 \cdot 10^{24}$	$9.486 \cdot 10^{23}$	$1.736 \cdot 10^{24}$	$1.309 \cdot 10^{24}$
security	128.0	125.9	126.8	126.4

Interesting

- Certicom challenge over a 239-bit prime field: estimate $1.4 \cdot 10^{27}$ Pentium-100 core years
- RSA-768 2000 core years. Estimate for RSA-3072:

$$\frac{N(3072)}{N(768)} T_{768} = 5 \cdot 10^{18} \cdot 2 \cdot 10^{3} = 10^{22} \text{ core years}$$

where $N(k) = \exp(1.923 \log(2^k)^{1/3} \log(\log(2^k))^{2/3})$

Conclusions

One should use the automorphism group of cardinality m when solving the DLP for genus $\{1, 2\}$ curves.

Conclusions

One should use the automorphism group of cardinality m when solving the DLP for genus $\{1, 2\}$ curves.

Advantage

Reduces the number of expected iterations by a factor $\sqrt{m}$

Conclusions

One should use the automorphism group of cardinality m when solving the DLP for genus $\{1, 2\}$ curves.

Advantage

Reduces the number of expected iterations by a factor $\sqrt{m}$

Disadvantages

- More expensive iteration function: find the unique representative
  - *Side-effect:* curves with larger automorphism groups loose less security then previously thought:
    e.g. BN254 $\sqrt{6} \rightarrow 0.790\sqrt{6} \approx \sqrt{3.7}$
- Deal with fruitless cycles: overhead

Conclusions

One should use the automorphism group of cardinality m when solving the DLP for genus $\{1, 2\}$ curves.

Advantage

Reduces the number of expected iterations by a factor $\sqrt{m}$

Disadvantages

- More expensive iteration function: find the unique representative
  - *Side-effect:* curves with larger automorphism groups loose less security then previously thought:
    e.g. BN254 $\sqrt{6} \rightarrow 0.790\sqrt{6} \approx \sqrt{3.7}$
- Deal with fruitless cycles: overhead

Given these estimates even the 131-bit prime field Certicom challenge seems out of reach by an academic effort.