Fast crypto in genus 2

Craig Costello

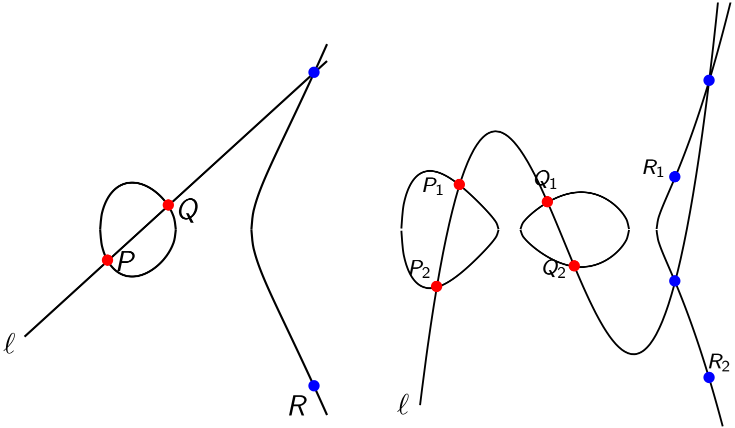
Microsoft Research

Joint work with Joppe Bos, Huseyin Hisil and Kristin Lauter

September 14, 2012

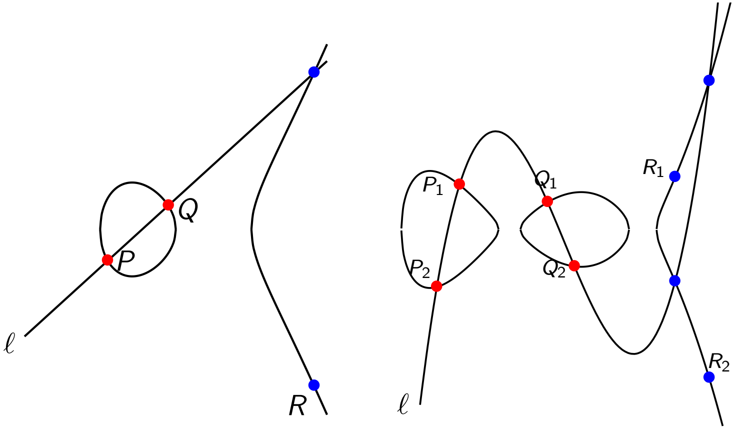
Genus 2: why bother?

- Everything is so much more complicated in genus 2



- Point counting, group law, underlying theory. . .

Genus 2: the reason to bother



Elliptic: $E : y^2 = x^3 + \ldots$ Hyperelliptic: $C : y^2 = x^5 + \ldots$

- $\#E$ and $\#C$ are close over same size field $\mathbb{F}_q$ ... BUT
- **Elliptic group size $\approx \#E$, whilst hyperelliptic group size $\approx \#C^2$**

Genus 2 uses smaller fields

- **g=1**: Bernstein's `curve25519`: $E/\mathbb{F}_p : y^2 = x^3 + \ldots$ over

  $$p = 2^{255} - 19 =$$

  57896044618658097711785492504343953926634992332820282019728792003956564819949

  has group order $\#E = 2^3 \cdot$

  7237005577332262213973186563042994240857116359379907606001950938285454250989 (253 bits)

