Faster Compact Diffie-Hellman: Endomorphisms on the x-line

Craig Costello

Joint work with Huseyin Hisil and Benjamin Smith

4th Annual MSR Privacy Workshop

Microsoft® Research

October 25, 2013

Elliptic curves

Q. Why do cryptographers fancy elliptic curves
A. They are as resilient as a "generic group"

- fastest attacks are "generic"
- other primitives (RSA, finite fields, etc) incomparable
- NSA: *". . . unlike the RSA and Diffie-Hellman cryptosystems that slowly succumbed to increasingly strong attack algorithms, elliptic curve cryptography has remained at its full strength since it was first presented in 1985"*.
- Nowadays: 256-bit ECDLP compared to 3072-bit DLP or RSA
- NSA: *"factor 10 speedup over others at* 128*-bit level"* . . .

Q. Why do number theorists fancy elliptic curves

A. They are beautiful, rich and deep objects

- Endless uses, from Gauss to Wiles
- Fermat's Last Theorem, BSD conjecture, etc etc
- Barry Mazur: "*These elliptic curves amply repay the obsessive interest that mathematicians have for them . . . elliptic curves seem to be designed to teach us things*"

Elliptic curves

Why do number-theoretic cryptographers fancy elliptic curves

A. The best attacks are generic, but elliptic curves couldn't be further from generic groups

- Ben Smith: *"they have a rich and concrete geometric structure, which should be exploited for fun and profit"*
- Can use all of the generic improvements for group exponentiation, but have access to several curve-specific optimisations:
  - endomorphisms, alternative models, coordinate systems, ...

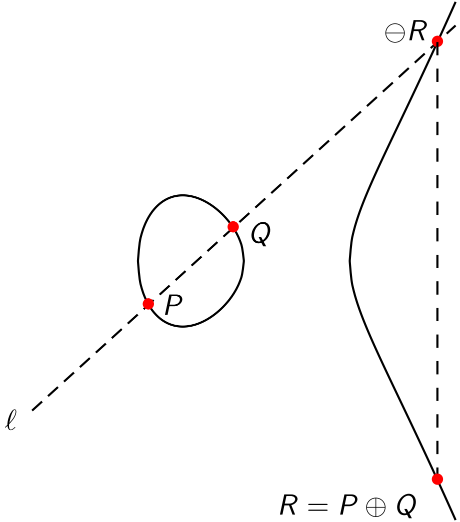
This work: turbocharged scalar multiplications

Combines two of the most powerful optimisations

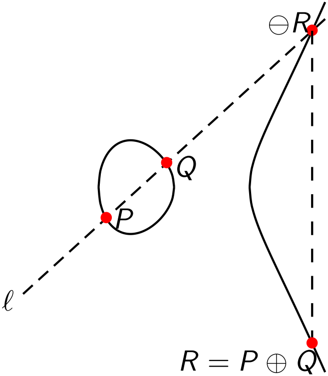
$\rightarrow$ the Montgomery model/ladder **and** endomorphisms

Elliptic curve group addition ...

Elliptic curve: $y^2 = x^3 + ax + b$



Peter: "why the y's?- we can do (scalar mults) without them"



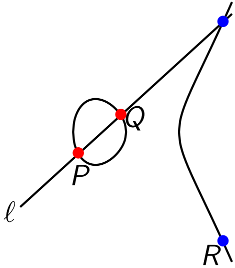
Montgomery's idea ...

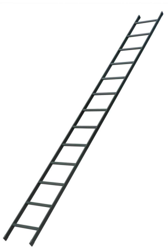
Peter: "why the y's?- we can do scalar mult. without them"

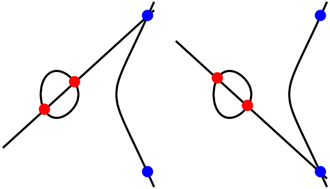
- x-line is a *pseudo-group*, allows only *pseudo-group* operations
- No longer technically a group, but enough to do scalar multiplications (e.g. Diffie-Hellman)

Montgomery ladder for elliptic curves . . .

