Selecting Elliptic Curves for Cryptography
"Real World" Issues

Michael Naehrig
Cryptography Research Group
Microsoft Research

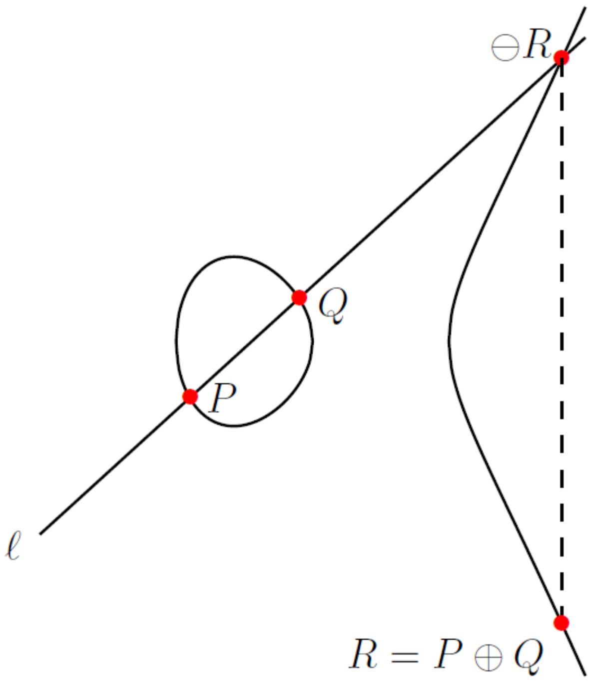
UW Number Theory Seminar
Seattle, 28 April 2015

Elliptic Curve Cryptography

- 1985: Neal Koblitz and Victor Miller propose to use elliptic curves for designing public-key crypto systems
- For example: key exchange and digital signatures

$$E: y^2 = x^3 + ax + b,$$
$$a, b \in \mathbb{F}_p, \mathrm{char}(\mathbb{F}_p) > 3$$

Elliptic Curve Cryptography



- Use group of rational points $E(\mathbb{F}_p)$ on E over finite field $\mathbb{F}_p$

- Fundamental operation: $(k, P) \mapsto [k]P$
  i.e. ``double-and-add':
  $k = (1, 0, 1, \ldots, 0, 0) \rightarrow (-, DBL, DBL + ADD, \ldots, DBL, DBL)$

- Security related to hardness of the discrete logarithm problem
  i.e. find k given $P, Q = [k]P$.

Why Elliptic Curves?

- Functionality: Can realize key exchange, encryption, signatures
- Security:
  - Best known algorithm for solving ECDLP is Pollard's rho
  - Expected run time $\sqrt{\pi r/4}$ in a subgroup of prime order r
- Performance:
  - Efficient representation of group elements
  - Efficient group operations and exponentiation
  - Much smaller key sizes than RSA or DL in finite fields

Why Elliptic Curves?

- Roughly equivalent levels of security

Security level	Symmetric Algorithms	RSA/ Finite Field DL	ECC
128 bits	AES-128, SHA-256	3072 bit modulus/field size	256 bit field size

- See various (slightly different) recommendations on http://www.keylength.com.

Standards – The NIST Curves

- (1999/2000) NIST standardizes a collection of elliptic curves
- For example P-256 given by $E : y^2 = x^3 - 3x + b$ modulo

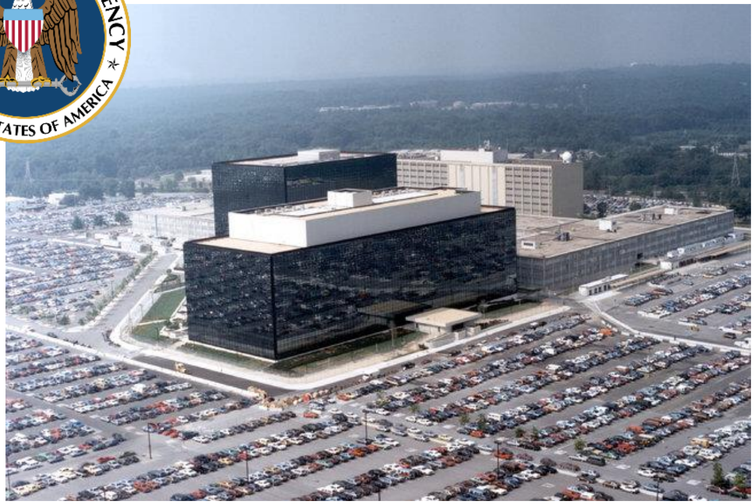
$$p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$$

