Fast Cryptography in Genus 2

Joppe W. Bos, Craig Costello, Huseyin Hisil and Kristin Lauter

EUROCRYPT 2013
Athens, Greece

May 27, 2013

Recall that curves are much better than $\mathbb{F}_q^*$

  '76

$\mathbb{F}_q^*$ (today $q \approx 3072$ bits)

  '85

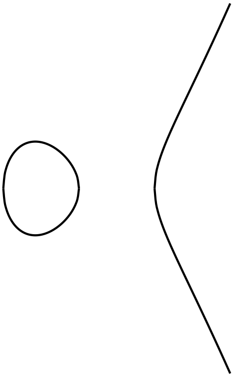
$E/\mathbb{F}_q$ (today $q \approx 256$ bits)

 '89

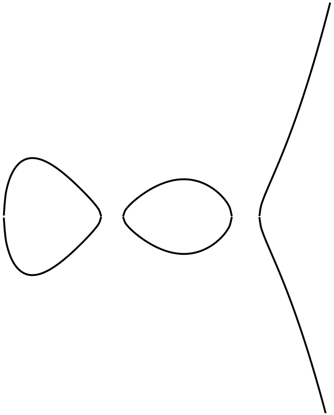
$\mathrm{Jac}(C_g/\mathbb{F}_q)$ (today, $g = 2$, $q \approx 128$ bits)

Why fields of half the size?

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



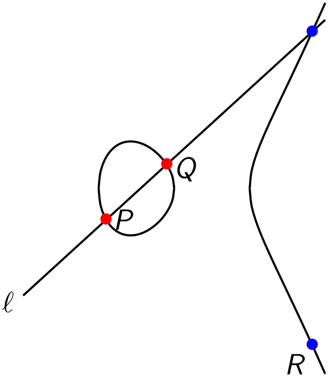
$y^2 = x^5 + b_4x^4 + \cdots + b_0$

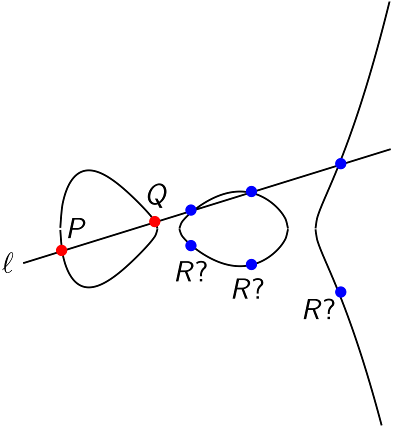


Both curves have around q points over $\mathbb{F}_q$

Why fields of half the size?

$y^2 = x^3 + a_2x^2 + a_1x + a_0$ $\qquad$ $y^2 = x^5 + b_4x^4 + \cdots + b_0$

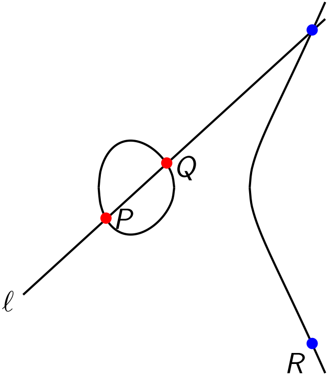


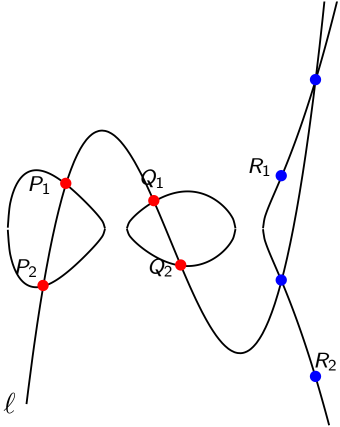


Can't do "chord-and-tangent" in genus 2

Why fields of half the size?

$y^2 = x^3 + a_2x^2 + a_1x + a_0$ $\qquad$ $y^2 = x^5 + b_4x^4 + \cdots + b_0$





Roughly speaking: group elements are pairs of points

$\#E(\mathbb{F}_q) \approx q$ **vs.** $\#\mathrm{Jac}(C)(\mathbb{F}_q) \approx q^2$

This work/talk

1. Finding cryptographically suitable curves
2. Arithmetic of general genus 2 curves
3. GLV decompositions
4. The Kummer surface
5. Results / Open Question

1. Finding cryptographically suitable genus 2 curves

Finding genus 2 curves

1. **Point Counting (Schoof-Pila)**
   - Until $<$ 5 years ago, 128-bit security still far out of reach
   - Gaudry-Schost'12 state-of-the-art
   - 1,000 CPU hours to find group order of any curve
   - 1,000,000+ CPU hours to find Kummer (used in this work)
2. **Real Multiplication (RM)**
   - Gaudry-Kohel-Smith'12
   - Accelerated Schoof-Pila for genus 2 curves with efficiently computable RM (endomorphism)
   - Finds 2-GLV curves
3. **Complex Multiplication (CM)**
   - Very practical for low-discriminant CM fields
   - Fix a prime p, fix quartic CM field $K = \mathbb{Q}[x]/(x^4 + Ax^2 + B)$
   - If p splits (nicely) in $\mathcal{O}_K$, can write down $\#\mathrm{Jac}(C/\mathbb{F}_p)$
   - Input $\mathbb{F}_p$-roots of Igusa Class Polynomials into Mestre to get C
   - Igusas are the hard part (although Kohel/Thomé databases)

