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

Craig Costello, Huseyin Hisil and Benjamin Smith

EUROCRYPT 2014
Copenhagen, Denmark
May 12, 2014

At a high level...

A software implementation of Diffie-Hellman key-exchange targeting 128-bit security:

- **Fast:** 148,000 cycles (Intel Core i7-3520M – Ivy Bridge) for

  `key_gen` and `shared_secret`

- **Compact:** 256-bit keys (*purely* x-coordinates only)
- **Constant-time:** execution independent of input – side-channel resistant

Software (in eBACS format) available at:

`http://hhisil.yasar.edu.tr/files/hisil20140318compact.tar.gz`

Outline

1. **Endomorphisms**

   replace single scalar with half-sized double-scalars

2. **The x-line**

   use x coordinates throughout, instead of (x, y) coordinates
   (and work on curve and twist simultaneously)

3. **Endomorphisms on the x-line**

   do both ...

Endomorphisms

A generic scalar multiplication

The fundamental ECC operation: scalar multiplication

given:	a scalar $[m]$ and an elliptic curve point P
compute:	$[m]P$

- Write the scalar in binary

$$m = (\ 1\ ,\ 0\ ,\ 1\ , \ldots ,\ 0\ ,\ 0\ ,\ 1\ )_2$$

and double-and-add

- Or use another addition chain . . .

Endomorphisms

What's an endomorphism?

In this talk, an endomorphism (on an elliptic curve $\mathcal{E}$) is a map

$$\psi \colon \mathcal{E} \to \mathcal{E}$$

- **some (trivial) examples:** multiplication-by-m map, $m \in \mathbb{Z}$

$$[-1], [2], [3], \ldots, [m]$$

- **real-world example:** curve used in bitcoin

$\mathcal{E}/\mathbb{F}_p \colon y^2 = x^3 + b$ with $p \equiv 1 \bmod 3$. Let $\zeta^3 = 1$ for $\zeta \neq 1$.

If $P = (x, y)$ on $\mathcal{E}$, so is $\psi(P) = (\zeta x, y)$

- Fact: $$\psi(P) = [\lambda]P$$

i.e. ψ's a shortcut to $[\lambda]$

What's a useful endomorphism?

ψ should be efficiently computable, and $[\lambda]$ should be large

i.e. ψ should be much faster than $[\lambda]$ (e.g. 1 mul vs. 3000+ muls)

How to use an endomorphism, part I

Scalar multiplication (in the presence of an endomorphism):

given: a scalar $[m]$ and two points P, $\psi(P)$ (order N)
compute: $[m]P = [a]P + [b]\psi(P)$

- many possible (a, b) pairs – find "short" one
- Use "zero decomposition lattice" $\mathcal{L}$: all pairs (c, d) such that
$$c + d\lambda \equiv 0 \bmod N$$
- Find $(v_1, v_2) \in \mathcal{L}$ close to $(m, 0)$: $(a, b) = (m, 0) - (v_1, v_2)$
- Short basis for $\mathcal{L} = \langle (N, 0), (-\lambda, 1) \rangle$ computed in advance, so
$$m \xrightarrow{\mathcal{L}} (a, b)$$
very cheap: i.e. less than 10 integer muls to compute

How to use an endomorphism, part II

- e.g.: 256-bit m's decompose into $\approx$ 128-bit a's and b's
- $m = 100162175736570768564527594834550209124031802653885759009988599962436827164086$

$$\downarrow \mathcal{L}$$

$$\begin{aligned} a &= 99172541169956320218199372915391025671 \\ b &= 224127230907715819133022922601979555751 \end{aligned}$$

- Multiexponentation to compute $[a]P + [b]\psi(P)$

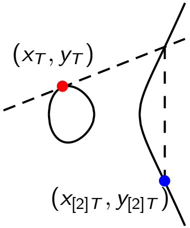
$$\begin{array}{ccccccccccccc} a = [& 1 & | & 0 & | & 1 & | & 0 & | & 1 & | & \dots \\ b = [& 0 & | & 1 & | & 0 & | & 0 & | & 1 & | & \dots \\ & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \\ [& P & | & \psi(P) & | & P & | & 0 & | & P + \psi(P) & | & \dots \end{array}$$

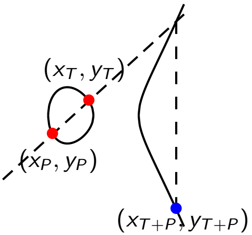
Summary: (at least in this case...)

half the doublings... and fewer additions too!