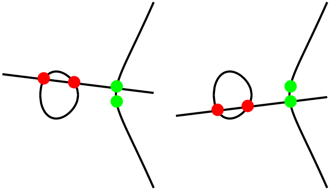
- **Key:** Can compute $P + Q$ from $\{P, Q, {\color{red}P - Q}\}$ without y-coords







vs.



same difference $\longrightarrow$ same result

different difference $\longrightarrow$ different result

An elliptic curve and its quadratic twist

Suppose $\mathbb{F}_p = \mathbb{F}_{19}$ (-1 is non square)

$E\colon y^2 = x^3 + 11x + 4$ $\qquad$ $E'\colon\ -y^2 = x^3 + 11x + 4$

An elliptic curve and its quadratic twist

Suppose $\mathbb{F}_p = \mathbb{F}_{19}$ (-1 is non square)

$E: y^2 = x^3 + 11x + 4$ $\qquad$ $E': -y^2 = x^3 + 11x + 4$

E		E'
	$x = 0$?	
$(0, 2),\ (0, 17)$	$x^3 + 11x + 4 = 4$ ✓	
	$x = 1$?	
$(1, 4),\ (1, 15)$	$x^3 + 11x + 4 = 16$ ✓	
	$x = 2$?	
	$x^3 + 11x + 4 = 15$ ✗	$(2, 2),\ (2, 17)$
	$x = 3$?	
$(3, 8),\ (3, 11)$	$x^3 + 11x + 4 = 7$ ✓	
	$x = 4$?	
	$x^3 + 11x + 4 = 17$ ✗	$(4, 6),\ (4, 13)$
$\vdots$	$\vdots$	$\vdots$
	$x = 18$?	
$(18, 7),\ (18, 12)$	$x^3 + 11x + 4 = 11$ ✓	

An elliptic curve and its quadratic twist

Suppose $\mathbb{F}_p = \mathbb{F}_{19}$ (-1 is non square)

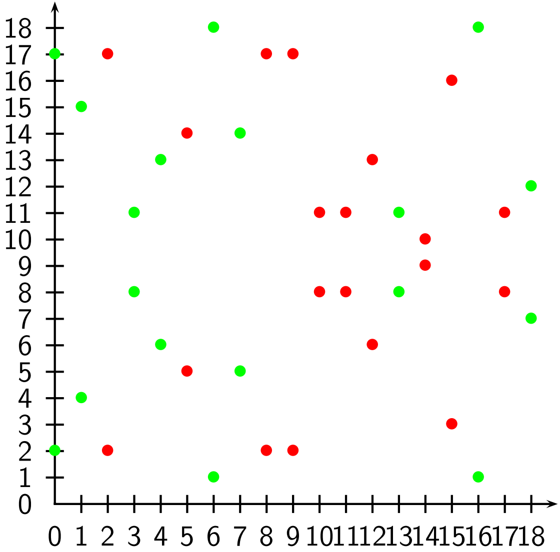
$E: y^2 = x^3 + 11x + 4$ $\quad$ $E': \; -y^2 = x^3 + 11x + 4$

E		E'
$(0,2),\ (0,17)$	$x = 0?$ $x^3 + 11x + 4 = 4$ ✓	
$(1,4),\ (1,15)$	$x = 1?$ $x^3 + 11x + 4 = 16$ ✓	
	$x = 2?$ $x^3 + 11x + 4 = 15$ ✗	$(2,2),\ (2,17)$
$(3,8),\ (3,11)$	$x = 3?$ $x^3 + 11x + 4 = 7$ ✓	
	$x = 4?$ $x^3 + 11x + 4 = 17$ ✗	$(4,6),\ (4,13)$
$\vdots$	$\vdots$	$\vdots$
$(18,7),\ (18,12)$	$x = 18?$ $x^3 + 11x + 4 = 11$ ✓	

$$\#E = 19 = \textit{prime} \longrightarrow ☺$$

$$\#E' = 21 = 3 \cdot 7 \longrightarrow ☹$$

The points on E and E'



