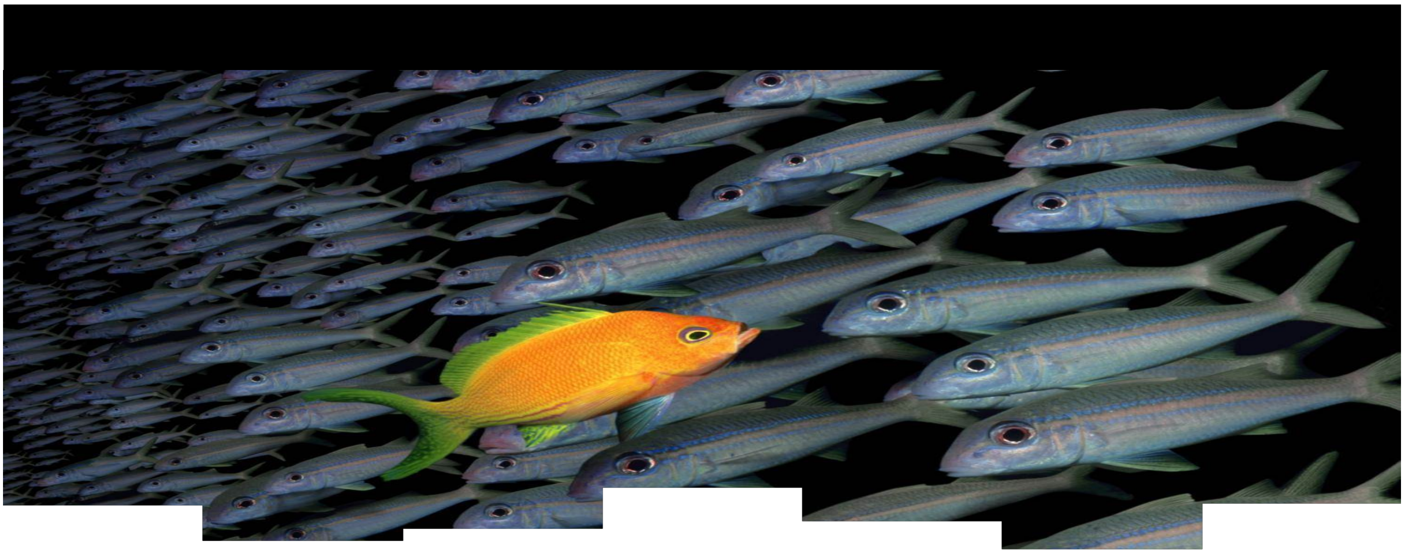
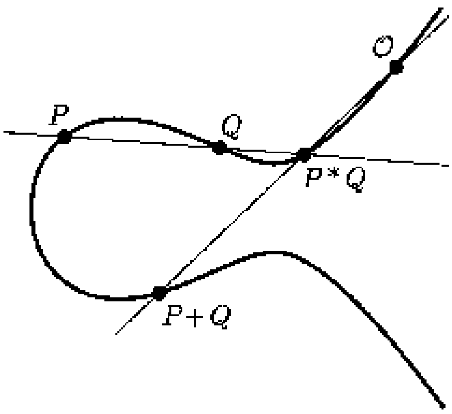
Inversion-Free Arithmetic on Elliptic Curves Through Isomorphisms



Marc Joye

Elliptic Curve Cryptography

- Invented [independently] by Neil Koblitz and Victor Miller in 1985



- Useful for key exchange, encryption and digital signature

Scalar Multiplication

Definition

Given scalar k and a point $\boldsymbol{P}$, compute $k\boldsymbol{P} = \underbrace{\boldsymbol{P} + \boldsymbol{P} + \cdots + \boldsymbol{P}}_{k \text{ times}}$

ECDLP Given $\boldsymbol{P}$ and $\boldsymbol{Q} = k\boldsymbol{P}$, recover k

- no subexponential algorithms are known to solve the ECDLP (in the *general* case)
- smaller key sizes can be used

