Efficient arithmetic on Jacobians of genus 2 curves

Craig Costello

Microsoft Research

Joint work with Joppe Bos, Huseyin Hisil and Kristin Lauter

January 12, 2013

AMS Joint Mathematics Meeting: San Diego, California

“Exponentiation” in groups is a fundamental operation in public-key cryptography

Discrete log based cryptography

- Take a (1024-bit) prime field $\mathbb{F}_q$, say

  $q =$

  $$179769313486231590772930519078902473361797697894230657273430081157732675805500963132708477322407536021120113879871393357658789768814416622492847430639474124377678934248654852763022196012460941194530829520850057688381506823424628814739131105408272371633505106845862982399472459384797163048353563296242241371111.$$

- Fix a generator $g \in \mathbb{F}_q$, say

  $g =$

  $$162149867078079733630570758365544581305875184269738045309267705124567666433207907274753905887825617560140047792693686357414478680328765786743749905370642417146006197307493686682988862410465086752278021697861829242290995507250313964248120082428034983083395177990742092372164638120893361575414427509982998153428.$$

- Choose a secret integer $0 < x < q$, say

  $x =$

  $$572871336499030024195022656911370864896103030777418241424051952387848976636846544947052524342960794029816515311002229966126084761065619088768613585981403294598890771814353731461999497073234268403770849337458446541268947397790519702411815948607604731745266637308807825500763821233543211163112011393469929719 8.$$

- Compute the exponentiation

  $g^x =$

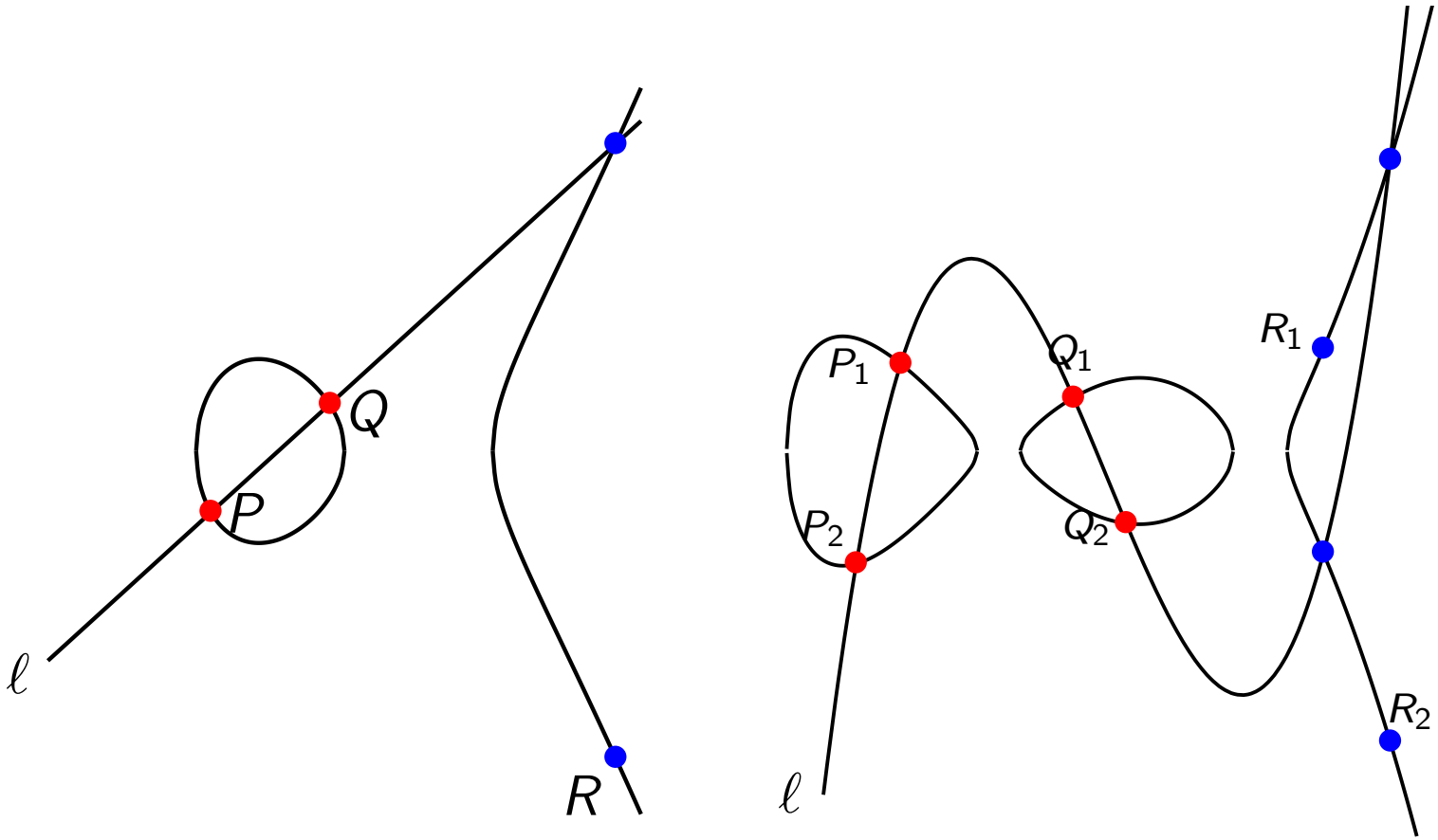
  $$171484610804308856443973256848864205154610094759100481705137473117260957903770235830205143005300918079799136878356275679370350702779083406294418921583985504920709760577750688303817508395654738131228965513407248476923986159686602053393567880169894204519748685033607634906681195287996116989213594533662093471782.$$