Dropping the y-coordinate

- Neither red or green sets are a group in their own right
- Montgomery's formulas don't differentiate between the two sets (they work identically on both)
- So let's (ignore many practical caveats for now and) not differentiate either, and work on the x-line

- Our x-coordinates will come from $\mathbb{F}_{p^2}$ where $p = 2^{127} - 1$.
- Think two 127-bit strings, or (more ignorance) a 254-bit string
- Use BHKL'13 "Elligator": - keys and transmissions all just random 254-bit strings

x-only needs twist-security . . .

- Consider NISTp224: $p = 2^{224} - 2^{96} + 1$, specific $b \in \mathbb{F}_p$

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

- $\#E = 2695994666715063 \ldots 21682722368061$ (224-bit prime)
- What about the order of the quadratic twist of NISTp224?
- $\#E' = 3^2 \cdot 11 \cdot 47 \cdot 3015283 \cdot 40375823 \cdot 267983539294927 \cdot$ $177594041488131583478651368420021457$ (118-bit prime)
- Not a problem if using both coordinates, just check $(x, y) \in E$
- If only dealing with x's, honest parties all work on E ☺. . .
  . . . but attackers could send x's on E' and solve DLP there ☹
- Or inject faults (FRLV'08) to convert x on E to x on E'
- **Solution: Use** *twist-secure* **curves: both** E **and** E' **strong**

Endomorphisms

- **Endomorphisms:** a powerful (non-generic) optimisation in curve-based cryptography
- Map P to "big multiple" $[\lambda]P$ somewhat immediately, on certain curves
- Simple example: on $E/\mathbb{F}_p : y^2 = x^3 + b$ for $p \equiv 1 \bmod 3$,

$$\psi\colon P \mapsto [\lambda]P, \quad (x, y) \mapsto (\xi x, y),$$

  where $\xi^3 = 1 \in \mathbb{F}_p$, but $\xi \neq 1$. Then scalar λ is big.

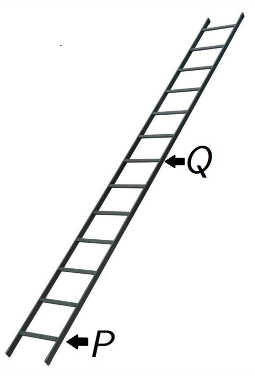
- Then what ...

Twist-security with endomorphisms

- Using Montgomery's fast/compact x-only arithmetic with endomorphisms has not been done
- **Why?** Two previous methods of endomorphism construction don't allow twist-security
- **GLV curves** are special - no hope of twist-secure GLV curves over best primes
  - e.g. $y^2 = x^3 + b$ - at most 6 isomorphism classes / group orders over any prime
- **GLS curves** remedy the sparseness, BUT still necessarily twist-insecure, e.g. $E/\mathbb{F}_{p^2}$ implies E' defined over $\mathbb{F}_p$
- **BUT:** Smith'13 gives a new endomorphism construction using $\mathbb{Q}$-curves: **can now achieve twist-secure curves with endomorphisms, over say, $\mathbb{F}_{p^2}$ with $p = 2^{127} - 1$**

Using endomorphisms in general (sketch)

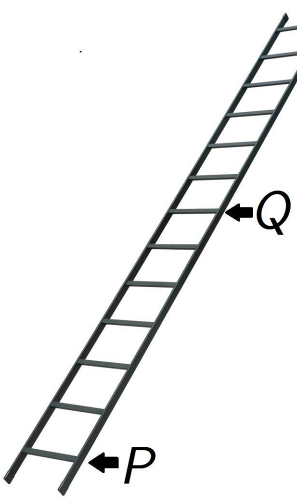
- Let $Q = \psi(P) = [\lambda]P$, perform multiscalar to get to $[k]P$ (very roughly) around twice as fast



- e.g. can start with $P + Q$, or $[2]P + Q$ or $[2]Q + P$, and crawl up in sync (Straus-Shamir)

