Faster Compact DiffieHellman: Endomorphisms on the x-line

Craig Costello

craigco@microsoft.com

Microsoft Resesarch
Redmond
Seattle, USA

Hüseyin Hışıl

huseyin.hisil@yasar.edu.tr

Computer Eng. Department
Yaşar University
İzmir, Turkey

Benjamin Smith

smith@lix.polytechnique.fr

INRIA, France
LIX, Ecole polytechnique,
France

ECC 2014, Chennai

At a high level...

A software implementation of Diffie-Hellman key-exchange targeting 128-bit security (EUROCRYPT 2013):

- **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 SUPERCOP format) available at:

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

Outline

1. **Endomorphisms**

   replace single scalar with half-sized double-scalars

2. **Selecting the curve**

   parameter fine tuning, twist security, large discriminant, . . .

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

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

4. **Fast finite field arithmetic**

   non-unique representation, assembly tricks, btrq, . . .

Standard definitions I [Silverman]

Let E_1 and E_2 be elliptic curves.

- An isogeny is a homomorphism

  $$\phi\colon E_1 \to E_2 \ \text{ with finite kernel satisfying } \ \phi(O) = O,\ \phi(E_1) \neq \{O\}.$$

- Let $P \in E_1$. Observe that the set

  $$\mathrm{Hom}(E_1, E_2) := \Big\{\text{isogenies } \phi\colon E_1 \to E_2\Big\}.$$

  becomes a group under the addition law

  $$(\phi + \psi)(P) = \phi(P) + \psi(P).$$

Standard definitions II [Silverman]

- Now let $E := E_1 = E_2$. An endomorphism is an element of

$$\mathrm{End}(E) := \mathrm{Hom}(E, E).$$

- $\mathrm{End}(E)$ is called the endomorphism ring of E since we have for all points on E;
  - the addition –homomorphism property–

$$(\phi + \psi)(P) = \phi(P) + \psi(P),$$

  - the multiplication –composition–

$$(\phi\psi)(P) = \phi(\psi(P)).$$

Classic examples for endomorphisms

- Multiplication-by-m map for $m \in \mathbb{Z}$.

$$[m]: \quad P \mapsto \underbrace{P + P + \ldots + P}_{m \text{ times}}.$$

Computing $[m](P)$ is the bottleneck for many curve based protocols.

Therefore, we want to speed up $[m](P)$.

Classic examples for endomorphisms

- Let $p \equiv 1 \pmod 4$ be a prime. Define

$$E\colon y^2 = x^3 + ax$$

over $\mathbb{F}_p$. Let $\kappa \in \mathbb{F}_p$ suct that $\kappa^2 = -1$. Then the map

$$\mu : \quad (x, y) \longmapsto (-x, \kappa y)$$

is an endomorphism with characteristic polynomial

$$\mathcal{P}(X) = X^2 + 1.$$

Suppose $N \mid \#E(\mathbb{F}_q)$ but $N^2 \nmid \#E(\mathbb{F}_q)$.

Now, $E(\mathbb{F}_q)$ contains exactly one subgroup of order N.

Assume $P \in E(\mathbb{F}_q)[N]$. Then $\mu(P) \in E(\mathbb{F}_q)[N]$.

Therefore, $\mu(P) = [\lambda]P$ for some $\lambda \in [1, N-1]$ when $P \neq \mathcal{O}$.

Furthermore, λ is a root modulo N of $\mathcal{P}(X)$.

Gallant/Lambert/Vanstone technique CRYPTO'01

Speeding up scalar multiplication with GLV:

Replace

$$(m, P) \mapsto [m](P)$$

with

$$\begin{aligned} ((a,b),P) \quad \longmapsto \quad & [a]P + [b]\mu(P) = \\ & [a]P + [b\lambda](P) = \\ & [m](P) \end{aligned}$$

where (a, b) is a short multiscalar decomposition of a random full-length scalar m.

Endomorphism examples by Gallant/Lambert/Vanstone'01 are only applicaple to a very limited set of elliptic curves.

- The q-power Frobenius endomorphism π_q (if E is defined over $\mathbb{F}_q$).

$$\pi_q : \quad (x, y) \mapsto (x^q, y^q)$$

where π_q satisfies the characteristic polynomial

$$\mathcal{P}(X) = X^2 - tX + q$$

where $t = q + 1 - \#E(\mathbb{F}_q)$.

We have $\pi_q(P) = P$ for all $P \in E(\mathbb{F}_q)$, i.e. the set of points fixed by π_q is exactly $E(\mathbb{F}_q)$.

Observe that $(X^2 - tX + q)$ mod $\#E$ factors as $(x - 1)(x - q)$.