Elliptic curve cryptography

- Take a (192-bit) field $\mathbb{F}_q$, say $q =$
  $$6277101735386680763835789423207666416083908700390324961279.$$
- Define a curve $E/\mathbb{F}_q : y^2 = x^3 - 3x + b$ with $b =$
  $$2455155546008943817740293915197451784769108058161191238065.$$
- E has cardinality $\#E =$
  $$6277101735386680763835789423176059013767194773182842284081.$$
- Fix a generator $P \in E(\mathbb{F}_q)$, $P = (P_x, P_y) =$
  $$\big(1978277224697059351202407728823149726111822159014162237135\ ,$$
  $$5597504791900977375028931377814390220171287037426932206878\big)$$
- Choose a secret exponent $x =$
  $$5987741930106895334218117330943111499705775828194129200173.$$
- Compute the "exponentiation" $Q = [x]P = (Q_x, Q_y) =$
  $$\big(3415048339127597876194206375324953976809958263486590111844,$$
  $$3762042547166032422588171589771998806109105333135206356446\big)$$

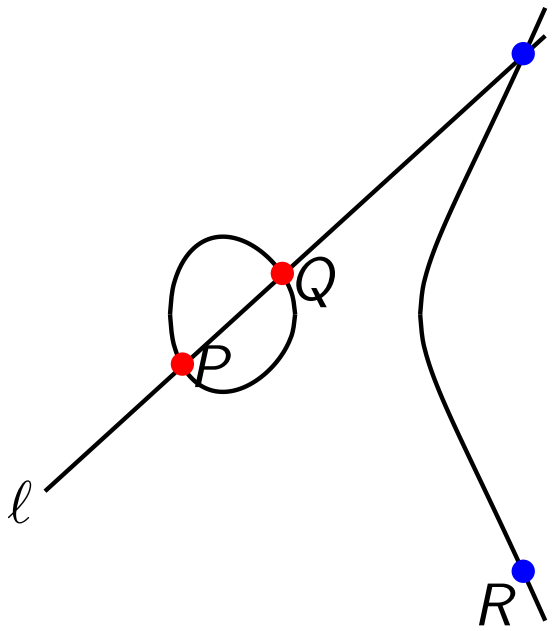
Why genus 2?

