

# Improved Algorithms for Efficient Arithmetic on Elliptic Curves using Fast Endomorphisms

Mathieu Ciet, Tanja Lange, Francesco Sica and Jean-Jacques Quisquater

# Outline of Talk

- Introduction to elliptic curves

# Outline of Talk

- Introduction to elliptic curves
- Methods to compute  $kP$  on elliptic curves

# Outline of Talk

- Introduction to elliptic curves
- Methods to compute  $kP$  on elliptic curves
- Curves with small endomorphism ring: Koblitz and GLV curves

# Outline of Talk

- Introduction to elliptic curves
- Methods to compute  $kP$  on elliptic curves
- Curves with small endomorphism ring: Koblitz and GLV curves
- The GLV and Frobenius methods

# Outline of Talk

- Introduction to elliptic curves
- Methods to compute  $kP$  on elliptic curves
- Curves with small endomorphism ring: Koblitz and GLV curves
- The GLV and Frobenius methods
- How can we combine the two? Base  $\phi$  decompositions

# Outline of Talk

- Introduction to elliptic curves
- Methods to compute  $kP$  on elliptic curves
- Curves with small endomorphism ring: Koblitz and GLV curves
- The GLV and Frobenius methods
- How can we combine the two? Base  $\phi$  decompositions
- Joint exponentiation: the JSF expansion

# Outline of Talk

- Introduction to elliptic curves
- Methods to compute  $kP$  on elliptic curves
- Curves with small endomorphism ring: Koblitz and GLV curves
- The GLV and Frobenius methods
- How can we combine the two? Base  $\phi$  decompositions
- Joint exponentiation: the JSF expansion
- Using JSF with endomorphism speedup: the  $\phi$ -JSF expansion

# Outline of Talk

- Introduction to elliptic curves
- Methods to compute  $kP$  on elliptic curves
- Curves with small endomorphism ring: Koblitz and GLV curves
- The GLV and Frobenius methods
- How can we combine the two? Base  $\phi$  decompositions
- Joint exponentiation: the JSF expansion
- Using JSF with endomorphism speedup: the  $\phi$ -JSF expansion
- Conclusion

# Elliptic Curve Definition

$E/\mathbb{F}_q$  is given by an equation of a plane curve:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \quad \text{with } a_i \in \mathbb{F}_q$$

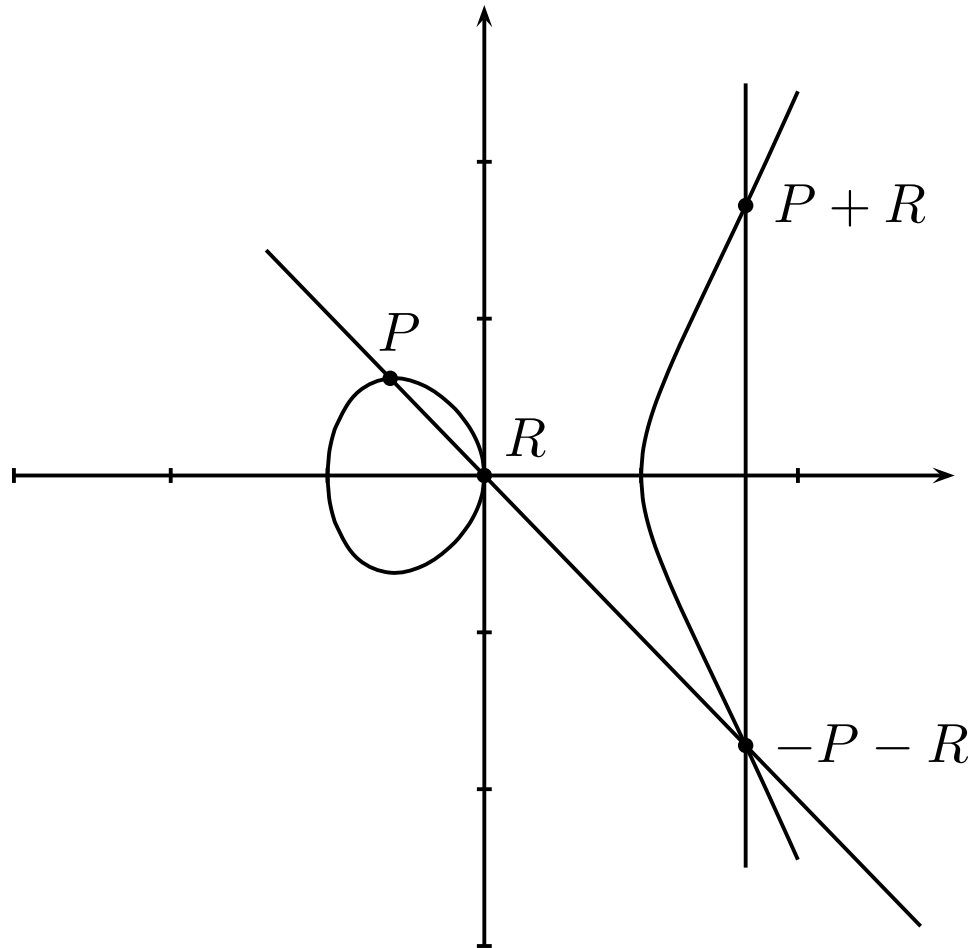
The set of solutions  $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$  together with the point “at infinity”  $\mathcal{O}$  is denoted by  $E(\mathbb{F}_q)$