Galbraith/Lin/Scott endomorphism EUROCRYPT'09

Ingredients for GLS construction **(just an overview)**:

1. E: an elliptic curve defined over $\mathbb{F}_p$ where $p > 3$
2. E': the quadratic twist of $E/\mathbb{F}_{p^2}$
3. ϕ: $E \to E'$: twisting $\mathbb{F}_{p^4}$-isomorphism
4. π_q: $E \to {}^{(q)}E$: q-power Frobenius isogeny; ${}^{(p)}E = E$, so $\pi_p \in End(E)$

Now define $$\psi := \phi \circ \pi_p \circ \phi^{-1}$$

- ψ is a (degree 2) $\mathbb{F}_{p^2}$-endomorphism of E' satisfying $\psi^2 = [-1]$
- If N is a prime such that $N \mid \#E(\mathbb{F}_{p^2})$ and $N > 2p$ then

$$\psi^2(P) + P = \mathcal{O} \quad \text{for} \quad P \in E'(\mathbb{F}_{p^2})[N]$$

- $\psi(P) = [\lambda]P$ for $P \in E'(\mathbb{F}_{p^2})[N]$ where $\lambda^2 \equiv -1 \pmod{N}$

Galbraith/Lin/Scott endomorphism EUROCRYPT'09

Ingredients for GLS construction **(just an overview)**:

1. E: an elliptic curve defined over $\mathbb{F}_p$ where $p > 3$
2. E': the quadratic twist of $E/\mathbb{F}_{p^2}$
3. ϕ: $E \to E'$: twisting $\mathbb{F}_{p^4}$-isomorphism
4. π_q: $E \to {}^{(q)}E$: q-power Frobenius isogeny; ${}^{(p)}E = E$, so $\pi_p \in End(E)$

Pros and cons (see Smith'13):

- Approximately p isomorphism classes ☺
- $\#E'(\mathbb{F}_{p^2})$ can be a prime ☺
- $\#E(\mathbb{F}_{p^2})$ cannot be a prime ☹
- Requires checking prohibited points on the quadratic twist ☹

see Bernstein'06, Fouque/Lercier/Réal/Valette'08

Let Δ be a square-free integer.

Quadratic $\mathbb{Q}$-curves

A quadratic $\mathbb{Q}$-curve of degree d:

- an elliptic curve $\widetilde{E}$ without complex multiplication
- $\widetilde{E}$ is defined over $\mathbb{Q}(\sqrt{\Delta})$
- existence of an isogeny of degree d

  from E to its Galois conjugate ${}^{\sigma}\widetilde{E}$,

  where

$$\langle\sigma\rangle = Gal(\mathbb{Q}(\sqrt{\Delta})/\mathbb{Q})$$

The Galois conjugate ${}^{\sigma}\widetilde{E}$ is the curve formed by applying σ to all of the coefficients of E.

Ingredients for the construction **(an overview of the degree 2 case)**:

1. $\widetilde{E}/\mathbb{Q}(\sqrt{\Delta})$: a quadratic $\mathbb{Q}$-curve of degree 2
2. E: the elliptic curve "$\widetilde{E}/\mathbb{Q}(\sqrt{\Delta})$ mod p" with $j(E/\mathbb{F}_{p^2}) \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$
3. $\phi\colon E \to {}^{(p)}E$: a degree 2 isogeny to (Galois) conjugate curve
4. $\pi_q\colon {}^{(q)}E \to E$: the q-power Frobenius isogeny

Now define $\psi := \pi_p \circ \phi$

- ψ is a (degree 2p) $\mathbb{F}_{p^2}$-endomorphism of E satisfying $\psi^2 = [\pm 2]\pi_{p^2}$
- If N is a prime such that $N \mid \#E(\mathbb{F}_{p^2})$ and $N^2 \nmid \#E(\mathbb{F}_{p^2})$ then

$$\psi^2(P) \pm r\psi(P) + 2p = \mathcal{O} \quad \text{for} \quad P \in E(\mathbb{F}_{p^2})[N]$$

  for some integer r.
- $\psi(P) = [\lambda]P$ for $P \in E'(\mathbb{F}_{p^2})[N]$ where $\lambda^2 \equiv \pm 2 \pmod{N}$

Smith's endomorphism ASIACRYPT'13

Ingredients for the construction **(an overview of the degree 2 case)**:

1. $\widetilde{E}/\mathbb{Q}(\sqrt{\Delta})$: a quadratic $\mathbb{Q}$-curve of degree 2
2. E: the elliptic curve "$\widetilde{E}/\mathbb{Q}(\sqrt{\Delta})$ mod p" with $j(E/\mathbb{F}_{p^2}) \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$
3. ϕ: $E \to {}^{(p)}E$: a degree 2 isogeny to (Galois) conjugate curve
4. π_q: ${}^{(q)}E \to E$: the q-power Frobenius isogeny

Pros and pros (see Smith'13):

- Approximately p isomorphism classes ☺
- $\#E(\mathbb{F}_{p^2})$ can be a prime ☺
- $\#E'(\mathbb{F}_{p^2})$ can be a prime ☺
- Immune to fault attacks exploiting insecure quadratic twists ☺

Writing the Smith's endomorphism explicitly I

Hasegawa family of elliptic curves over $\mathbb{Q}(\sqrt{\Delta})$:

$$\widetilde{E}_W \colon y^2 = x^3 - 6(5 - 3s\sqrt{\Delta})x + 8(7 - 9s\sqrt{\Delta}).$$

$$\begin{array}{rrcl} \hat{\phi}_W : & \widetilde{E}_W & \longrightarrow & \widetilde{E}_W/\langle(4,0)\rangle = ({}^{\sigma}\widetilde{E})^{\sqrt{-2}}, \\ & (x,y) & \longmapsto & \left(x + 2\dfrac{9(1+s\sqrt{\Delta})}{x-4}, y\left(1 - 2\dfrac{9(1+s\sqrt{\Delta})}{(x-4)^2}\right)\right) \end{array}$$

$$\delta_W : \quad \widetilde{E}_W/\langle(4,0)\rangle \longrightarrow {}^{\sigma}\widetilde{E}_W, \quad (x,y) \longmapsto (\lambda^2 x, \lambda^3 y)$$

$$\widetilde{\phi}_W : \quad \widetilde{E}_W \longrightarrow {}^{\sigma}\widetilde{E}_W, \quad (x,y) \longmapsto \delta_W(\hat{\phi}_W(x,y))$$

- $\widetilde{\phi}_W$ is defined over $\mathbb{Q}(\sqrt{\Delta}, \sqrt{-2})$
- ${}^{\sigma}\widetilde{\phi}_W \circ \widetilde{\phi}_W = [2]$ if ${}^{\sigma}(\sqrt{-2}) = -\sqrt{-2}$ and $[-2]$ if ${}^{\sigma}(\sqrt{-2}) = \sqrt{-2}$.