- **g=2**: One curve we're using: $C/\mathbb{F}_p : y^2 = x^5 + \ldots$ over

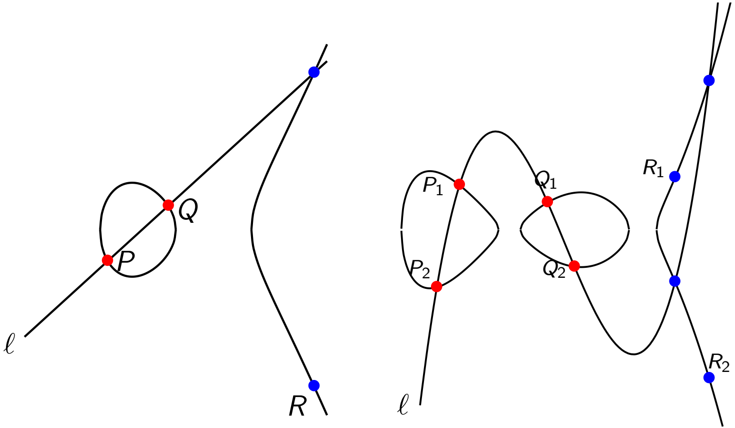
  $$p = 2^{128} - 173 =$$

  340282366920938463463374607431768211283

  has group order $\#\mathrm{Jac}(C) =$

  115792089237316195429342203801033554170931615651881657307308068079702089951781 (257 bits)

Group law complexity in general case



per bit: $\approx$ **10** $\times$ 256-bit muls vs. $\approx$ **50** $\times$ 128-bit muls

- unfortunately: 256-bit mul $\ll 4 \times$ 128-bit mul
- BUT genus 1 estimate uses all the known tricks (genus 2's doesn't)

A fair fight

- The very best curves in genus 2 have not been available
- Bernstein ECC'06:

> *"Standardise genus 2 curves for cryptography? I think that's premature... let's wait for point counting to catch up, then standardize..."*

- **Good news: point counting has caught up!** even in the most general case
- Thanks Gaudry-Schost'12 - and many others!

- Elliptic vs. Hyperelliptic: it's time for a fair fight.

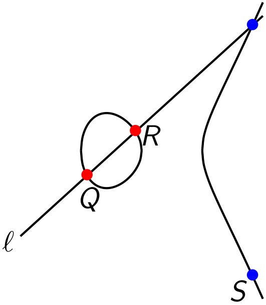
This work: all the tricks

1. The Kummer surface: Gaudry's analogue of Montgomery ladder in genus 1
2. GLV scalar decomposition: genus 2 gets twice as big (dimension) scalar decomposition than genus 1
3. Combine the two?
4. Many other options documented (taxonomy): classic Kummer surface formulas, generic curves, real hyperelliptic curves...

1. The Kummer surface

A wise man once said...

Who needs the y-coordinate?



- Don't use (Q_x, Q_y) and (R_x, R_y) to get (S_x, S_y)
- Instead, use $Q_x, R_x, (Q-R)_x$ to get $(Q+R)_x$
- Enough to define scalar multiplication: Montgomery ladder
- To compute $[k]P$, always keep $Q=[n+1]P$, $R=[n]P$, so we have $Q-R=P$

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

- For $P = (x_P, y_P)$, Montgomery took $P \mapsto P_x$ (two-to-one)

- There is a map $\mathrm{Jac}(C) \to \mathcal{K}$ that is two-to-one

$$\mathcal{K}: \quad (x^4 + y^4 + z^4 + t^4) + 2Exyzt - F(x^2t^2 + y^2z^2) \\ - G(x^2z^2 + y^2t^2) - H(x^2y^2 + z^2t^2) = 0$$

- We lose information, but on the other hand can enjoy beautiful symmetries that exist on $\mathcal{K}$...

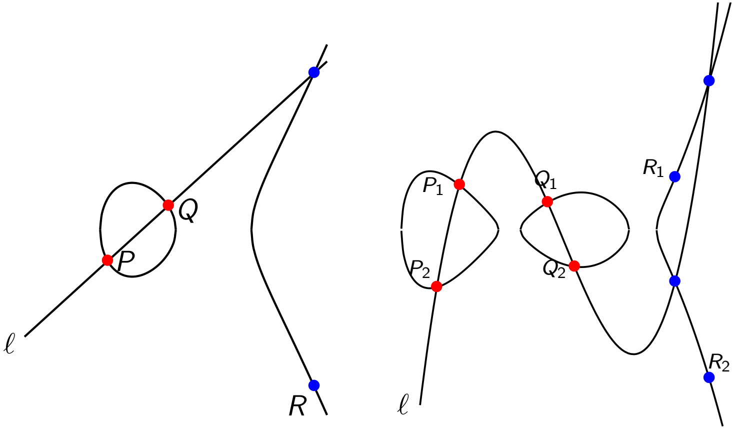
The genus 2 analogue: the Kummer surface $\mathcal{K}$