$$q = 2^l \quad \text{binary curve} \quad y^2 + xy = x^3 + a_2x^2 + a_6$$

$$q = p \geq 5 \quad \text{prime curve} \quad y^2 = x^3 + a_4x + a_6$$

# Group Law in $E(\mathbb{R})$

$$y^2 = x^3 - x$$



# Computation of $Q = kP$ : Classical

Double and add (analogue of square and multiply): let

$$k = \langle k_t k_{t-1} \dots k_0 \rangle = k_t 2^t + k_{t-1} 2^{t-1} + \dots + k_0.$$

1.  $Q = \mathcal{O}$
2. for  $i = t$  down to 0
  - (a)  $Q = 2Q$
  - (b) if  $k_i \neq 0$  then  $Q = Q + P$
3. return  $Q$

Does not exploit the elliptic curve structure!

Average Hamming weight of  $\langle k_t k_{t-1} \dots k_0 \rangle = t/2$

# Computation of $Q = kP$ : NAF

NAF: based on the fact that computing  $-P$  knowing  $P$  is very simple.

$$k = \langle k_m k_{m-1} \dots k_0 \rangle = k_m 2^m + k_{m-1} 2^{m-1} + \dots + k_0,$$

with  $k_i = 0, \pm 1$  and  $k_i k_{i+1} = 0$

Advantage: average density 33% (compared to 50% for Double& Add) leading to 17% saving in additions. Also,  $m \leq t+1$ .

# NAF Algorithm

Let  $k = \langle k_m k_{m-1} \dots k_0 \rangle$ .

1.  $Q = \mathcal{O}$
2. for  $i = m$  down to 0
  - (a)  $Q = 2Q$
  - (b) if  $k_i \neq 0$  then  $Q = Q + k_i P$
3. return  $Q$

Average Hamming weight of  $\langle k_m k_{m-1} \dots k_0 \rangle$  is  $m/3$ .