	Bit security				
	80	112	**128**	192	256
ECC	160	224	256	384	512
RSA	1024	2048	3072	8192	15360

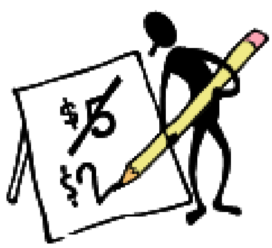
technicolor

This Talk

Goal

Generalization of Meloni's co-Z arithmetic on elliptic curves

- all elliptic curve models
- all scalar multiplication algorithms
- (suitable for memory-constrained devices)



technicolor

Outline

1 Arithmetic on Elliptic Curves
- Jacobian coordinates
- Co-Z point addition

2 Binary Scalar Multiplication Algorithms

3 New Implementations
- Conjugate point addition
- Binary ladders with co-Z trick

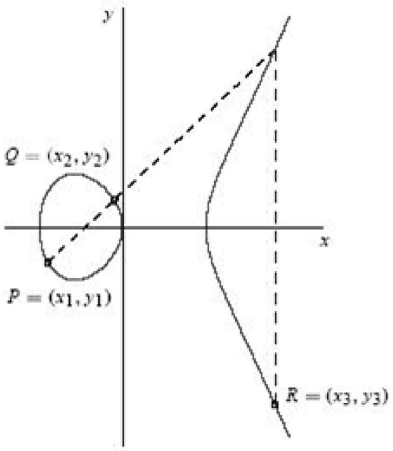
4 Generalization
- Meloni's technique revisited
- Inversion-free arithmetic through isomorphisms
- New operations
- Some results

5 Conclusion

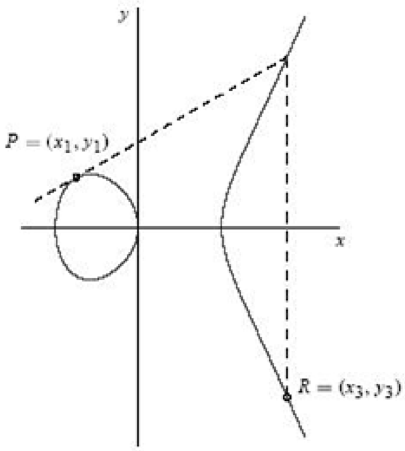
Elliptic Curves

Weierstraß equation (affine coordinates)

Let $E: y^2 = x^3 + ax + b$ define over $\mathbb{F}_q$ ($char \neq 2, 3$) with discriminant $\Delta = -16(4a^3 + 27b^2) \neq 0$



(a) Addition: $P + Q = R$.



(b) Doubling: $P + P = R$.

Group Law

$$E(\mathbb{F}_q) = \{y^2 = x^3 + ax + b\} \cup \{\boldsymbol{O}\}$$

- Let $\boldsymbol{P} = (x_1, y_1)$ and $\boldsymbol{Q} = (x_2, y_2)$
- Group law
  - $\boldsymbol{P} + \boldsymbol{O} = \boldsymbol{O} + \boldsymbol{P} = \boldsymbol{P}$
  - $-\boldsymbol{P} = (x_1, -y_1)$
  - $\boldsymbol{P} + \boldsymbol{Q} = (x_3, y_3)$ where

$$x_3 = \lambda^2 - x_1 - x_2, \quad y_3 = (x_1 - x_3)\lambda - y_1$$

$$\text{with } \lambda = \begin{cases} \dfrac{y_1 - y_2}{x_1 - x_2} & \text{[addition]} \\ \dfrac{3x_1^2 + a}{2y_1} & \text{[doubling]} \end{cases}$$

technicolor

Jacobian Coordinates

- To avoid computing inverses in $\mathbb{F}_q$
  - affine point $(x, y) \to$ projective point $(X : Y : Z)$ such that $x = X/Z^2$ and $y = Y/Z^3$

Weierstraß equation (projective Jacobian coordinates)

Let $E : Y^2 = X^3 + aXZ^4 + bZ^6$ define over $\mathbb{F}_q$ ($char \neq 2, 3$) with discriminant $\Delta = -16(4a^3 + 27b^2) \neq 0$