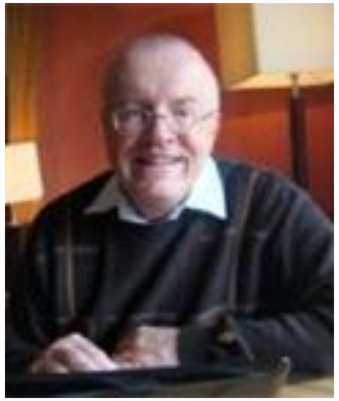
- with 256-bit prime order $r = \#E(\mathbb{F}_p)$, where

$$b = \sqrt{-27/\mathrm{SHA1}(s)},$$
$$s = \text{c49d360886e704936a6678e1139d26b7819f7e90}$$

- … so the curve is "verifiably random"…

One in a million?





“Consider now the possibility that one in a million of all curves have an exploitable structure that "they" know about, but we don't. Then "they" simply generate a million random seeds until “they” find one that generates one of "their" curves…
… So, sigh, why didn't they do it that way? Do they want to be distrusted?"

Mike Scott ‘99

Other voices

- 2008 – Koblitz and Menezes: “However, in practice the NSA has had the resources and expertise to dominate NIST, and NIST has rarely played a significant independent role.”
- 2013 - Bernstein and Lange talk “The security dangers of the NIST curves”:
  “Jerry Solinas at the NSA used this [random method] to generate the NIST curves … or so he says…”

Dual_EC_DRBG

- Example of a weakened standard?
- Possibility of a back door seems to have been known by 2005.
- 2007 – Shumow and Ferguson: “We don’t know how $Q = [d]P$ was chosen, so we don’t know if the algorithm designer [NIST] knows [the backdoor] d.”
- Change to the standard in 2007, making the attack easier.

Snowden

- Confirmed some of the suspicions
- Cryptography standards may have been influenced by the NSA
- E.g. DUAL_EC_DRBG

The New York Times

"… the NSA had written the [crypto] standard and could break it."

"I no longer trust the constants. I believe the NSA has manipulated them through their relationships with industry."

Schneier '13 (post-Snowden)

What about some new curves?

Rigidity

- Give reasoning for all parameters and minimize “choices” that could allow room for manipulation
- Hash function needs a seed (digits of e, π, etc), but do choice of seed and choice of hash function themselves introduce more wiggle room?
- Goal: Justify all choices with (hopefully) undisputable efficiency arguments,
  e.g. choose fast prime field and take smallest curve constant that gives “optimal” group order [Bernstein‘06].

Rigid curve generation

Define a short Weierstrass curve

$$E_b/\mathbb{F}_p\colon y^2 = x^3 - 3x + b$$

as follows.

1. Pick a prime p according to well-defined efficiency/security criteria.
2. Find smallest $|b| > 0$, such that $\#E_b(\mathbb{F}_p) = r$ is prime.

What about these?

Replacement curve	Prime p	Constant b
(NEW) Curve P-256	$2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$	2627
(NEW) Curve P-384	$2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$	14060
(NEW) Curve P-521	$2^{521} - 1$	167884

- Same fields and equations ($E_b : y^2 = x^3 - 3x + b$) as NIST curves
- BUT smallest constant b such that $\#E_{\mathrm{b}}(\mathbb{F}_p)$ and $\#E'(\mathbb{F}_p)$ are prime
- So, simply change curve constants, and we're done, right???

Is that all? Motivations

- Curves that regain confidence:
  - rigid generation / nothing up my sleeves,
  - public approval and acceptance.
- 15 years on, we can do much better than the NIST curves (and this is true regardless of NIST-curve paranoia!):
  - faster finite fields and modular reduction,
  - side-channel resistance,
  - a whole new world of curve models.