# Endomorphisms Examples

Homomorphisms from  $E \rightarrow E$  expressed as rational functions of coordinates. For instance  $k: E \rightarrow E, P \mapsto kP$

- (Koblitz) When  $E$  has equation  $y^2 + xy = x^3 + ax^2 + 1$  (and  $a = 0, 1$ ) over  $\mathbb{F}_{2^\ell}$ . Note that  $\phi^2 + (-1)^a\phi + 2 = 0$ .

$$\phi = \tau: E \rightarrow E$$

$$(x, y) \mapsto (x^2, y^2)$$

- (GLV) When  $E$  has equation  $y^2 = x^3 + ax$  over  $\mathbb{F}_p$  with  $p \equiv 1 \pmod{4}$ . Note that  $\phi^2 + 1 = 0$ .

$$\phi: E \rightarrow E$$

$$(x, y) \mapsto (-x, \sqrt{-1}y)$$

# Endomorphisms Examples (cont.)

- (GLV) When  $E$  has equation  $y^2 = x^3 - 3x^2/4 - 2x - 1$  over  $\mathbb{F}_p$  with  $p \equiv 1, 2$  or  $4 \pmod{7}$ . Note that  $\phi^2 - \phi + 2 = 0$ . Denote  $\xi = (1 + \sqrt{-7})/2$  and  $a = (\xi - 3)/4$ .

$$\phi: E \rightarrow E$$

$$(x, y) \mapsto \left( \frac{x^2 - \xi}{\xi^2(x - a)}, \frac{y(x^2 - 2ax + \xi)}{\xi^3(x - a)^2} \right)$$

In last case, computing  $\phi(P)$  can take less operations than  $2P$  (depending on coordinate system). In previous slide, computing  $\phi(P)$  is very fast.

# Computation of $Q = kP$ : $\tau$ -NAF

Most effective methods use fast nontrivial endomorphism  $\phi$   
Char 2 (Meier-Staffelbach, Solinas): where  $E/\mathbb{F}_{2^\ell}$  is a  
Koblitz curve (defined over  $\mathbb{F}_2$ ) and  $\phi = \tau$  is the 2-Frobenius  
 $(x, y) \mapsto (x^2, y^2)$ . Use a complex NAF

$$kP = k_0P + k_1\tau(P) + \cdots + k_l\tau^l(P)$$

$$\text{with } k_i = 0, \pm 1, \quad k_i k_{i+1} = 0$$

Again in practice  $l \leq \ell - 1$ .

# $\tau$ -NAF Algorithm

Let  $k = \langle k_l k_{l-1} \dots k_0 \rangle$ .

1.  $Q = \mathcal{O}$
2. for  $i = l$  down to 0
  - (a)  $Q = \tau(Q)$
  - (b) if  $k_i \neq 0$  then  $Q = Q + k_i P$
3. return  $Q$

Average Hamming weight of  $\langle k_l k_{l-1} \dots k_0 \rangle = l/3$ .

# Computation of $Q = kP$ : GLV

Char  $p \geq 5$ : Gallant-Lambert-Vanstone method where  $\phi \in \text{End}_{\mathbb{F}_p}(E)$  is different from the  $p$ -Frobenius

$$kP = k^{(0)}P + k^{(1)}\phi(P), \quad \max(|k^{(0)}|, |k^{(1)}|) = O(\sqrt{n})$$

Key point:  $k^{(i)}$  have half length with respect to  $k \in [1, n]$ , so use of parallel computation is effective (alternatively use Joint Sparse Form)

# Base $\phi$ Expansions

- Valid in characteristic 2 and  $p$ . Let  $\phi^2 + r\phi + s = 0$ . We can decompose the scalar multiplication  $kP$  for any  $k \in [1, n]$  as

$$kP = k_0P + k_1\phi(P) + \dots + k_v\phi^v(P) \quad \text{with } k_i \in \{-\lfloor s/2 \rfloor, \dots, \lfloor s/2 \rfloor\}$$

- Effective only when  $s$  is small, in which case  $\phi$  can also be computed quickly.
- Using GLV, we can choose
$$v \leq \lceil 2 \log_s 2\sqrt{1 + |r| + s} + \log_s n \rceil + 3 \leq t + 9.$$
- When  $s = 2$  (very small), can compute  $\phi$ -NAF expansion. In general, need large coefficient set ( $\varphi(s^2) + 1$  coefficients).