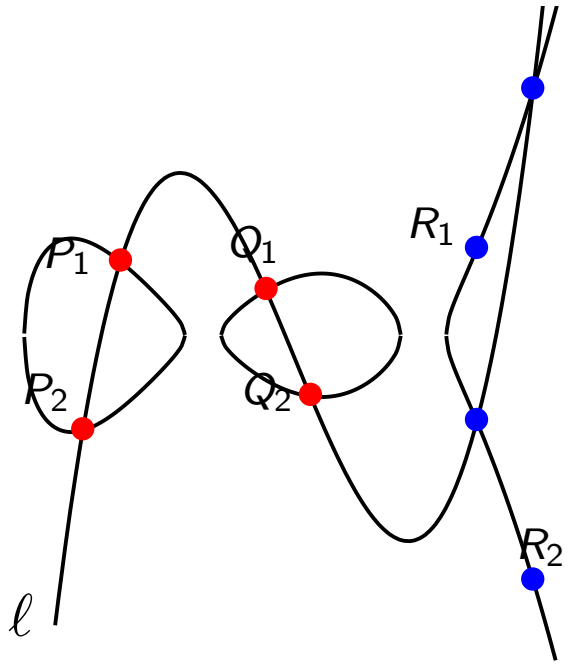
- Everything is so much more complicated in genus 2



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

Why genus 2? (cont)





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

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

- $\#E$ and $\#C$ are close over same size field $\mathbb{F}_q$ ...
- **Elliptic (genus 1) group size $\approx q$**
- **Hyperelliptic (genus 2) Jacobian group size $\approx q^2$**

Fields of half the size . . .

- **g=1**: Bernstein's curve25519: $E/\mathbb{F}_p : y^2 = x^3 + \ldots$ over
$$p = 2^{255} - 19 =$$
57896044618658097711785492504343953926634992332820282019728792003956564819949
has group order $\#E = 2^3 \cdot$
72370055773322622139731865630429942408571163593799076060019509382854542509 89 (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 $\#\mathrm{Jac}(C) =$
1157920892373161954293422038010335541709316156518816573073080680797020899517 81 (257 bits)