- e.g. to get from $P = (x, y, z, t)$, $Q = (\underline{x}, \underline{y}, \underline{z}, \underline{t})$, $P - Q = (\overline{x}, \overline{y}, \overline{z}, \overline{t})$ to $P + Q = (X, Y, Z, T)$

$$
\begin{aligned}
x' &= (x^2 + y^2 + z^2 + t^2) \cdot (\underline{x}^2 + \underline{y}^2 + \underline{z}^2 + \underline{t}^2) \\
y' &= (x^2 + y^2 - z^2 - t^2) \cdot (\underline{x}^2 + \underline{y}^2 - \underline{z}^2 - \underline{t}^2) \\
z' &= (x^2 - y^2 + z^2 - t^2) \cdot (\underline{x}^2 - \underline{y}^2 + \underline{z}^2 - \underline{t}^2) \\
t' &= (x^2 - y^2 - z^2 + t^2) \cdot (\underline{x}^2 - \underline{y}^2 - \underline{z}^2 + \underline{t}^2) \\
X &= (x'^2 + y'^2 + z'^2 + t'^2)/\overline{x} \\
Y &= (x'^2 + y'^2 - z'^2 - t'^2)/\overline{y} \\
Z &= (x'^2 - y'^2 + z'^2 - t'^2)/\overline{z} \\
T &= (x'^2 - y'^2 - z'^2 + t'^2)/\overline{t}
\end{aligned}
$$

- Thanks again to Gaudry! (and Chudnovsky brothers, and theta functions)... doubling even nicer!
- $\mathcal{K}$ not a group, but "pseudo-group" - enough to define scalar multiplications via ladder (and do Diffie-Hellman)
- Total per bit (DBL+ADD) of scalar: **25** $\times$ $\mathbb{F}_p$ **multiplications!!!**

Things don't look so bad for $g = 2$ anymore



per bit: $\approx 10 \times$ 256-bit muls vs. $\approx \cancel{50}\ 25 \times$ 128-bit muls

Generic vs. Kummer: $p = 2^{127} - 1$

- `generic1271`: (CM method) $\#\mathrm{J} = 254$ bit prime

$$C/\mathbb{F}_p : y^2 = x^5 + f_3x^3 + f_2x^2 + f_1x + f_0$$

$$\begin{aligned} f_3 &= 34744234758245218589390329770704207149, & f_2 &= 132713617209345335075125059444256188021, \\ f_1 &= 90907655901711006083734360528442376758, & f_0 &= 6667986622173728337823560857179992816. \end{aligned}$$

$$\#\mathrm{J} = 28948022309329048848169239995659025138451177973091551374101475732892580332259$$

- `kummer1271`: (Gaudry-Schost'12) $\#\mathrm{J} = 16 \cdot r$ (251-bit prime)

$$\begin{aligned} \mathcal{K}'/\mathbb{F}_p : E \cdot xyzt - ((x^2 + y^2 + z^2 + t^2) - F(xt + yz) \\ - G(xz + yt) - H(xy + zt))^2 = 0. \end{aligned}$$

$$\begin{aligned} E &= 34744234758245218589390329770704207149, & F &= 132713617209345335075125059444256188021, \\ G &= 90907655901711006083734360528442376758, & H &= 6667986622173728337823560857179992816. \end{aligned}$$

$$\#\mathrm{J} = 2^4 \cdot 1809251394333065553571917326471206521441306174399683558571672623546356726339$$

Generic vs. Kummer: $p = 2^{127} - 1$

- Microsoft's genus 2 library ($\approx$ 128-bit sec) - Intel core i7-3520M (2.90 GHz)
  1. `generic1271`: 352,000 cycles (and $\downarrow$)
  2. `kummer1271`: 150,000 cycles (and $\downarrow$)
  3. ...