Hasegawa family of elliptic curves over $\mathbb{Q}(\sqrt{\Delta})$:

$$\widetilde{E}_W : y^2 = x^3 - 6(5 - 3s\sqrt{\Delta})x + 8(7 - 9s\sqrt{\Delta}).$$

$$\begin{array}{llcll} \hat{\phi}_W : & \widetilde{E}_W & \longrightarrow & \widetilde{E}_W/\langle(4,0)\rangle & = ({}^{\sigma}\widetilde{E})^{\sqrt{-2}}, \\ & (x,y) & \longmapsto & \left(x + 2\dfrac{9(1+s\sqrt{\Delta})}{x-4}, y\left(1 - 2\dfrac{9(1+s\sqrt{\Delta})}{(x-4)^2}\right)\right) & \end{array}$$

$$\delta_W : \quad \widetilde{E}_W/\langle(4,0)\rangle \longrightarrow {}^{\sigma}\widetilde{E}_W, \quad (x,y) \longmapsto (\lambda^2 x, \lambda^3 y)$$

$$\widetilde{\phi}_W : \quad \widetilde{E}_W \longrightarrow {}^{\sigma}\widetilde{E}_W, \quad (x,y) \longmapsto \delta_W(\hat{\phi}_W(x,y))$$

- $\widetilde{\phi}_W$ is defined over $\mathbb{Q}(\sqrt{\Delta}, \sqrt{-2})$
- ${}^{\sigma}\widetilde{\phi}_W \circ \widetilde{\phi}_W = [2]$ if ${}^{\sigma}(\sqrt{-2}) = -\sqrt{-2}$ and $[-2]$ if ${}^{\sigma}(\sqrt{-2}) = \sqrt{-2}$.

Writing the Smith's endomorphism explicitly I

Hasegawa family of elliptic curves over $\mathbb{Q}(\sqrt{\Delta})$:

$$\widetilde{E}_W : y^2 = x^3 - 6(5 - 3s\sqrt{\Delta})x + 8(7 - 9s\sqrt{\Delta}).$$

$$\begin{array}{llcl} \hat{\phi}_W : & \widetilde{E}_W & \longrightarrow & \widetilde{E}_W/\langle(4,0)\rangle = ({}^{\sigma}\widetilde{E})^{\sqrt{-2}}, \\ & (x,y) & \longmapsto & \left(x + 2\dfrac{9(1+s\sqrt{\Delta})}{x-4}, y\left(1 - 2\dfrac{9(1+s\sqrt{\Delta})}{(x-4)^2}\right)\right) \end{array}$$

$$\delta_W : \quad \widetilde{E}_W/\langle(4,0)\rangle \longrightarrow {}^{\sigma}\widetilde{E}_W, \quad (x,y) \longmapsto (\lambda^2 x, \lambda^3 y)$$

$$\widetilde{\phi}_W : \quad \widetilde{E}_W \longrightarrow {}^{\sigma}\widetilde{E}_W, \quad (x,y) \longmapsto \delta_W(\hat{\phi}_W(x,y))$$

- $\widetilde{\phi}_W$ is defined over $\mathbb{Q}(\sqrt{\Delta}, \sqrt{-2})$
- ${}^{\sigma}\widetilde{\phi}_W \circ \widetilde{\phi}_W = [2]$ if ${}^{\sigma}(\sqrt{-2}) = -\sqrt{-2}$ and $[-2]$ if ${}^{\sigma}(\sqrt{-2}) = \sqrt{-2}$.

We reduce $\widetilde{E}_W$ and $\widetilde{\phi}_W$ modulo a "good" p and obtain E_W and ϕ.

We see that

$$ {}^{\sigma}\widetilde{E}_W \quad \text{reduces to} \quad {}^{(p)}E_W $$

and

$$ \widetilde{\phi}_W \colon \widetilde{E}_W \to {}^{\sigma}\widetilde{E}_W \quad \text{reduces to} \quad \phi_W \colon E_W \to {}^{(p)}E_W. $$

$$ \pi_p : \qquad {}^{(p)}E_W \longrightarrow E_W \quad (x, y) \longmapsto \left({}^{(p)}x, {}^{(p)}y\right) $$

$$ \begin{aligned} \psi_W : \ E_W &\longrightarrow E_W, \\ (x, y) &\longmapsto \pi_p(\phi_W(x, y)) = \\ & \left(\frac{-x^p}{2} - \frac{9(1 + s\sqrt{\Delta})}{x^p - 4}, \ \frac{y^p}{\sqrt{-2}} \left(\frac{-1}{2} + \frac{9(1 + s\sqrt{\Delta})}{(x^p - 4)^2} \right) \right) \end{aligned} $$

