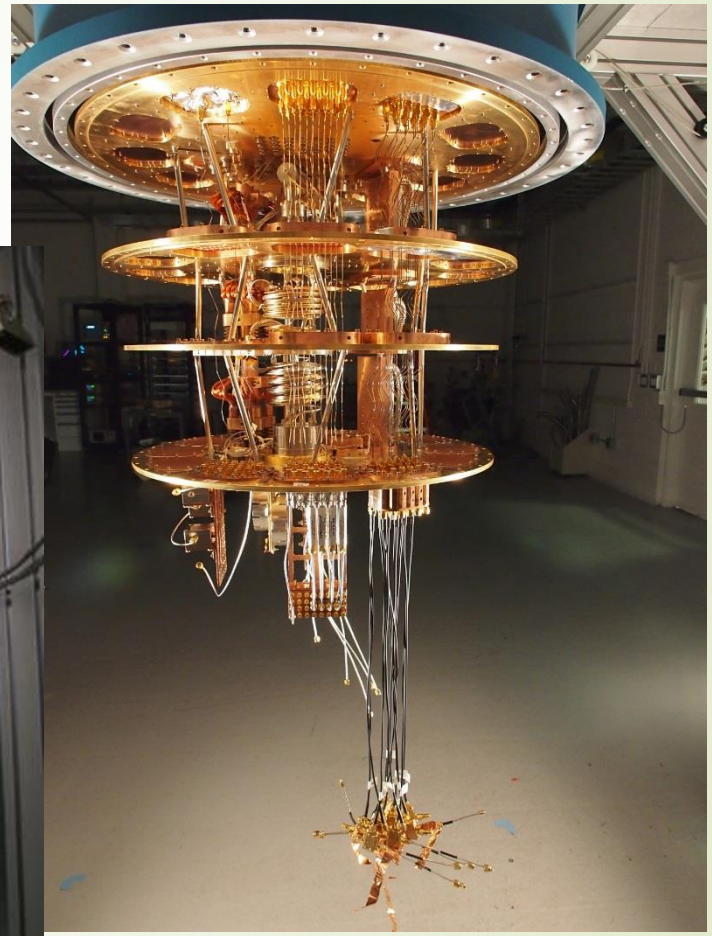
Fast Arithmetic Modulo $2^x p^y \pm 1$

Joppe W. Bos and **Simon Friedberger**

Why these strange primes?

- Quantum computers
- NIST call for PQC standards

[2]



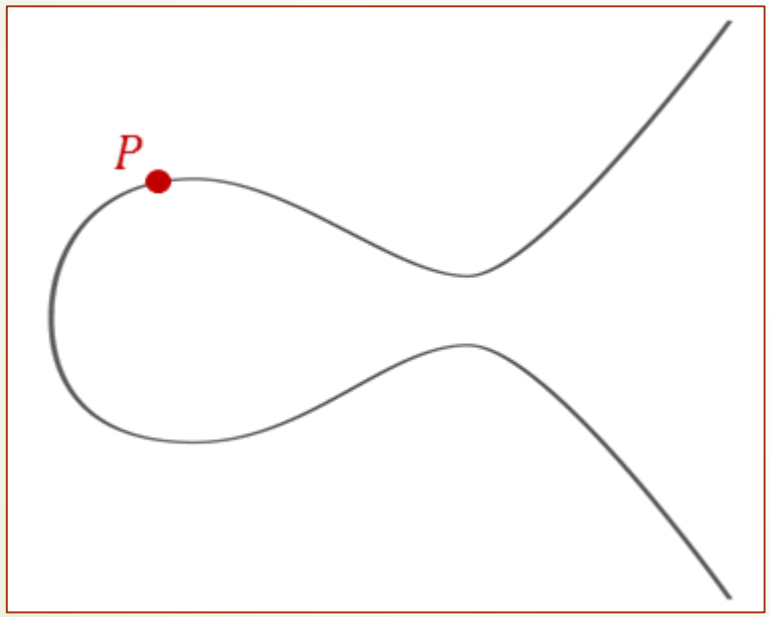
[1]

Post-Quantum Cryptography

- Lattice-based
- Code-based
- MQ-based
- Hash-based
- **Isogeny-based**
  - Little data (330 B / 10 x smaller)
  - Very slow (1000 x slower)
  - Requires more cryptanalysis (published 2011)
  - …but it has elliptic curves!