2. GLV scalar decomposition

GLV: e.g. Buhler-Koblitz curves

- Let $p = 1 + 2^{64} - 2^{66} + 2^{68} - 2^{70} + 2^{72} + 2^{74} + 2^{76} - 2^{79} + 2^{127}$
- Consider the prime order (254-bit) Buhler-Koblitz curve:
$$C/\mathbb{F}_p : y^2 = x^5 + 17$$
- $\#J = 28948022309328876595115567994214488524823328209723866335483563634241778912751$
- There is a map on C , $\phi : (x, y) \mapsto (\xi_5 x, y)$ where $\xi_5^5 = 1$
- It induces a map on $\mathrm{Jac}(C)$ (Mumford coordinates):
$$\phi : (u_1, u_0, v_1, v_0) \mapsto (\xi_5 u_1, \xi_5^2 u_0, \xi_5^4 v_1, v_0)$$
- For $D \in \mathrm{Jac}(C)$, $\phi(D)$ is a scalar multiple $[\lambda]D$ of D
- Minimal polynomial $\phi^4 + \phi^3 + \phi^2 + \phi + 1$, so $\phi^2(D)$ and $\phi^3(D)$ will also be useful

GLV: e.g. Buhler-Koblitz curves

- Take a random $D = (u_1, u_0, v_1, v_0)$, assume we have to compute the scalar multiplication by $k = 23477399837278936923599493713286470955314785798347519197199578120259089016680$
- The endomorphism ϕ corresponds to multiplication by $\lambda = 78315468676855127052976159806517945867532292413107653204061477837087562856 46$
- So (essentially) for free we get

$$D, \qquad \phi(D) = [\lambda]D, \qquad \phi^2(D) = [\lambda^2]D, \qquad \phi^3(D) = [\lambda^3]D$$

- How best to combine the 4 scalar multiples?... find the minimum k_0, k_1, k_2, k_3 such that

$$[k]D = [k_0]D + [k_1]\phi(D) + [k_2]\phi^2(D) + [k_3]\phi^3(D)$$

GLV: e.g. Buhler-Koblitz curves

- $k = 2347739983727893692359949371328647095531478579834751919719957812025908 9016680$
- Finding k_0, k_1, k_2, k_3 s.t. $[k]D = [k_0]D + [k_1]\phi(D) + [k_2]\phi^2(D) + [k_3]\phi^3(D)$ involves solving a shortest-vector in a lattice problem
- We implement Park-Jeong-Lim (EuroCrypt'02) division in $\mathbb{Z}[\alpha]$ algorithm, so that (in $\approx 20 \times \mathbb{F}_p$ muls), we get

$$k_0 = -6344646642321980551 \quad (63 \text{ bits})$$
$$k_1 = -3170471730617986668 \quad (62 \text{ bits})$$
$$k_2 = -4387949940648063094 \quad (62 \text{ bits})$$
$$k_3 = 3721725683392112311 \quad (62 \text{ bits})$$

- How to proceed?...

GLV: e.g. Buhler-Koblitz curves

- $[k]D = [k_0]D + [k_1]\phi(D) + [k_2]\phi^2(D) + [k_3]\phi^3(D)$
- Stack the binary sequences on top of each other
- Precompute $[[b_0]D, [b_1]D_1, [b_2]D_2, [b_3]D_3]$ for $b_i \in \{0, 1\}$

$$\begin{aligned}
k_0 &= [1,0,1,1,0,0,0,0,0,0,0,1, && \dots (63\ bits) \\
k_1 &= [0,1,0,1,0,1,1,1,1,1,1,1, && \dots (63\ bits) \\
k_2 &= [0,1,1,1,1,0,0,1,1,1,0,0, && \dots (63\ bits) \\
k_3 &= [0,1,1,0,0,1,1,1,0,1,0,0, && \dots (63\ bits)
\end{aligned}$$

- Instead of 254 doublings and approx. 127 additions, we have 63 doublings and 80 additions
- Thanks to Patrick for point out: if window size is bigger than dimension of decomposition (e.g. $w > 4$), windowing is faster nice!