Smith's endomorphism for Montgomery form I

- Assume that $8/A^2 = 1 + s\sqrt{\Delta}$ from now on.
- We define $\mathcal{E}$ to be the elliptic curve over $\mathbb{F}_{p^2}$ with affine Montgomery model

$$\mathcal{E}\colon y^2 = x(x^2 + Ax + 1)$$

- If the element $12/A$ is not a square in $\mathbb{F}_{p^2}$, the curve over $\mathbb{F}_{p^2}$ defined by

$$\mathcal{E}'\colon (12/A)y^2 = x(x^2 + Ax + 1)$$

  is a model of the quadratic twist of $\mathcal{E}$.
- The twisting $\mathbb{F}_{p^4}$-isomorphism $\delta : \mathcal{E} \to \mathcal{E}'$ is defined by

$$\delta\colon (x, y) \mapsto (x, y\sqrt{A/12}).$$

Smith's endomorphism for Montgomery form II

- The map

$$\delta_1 \colon (x, y) \mapsto (x_W, y_W) = (\frac{12}{A}x + 4, \frac{12^2}{A^2}y)$$

defines an $\mathbb{F}_{p^2}$-isomorphism between $\mathcal{E}'$ and the Hasegawa curve in Weierstrass form.

- Applying the isomorphisms δ and δ_1, we define efficient $\mathbb{F}_{p^2}$-endomorphisms

$$\psi := (\delta_1\delta)^{-1}\psi_W\delta_1\delta \qquad \text{and} \qquad \psi' := \delta\psi\delta^{-1} = \delta_1^{-1}\psi_W\delta_1$$

of degree $2p$ on $\mathcal{E}$ and $\mathcal{E}'$, respectively, each with kernel $\langle(0, 0)\rangle$.

- More explicitly, ψ and ψ' reads as follows:

$$\psi\colon\ (x,y) \longmapsto \left(s(x)\ ,\ \frac{-12^{(p-1)/2}}{A^{(p-1)/2}\sqrt{-2}}\frac{y^p m(x)^p}{d(x)^{2p}} \right)\ ,$$

$$\psi'\colon\ (x,y) \longmapsto \left(s(x)\ ,\ \frac{-12^{p-1}\sqrt{-2}}{A^{p-1}}\frac{y^p r(x)^p}{d(x)^{2p}} \right)$$

where

$$n(x) := \tfrac{A^p}{A}\left(x^2 + Ax + 1\right)\ ,\quad d(x) := -2x\ ,\quad s(x) := n(x)^p / d(x)^p\ ,$$

$$r(x) := \tfrac{A^p}{A}(x^2 - 1)\ ,\quad m(x) := n'(x)d(x) - n(x)d'(x)\ .$$

Selecting a secure Montgomery curve $y^2 = x^3 + Ax + x$

We are at a point to fix all free parameters for cryptographic concern:

- We set $\Delta = \sqrt{-1} = i$, $p = 2^{127} - 1$, and $\mathbb{F}_{p^2} = \mathbb{F}_p[x]/\langle i^2 + 1 \rangle$.
- We fix $\sqrt{-2} := 2^{64} \cdot i$.
- We chose $s = 86878915556079486902897638486322141403$.
- Then, we get $A = A_0 + A_1 \cdot i$ where

$$\begin{cases} A_0 = 45116554344555875085017627593321485421 \ , \\ A_1 = 2415910908 \qquad \text{satisfying } 8/A^2 = 1 + s\sqrt{\Delta}. \end{cases}$$

- We define $u := 1466100457131508421$.
- We define $v := (p-1)/2 = 2^{126} - 1$ and $w := (p+1)/4 = 2^{125}$.
- We get

$$\#\mathcal{E} = 4 \cdot N \quad \text{and} \quad \#\mathcal{E}' = 8 \cdot N'$$

where N is a 252 bit and N' is a 251 bit prime.

$$N = v^2 + 2u^2 \qquad \text{and} \qquad N' = 2w^2 - u^2.$$

Targeting 128-bit security level

- Large embedding degrees of $\mathcal{E}$ and $\mathcal{E}'$; Menezes/Okamoto/Vanstone'93 or Frey/Rück'99 attacks are not a threat.
- The trace of $\mathcal{E}$ is $p^2 + 1 - 4N \neq \pm 1$, so neither $\mathcal{E}$ nor $\mathcal{E}'$ are amenable to the Smart–Satoh–Araki–Semaev'98 -'99 attacks.
- The Weil restriction of $\mathcal{E}$ (or $\mathcal{E}'$) to $\mathbb{F}_p$ as in the Gaudry/Hess/Smart'02 produces a simple abelian surface over $\mathbb{F}_p$; which is also secure.
- $\mathrm{End}(\mathcal{E}) = \mathbb{Z}[\psi]$, see the paper.
- The `safecurves` specification suggests that the discriminant of the CM field should have at least 100 bits; our $\mathcal{E}$ easily meets this requirement, since D_K has 130 bits.