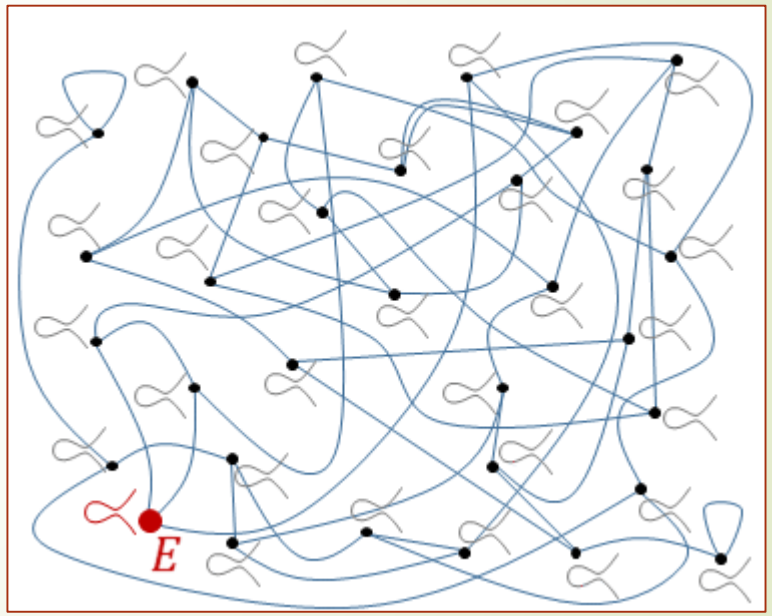
ECC vs SIDH

ECC $[n]P = Q$



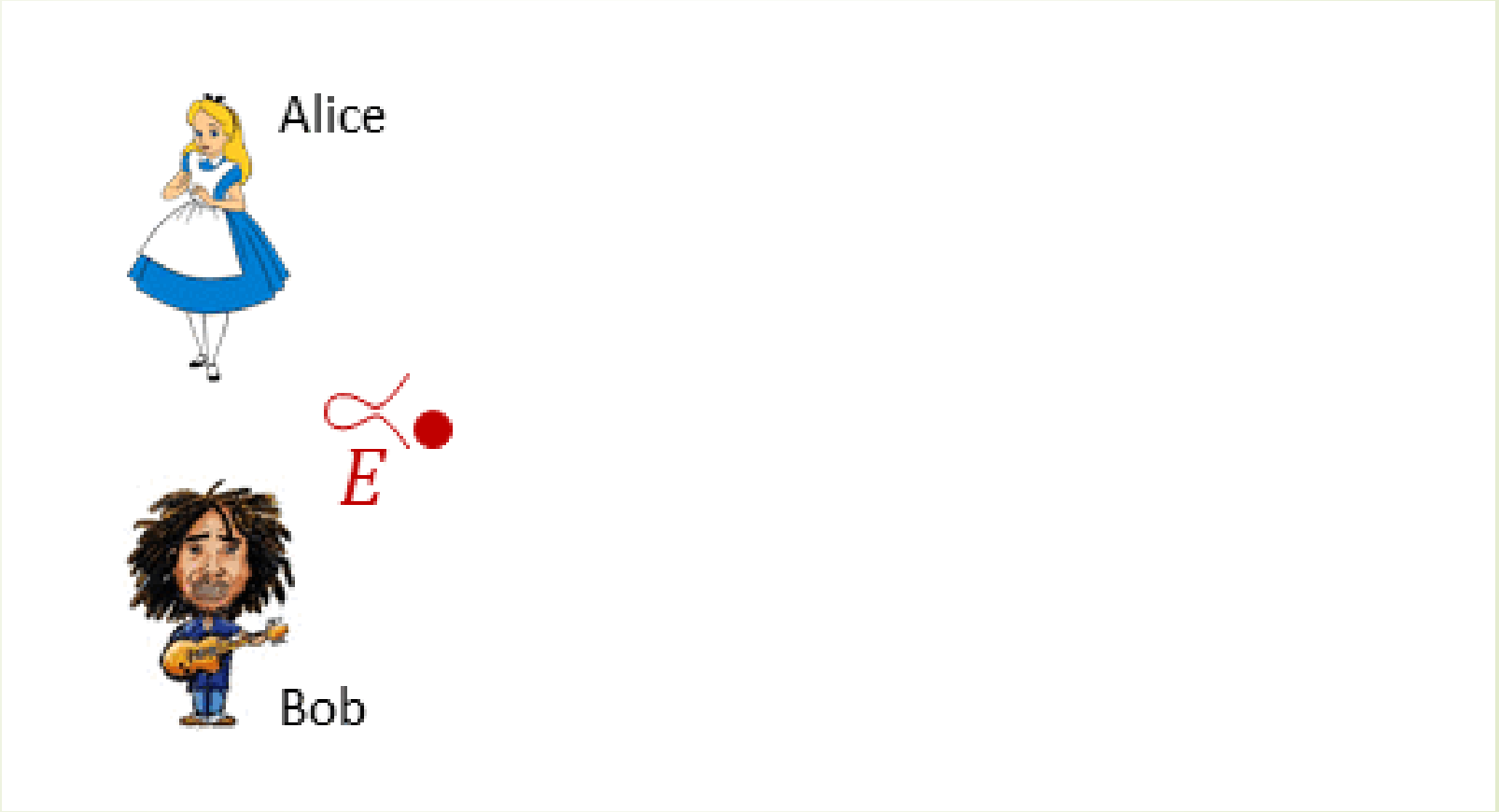
[3]