Prime selection

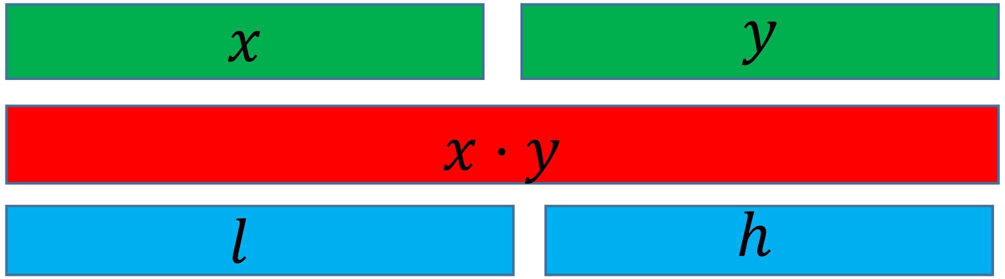
There are several alternatives for primes:

- pseudo-random primes,
- pseudo-Mersenne primes $p = 2^m - s$, $0 < |s| < 2^{\lfloor m/2 \rfloor}$,
- Solinas-primes $p = 2^a \pm 2^b \pm 1$, $0 < b < a$,
- etc.

Efficiency criterium: take prime with fastest modular reduction!

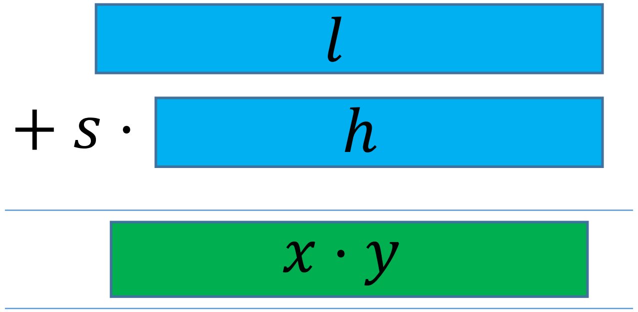
Arithmetic for pseudo-Mersenne primes

- Constant time modular multiplication



input: $0 \le x, y < 2^m - s$

$$
\begin{aligned}
& x \cdot y \in \mathbf{Z} \\
&= h \cdot 2^m + l \\
&\equiv h \cdot 2^m + l \; - h(2^m - s) \bmod (2^{\mathrm{m}} - s) \\
&= l + s \cdot h
\end{aligned}
$$



output: $x \cdot y \;\bmod\; (2^m - s)$

(after fixed, worst-case number of reduction rounds)

- Constant time modular inversion: $a^{-1} \equiv a^{p-2} \bmod p$
- Constant time modular square-root: $\sqrt{a} \equiv a^{(p+1)/4} \bmod p$

Favorite primes

- Bernstein and Lange: Curve25519, Curve41417, E-521

$$p = 2^{255} - 19, \qquad p = 2^{414} - 17, \qquad p = 2^{521} - 1$$

- Hamburg: Ed448-Goldilocks, Ed480-Ridinghood

$$p = 2^{448} - 2^{224} - 1, \qquad p = 2^{480} - 2^{240} - 1$$

- Brainpool: brainpoolP256t1, brainpoolP384t1, etc

$p = 76884956397045344220809746629001649093037950200943055203735601445031516197751$

- Bos, Costello, Longa, N.:

$$p = 2^{256} - 189, p = 2^{379} - 19, p = 2^{384} - 317, p = 2^{512} - 569$$

A world of curve models

$$y^2 = x^3 + ax + b$$
short Weierstrass curves

$$y^2 = x^4 + 2ax^2 + 1$$
Jacobi quartics

$$ax^3 + y^3 + 1 = dxy$$
(twisted) Hessian curves

$$By^2 = x^3 + Ax^2 + x$$
Montgomery curves

$$ax^2 + y^2 = 1 + dx^2y^2$$
(twisted) Edwards curves

$$y^2 = x^3 + ax^2 + 16ax$$
Doubling-oriented DIK curves

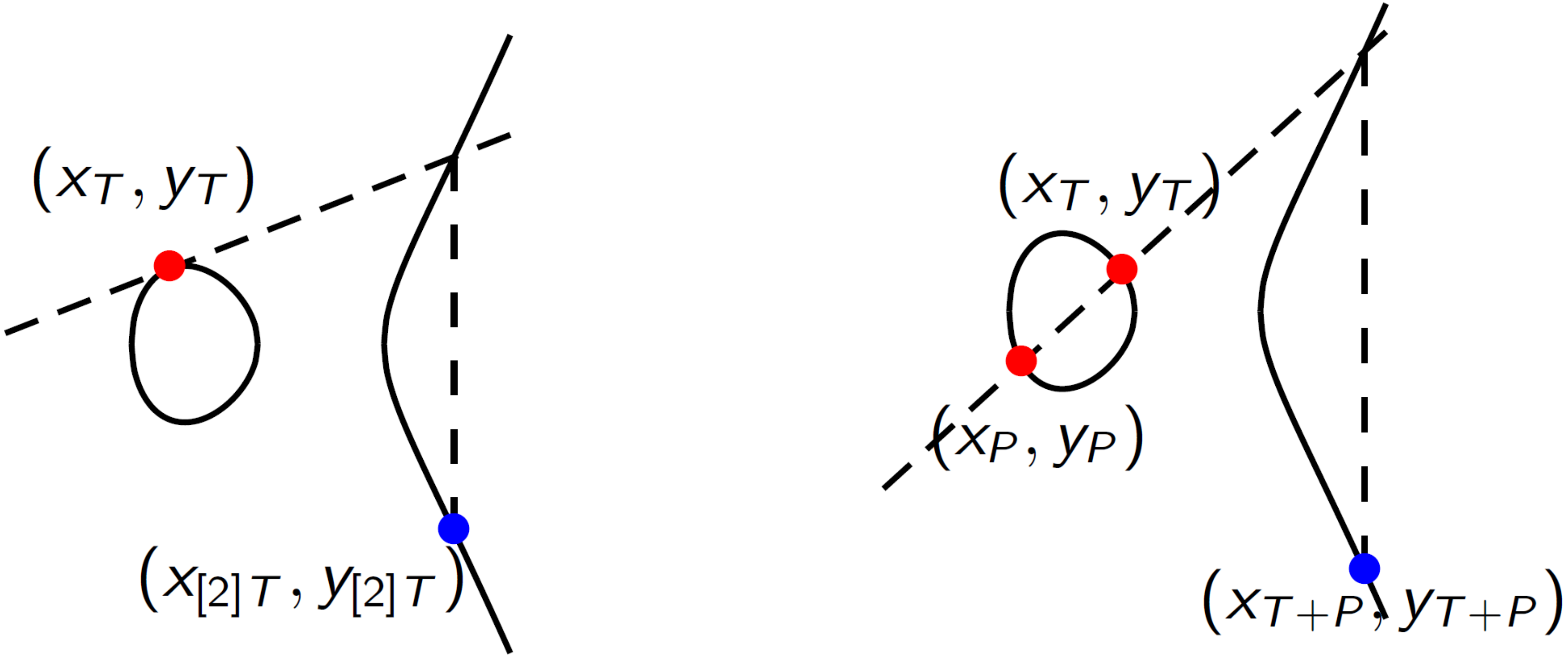
$$s^2 + c^2 = 1 \quad \cap \quad as^2 + d^2 = 1$$
Jacobi intersections

See Bernstein and Lange's Explicit-Formulas Database (EFD) and/or Hisil's PhD thesis

Curve models

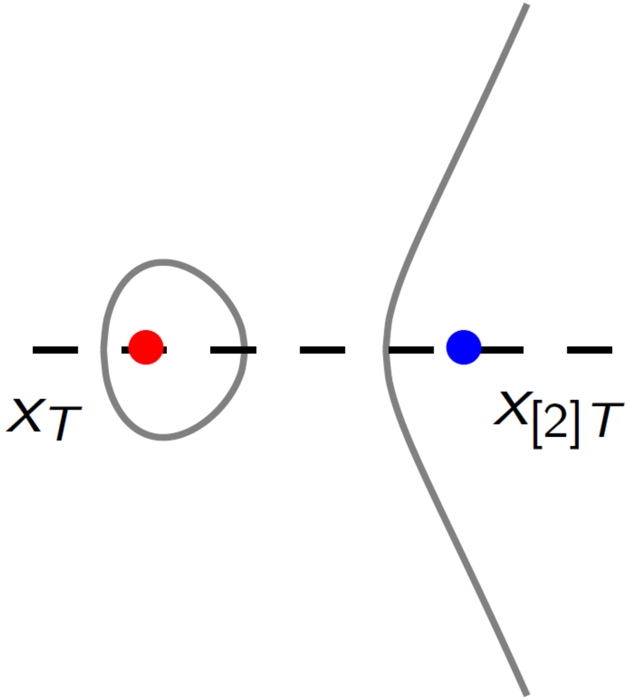
- Many different curve models and coordinate systems
- Many different formulas, ways to compute the group law
- Projective coordinates to avoid modular inversion
- Efficient formulas on Weierstrass model do not work for all points, they are actually sets of formulas