- Brainpool requires the ideal class number of K to be larger than 10^7; $\mathcal{E}$ easily meets this requirement: the class number of $\mathrm{End}(\mathcal{E})$ is

$$h(\mathrm{End}(\mathcal{E})) = h(D_K) = 2^7 \cdot 31 \cdot 37517 \cdot 146099 \cdot 505117 \sim 10^{19} \ .$$

- Both $\mathcal{E}$ and $\mathcal{E}'$ are compatible with the Elligator 2 construction, see Bernstein/Hamburg/Krasnova/Lange'13
- Theorem 5 of Elligator: invertible injective maps $\mathbb{F}_{p^2} \to \mathcal{E}(\mathbb{F}_{p^2})$ and $\mathbb{F}_{p^2} \to \mathcal{E}'(\mathbb{F}_{p^2})$. $\mathcal{E}$ and/or $\mathcal{E}'$ can be encoded in such a way that they are indistinguishable from uniformly random 254-bit strings.
- Twist secure, so immune to Fouque/Lercier/Réal/Valette'08 fault attacks

The importance of twist-security

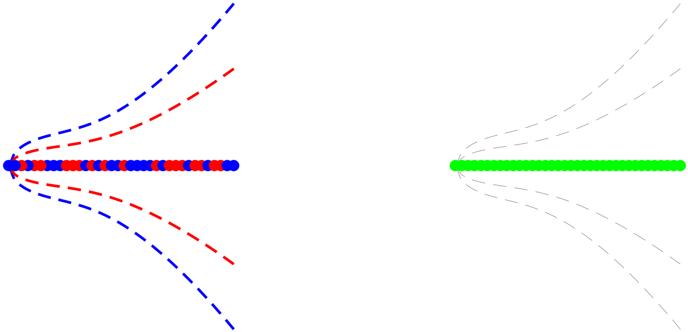
Compact scalar multiplications:

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

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

- 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

x-line scalar multiplication without endomorphisms

```
//        MONTGOMERY CURVE: Y^2*Z = X^3 + A*X^2*Z + X*Z^2
```

```
function LADDER(k,X1,Z1,A)                              //MONTGOMERY LADDER
  X2:=(X1^2-Z1^2)^2;          Z2:=4*X1*Z1*(X1^2+A*X1*Z1+Z1^2);
  X3:=X1;                     Z3:=Z1;
  for j:=#k-1 to 1 by -1 do
    if k[j] eq 1 then
      X2,Z2,X3,Z3:=DBLADD(X2,Z2,X3,Z3,X1,Z1,A);
    else
      X3,Z3,X2,Z2:=DBLADD(X3,Z3,X2,Z2,X1,Z1,A);
    end if;
  end for;
  return X3,Z3;
end function;
```

x-line scalar multiplication without endomorphisms

```
//       MONTGOMERY CURVE: Y^2*Z = X^3 + A*X^2*Z + X*Z^2

DBLADD:=function(X2,Z2,X3,Z3,X1,Z1,A)
  X4:=(X2^2-Z2^2)^2;          Z4:=4*X2*Z2*(X2^2+A*X2*Z2+Z2^2); //DBL
  X5:=Z1*(X2*X3-Z2*Z3)^2;     Z5:=X1*(X2*Z3-Z2*X3)^2;              //ADD
  return X4,Z4,X5,Z5;
end function;

function LADDER(k,X1,Z1,A)                              //MONTGOMERY LADDER
  X2:=(X1^2-Z1^2)^2;          Z2:=4*X1*Z1*(X1^2+A*X1*Z1+Z1^2);
  X3:=X1;                     Z3:=Z1;
  for j:=#k-1 to 1 by -1 do
    if k[j] eq 1 then
      X2,Z2,X3,Z3:=DBLADD(X2,Z2,X3,Z3,X1,Z1,A);
    else
      X3,Z3,X2,Z2:=DBLADD(X3,Z3,X2,Z2,X1,Z1,A);
    end if;
  end for;
  return X3,Z3;
end function;
```

Scalar decomposition I

We want to evaluate scalar multiplications $[m]P$ as $[a]P \oplus [b]\psi(P)$, where

$$m \equiv a + b\lambda \pmod{N}$$

and the multiscalar (a, b) has a significantly shorter bitlength than m.

Two extra requirements on (a, b), so as to add a measure of side-channel resistance:

1. both a and b must be **positive**, to avoid branching and to simplify our algorithms; and
2. the multiscalar (a, b) must have **constant bitlength** (independent of m as m varies over $\mathbb{Z}$), so that multiexponentiation can run in constant time.

Scalar decomposition II

The usual technique:

1. Compute a reduced basis for

$$\mathcal{L} = \langle (N, 0), (-\lambda, 1) \rangle \qquad \text{and} \qquad \mathcal{L}' = \big\langle (N', 0), (-\lambda', 1) \big\rangle$$

   using one of the available techniques e.g. LLL algorithm.

2. Compute the unique $(\alpha, \beta) \in \mathbb{Q}^2$ satisfying

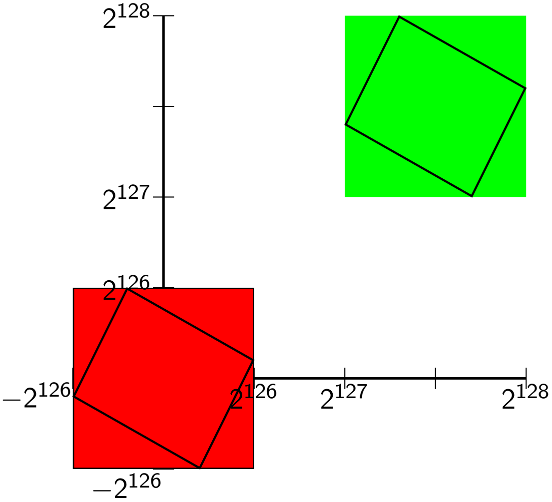
$$\alpha \mathbf{e}_1 + \beta \mathbf{e}_2 = (m, 0).$$