SIDH $E_2 = \Phi_n \circ \cdots \circ \Phi_1(E_1)$



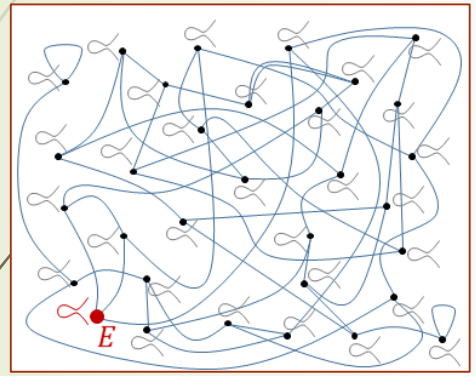
[3]

Key exchange



[3]

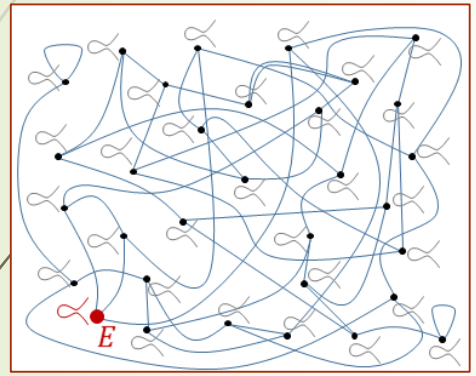
Fast Arithmethic modulo $2^x p^y \pm 1$



$$\#\mathrm{E}(\mathbb{F}_{q^2}) = (2^x p^y)^2$$

$$\mathrm{q} = 2^{\mathrm{x}}\mathrm{p}^{\mathrm{y}} \pm 1$$

Fast Arithmethic modulo $2^x p^y \pm 1$



$$\#\mathrm{E}(\mathbb{F}_{q^2}) = (2^x p^y)^2$$

$$\mathrm{q} = 2^{\mathrm{x}}\mathrm{p}^{\mathrm{y}} \pm 1$$

Compared approaches

- Montgomery reduction
- Barrett division
- Modular simplification
- Shifting
- Special radix
- …

Montgomery reduction

- Calculate $\tilde{a}\tilde{b}/R = abR \; mod \; m$
- Montgomery multiplication
$$cR^{-1} = (c + (\mu ab \; mod \; R)m)/R \; (mod \; m)$$
- Prime shape optimizations:
  - $\mu = -m^{-1} \equiv 1$ *for* $m \equiv \pm 1$
  - $xm = x(2^x p^y \pm 1) = (xp^y)2^x \pm x$
- Costs $n^2 + n$ optimized to $\frac{n^2}{2}$M