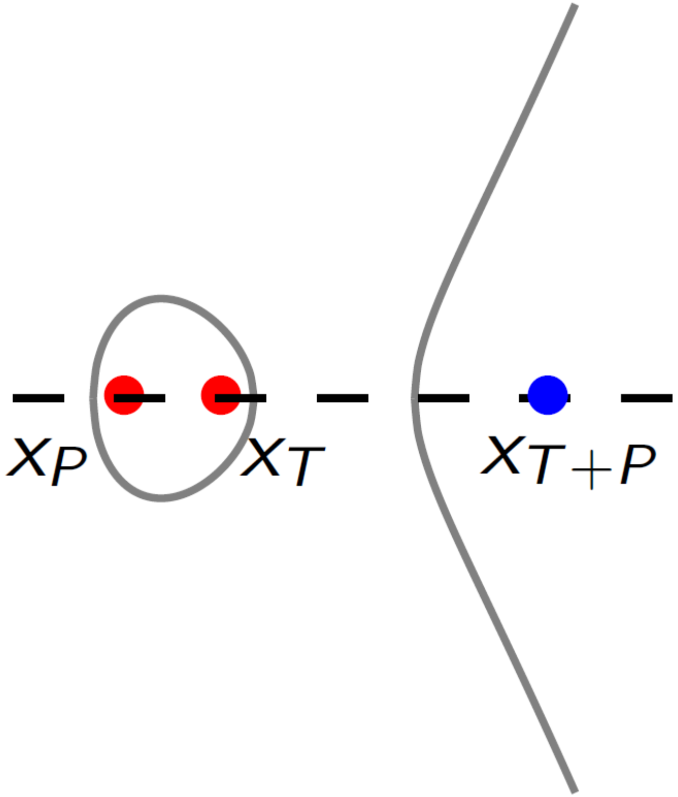
Text book arithmetic on $y^2 = x^3 + ax + b$



$(x_{[2]T}, y_{[2]T}) = DBL(x_T, y_T)$ $(x_{T+P}, y_{T+P}) = ADD(x_T, y_T, x_P, y_P)$

Montgomery's arithmetic on $By^2 = x^3 + Ax^2 + x$

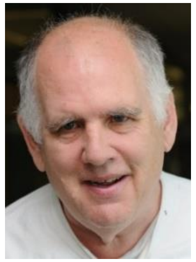




$$x_{[2]T} = DBL(x_T)$$

$$x_{T+P} = DIFFADD(x_T, x_P, x_{T-P})$$

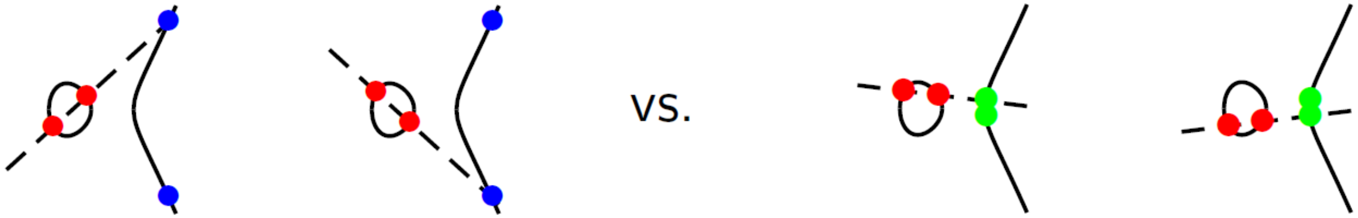
The Montgomery Ladder on $By^2 = x^3 + Ax^2 + x$