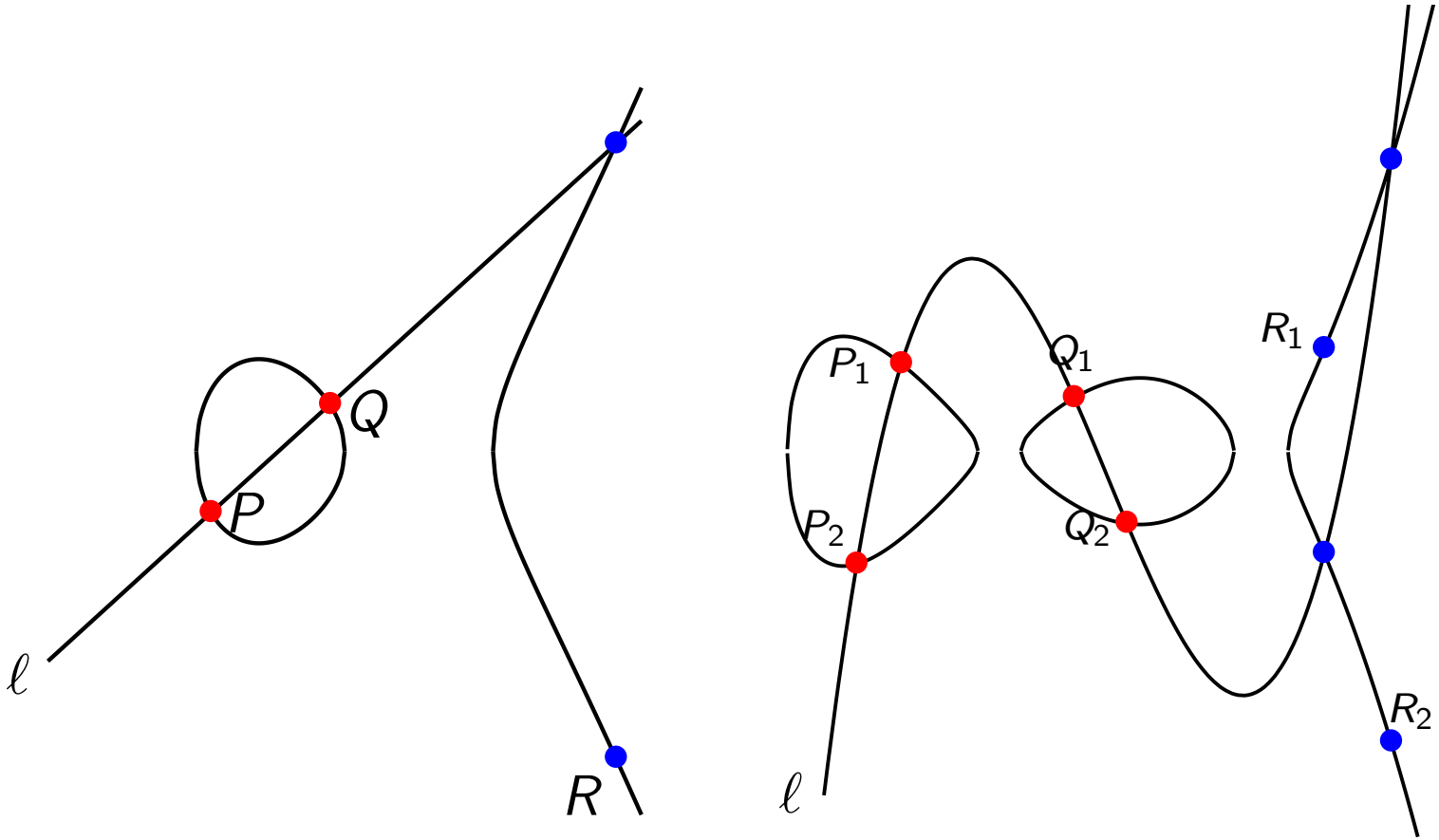
Genus 1 vs. Genus 2

- $g = 2$ group elements require 4 coordinates, whilst $g = 1$ group elements require 2 coordinates (of twice the size)
  $\rightarrow$ **same key sizes/storage**
- same attack complexity (generic Pollard ρ - exponential)
- scalars (exponents) are of the same size. . . so . . .

Comparison boils down to efficiency . . .

in which setting can we compute $[k]D$ faster?

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

This work: all the best (known) tricks in genus 2

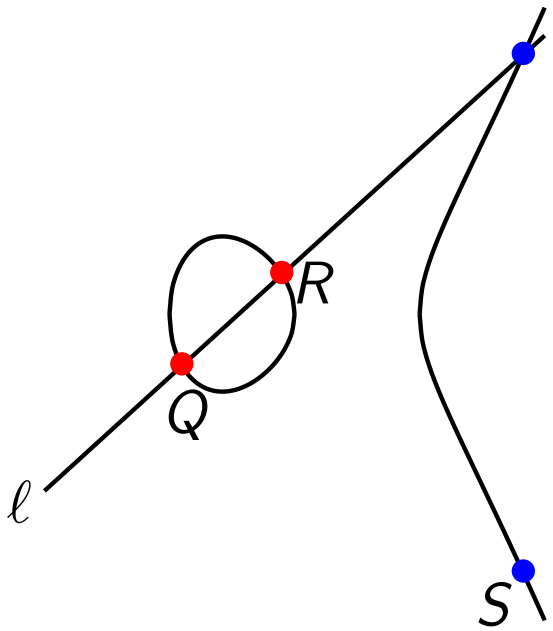
Including. . .

1. The Kummer surface: analogue of Montgomery ladder in genus 1
2. GLV scalar decomposition: genus 2 gets twice as big (dimension) scalar decomposition than genus 1
3. Open problem: combine the above?
4. Much more (see `http://eprint.iacr.org/2012/670.pdf`)

1. The Kummer surface

Montgomery ladder for elliptic curves . . .