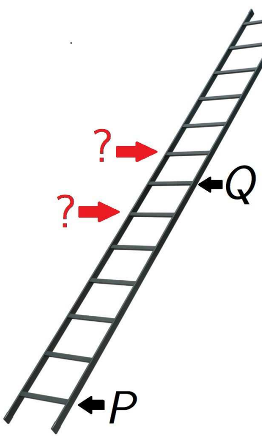
Using endomorphisms with x-only

- BUT: In our case, can't add P and Q to kickstart
- Can't move anywhere with just P and Q...



Using endomorphisms with x-only

- Need $Q \pm P$ or $(\psi \pm 1)(P)$ to move quickly to $[k]P$



- Other people have run into this problem and halted

Computing $(\psi \pm 1)(P)$: a fortunate exponent

- Smith'13: Let $P = (x_P, y_P)$ be a point on Montgomery form $By^2 = x^3 + Ax^2 + x$ of special Hasegawa $\mathbb{Q}$-curve of degree two over $\mathbb{F}_{p^2}$. Then $\psi(P) = (x_Q, y_Q) = Q$, where

$$x_Q = c_1 \left(\frac{x_P^2 + Ax_P + 1}{x_P} \right)^p, \quad y_Q = c_2 \left(\frac{y_P(x_P^2 - 1)}{x_P^2} \right)^p \tag{1}$$

  for constants c_1 and c_2

- On the general Montgomery curve $By^2 = x^3 + Ax^2 + x$

$$x_{Q \pm P} = \frac{B\left(x_P y_Q \mp x_Q y_P\right)^2}{x_P x_Q \left(x_P - x_Q\right)^2}. \tag{2}$$

- **Sub (2) into (1): everything simplifies to be relatively efficient and all y_P's trivially vanish (using curve equation), except for one term: y_P^{p+1}**

- Looks very unwieldy, but ...

Computing $(\psi \pm 1)(P)$: a fortunate exponent

$$y^{p+1} \quad = \quad (y^2)^{(p+1)/2} \quad = \quad \left(\frac{x^3 + Ax^2 + x}{B}\right)^{(p+1)/2}$$

- BUT: in our case $p = 2^{127} - 1$, so exponent is 2^{126}
- Exponentiation is 126 squarings in $\mathbb{F}_{p^2}$
- In total, computing the values

  $$x_Q = \psi(x_P), \quad x_{Q+P} = (\psi + 1)(x_P), \quad x_{Q-P} = (\psi - 1)(x_P)$$

  costs 129 squarings and 15 multiplications
- Not as cheap as traditional endomorphisms, or standalone group operations, but could still be worth it . . .

Two dimensional differential addition chains. . .

- Two dimensional **differential** addition chains are already in the literature (for other purposes)
- Equipped with ψ, we implemented 3 of them

chain	dim.	endomorphisms ψ_x, $(\psi \pm 1)_x$	#DBL's	#ADD's
LADDER	1	—	254	253
DJB	2	affine	128	255
AK	2	affine	≈ 181	≈ 181
PRAC	2	projective	≈ 74	≈ 187

- DBL's take roughly 4 multiplications, ADD's take roughly 6.
- So endomorphisms ψ_x, $(\psi \pm 1)_x$ cost around 25 ADD's
- (modulo many caveats) Clearly some speedups on the cards from using ψ . . .

How fast are we talking?

- **Disclaimer**: There are several others (Bos *et al.*, Longa *et al.*, Oliveira *et al.*) who are faster
- But we are simply talking x-only...

Table: Intel i7-3520M (Ivy-Bridge) cycles per scalar multiplication at 128-bit security level for x-coordinate only implementations

addition chain	dimension	uniform?	constant time?	cycles
Bernstein (`curve25519`)	1	✓	✓	182,000
LADDER	1	✓	✓	152,000
DJB	2	✓	✓	145,000
AK	2	✓	✗	130,000
PRAC	2	✗	✗	110,000