Finding genus 2 curves with CM over fast primes

Picky with prime $p \longrightarrow$ flexible with CM field

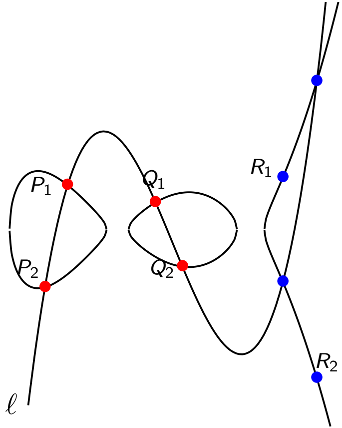
- **Mersenne prime** $p = 2^{127} - 1$ performs fastest at this level
- Search many CM fields to find good curves
- Can't hope to find secure curve for particular family (CM field)

Picky with CM field $\longrightarrow$ flexible with prime p

- Sometimes need particular CM field (e.g. if you want endomorphisms)
- **Montgomery-friendly primes:** $p = 2^{64} \cdot (2^{63} - c) + 1$
- **NIST-friendly primes:** $p = 2^{128} - c$
- Both take $c \ll 2^{63}$ (more than enough flexibility)

2. Arithmetic of general genus 2 curves

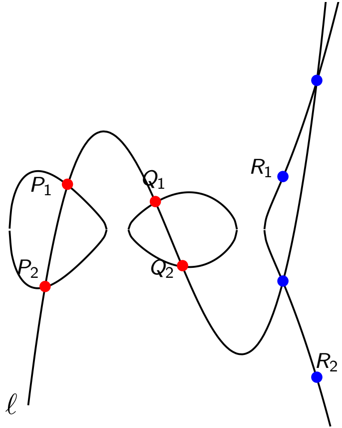
Mumford coordinates



$$\mathit{sextic} = (x - x_{P_1})(x - x_{P_2})(x - x_{Q_1})(x - x_{Q_2})(x - x_{R_1})(x - x_{R_2}) = 0$$
$$\rightarrow \mathit{quadratic} = (x - x_{R_1})(x - x_{R_2}) = 0$$

Computing with actual points means root finding in $\mathbb{F}_q$

Mumford coordinates



$$sextic = (x^2 + \alpha_P x + \beta_P)(x^2 + \alpha_Q x + \beta_Q)(x^2 + \alpha_R x + \beta_R) = 0$$
$$\rightarrow quadratic = (x^2 + \alpha_R x + \beta_R) = 0$$

Mumford coordinates avoid root finding

The cost of genus 2 group operations

- Based on C-Lauter'11, we optimized genus 2 formulas for 128-bit fields

op.	Divisor doubling	Divisor addition	Divisor mix add.
$g = 2$	$34\mathbf{M} + 6\mathbf{S} + 34a$	$44\mathbf{M} + 4\mathbf{S} + 29a$	$37\mathbf{M} + 5\mathbf{S} + 29a$

$\mathbb{F}_p$ operations for common divisor operations in genus 2

implementation	prime p	cycles/scalar mult.
`generic128`	$2^{128} - 173$	364,000
`generic127`	$2^{127} - 1$	248,000

Timings on Intel Core i7-3520M (Ivy Bridge) at 2893.484 MHz

3. GLV scalar decomposition

4-GLV: e.g. Buhler-Koblitz curves

- Let $p = 2^{64} \cdot (2^{63} - 27443) + 1$, and let
$$C/\mathbb{F}_p : y^2 = x^5 + 17$$
- $\#\mathrm{Jac} = 28948022309328876595115567994214488524823328209723866335483563634241778912751$
- Notice that $(x, y) \in C \implies (\xi_5 x, y) \in C$, where $\xi_5^5 = 1$,
$\implies$ "easy to compute" map ϕ on $\mathrm{Jac}(C)$
- For $D \in \mathrm{Jac}(C)$, we get the scalar multiples $\phi(D) = [\lambda]D$, $\phi^2(D) = [\lambda^2]D$ and $\phi^3(D) = [\lambda^3]D$ for free
- $[k]D$ as $[k]D = [k_0]D + [k_1]\phi(D) + [k_2]\phi^2(D) + [k_3]\phi^3(D)$
- eg. $k = 23477399837278936923599493713286470955314785798347519197199578120259089016680$
$(k_0, k_1, k_2, k_3) =$
$(-6344646642321980551, -3170471730617986668, -4387949940648063094, 3721725683392112311)$