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

$$C: \quad y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu)$$

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

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

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

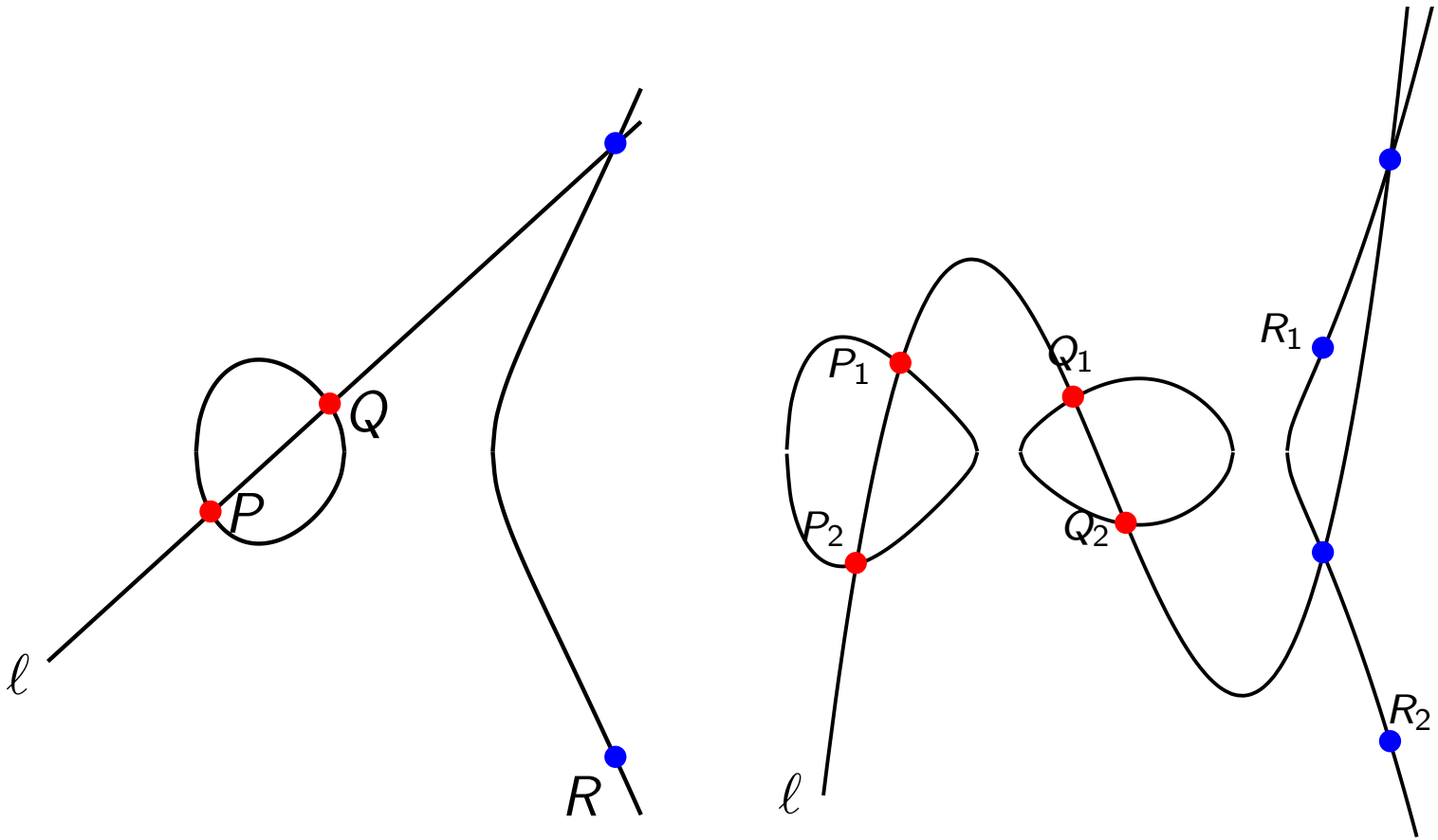
- e.g. to get from $P = (x, y, z, t)$, $Q = (\underline{x}, \underline{y}, \underline{z}, \underline{t})$, $P - Q = (\overline{x}, \overline{y}, \overline{z}, \overline{t})$ to $P + Q = (X, Y, Z, T)$

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

- Comes from identities on theta functions . . . doubling even nicer!
- $\mathcal{K}$ not a group, but "pseudo-group" - enough to define scalar multiplications via ladder (and do crypto. key exchange)
- 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. $\approx$ ~~50~~ 25 $\times$ 128-bit muls

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

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

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

$f_3 = 347442347582452185893903297770704207149,$ $f_2 = 132713617209345335075125059444256188021,$
$f_1 = 909076559017110060837343605284423767 58,$ $f_0 = 666798662217372833782356085717 9992816.$

$\#\mathrm{J} = 289480223093290488481692399956590251384511779730915513741014757328925803322 59$