3. Use Babai rounding to transform each scalar m into the multiscalar $(\tilde{a}, \tilde{b})$ by

$$(\tilde{a}, \tilde{b}) := (m, 0) - \lfloor \alpha \rceil \mathbf{e}_1 - \lfloor \beta \rceil \mathbf{e}_2.$$

- **Consequence:** Bitlength of $\tilde{a}$ and $\tilde{b}$ can be at most 126 bits.
- **Problem:** Bitlength of $\tilde{a}$ and $\tilde{b}$ can be less than 126 bits.
- **Problem:** $\tilde{a}$ or $\tilde{b}$ can be negative.

Scalar decomposition III



Scalar decomposition IV

- **Solution:** Add a carefully selected offset vector to $(\tilde{a}, \tilde{b})$.

$$(a, b) := (m, 0) - \lfloor \alpha \rceil \mathbf{e}_1 - \lfloor \beta \rceil \mathbf{e}_2 + 3(\mathbf{e}_1 + \mathbf{e}_2).$$

- **Consequence:** Bitlength of a and b are exactly 128 bits.
- **Consequence:** Both a and b are positive.

Theorem

Given an integer m, let (a, b) be the multiscalar defined by

$$a := m + (3 - \lfloor (v/N)m \rceil)\, v - 2\,(3 - \lfloor -(u/N)m \rceil)\, u$$
$$b := (3 - \lfloor (v/N)m \rceil)\, u + (3 - \lfloor -(u/N)m \rceil)\, v$$

We have $2^{127} < a, b < 2^{128}$, and

$$m \equiv a + b\lambda \pmod{N}.$$

- One dimensional (1-D) ladder:

$$m, x(P) \longmapsto x([m]P)$$

- Two-dimensional (2-D) ladder:

$$a, b, x(P), x(\psi(P)), x(\psi(P) - P) \longmapsto x([a]P + [b]\psi(P))$$

- Three 2-D ladders 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

$$\ell = \max\{\lfloor \log_2 a \rfloor, \lfloor \log_2 b \rfloor\} + 1$$

Kickstarting 2-D addition chains . . .

- All three chains requires a computation of

$$x(\psi(P) - P) = x\left((\psi - 1)(P)\right)$$

Computing the initial difference:

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

where f and g have low degree.

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

Projective ψ and $\psi + 1$

- The pseudo-doubling on $\mathbb{P}^1$ is

$$[2]_x((X:Z)) = \left((X+Z)^2(X-Z)^2 : (4XZ)\left((X-Z)^2 + \tfrac{A+2}{4}\cdot 4XZ\right)\right).$$

- Our endomorphism ψ induces the pseudo-endomorphism

$$\psi_x((X:Z)) = \left(A^p\left((X-Z)^2 - \tfrac{A+2}{2}(-2XZ)\right)^p : A(-2XZ)^p\right).$$

- Composing ψ_x with itself, we confirm that $\psi_x\psi_x = -[2]_x(\pi_q)_x$.
- $\psi + 1$ is as follows:

$$\begin{aligned}(\psi - 1)_x(x) &= (\psi' - 1)_x(x) \\ &= \frac{2s^2nd^{4p} - x(xn)^p m^{2p} A^{p-1}}{2s(x-s)^2 d^{4p} A^{p-1}} \mp \frac{m^p (xn)^{(p+1)/2}\sqrt{-2}}{A^{(p-1)/2}(x-s)^2 d^{2p}}.\end{aligned}$$

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*

- 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

- Yanik/Tugrul/Koc'02, Longa/Miri'08
  - Inputs come from range $[0, p-1]$.
  - Outputs are generated in range $[0, 2^b - 1]$.
  - An addition is prohibited to be followed by another addition
- This restriction can be eliminated for $p = 2^{127} - 1$:
  - Inputs come from range $[0, 2^{127} - 1]$.
  - Outputs are generated in range $[0, 2^{127} - 1]$.
  - An addition can be followed by another addition

Semi-reduced addition modulo $p = 2^{127} - 1$

The operation $f := (a + b) \bmod p$ is replaced by the following algorithm:

Let $a, b \in \mathbb{Z}$ such that $0 \leq a, b \leq p$

1. $c := (a + b) \bmod 2^{128}$
2. $d := (c_0, c_1, \ldots, c_{126})$, $e := (c_{127})$
3. $f := (d + e) \bmod 2^{128}$

- **Line-1:** Notice that $0 \leq c = a + b \leq 2p < 2^{128}$.