4-GLV: e.g. Buhler-Koblitz curves

- k was 254 bits, but instead we multiexponentiate by

$$\begin{array}{rll} D & k_0 = [1,0,1,1,0,0,0,0,0,0,0,1,\dots] & (63\ bits) \\ \phi(D) & k_1 = [0,1,0,1,0,1,1,1,1,1,1,1,\dots] & (63\ bits) \\ \phi^2(D) & k_2 = [0,1,1,1,1,0,0,1,1,1,0,0,\dots] & (63\ bits) \\ \phi^3(D) & k_3 = [0,1,1,0,0,1,1,1,0,1,0,0,\dots] & (63\ bits) \end{array}$$

- 254 `DBL` + 127 `ADD` $\rightarrow$ 63 `DBL` + 80 `ADD` (Straus-Shamir)

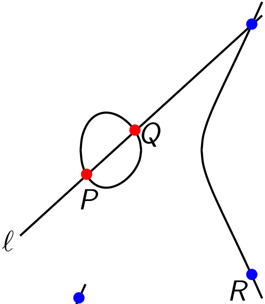
implementation	prime p	cycles/scalar mult.
`generic128`	$2^{128} - 173$	364,000
`4GLV-BK`	$2^{128} - 24935$	164,000
`4GLV-BK`	$2^{64} \cdot (2^{63} - 27443) + 1$	156,000

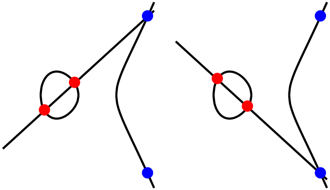
Timings on Intel Core i7-3520M (Ivy Bridge) at 2893.484 MHz

4. The Kummer surface

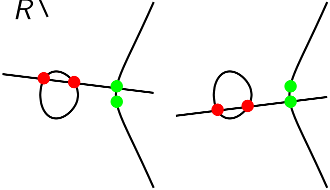
Montgomery ladder for elliptic curves . . .

- Can compute $P + Q$ from $\{P, Q, P - Q\}$ without y-coords
- **Key:** to compute $[k]P$, have $[n + 1]P$ and $[n]P$ at each stage





vs.



same difference → same result

different difference → different result

Genus 2 analogue: the Kummer surface $\mathcal{K}$

- Montgomery identified $P = (P_x, P_y)$ and $-P = (P_x, -P_y)$
- Smart-Siksek'99: $g = 2$ analogue... $\mathrm{Jac}(C) \to \mathcal{K}$ is 2-to-1

- Gaudry'07: much better Kummer surface from theta theory
- The "Squares-only" Kummer is best (Cosset'10)

  $\mathcal{K} : Exyzt = ((x^2+y^2+z^2+t^2) - F(xt+yz) - G(xz+yt) - H(xy+zt))^2$
- No longer a group, but enough to do secure crypto (e.g. DH)
- Each ladder step needs $\mathtt{DBL}_{\mathcal{K}} +$ "ADD"$_{\mathcal{K}}$ – **only 25 $\mathbb{F}_p$ muls !!!**
- Compare to non-Kummer – DBL ≈ 40 and ADD ≈ 50

The Kummer surface

implementation	prime p	cycles/scalar mult.
`generic128`	$2^{128} - 173$	364,000
`generic127`	$2^{127} - 1$	248,000
`4GLV-BK`	$2^{128} - 24935$	164,000
`4GLV-BK`	$2^{64} \cdot (2^{63} - 27443) + 1$	156,000
Kummer	$2^{128} - 237$	166,000
Kummer	$2^{127} - 1$	117,000

Timings on Intel Core i7-3520M (Ivy Bridge) at 2893.484 MHz

- See eBACS for more performance numbers ...

  `http://bench.cr.yp.to`

5. Results / Open Question

Results: genus 1 and 2 implementations over prime fields

who	primitive	g	constant time	10^3 cycles
OpenSSL	NISTp256	1	?	658
Hisil	ecfp256e	1	X	227
Bernstein	curve25519	1	✓	182
Longa-Sica	GLV-2	1	X	145
this work	**generic-p1271**	**2**	**X**	**248**
	GLV-4-BK-Mont	**2**	**X**	**156**
	Kummer-p1271	**2**	✓	**117**

- Kummer offers fastest Diffie-Hellman over prime fields
- Bonus: Kummer also runs in constant-time
- **Kummer chameleons**: curves which can be Kummer or 4GLV, depending on scenario (see full version)

Open question

- Using endomorphisms (GLV) gives big speedups:

  $364{,}000 \longrightarrow 156{,}000$

- Using Kummer surface gives big speedups:

  $248{,}000 \longrightarrow 117{,}000$

- **Question: can we do GLV on the Kummer surface?**
  - Gaudry also noticed that certain Kummers can have an endomorphism ϕ
  - We found some cryptographically sized examples
  - But GLV on pseudo-groups is harder (several caveats)
  - Nevertheless, big speedups possible
  - See the full version - eprint 2012/670