- Point at infinity $\boldsymbol{O} = (1 : 1 : 0)$
- If $\boldsymbol{P} = (X_1 : Y_1 : Z_1) \in E$ then $-\boldsymbol{P} = (X_1 : -Y_1 : Z_1)$

technicolor

Best Addition Formulæ

- Jacobian point addition: 11M + 5S
- Jacobian point doubling: 1M + 8S + 1c

Co-Z Point Addition (ZADD)

- Introduced by Meloni [WAIFI 2007]
- Addition of two distinct points with the same Z-coordinate

Co-Z point addition

Let $\boldsymbol{P} = (X_1 : Y_1 : Z)$ and $\boldsymbol{Q} = (X_2 : Y_2 : Z)$. Then $\boldsymbol{P} + \boldsymbol{Q} = (X_3 : Y_3 : Z_3)$ where

$$X_3 = D - W_1 - W_2, \quad Y_3 = (Y_1 - Y_2)(W_1 - X_3) - A_1, \quad Z_3 = Z(X_1 - X_2)$$

with $A_1 = Y_1(W_1 - W_2)$, $W_1 = X_1C$, $W_2 = X_2C$, $C = (X_1 - X_2)^2$ and $D = (Y_1 - Y_2)^2$

- Cost of ZADD: 5M + 2S

Co-Z Point Addition with Update (ZADDU)

- Main advantage of Meloni's addition

Equivalent representation of $\boldsymbol{P}$

Evaluation of $\boldsymbol{R} = \mathrm{ZADD}(\boldsymbol{P}, \boldsymbol{Q})$ yields for free

$$\boldsymbol{P}' = (X_1(X_1 - X_2)^2 : Y_1(X_1 - X_2)^3 : Z_3) = (W_1 : A_1 : Z_3) \sim \boldsymbol{P}$$

that is, $\mathrm{Z}(\boldsymbol{P}') = \mathrm{Z}(\boldsymbol{R})$

- Notation: $(\boldsymbol{R}, \boldsymbol{P}') = \mathrm{ZADDU}(\boldsymbol{P}, \boldsymbol{Q})$
- Cost of ZADDU: $5\mathrm{M} + 2\mathrm{S}$

Classical Methods

Algorithm 1 Left-to-right binary method

Input: $\boldsymbol{P} \in E(\mathbb{F}_q)$ and $k = (k_{n-1}, \ldots, k_0)_2 \in \mathbb{N}$
Output: $\boldsymbol{Q} = k\boldsymbol{P}$

$\boldsymbol{R_0} \leftarrow \boldsymbol{O}$; $\boldsymbol{R_1} \leftarrow \boldsymbol{P}$
for $i = n - 1$ down to 0 **do**
  $\boldsymbol{R_0} \leftarrow 2\boldsymbol{R_0}$
  if $(k_i = 1)$ **then** $\boldsymbol{R_0} \leftarrow \boldsymbol{R_0} + \boldsymbol{R_1}$
end for
return $\boldsymbol{R_0}$

Algorithm 2 Montgomery ladder

Input: $\boldsymbol{P} \in E(\mathbb{F}_q)$ and $k = (k_{n-1}, \ldots, k_0)_2 \in \mathbb{N}$
Output: $\boldsymbol{Q} = k\boldsymbol{P}$

$\boldsymbol{R_0} \leftarrow \boldsymbol{O}$; $\boldsymbol{R_1} \leftarrow \boldsymbol{P}$
for $i = n - 1$ down to 0 **do**
  $b \leftarrow k_i$; $\boldsymbol{R_{1-b}} \leftarrow \boldsymbol{R_{1-b}} + \boldsymbol{R_b}$
  $\boldsymbol{R_b} \leftarrow 2\boldsymbol{R_b}$
end for
return $\boldsymbol{R_0}$

Conjugate co-Z Point Addition (ZADDC)