Montgomery reduction

- Calculate $\tilde{a}\tilde{b}/R = abR \; mod \; m$
- Montgomery multiplication
$$cR^{-1} = (c + (\mu ab \; mod \; R)m)/R \; (mod \; m)$$
- Prime shape optimizations:
  - $\mu = -m^{-1} \equiv 1$ for $m \equiv \pm 1$
  - $xm = x(2^x p^y \pm 1) = (xp^y)2^x \pm x$
- Costs $n^2 + n$ optimized to $\frac{n^2}{2}$M

Montgomery reduction

- Calculate $\tilde{a}\tilde{b}/R = abR \bmod m$
- Montgomery multiplication
$$cR^{-1} = (c + (\mu ab \bmod R)m)/R \pmod m$$
- Prime shape optimizations:
  - $\mu = -m^{-1} \equiv 1$ *for* $m \equiv \pm 1$
  - $xm = x(2^x p^y \pm 1) = (xp^y)2^x \pm x$
- Costs $n^2 + n$ optimized to $\frac{n^2}{2}\mathrm{M}$

Barrett division

- Calculate $c \bmod m$ as $c - \lfloor c/m \rfloor m$
- Approximate $\left\lfloor \frac{c}{m} \right\rfloor$ as $\left\lfloor \frac{c}{R} \left\lfloor \frac{R}{m} \right\rfloor \right\rfloor$
- Error of at most m, or at most $3m$ after some more optimizations
- Also gives the fraction not just the remainder

- Costs $n^2 + 4n + 1$ optimized to $\frac{5}{8}n^2 + \frac{13}{4}n + 1\ M$

Barrett division

- Calculate $c \; mod \; m$ as $c - \lfloor c/m \rfloor m$
- Approximate $\left\lfloor \frac{c}{m} \right\rfloor$ as $\left\lfloor \frac{c}{R} \left\lfloor \frac{R}{m} \right\rfloor \right\rfloor$
- Error of at most m, or at most $3m$ after some more optimizations
- Also gives the fraction not just the remainder

- Costs $n^2 + 4n + 1$ optimized to $\frac{5}{8}n^2 + \frac{13}{4}n + 1 \; M$

Simplified Modulus

- Pick $R = m + 1 = 2^x p^y$
- $c = c_1 R + c_0 = c_1 m + c_1 + c_0 \equiv c_1 + c_0$
- Need to divide $\frac{c}{R}$ and suppose $R = 2^x R'$

- Idea: Use Barrett division with special modulus
- If $c = c_1' 2^x + c_0'$ and $c_1' = uR' + v$ it follows that
- $c = u2^x R' + v2^x + c_0{}'$
- It follows that $v2^x + c_0' = c_0$ and $u = c_1$

- Cost $\mathcal{B}\left(\frac{3}{2}n, \frac{1}{2}n\right) = \frac{5}{8}n^2 + \frac{13}{4}n + 1\ M$

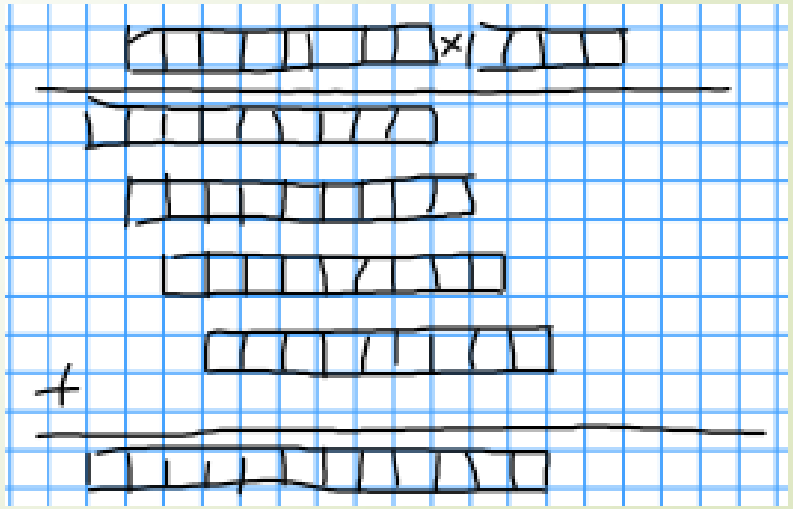
Folding

- Save time on the reduction by computing a multiplication first
- With precomputed $\mu = R \; mod \; m$
- Transform $c = c_1 R + c_0$ it is clear that $c \equiv c_1 \mu + c_0 \; (mod \; m)$
- Picking R appropriately will reduce the size of the number to reduce