Semi-reduced addition modulo $p = 2^{127} - 1$

The operation $f := (a + b) \bmod p$ is replaced by the following algorithm:

Let $a, b \in \mathbb{Z}$ such that $0 \le a, b \le p$

1. $c := (a + b) \bmod 2^{128}$
2. $d := (c_0, c_1, \ldots, c_{126})$, $e := (c_{127})$
3. $f := (d + e) \bmod 2^{128}$

- **Line-1:** Notice that $0 \le c = a + b \le 2p < 2^{128}$.
- **Line-2:** Write $c = d + 2^{127}e$ for integers $0 \le d < 2^{127}$ and e. There are two cases to investigate:
  - Case 1: Assume that $a + b \le p$. The bounds on c and d imply that $\lfloor 0/2^{127} \rfloor \le \lfloor c/2^{127} \rfloor = \lfloor (d + 2^{127}e)/2^{127} \rfloor = \lfloor d/2^{127} \rfloor + \lfloor 2^{127}e/2^{127} \rfloor = e \le \lfloor p/2^{127} \rfloor$, so $e = 0$. Thus $a + b \equiv d + 2^{127}e \equiv d + 2^{127} \cdot 0 \equiv d + 0 \equiv \underline{d + e} \pmod{p}$.

Semi-reduced addition modulo $p = 2^{127} - 1$

The operation $f := (a + b) \bmod p$ is replaced by the following algorithm:

Let $a, b \in \mathbb{Z}$ such that $0 \leq a, b \leq p$

1. $c := (a + b) \bmod 2^{128}$
2. $d := (c_0, c_1, \ldots, c_{126}),\ e := (c_{127})$
3. $f := (d + e) \bmod 2^{128}$

- **Line-1:** Notice that $0 \leq c = a + b \leq 2p < 2^{128}$.
- **Line-2:** Write $c = d + 2^{127}e$ for integers $0 \leq d < 2^{127}$ and e. There are two cases to investigate:
  - Case 2: Assume that $a + b > p$. Then $p < c \leq 2p$. The bounds on c and d imply that $\lfloor (p+1)/2^{127} \rfloor \leq e \leq \lfloor 2p/2^{127} \rfloor$, so $e = 1$. The bounds on c also imply that $p - 2^{127} < c - 2^{127} \leq 2p - 2^{127}$ and we have $d = c - 2^{127}e = c - 2^{127}$, so $0 \leq d < p$. Thus $a + b \equiv d + 2^{127}e \equiv d + 2^{127} \cdot 1 \equiv d + 1 \equiv \underline{d + e} \pmod{p}$.

Semi-reduced addition modulo $p = 2^{127} - 1$

The operation $f := (a + b) \bmod p$ is replaced by the following algorithm:

Let $a, b \in \mathbb{Z}$ such that $0 \leq a, b \leq p$

1. $c := (a + b) \bmod 2^{128}$
2. $d := (c_0, c_1, \ldots, c_{126})$, $e := (c_{127})$
3. $f := (d + e) \bmod 2^{128}$

- **Line-1:** Notice that $0 \leq c = a + b \leq 2p < 2^{128}$.
- **Line-3:** A semi-reduced output is given by $f := (d + e) \bmod 2^{128}$, observing that $0 \leq f \leq p$.

Semi-reduced addition modulo $p = 2^{127} - 1$

Max 9 instructions:

```
movq 8*0+OPERAND1, %r12
addq 8*0+OPERAND2, %r12
movq 8*1+OPERAND1, %rsi
adcq 8*1+OPERAND2, %rsi
btrq $63, %rsi
adcq $0, %r12
movq %r12, 8*0+OUTPUT
adcq $0, %rsi
movq %rsi, 8*1+OUTPUT
```

Semi-reduced subtraction modulo $p = 2^{127} - 1$

The operation $f := (a - b) \bmod p$ is replaced by the following algorithm:

$a, b \in \mathbb{Z}$ such that $0 \le a, b \le p$

1. $c := (a - b) \bmod 2^{128}$
2. $d := (c_0, c_1, \ldots, c_{126})$, $e := (c_{127})$
3. $f := (d - e) \bmod 2^{128}$

- **Line-1:** Notice that $0 \le c < 2^{128}$.

Semi-reduced subtraction modulo $p = 2^{127} - 1$

The operation $f := (a - b) \bmod p$ is replaced by the following algorithm:

$a, b \in \mathbb{Z}$ such that $0 \leq a, b \leq p$

1. $c := (a - b) \bmod 2^{128}$
2. $d := (c_0, c_1, \ldots, c_{126})$, $e := (c_{127})$
3. $f := (d - e) \bmod 2^{128}$

- **Line-1:** Notice that $0 \leq c < 2^{128}$.
- **Line-2:** Write $c = d + 2^{127}e$ for integers $0 \leq d < 2^{127}$ and e. There are two cases to investigate:
  - Case 1: Assume that $a \geq b$. Then $0 \leq c = a - b \leq p$. The bounds on c and d imply that $\lfloor 0/2^{127} \rfloor \leq \lfloor c/2^{127} \rfloor = \lfloor (d + 2^{127}e)/2^{127} \rfloor = e \leq \lfloor p/2^{127} \rfloor$, so $e = 0$. Thus $a - b \equiv d + 2^{127}e \equiv \underline{d - e} \pmod{p}$.