- New co-Z point operation
  - using caching techniques

Conjugate co-Z point addition

From $-\boldsymbol{Q} = (X_2 : -Y_2 : Z_2)$, evaluation of $\boldsymbol{R} = \text{ZADD}(\boldsymbol{P}, \boldsymbol{Q})$ allows one to get $\boldsymbol{S} := \boldsymbol{P} - \boldsymbol{Q} = (\overline{X_3}, \overline{Y_3}, Z_3)$ where

$$\overline{X_3} = (Y_1 + Y_2)^2 - W_1 - W_2, \quad \overline{Y_3} = (Y_1 + Y_2)(W_1 - \overline{X_3})$$

with an additional cost of 1M + 1S

- Notation: $(\boldsymbol{P} + \boldsymbol{Q}, \boldsymbol{P} - \boldsymbol{Q}) = \text{ZADDC}(\boldsymbol{P}, \boldsymbol{Q})$
- Total cost of ZADDC: 6M + 3S

Left-to-Right Binary Ladder With co-Z Trick

Algorithm 3 Montgomery ladder with co-Z formulæ

Input: $\boldsymbol{P} \in E(\mathbb{F}_q)$ and $k = (k_{n-1}, \ldots, k_0)_2 \in \mathbb{N}$ with $k_{n-1} = 1$
Output: $\boldsymbol{Q} = k\boldsymbol{P}$

1: $\boldsymbol{R_0} \leftarrow \boldsymbol{O}$; $\boldsymbol{R_1} \leftarrow \boldsymbol{P}$
2: **for** $i = n - 1$ down to 0 **do**
3: $\quad b \leftarrow k_i$; $\boldsymbol{R_{1-b}} \leftarrow \boldsymbol{R_{1-b}} + \boldsymbol{R_b}$
4: $\quad \boldsymbol{R_b} \leftarrow 2\boldsymbol{R_b}$
5: **end for**
6: **return** $\boldsymbol{R_0}$

- Cost per bit: (6M + 3S) + (5M + 2S) = 11M + 5S

Improved version: 8M + 6S

Can We Generalize the Approach?

N. Meloni
New point addition formulæ for ECC applications
Proc. of WAIFI 2007, LNCS 4537, pp. 189-201, Springer, 2007

R. Goundar, M. Joye, A. Miyaji, M. Rivain, and A. Venelli
Scalar multiplication on Weierstraß elliptic curves from co-*Z* arithmetic
J. Cryptographic Engineering **1**(2):161-176, 2011

Isomorphisms of Elliptic Curves

Theorem (Char $\mathbb{K} \neq 2, 3$)

Any two elliptic curves given the Weierstraß equations

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\,, \text{ and}$$
$$E' : y^2 + a'_1xy + a'_3y = x^3 + a'_2x^2 + a'_4x + a'_6$$

are isomorphic over $\mathbb{K}$ if and only if there exist $u, r, s, t \in \mathbb{K}$, $u \neq 0$, such that the change of variables $(x, y) \leftarrow (u^2x + r, u^3y + u^2sx + t)$ transforms E into E', and where

$$\begin{cases} ua'_1 = a_1 + 2s \\ u^2a'_2 = a_2 - sa_1 + 3r - s^2 \\ u^3a'_3 = a_3 + ra_1 + 2t \\ u^4a'_4 = a_4 - sa_3 + 2ra_2 - (t + rs)a_1 + 3r^2 - 2st \\ u^6a'_6 = a_6 + ra_4 + r^2a_2 + r^3 - ta_3 - t^2 - rta_1 \end{cases}$$

Meloni's Technique Revisited (1/2)

- For any $u \neq 0$, elliptic curve

$$E_1 : y^2 = x^3 + ax + b$$

is $\mathbb{K}$-isomorphic to

$$E_u : y^2 = x^3 + au^4x + bu^6$$

- Jacobian coordinates: $x = X/Z^2$ and $y = Y/Z^3$

$$E_1 : Y^2 = X^3 + aXZ^4 + bZ^6$$

Observation

- A finite point $\boldsymbol{P} = (x_1, y_1) \in E_1$ is represented as $(X_1 : Y_1 : Z_1)$ with $X_1 = x_1{Z_1}^2$ and $Y_1 = y_1{Z_1}^3$, for any $Z_1 \in \mathbb{K}^*$
- Point (X_1, Y_1) can be seen as a point on isomorphic elliptic curve E_{Z_1}

technicolor

Meloni's Technique Revisited (2/2)