Rather than computing: $x_{Q+R} = f(x_Q, y_Q, x_R, y_R)$

$y_{Q+R} = g(x_Q, y_Q, x_R, y_R)$

It's much faster to compute: $x_{Q+R} = h(x_Q, x_R, x_{Q-R})$

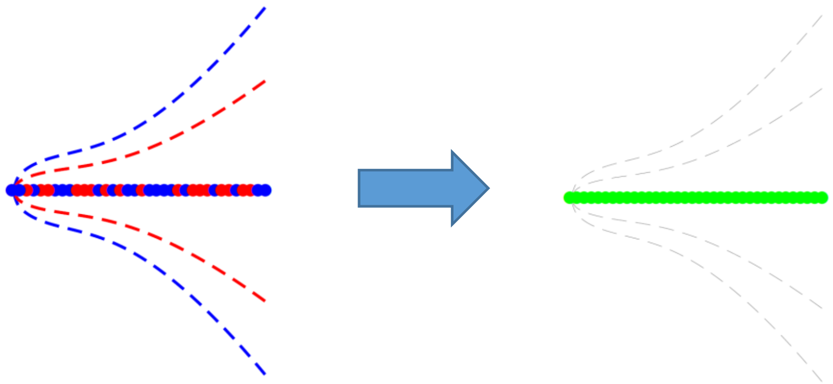


Key: so that we've always got x_{Q-R} , fix $Q - R = P$, the input point!

$[n+1]P$ $\xrightarrow{\text{One ``rung'' of the ladder}}$ $[2n+1]P$

$[n]P$ $\qquad$ $[2n]P$ or $[2n+2]P$

Twist-security



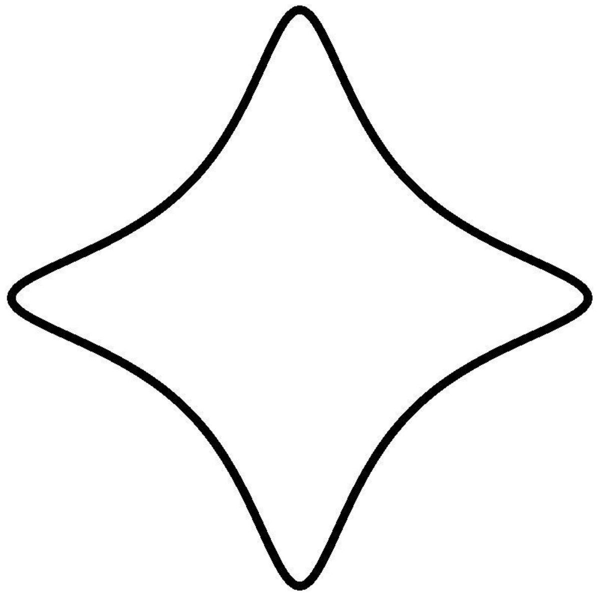
- Ladder gives scalar multiplications on E: $By^2 = x^3 + Ax^2 + x$ as
$$x([k]P) = LADDER(x(P), k, A)$$
- Independent of B, i.e. works on E': $B'y^2 = x^3 + Ax^2 + x$ for any B'
- Up to isomorphism, there are only two possibilities for fixed A:
$$E \text{ and its quadratic twist } E'$$
- If E and E' are both secure, no need to check $P \in E$ for any $x(P) \in K$, as $LADDER(x, k, A)$ gives result on E or E' for all $x \in K$
- Twist-security only really useful when doing x-only computations, but why not have it anyway?

Curve25519

$$E: y^2 = x^3 + Ax^2 + x,$$
$$p = 2^{255} - 19, A = 486662$$

- Dan Bernstein (2005)
- Diffie-Hellman key exchange using the Montgomery ladder
- Simple, constant-time x-only scalar multiplication
- Twist-secure, i.e. all x-coordinates work, avoids check of curve equation
- Montgomery coordinates not useful for signatures (ECDSA verification needs general point addition)
- $\#E(\mathbb{F}_p) = 8 \cdot r$, $\#E'(\mathbb{F}_p) = 4 \cdot r'$, r, r' are both prime.