Generic vs. Kummer vs. GLV

- Microsoft's genus 2 library ($\approx$ 128-bit sec) - Intel core i7-3520M (2.90 GHz)
    i. `generic1271`: 352,000 cycles (and $\downarrow$)
    ii. `kummer1271`: 150,000 cycles (and $\downarrow$)
    iii. `GLV4-127eps`: 180,000 cycles (and $\downarrow$)
    iv. ...

3. GLV on the Kummer surface (the Holy Grail in genus 2?)

Endomorphisms on the Kummer surface

- Using the **Kummer surface** improved cycles from 352,000 to 150,000
- Exploiting **endomorphisms** improved cycles from 352,000 to 180,000
- Natural question: what if there were **endomorphisms** we could exploit on the **Kummer surface**?

Endomorphisms on the Kummer surface

- Again, Gaudry to the rescue: he saw an endomorphism that could possibly exist
- Consider the doubling $[2](x, y, z, t) = (X, Y, Z, T)$ on $\mathcal{K}$

$$\begin{aligned}
x' &= (x^2 + y^2 + z^2 + t^2) \\
y' &= y_0'(x^2 + y^2 - z^2 - t^2) \\
z' &= z_0'(x^2 - y^2 + z^2 - t^2) \\
t' &= t_0'(x^2 - y^2 - z^2 + t^2) \\
X &= (x'^2 + y'^2 + z'^2 + t'^2) \\
Y &= y_0(x'^2 + y'^2 - z'^2 - t'^2) \\
Z &= z_0(x'^2 - y'^2 + z'^2 - t'^2) \\
T &= t_0(x'^2 - y'^2 - z'^2 + t'^2)
\end{aligned}$$

where $y_0', z_0', t_0', y_0, z_0, t_0$ are all constants that depend on the Kummer surface.

- What if we can find a Kummer with $y_0' = y_0$, $t_0' = t_0$, $z_0' = z_0$?
- Then doubling is the same operation on top of itself

Endomorphisms on the Kummer surface

- Again, Gaudry to the rescue: he saw an endomorphism that could possibly exist
- Consider the doubling $[2](x, y, z, t) = (X, Y, Z, T)$ on $\mathcal{K}$

$$
\begin{aligned}
x' &= (x^2 + y^2 + z^2 + t^2) \\
y' &= y_0(x^2 + y^2 - z^2 - t^2) \\
z' &= z_0(x^2 - y^2 + z^2 - t^2) \\
t' &= t_0(x^2 - y^2 - z^2 + t^2)
\end{aligned}
$$

pause

$$
\begin{aligned}
X &= (x'^2 + y'^2 + z'^2 + t'^2) \\
Y &= y_0(x'^2 + y'^2 - z'^2 - t'^2) \\
Z &= z_0(x'^2 - y'^2 + z'^2 - t'^2) \\
T &= t_0(x'^2 - y'^2 - z'^2 + t'^2)
\end{aligned}
$$

where $y_0', z_0', t_0', y_0, z_0, t_0$ are all constants that depend on the Kummer surface.

- What if we can find a Kummer with $y_0' = y_0$, $t_0' = t_0$, $z_0' = z_0$?
- Then doubling is the same operation on top of itself
- i.e. $\phi(\phi(P)) = [2]P$, so we must have $\phi = [\sqrt{2}]$ endo.

What curves can have this nice property?

- If these parameter choices on $\mathcal{K}$ imply $[\sqrt{2}]$ endomorphism on $\mathcal{K}$, then ...
- ... perhaps all families whose Jacobians have RM by $\sqrt{2}$ can find $\mathcal{K}$'s with this endomorphism
- **TRUE!** many such "families"
- e.g. Van-Wamelen family with quartic CM field $\mathbb{Q}(\sqrt{-2+\sqrt{2}})$

$$C_{VW} : y^2 = -x^5 + 3x^4 + 2x^3 - 6x^2 - 3x + 1.$$

gives $\mathcal{K}$ with $y_0' = y_0$, $t_0' = t_0$, $z_0' = z_0$ and therefore $\phi = [\sqrt{2}]$ endomorphism on $\mathcal{K}$

Endomorphisms on the Kummer surface

- To compute $[k]P$ on $\mathcal{K}$, compute $Q = \phi(P) = [\sqrt{2}]P$ decompose as

$$[k]P = [k_0]P + [k_1]Q,$$

where k_0, k_1 are both half the size of k.