Meloni On a short Weierstraß curve E_1, two finite points $\boldsymbol{P} = (X_1 : Y_1 : Z)$ and $\boldsymbol{Q} = (X_2 : Y_2 : Z)$ given in Jacobian coordinates and sharing the same Z-coordinate can be added faster to get $\boldsymbol{R} = \boldsymbol{P} + \boldsymbol{Q} = (X_3 : Y_3 : Z_3) \in E_1$

New interpretation Two points (X_1, Y_1) and (X_2, Y_2) given in affine coordinates on a same isomorphic curve E_1 (i.e., on E_Z with $Z = 1$) can be added faster to get

$$\tilde{\boldsymbol{R}} := \Psi_\varphi(\boldsymbol{P} + \boldsymbol{Q})$$

where $\Psi_\varphi : E_1 \xrightarrow{\sim} E_\varphi, (x, y) \mapsto (\varphi^2 x, \varphi^3 y)$

technicolor

Application: Point Addition

- Let $\boldsymbol{P} = (x_1, y_1)$ and $\boldsymbol{Q} = (x_2, y_2) \in E_1 \setminus \{\boldsymbol{O}\}$ with $\boldsymbol{P} \neq \pm\boldsymbol{Q}$

Reminder: if $x_1 \neq x_2$ then

$$(x_1, y_1) + (x_2, y_2) = (x_3, y_3) = (\lambda^2 - x_1 - x_2, (x_1 - x_3)\lambda - y_1)$$

where $\lambda = \frac{y_1 - y_2}{x_1 - x_2}$

- Define $\varphi = x_1 - x_2$. Then $\tilde{\boldsymbol{R}} := \Psi_\varphi(\boldsymbol{P} + \boldsymbol{Q}) = (\varphi^2 x_3, \varphi^3 y_3) \in E_\varphi$ with

$$\begin{cases} \varphi^2 x_3 = (y_1 - y_2)^2 - \varphi^2 x_1 - \varphi^2 x_2 \\ \varphi^3 y_3 = (\varphi^2 x_1 - \varphi^2 x_3)(y_1 - y_2) - \varphi^3 y_1 \end{cases}$$

  - Cost of iADD: $4M + 2S$
  - Cost of iADDU: $4M + 2S$
  - Cost of iADDC: $5M + 3S$

technicolor

Application: Point Doubling

- Let $\boldsymbol{P} = (x_1, y_1) \in E_1 \setminus \{\boldsymbol{O}\}$ with $\boldsymbol{P} \neq -\boldsymbol{P}$

Reminder: if $y_1 \neq 0$ then

$$2(x_1, y_1) = (x_3, y_3) = (\lambda^2 - 2x_1, (x_1 - x_3)\lambda - y_1)$$

where $\lambda = \frac{3x_1^2 + a}{2y_1}$

- Define $\varphi = 2y_1$. Then $\tilde{\boldsymbol{R}} := \Psi_\varphi(2\boldsymbol{P}) = (\varphi^2 x_3, \varphi^3 y_3) \in E_\varphi$ with

$$\begin{cases} \varphi^2 x_3 = (3x_1^2 + a)^2 - 2\varphi^2 x_1 \\ \varphi^3 y_3 = (\varphi^2 x_1 - \varphi^2 x_3)(3x_1^2 + a) - \varphi^3 y_1 \end{cases}$$