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

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

$E = 347442347582452185893903297770704207149,$ $F = 132713617209345335075125059444256188021,$
$G = 909076559017110060837343605284423767 58,$ $H = 666798662217372833782356085717 9992816.$

$\#\mathrm{J} = 2^4 \cdot 18092513943330655535719173264712065214413061743996835585716726235463567263 39$

Numbers for implementations over prime fields

- Previous implementations in genus 1 ($\approx$ 128-bit sec) - Intel core i7-3520M (2.90 GHz)
  i. NIST `p256` (generic): 658,000 cycles
  ii. Bernstein's `curve25519` (Montgomery): 182,000 cycles
  iii. Longa-Sica `2GLV`: 145,000 cycles

- Our implementations in genus 2 ($\approx$ 128-bit sec) - Intel core i7-3520M (2.90 GHz)
  i. `generic1271`: 248,000 cycles
  ii. `kummer1271`: 117,000 cycles
  iii. ...

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
  $k = 234773998372789369235994937132864709553147857983475191971995781202590890166 80$
- The endomorphism ϕ corresponds to multiplication by
  $\lambda = 783154686768551270529761598065179458675322924131076532040614778370875628 5646$
- 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

- $k =$ 23477399837278936923599493713286470955314785798347519197199578120259089016680
- 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
- In $\approx 20 \times \mathbb{F}_p$ muls (less than a group operation), we get

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

- 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 $b_i \in \{0, 1\}$

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

- Instead of 254 doublings and approx. 127 additions, we have 63 doublings and 80 additions

Numbers for implementations over prime fields

- Previous record implementations in genus 1 ($\approx$ 128-bit sec) - Intel core i7-3520M (2.90 GHz)
  i. NIST `p256` (generic): 658,000 cycles
  ii. Bernstein's `curve25519` (Montgomery): 182,000 cycles
  iii. Longa-Sica `2GLV`: 145,000 cycles

- Our implementations in genus 2 ($\approx$ 128-bit sec) - Intel core i7-3520M (2.90 GHz)
  i. `generic1271`: 248,000 cycles
  ii. `BK-4GLV`: 156,000 cycles
  iii. `kummer1271`: 117,000 cycles

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

Endomorphisms on the Kummer surface

- Using the **Kummer surface** improved cycles from 248,000 to 117,000
- Exploiting **endomorphisms** improved cycles from 248,000 to 156,000
- Natural question: what if there were **endomorphisms** we could exploit on the **Kummer surface**?

Endomorphisms on the Kummer surface

- Gaudry pointed out 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 twice

Endomorphisms on the Kummer surface

- 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.

How to find curves whose Kummers have *RM* by $\sqrt{2}$???

- If these parameter choices on $\mathcal{K}$ imply $[\sqrt{2}]$ endomorphism on $\mathcal{K}$, then ...
- ...perhaps some families whose Jacobians have RM by $\sqrt{2}$ can find $\mathcal{K}$'s with this endomorphism
- 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 $[k]P$
- Such a chain needs as input P, Q **and** $Q - P$
- **Open problem: what is $Q - P$**
  (rephrase: how does $(\phi - 1)$ act on $\mathcal{K}$)

Summary...

Genus 2 currently offers the fastest side-channel resistant scalar multiplications for 128-bit secure scalar multiplications

see "Two is Greater than One"
`http://eprint.iacr.org/2012/670.pdf`

Follow-up work: see http://eprint.iacr.org/2013/???.pdf

- All this work was over prime fields $C/\mathbb{F}_p$ with $2^{126} < p < 2^{128}$
- Working over quadratic extension fields allows us to take $C/\mathbb{F}_p$ with $p < 2^{64}$ (fields fit into one computer word)
- Small fields: particularly suitable to embedding devices using ARM (32-bit) processors (tablets/smartphones)
- In addition, Frobenius endomorphism π_q is no longer trivial so can be used to give 8-dimensional GLV
- Wary of Weil Descent attacks - discrete log in genus 2 over $\mathbb{F}_{p^2}$ could possibly be transformed to DLP in genus 4 over $\mathbb{F}_p$ (faster than exponential attacks)

Thanks!