- Beware: can't compute regular additions on $\mathcal{K}$, must use 2-dimensional differential addition chain to compute $[k_0]P + [k_1]Q$
- Many fewer operations than $[k]P$... this is the hope
- Such a chain needs as input P (got it), Q (got it) **and** $Q - P$ (need it)
- **My main Summer headache: what is $Q - P$ (rephrase: how does $(\phi - 1)$ act on $\mathcal{K}$)**

What I said I'd do...

- **Weeks 1-3:** Point counting!
  - Implement genus 2 CM method that crawls (Igusa class polynomial) databases: Kohel's Echidna database
  - Special (fast) point counting algorithms for Buhler-Koblitz ($y^2 = x^5 + a$) and FKT ($y^2 = x^5 + ax$) curves
  - Find optimal curves to match Joppe/Huseyin's favorite primes!
- **Weeks 4:** Implement GLV algorithms
  - Write algorithm for decomposition $k \mapsto k_0, k_1, k_2, k_3$ etc
  - Implement 4 dimension GLV on BK $y^2 = x^5 + a$
  - Implement 2 and 4 dimension GLV on BK $y^2 = x^5 + ax$
- **Week 5-8:** Kummer ... -Kummer-Kummer-Kummer-Kummer-chameleonnnn
  - Optimized implementation on curve we found (standard)
  - Same, but with squares-only and Gaudry-Schost curve
  - Investigate hybrid GLV/Kummer
- **Week 9-12:** Write up
  - Target Eurocrypt submission ($\approx$ one month)

What I actually did. . .

- **Weeks 1-3:** Point counting!
  - Implement genus 2 CM method that crawls (Igusa class polynomial) databases: Kohel's Echidna database
  - Special (fast) point counting algorithms for Buhler-Koblitz ($y^2 = x^5 + a$) and FKT ($y^2 = x^5 + ax$) curves
  - Find optimal curves to match Joppe/Huseyin's favorite primes!
- **Weeks 4:** Implement GLV algorithms
  - Write algorithm for decomposition $k \mapsto k_0, k_1, k_2, k_3$ etc
  - Implement 4 dimension GLV on BK $y^2 = x^5 + a$
  - Implement 2 and 4 dimension GLV on BK $y^2 = x^5 + ax$
- **Week 5-8:** Kummer . . . -Kummer-Kummer-Kummer-Kummer-chameleonnnn
  - Optimized implementation on curve we found (standard)
  - Same, but with squares-only and Gaudry-Schost curve
  - Investigate hybrid GLV/Kummer
- **Week 9-12:** Investigate hybrid GLV/Kummer
  - *Become increasingly angry that I can't find* ($\phi - 1$) *map*
  - *Crash Maple trying to solve 4 equations in 4 unknowns* (not what it sounds like)
  - *Crash Magma trying to solve 4 equations in 4 unknowns*

Timeline cont. . .

- **Weeks 13 $\rightarrow$ deadline (and beyond):**
  - Write the paper! (subtext: find the map)
- Other things we did (see the paper):
  - Analytic theory: discovered new ways to define Kummer surface directly from period matrix (possibly avoids cofactor 16 imposed by Gaudry / Rosenhain invariants) also finds nice surfaces
  - Classical Kummer surface: maps $\mathrm{Jac}(C) \leftrightarrow \mathcal{K}_{classic}$ so much nicer (formulas slower though)
  - Real and imaginary generic curves (improved formulas)

Summary. . .

- Microsoft's genus 2 library fastest in world
- Also beats all benchmarks (over all curves) on eBACS (Bernstein's `curve25519`:180,000+ cycles, etc)
- **Final frontier**: can we push Kummer (150,000 cycles) past Longa/Sica-4GLV (145,000) to have fastest over prime fields?
- . . . suggest standards???