  - Cost of iDBL: $1M + 5S$
  - Cost of iDBLU: $1M + 5S$

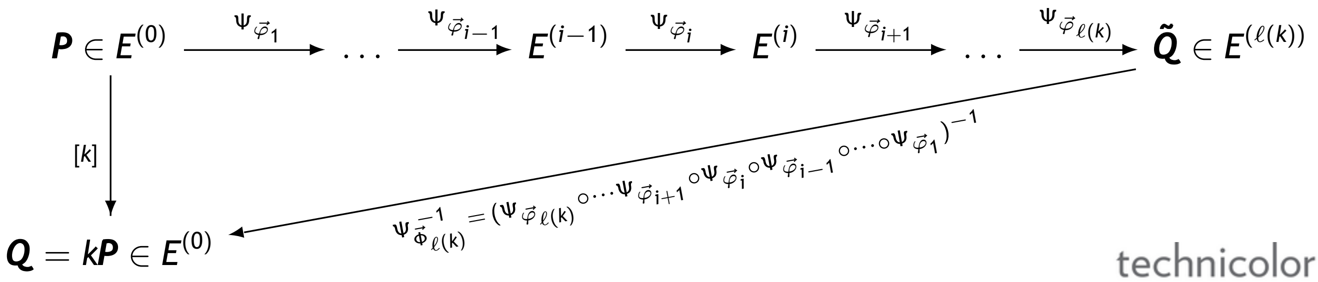
technicolor

Inversion-Free Arithmetic Through Isomorphisms

- Addition chain for k when computing $\boldsymbol{Q} = k\boldsymbol{P}$
  - $a_0 = 1, a_1, \ldots, a_\ell = k$ such that $\forall i \geqslant 1$, $\exists u, v$ with $1 \leqslant u, v < i$ and $a_i = a_u + a_v$
- Define

$$\begin{cases} E^{(0)} = E_{\mathbb{1}} & \text{original elliptic curve} \\ E^{(i)} = E_{\vec{\Phi}_i} & \text{current elliptic curve at Step } i \\ E^{(\ell(k))} = E_{\vec{\Phi}_{\ell(k)}} & \text{final elliptic curve} \end{cases}$$

- Then $\tilde{\boldsymbol{Q}} := k\big((\Psi_{\vec{\varphi}_{\ell(k)}} \circ \cdots \circ \Psi_{\vec{\varphi}_i} \circ \cdots \circ \Psi_{\vec{\varphi}_1})\boldsymbol{P}\big) \in E^{(\ell(k))}$



Composition of Isomorphisms (1/2)

- $\tilde{\boldsymbol{Q}} = k\big(\Psi_{\vec{\Phi}_{\ell(k)}}(\boldsymbol{P})\big) = k\big((\Psi_{\vec{\varphi}_{\ell(k)}} \circ \cdots \circ \Psi_{\vec{\varphi}_i} \circ \cdots \circ \Psi_{\vec{\varphi}_1})\boldsymbol{P}\big)$
  $= \Psi_{\vec{\Phi}_{\ell(k)}}(k\boldsymbol{P}) \implies \boldsymbol{Q} = \Psi^{-1}_{\vec{\Phi}_{\ell(k)}}(\tilde{\boldsymbol{Q}})$
- $\Psi_{\vec{\Phi}_{\ell(k)}}$ is obtained iteratively

$$\Psi_{\vec{\Phi}_i} = \Psi_{\vec{\varphi}_i} \circ \Psi_{\vec{\Phi}_{i-1}}$$

  with $\Psi_{\vec{\Phi}_0} = \mathrm{Id}$
- ...or slightly abusing the notation— since $\vec{\Phi}_i = \mathrm{desc}(\Psi_{\vec{\Phi}_i})$

$$\vec{\Phi}_i = \vec{\varphi}_i \circ \vec{\Phi}_{i-1}$$

  with $\vec{\Phi}_0 = \mathrm{desc}(\mathrm{Id}) := \mathbb{1}$

Composition of Isomorphisms (2/2)