# What happens if $s = 1$ ?

Previous analysis fails theoretically and in practice one must have exponentially long decompositions ( $v \approx n \approx 2^t$ ).

# Performance using $\phi$ -NAF

Average number of operations to perform  $kP$  when  $k = \langle k_0 k_1 \dots k_t \rangle$  in binary or equivalent ( $s = 2$ ).

NAF	$\phi$ -NAF	GLV with joint-NAF	GLV with JSF
$t$ doubles, $t/3$ adds	$t$ $\phi$ 's, $t/3$ adds	$t/2$ doubles, $t/3$ adds	$t/2$ doubles, $t/4$ adds
$\emptyset$	$\emptyset$	$\phi(P)$	$\phi(P),$ $\phi(P) \pm P$

Advantage when  $\phi$  is cheaper than a doubling. However Straus-Shamir Trick (joint double and add) produces significant speedup.

Can we combine SST (especially JSF) with fast computation of  $\phi$ ?

# Answer

# Answer

YES!

# The Straus-Shamir Trick: Example

The fast Straus-Shamir method to compute  $19P + 14Q$  from a joint NAF works as follows

$19 =$	$1$	$0$	$1$	$0$	$-1$
$14 =$	$1$	$0$	$0$	$-1$	$0$
$\times 2$	$2P + 2Q$		$4P + 4Q$	$10P + 8Q$	$20P + 14Q$
$P$			$5P + 4Q$	$19P + 14Q$	
$Q$			$10P + 7Q$		
$P + Q$	$P + Q$				
$P - Q$					

# $k^{(0)}P + k^{(1)}Q$ with Straus-Shamir

Let

$$k^{(0)} = \langle k_{0,u} k_{0,u-1} \dots k_{0,0} \rangle$$

$$k^{(1)} = \langle k_{1,u} k_{1,u-1} \dots k_{1,0} \rangle$$

1.  $R = k_{0,u}P + k_{1,u}Q$
2. for  $i = u - 1$  down to 0
  - (a)  $R = 2R$
  - (b) if  $(k_{0,i}, k_{1,i}) \neq (0, 0)$  then  $R = R + k_{0,i}P + k_{1,i}Q$   
*// one addition with precomputed points*
3. return  $R$

Using NAF expansions: on average,  $u$  doublings and  $2u/3$  (resp.  $5u/9$ ) additions precomputing  $P, Q$  (resp.  $P, Q, P \pm Q$ ).

# Solinas' Joint Sparse Form

When computing  $k^{(0)}P + k^{(1)}Q$  allowing 2 precomputed values besides  $P$  and  $Q$ , i.e.  $P \pm Q$ , want to use minimal number of doublings and additions  $\rightarrow$  minimise Joint Hamming weight (number of nonzero columns). Consider signed binary representations of

$$k^{(0)} = \langle k_{0,u} k_{0,u-1} \dots k_{0,0} \rangle$$

$$k^{(1)} = \langle k_{1,u} k_{1,u-1} \dots k_{1,0} \rangle$$

The double representation is a JSF if

**(JSF 1)** Of any three consecutive columns at least one is a zero column.

**(JSF 2)** It is never the case that  $k_{i,j+1}k_{i,j} = -1$ .

**(JSF 3)** If  $k_{i,j+1}k_{i,j} \neq 0$  then  $k_{1-i,j+1} = \pm 1$  and  $k_{1-i,j} = 0$ .

# Properties of Joint Sparse Form

- JSF of any two integers exists and is unique and can be efficiently computed by Solinas' algorithm.
- JSF has minimal joint Hamming weight among all joint signed binary expansions.
- Length of JSF is at most one bit longer than ordinary binary expansion of  $\max(k^{(0)}, k^{(1)})$ , i.e.  $u \leq t + 1$ .
- If  $k^{(0)}$  and  $k^{(1)}$  have maximal length  $u$ , then the joint double and add (SST) algorithm computes  $k^{(0)}P + k^{(1)}Q$  from the JSF with an average of  $u$  doublings and  $u/2$  additions of either  $\pm P$ ,  $\pm Q$ ,  $\pm(P + Q)$  or  $\pm(P - Q)$  (Joint Hamming density is  $1/2$ ).