- Costs: For R 1.5 times as long as m we get
- c is reduced in length by 25 %
- Cost $\frac{n^2}{2} M$
- Folding + Barrett Cost $\frac{n^2}{2} + \frac{5}{4} n + 1 \; M$

Interleaved vs Non-interleaved

- Interleave multiplication and reduction
  - Uses less memory
- Multiply and reduce separately
  - Allows asymptotically fast multiplication algorithms
- SIDH: Arithmetic in $\mathbb{F}_{q^2}$
  - $(a + ib)(c + id)$
  - Interleaved: 4 M&R, Non-interleaved: 4 M + 2 R
  - Using Karatsuba: 3 M&R vs 3 M + 2 R
- Non-interleaved is to be preferred for SIDH



Modulus based Radix

- Recent approach from WAIFI
- Pick $R = \sqrt{m}$ and representation $a = a_1 R + a_0$ this gives
- $ab = a_1 b_1 R^2 + (a_1 b_0 + a_0 b_1)R + a_0 b_0 = (a_1 b_0 + a_0 b_1)R + a_1 b_1 + a_0 b_0$
- Reduce both parts again using Barrett division

- Costs: $\frac{17}{16} n^2 + \frac{13}{4} n + 2\ M$

- Unfortunately interleaved

Results (interleaved)

approach	moduli family	# muls
Montgomery	generic	$2n^2 + n$
	$2^x p^y - 1$	$\frac{3}{2}n^2$
use radix directly	$2 \cdot 2^x 3^y - 1$	$\frac{17}{16}n^2 + \frac{13}{4}n + 2$
use radix directly	$2^x p^y - 1$	$\frac{39}{32}n^2 + \frac{39}{8}n + 3$

(Costs for multiplication and reduction)

Results (non-interleaved)

approach	moduli family	# muls
Barrett	generic	$n^2 + 4n + 1$
	$2^x p^y \pm 1$	$\frac{5}{8}n^2 + \frac{13}{4}n + 1$
Montgomery	generic	$n^2 + n$
	$2^x p^y - 1$	$\frac{n^2}{2}$
use radix directly	$2^x p^y - 1$	$\frac{1}{2}n^2 + \frac{5}{4}n + 1$

(Costs for reduction only)

Shifting

- $2^{372}3^{239} - 1$
- $2^{372}3^{239}$ has 372 zero bits
- 5 words of 64 bit and another 52 bits
- 3^{239} fits into 6 words but it actually uses 7 now
- We can properly align the powers of three

- Costs: several shifts by 52 bits

SIDH friendly primes

- Conditions for our search

1. $p \in \{3,5,7,11,13,17,19\}$
2. $384 \leq x < 450$ and $2^{300} < p^y < 2^{450}$
3. $2^{740} < 2^x p^y \pm 1 < 2^{768}$
4. $|2^x - p^y| < 2^{40}$
5. $2^x p^y + 1$ or $2^x p^y - 1$ is prime

New prime suggestions

Prime	Security
$\mathbf{2^{385}3^{227}-1}$	**120**
$2^{394}5^{154}+1$	119
$2^{394}5^{155}-1$	120
$2^{396}7^{131}+1$	123
$2^{393}17^{91}+1$	124
$\mathbf{2^{391}19^{88}-1}$	**125**

Benchmarking results

	#cycle	#mul	#add	#mov	#other
$2^{372}3^{239} - 1$ $(B = 1)$	254.9 ± 9.5	84	332	157	41
$2^{372}3^{239} - 1$ $(B = 2)$	275.3 ± 11.2	84	358	202	59
$2^{372}3^{239} - 1$ (shifted)	240.2 ± 10.9	72	299	223	85
$2^{391}19^{88} - 1$	224.5 ± 8.8	72	292	145	38

Questions?

https://github.com/sidh-arith/

References

1. https://www.technologyreview.com/s/602283/googles-quantum-dream-may-be-just-around-the-corner/
2. https://bits.blogs.nytimes.com/2013/05/16/google-buys-a-quantum-computer/?_r=0
3. https://www.esat.kuleuven.be/cosic/elliptic-curves-are-quantum-dead-long-live-elliptic-curves/