The x-line

x–coordinate only arithmetic





Classical formulas: $y^2 = x^3 + ax + b$

$$x_{[2]T}, y_{[2]T} = \mathtt{DBL}(x_T, y_T, a)$$

$$x_{T+P}, y_{T+P} = \mathtt{ADD}(x_T, y_T, x_P, y_P)$$

Montgomery's formulas: $By^2 = x^3 + Ax^2 + x$

$$x_{[2]T} = \mathtt{DBL}(x_T, A)$$

$$x_{T+P} = \mathtt{PSEUDOADD}(x_T, x_P, x_{T-P})$$

x–coordinate only arithmetic

Classical formulas: $y^2 = x^3 + ax + b$

$$x_{[2]T}, y_{[2]T} = \mathtt{DBL}(x_T, y_T, a)$$

$$x_{T+P}, y_{T+P} = \mathtt{ADD}(x_T, y_T, x_P, y_P)$$

Montgomery's formulas: $By^2 = x^3 + Ax^2 + x$

$$x_{[2]T} = \mathtt{DBL}(x_T, A)$$

$$x_{T+P} = \mathtt{PSEUDOADD}(x_T, x_P, x_{T-P})$$

- opposite y's give different x-coordinate than same-sign y's
- decide between them with difference x_{T-P}
- **Differential additions:** $x_{T+P} = \mathtt{PSEUDOADD}(x_T, x_P, x_{T-P})$

The importance of twist-security

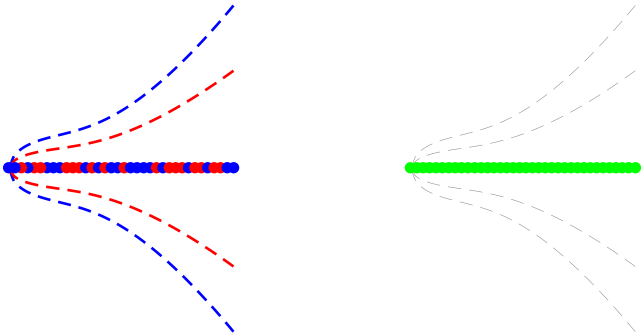
Compact scalar multiplications on:

$$\mathcal{E}/\mathbb{F}_q: \ By^2 = x^3 + Ax^2 + x$$

$$x([m]P) \ = \ \texttt{LADDER}(m, x(P), A)$$

- Now just $\mathbb{F}_q$ values (hard ECDLP underneath)
- BUT only $\approx$ half of $x \in \mathbb{F}_q$ give point on $By^2 = x^3 + Ax^2 + x$
- Other $\approx$ half give point on twist $\mathcal{E}': \ B'y^2 = x^3 + Ax^2 + x$
- Bernstein '01: $\texttt{LADDER}(m, x, A)$ will give hard ECDLP for all $x \in \mathbb{F}_q$ if $\mathcal{E}$ and $\mathcal{E}'$ are both secure (i.e. same A for $\mathcal{E}$, $\mathcal{E}'$)

The picture



- All possible $x \in \mathbb{F}_q$ “partitioned” to $\mathcal{E}$ or $\mathcal{E}'$
- But $\texttt{LADDER}(m, x, A)$ doesn't distinguish: so users needn't
- Bernstein'06: `curve25519` built on this notion

Endomorphisms on the x-line

Curve constructions

We need a curve that:

i. is defined over fast field
ii. has a useful endomorphism
iii. is twist-secure

- (ii) and (iii): Gallant-Lambert-Vanstone (GLV) – CRYPTO'01
- (i) and (ii): Galbraith-Lin-Scott - (GLS) – EUROCRYPT'09

(i), (ii) and (iii): Benjamin Smith - ASIACRYPT'13

Fast families of elliptic curves from $\mathbb{Q}$-curves

The curve: targeting 128-bit security level

- **the field:**

$$\mathbb{F}_{p^2} = \mathbb{F}_p(i),\ i^2 + 1 = 0 \text{ and } p = 2^{127} - 1$$