# Definition of $\phi$ -Joint Sparse Form

**Idea:** Replace doublings by  $\phi$ 's to perform  $k^{(0)}P + k^{(1)}Q$  with joint  $\phi$  and add algorithm! As before, can use 2  $\phi$ -NAF's.

**Goal:** to get optimal expansion for maximum performance  
→ invent a  $\phi$ -JSF!

Focus on case when  $\phi^2 - \epsilon\phi + 2 = 0$  and  $\epsilon = \pm 1$  (e.g. Koblitz curves). A double signed  $\phi$ -expansion is a  $\phi$ -JSF if

**( $\phi$ -JSF 1)** Among three consecutive columns at least one is a double zero.

**( $\phi$ -JSF 2)** It is never the case that  $k_{i,j+1} k_{i,j} = \epsilon$ .

**( $\phi$ -JSF 3)** If  $k_{i,j+1} k_{i,j} \neq 0$  then  $k_{1-i,j+1} = \pm 1$  and  $k_{1-i,j} = 0$ .

# Properties of $\phi$ -Joint Sparse Form

- $\phi$ -JSF of any two integers exists and is unique. It can be constructed by our Algorithm 1, which is the analogue of Solinas' algorithm.
- $\phi$ -JSF has almost minimal joint Hamming weight among all joint signed  $\phi$ -expansions.
- Length of  $\phi$ -JSF is at most 3 bits longer than length of unsigned  $\phi$  expansion, i.e.  $u \leq v + 3$ .
- Joint Hamming density is  $1/2$ .

*Remark:* the  $\phi$ -JSF of  $(\langle 1, 0, -1 \rangle, \langle 0, \epsilon, 0 \rangle)$  is  $(\langle -\epsilon, 0, -\epsilon, 0, -\epsilon, 1 \rangle, \langle 0, 0, 0, 0, \epsilon, 0 \rangle)$ .

# Performance of $\phi$ -Joint Sparse Form

Average number of operations to perform  $k^{(0)}P + k^{(1)}Q$  or  $kP$  when  $k = \langle k_0 k_1 \dots k_t \rangle$  or  $k^{(0)} = \langle k_{0,u} k_{0,u-1} \dots k_{0,0} \rangle$  and  $k^{(1)} = \langle k_{1,u} k_{1,u-1} \dots k_{1,0} \rangle$  in binary or equivalent.

JSF	$\phi$ -JSF	GLV with JSF	Lim-Lee with $\phi$ -JSF
$u$ doubles, $u/2$ adds	$u$ $\phi$ 's, $u/2$ adds	$t/2$ doubles, $t/4$ adds	$t/2$ $\phi$ 's, $t/4$ adds
$P, Q, P \pm Q$	$P, Q, P \pm Q$	$P, \phi(P),$ $P \pm \phi(P)$	$P, \phi^{\lceil \frac{(v+1)}{2} \rceil}(P),$ $P \pm \phi^{\lceil \frac{(v+1)}{2} \rceil}(P)$

# Conclusion

- Extension of  $\tau$ -expansions (Meier-Staffelbach ...) to other endomorphisms over large characteristics to compute  $kP$ : effective when no precomputations allowed.
- When precomputing 3 points, Lim-Lee method with  $\phi$ -JSF expansion leads to best results (large characteristic, see paper) if  $\phi$  is faster than doubling.
- For signature verification,  $\phi$ -JSF is very effective, most dramatically when  $\phi = \tau$ , e.g. on Koblitz curves, when applying  $\tau$  is very low-cost.