Semi-reduced subtraction modulo $p = 2^{127} - 1$

The operation $f := (a - b) \bmod p$ is replaced by the following algorithm:

$a, b \in \mathbb{Z}$ such that $0 \le a, b \le p$

1. $c := (a - b) \bmod 2^{128}$
2. $d := (c_0, c_1, \ldots, c_{126})$, $e := (c_{127})$
3. $f := (d - e) \bmod 2^{128}$

- **Line-1:** Notice that $0 \le c < 2^{128}$.
- **Line-2:** Write $c = d + 2^{127}e$ for integers $0 \le d < 2^{127}$ and e. There are two cases to investigate:
  - Case 2: Assume that $a < b$. Then $c = 2^{128} + a - b$ and $-p \le a - b < 0$. So, $2^{127} < c < 2^{128}$. The bounds on c and d imply that $\lfloor (2^{127} + 1)/2^{127} \rfloor \le e \le \lfloor (2^{128} - 1)/2^{127} \rfloor$, so $e = 1$. The bounds on c also imply that $2^{127} - 2^{127} < c - 2^{127} < 2^{128} - 2^{127}$, and we have $d = c - 2^{127}e = c - 2^{127}$. So, $0 < d \le p$ and $d \ge e$. Thus $a - b \equiv (2^{128} + a - b) - 2^{128} \equiv c - 2^{128} \equiv d + 2^{127}e - 2^{128} \equiv \underline{d - e} \pmod{p}$.

Semi-reduced subtraction modulo $p = 2^{127} - 1$

The operation $f := (a - b) \bmod p$ is replaced by the following algorithm:

$a, b \in \mathbb{Z}$ such that $0 \leq a, b \leq p$

1. $c := (a - b) \bmod 2^{128}$
2. $d := (c_0, c_1, \ldots, c_{126})$, $e := (c_{127})$
3. $f := (d - e) \bmod 2^{128}$

- **Line-1:** Notice that $0 \leq c < 2^{128}$.
- **Line-3:** A semi-reduced output is given by $f := (d - e) \bmod 2^{128}$, observing that $0 \leq f \leq p$.

Semi-reduced subtraction modulo $p = 2^{127} - 1$

Max 9 instructions:

```
movq 8*0+OPERAND1, %r12
subq 8*0+OPERAND2, %r12
movq 8*1+OPERAND1, %rsi
sbbq 8*1+OPERAND2, %rsi
btrq $63, %rsi
sbbq $0, %r12
movq %r12, 8*0+OUTPUT
sbbq $0, %rsi
movq %rsi, 8*1+OUTPUT
```

Semi-reduced multiplication modulo $p = 2^{127} - 1$

The operation $f := (a \cdot b) \bmod p$ is replaced by the following algorithm:

Let $a, b \in \mathbb{Z}$ such that $0 \leq a, b \leq p$

1. $c := (ab) \bmod 2^{256}$
2. $d := (c_0, c_1, \ldots, c_{126})$, $e := (c_{127}, c_{128}, \ldots, c_{253})$
3. $f := \text{semi-add}(d, e)$

- **Line-1:** Notice that $0 \leq c = ab \leq p^2 < 2^{256}$.

Semi-reduced multiplication modulo $p = 2^{127} - 1$

The operation $f := (a \cdot b) \bmod p$ is replaced by the following algorithm:

Let $a, b \in \mathbb{Z}$ such that $0 \leq a, b \leq p$

1. $c := (ab) \bmod 2^{256}$
2. $d := (c_0, c_1, \ldots, c_{126})$, $e := (c_{127}, c_{128}, \ldots, c_{253})$
3. $f := \text{semi-add}(d, e)$

- **Line-1:** Notice that $0 \leq c = ab \leq p^2 < 2^{256}$.
- **Line-2:** Write $c = d + 2^{127}e$ for integers $0 \leq d < 2^{127}$ and e. The bounds on c and d imply that $\lfloor 0/2^{127} \rfloor \leq \lfloor c/2^{127} \rfloor = \lfloor (d + 2^{127}e)/2^{127} \rfloor = e \leq \lfloor p^2/2^{127} \rfloor$, so $0 \leq e < p$.

Semi-reduced multiplication modulo $p = 2^{127} - 1$

The operation $f := (a \cdot b) \bmod p$ is replaced by the following algorithm:

Let $a, b \in \mathbb{Z}$ such that $0 \le a, b \le p$

1. $c := (ab) \bmod 2^{256}$
2. $d := (c_0, c_1, \ldots, c_{126})$, $e := (c_{127}, c_{128}, \ldots, c_{253})$
3. $f := \text{semi-add}(d, e)$

- **Line-1:** Notice that $0 \le c = ab \le p^2 < 2^{256}$.
- **Line-3:** Noting that $ab \equiv d + 2^{127}e \equiv d + (2^{127} - 1)e + e \equiv d + pe + e \equiv d + e \pmod{p}$, that $0 \le d, e \le p$, and that $0 \le d + e \le 2p$, a semi-reduced output is obtained by semi-reduced addition applied on the operands d and e.

Semi-reduced multiplication modulo $p = 2^{127} - 1$

Max 27 instructions:

```
movq 8*0+OPERAND1, %rax
mulq 8*1+OPERAND2
movq %rdx, %r10
movq %rax, %rsi
movq 8*1+OPERAND1, %rax
mulq 8*0+OPERAND2
addq %rax, %rsi
adcq %rdx, %r10
movq 8*0+OPERAND2, %rax
mulq 8*0+OPERAND1
addq %rdx, %rsi
movq %rax, %r12
adcq $0, %r10
movq 8*1+OPERAND1, %rax
mulq 8*1+OPERAND2
addq %r10, %rax
adcq $0, %rdx
addq %rax, %rax
adcq %rdx, %rdx
btrq $63, %rsi
adcq %rax, %r12
adcq %rdx, %rsi
btrq $63, %rsi
adcq $0, %r12
movq %r12, 8*0+OUTPUT
adcq $0, %rsi
movq %rsi, 8*1+OUTPUT
```

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/