- General Weierstraß elliptic curves ($char \neq 2, 3$)

$$\Psi_{\vec{\Phi}_{i-1}} : E^{(0)} \xrightarrow{\sim} E^{(i-1)},$$
$$(x, y) \longmapsto ({U_{i-1}}^2 x + R_{i-1}, {U_{i-1}}^3 y + {U_{i-1}}^2 S_{i-1} x + T_{i-1})$$

$$\Psi_{\vec{\varphi}_i} : E^{(i-1)} \xrightarrow{\sim} E^{(i)}, (x, y) \longmapsto ({u_i}^2 x + r_i, {u_i}^3 y + {u_i}^2 s_i x + t_i)$$

where $\vec{\Phi}_{i-1} = (U_{i-1}, R_{i-1}, S_{i-1}, T_{i-1})$ and $\vec{\varphi}_i = (u_i, r_i, s_i, t_i)$

- Operation $\vec{\Phi}_i = \vec{\varphi}_i \circ \vec{\Phi}_{i-1}$ translates into $(U_i, R_i, S_i, T_i) = (u_i, r_i, s_i, t_i) \circ (U_{i-1}, R_{i-1}, S_{i-1}, T_{i-1})$ with

$$\begin{cases} U_i = U_{i-1} u_i \\ R_i = {u_i}^2 R_{i-1} + r_i \\ S_i = u_i S_{i-1} + s_i \\ T_i = {u_i}^3 T_{i-1} + {u_i}^2 s_i R_{i-1} + t_i \end{cases}$$

for $i \geqslant 1$, and $(U_0, R_0, S_0, T_0) := \mathbb{1} = (1, 0, 0, 0)$

technicolor

New Operations

- Given two elliptic curves $E_{\vec{\Phi}}$ and $E_{\vec{\Phi}'}$ being isomorphic to $E_{\mathbb{1}}$, if

$$\Psi_{\vec{\varphi}} : E_{\vec{\Phi}} \xrightarrow{\sim} E_{\vec{\Phi}'}$$

denotes the isomorphism between $E_{\vec{\Phi}}$ and $E_{\vec{\Phi}'}$, we define

$$\begin{cases} \text{iADD}_{\vec{\Phi}} : & (P_1, P_2) \mapsto \left(\Psi_{\vec{\varphi}}(P_1 + P_2), \vec{\varphi}\right) \\ \text{iADDU}_{\vec{\Phi}} : & (P_1, P_2) \mapsto \left(\Psi_{\vec{\varphi}}(P_1 + P_2), \Psi_{\vec{\varphi}}(P_1), \vec{\varphi}\right) \\ \text{iADDC}_{\vec{\Phi}} : & (P_1, P_2) \mapsto \left(\Psi_{\vec{\varphi}}(P_1 + P_2), \Psi_{\vec{\varphi}}(P_1 - P_2), \vec{\varphi}\right) \\ \text{iDBL}_{\vec{\Phi}} : & P_1 \mapsto \left(\Psi_{\vec{\varphi}}(2P_1), \vec{\varphi}\right) \\ \text{iDBLU}_{\vec{\Phi}} : & P_1 \mapsto \left(\Psi_{\vec{\varphi}}(2P_1), \Psi_{\vec{\varphi}}(P_1), \vec{\varphi}\right) \end{cases}$$

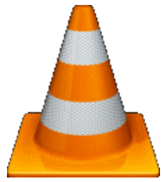
technicolor

Some Results

- Short Weierstraß model

Algorithm	Cost/bit
Montgomery ladder	$\underline{8M + 6S}$
Double-and-add	$7M + 8.5S$
Double-and-add + NAF	$\underline{6M + 6.33S}$

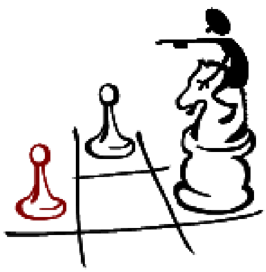
- Twisted Edwards model
  - unified iADD: $10M + 1S$
  - unified iADDU: $12M + 1S$
  - unified iADDC: $13M + 1S$



technicolor

Summary

- Re-casting and generalization of Meloni's technique using elliptic curve isomorphisms
- New strategies for evaluating scalar multiplications on elliptic curves
  - without inversion
  - applicable to any scalar multiplication algorithm
  - applicable to any elliptic curve model
  - (nicely combine with certain countermeasures)



technicolor