- **the curve (and twist):** defined by $A \in \mathbb{F}_{p^2}$

$$\mathcal{E}\colon y^2 = x^3 + Ax^2 + x, \qquad \mathcal{E}'\colon (\tfrac{12}{A})y^2 = x^3 + Ax^2 + x$$

- **the group orders:**

$$\#\mathcal{E} = 4N, \qquad\qquad \#\mathcal{E}' = 8N',$$

252-bit prime N and 251-bit prime N'

- **security properties:**

MOV deg, $\mathrm{disc}(\mathrm{End}(\mathcal{E}))$, $h(\mathrm{End}(\mathcal{E}))$ – all huge ...

The (x-only) endomorphism ψ_x

$$\psi_x(x) = \frac{A^p\big((x-1)^2+(A+2)x\big)^p}{-2Ax^p}$$

2-dimensional differential addition chains

- Requirement: difference $U - V$ must be in chain before computing $U + V$
- One dimensional ladder: $m, x(P) \;\mapsto\; x([m]P)$
- We need two-dimensional version:

$$a, b, x(P), x(\psi(P)) \;\mapsto\; x([a]P + [b]\psi(P))$$

- Three variants chosen from the literature ...

chain	by	# steps	ops per step
`PRAC`	Montgomery	$\approx 0.9\ell$	≈ 1.6 `ADD` $+ 0.6$ `DBL`
`AK`	Azarderakhsh & Karabina	$\approx 1.4\ell$	1 `ADD` $+ 1$ `DBL`
`DJB`	Bernstein	ℓ	2 `ADD` $+ 1$ `DBL`

- Note: easy to force $\ell = \max\{\lceil \log_2 a \rceil, \lceil \log_2 b \rceil\}$ to be of constant length for constant-time chains

Kickstarting addition chains . . .

- All three chains require inputs $x(P)$, $x(\psi(P))$, and one of

$$x\left((\psi \pm 1)(P)\right)$$

i.e. can't add two points without their difference

Computing the initial difference:

$$(\psi \pm 1)_x(x) = f(x) + g(x) \cdot x^{(p+1)/2},$$

where f and g have low degree.

- Exponentiation to $(p+1)/2 = 2^{126} \longrightarrow 126$ squarings
- $(\psi \pm 1)_x$ not as fast as ψ_x, or other endomorphisms around, but it could be worse . . .

Performance results (Ivy Bridge)

The routine

Input: scalar $m \in \mathbb{Z}$ and $x(P) \in \mathbb{F}_{p^2}$

1. $a, b \leftarrow \texttt{DECOMPOSE}(m)$
2. $x(\psi(P)), x((\psi - 1)(P)) \leftarrow \texttt{ENDO}(x(P))$
3. $x([m]P) \leftarrow \texttt{CHAIN}\,(x(P), x(\psi(P)), x((\psi - 1)(P))$

Output: $x([m]P)$

`CHAIN`	dimension	uniform?	constant time?	cycles
`LADDER`	1	✓	✓	159,000
`DJB`	2	✓	✓	148,000
`AK`	2	✓	✗	133,000
`PRAC`	2	✗	✗	109,000

Compare to `curve25519` *(✓ & ✓): 182,000 cycles*

Variants / alternatives / spin-offs . . .

- Slightly faster/simpler if choosing (a, b) at random (see paper)

- Faster `key_gen` in ephemeral Diffie-Hellman: Alice may want to exploit pre-computations on the public generator $x(P)$:
    - precompute $x(\psi(P))$ and $x((\psi+1)P)$, or
    - Alice works on twisted Edwards form of $\mathcal{E}$ before pushing to x-line for Bob

- Genus 2 analogue still open: even more attractive on the Kummer surface

Full version

`http://eprint.iacr.org/2013/692`

C-and-assembly software implementation

`http://hhisil.yasar.edu.tr/files/hisil20140318compact.tar.gz`

Magma scripts

`http://research.microsoft.com/en-us/downloads/ef32422a-af38-4c83-a033-a7aafbc1db55/`