Complete addition on Edwards curves



Let $d \neq \square$ in $\mathbb{F}_p$ and consider the Edwards curve

$$E/\mathbb{F}_p\colon x^2 + y^2 = 1 + dx^2y^2$$

For all (!!!) $\quad P_1 = (x_1, y_1), P_2 = (x_2, y_2) \in E(\mathbb{F}_p)$

$$P_1 + P_2 =: P_3 = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2}\right)$$

Denominators never zero, neutral element rational $= (0,1)$, etc..

(Bernstein-Lange, AsiaCrypt 2007)

Models considered for use in practice

Weierstrass curves

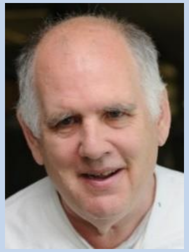
$$y^2 = x^3 + ax + b$$

- Most general form
- Prime order possible
- Exceptions in group law
- NIST and Brainpool curves

Montgomery curves

$$By^2 = x^3 + Ax^2 + x$$

- Subset of curves
- Not prime order
- Fast Montgomery ladder
- ≈ Exception free



(twisted) Edwards curves

$$ax^2 + y^2 = 1 + dx^2y^2$$

- Subset of curves
- Not prime order
- Fastest addition law
- Some have complete group law

The NUMS curves

Security $s =$	Prime $p =$	Weierstrass $b =$	Twisted Edwards $d =$	Montgomery $A =$
128	$2^{256} - 189$	152961	15342	-61370
192	$2^{384} - 317$	-34568	333194	-1332778
256	$2^{512} - 569$	121243	637608	-2550434

- Primes: Largest $p = 2^{2s} - \gamma \equiv 3 \bmod 4$ (here: largest primes, full stop)
- Weierstrass: Smallest $|b|$ such that $\#E$ and $\#E'$ both prime
- Twisted Edwards: Smallest $d > 0$ such that $\#E$ and $\#E'$ both 4 times a prime, and $d > 0$ corresponds to $t > 0$.

Small constants for $p \equiv 3 \bmod 4$

$$M_A : y^2 = x^3 + Ax^2 + x \qquad E_{a,d} : ax^2 + y^2 = 1 + dx^2y^2$$

$$\begin{tikzcd}
M_A \arrow[r, leftrightarrow, "\cong"] \arrow[d, leftrightarrow, "\text{twist}"'] & E_{-1,d_1} \arrow[r, leftrightarrow, "\text{isogeny}"] & E_{-1,d_0} \arrow[dr, leftrightarrow, "\text{twist}"'] & E_{1,\, d_0+1} \arrow[dl, leftrightarrow] \\
M_{-A} \arrow[r, leftrightarrow, "\cong"] & E_{-1,1/d_1} \arrow[r, leftrightarrow, "\text{isogeny}"] & E_{-1,-(d_0+1)} & E_{1,-d_0}
\end{tikzcd}$$

$d_1 = -\frac{A-2}{A+2}$ (big)

Both non-squares

$d_0 = -\frac{A+2}{4}$ (small)

Search that minimizes Montgomery constant size also minimizes size of both twisted Edwards and Edwards constants.

Real world discussions

- TLS WG requested recommendations for new elliptic curves from the CFRG
  See mailing list on https://irtf.org/cfrg.

  TLS 1.3 will have new cipher suites with Curve25519 and a curve using $p = 2^{448} - 2^{244} - 1$.

- NIST is holding a workshop on the standardization of new elliptic curves in June, see http://www.nist.gov/itl/csd/ct/ecc-workshop.cfm.

Some References

- Bos, Costello, Longa, N.:
  Selecting elliptic curves for cryptography – an efficiency and security analysis
  http://eprint.iacr.org/2014/130

- Longa: MSR ECCLib
  http://research.microsoft.com/en-us/projects/nums/default.aspx

- Bernstein, Lange; Safecurves web site: http://safecurves.cr.yp.to/

- Bernstein, Lange: Explicit Formulas Database (EFD)
  http://www.hyperelliptic.org/EFD/
  Formulas and operation counts for elliptic curve operations on many different curve models

- Bernstein, Curve25519: http://cr.yp.to/ecdh/curve25519-20060209.pdf

- Hisil, PhD thesis: http://eprints.qut.edu.au/33233/