Preventing Differential Analysis in GLV Elliptic Curve Scalar Multiplication

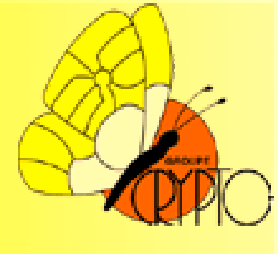
Mathieu Ciet*, Jean-Jacques Quisquater and Francesco Sica,

UCL Crypto Group

http://www.dice.ucl.ac.be/crypto/

Outline of Talk

- Introduction to elliptic curves
- Introduction to DPA
- The GLV method
- Randomising the GLV method
- Entropy analysis
- Performance analysis
- Affine generalisations
- Concluding remarks



What is an elliptic curve?

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

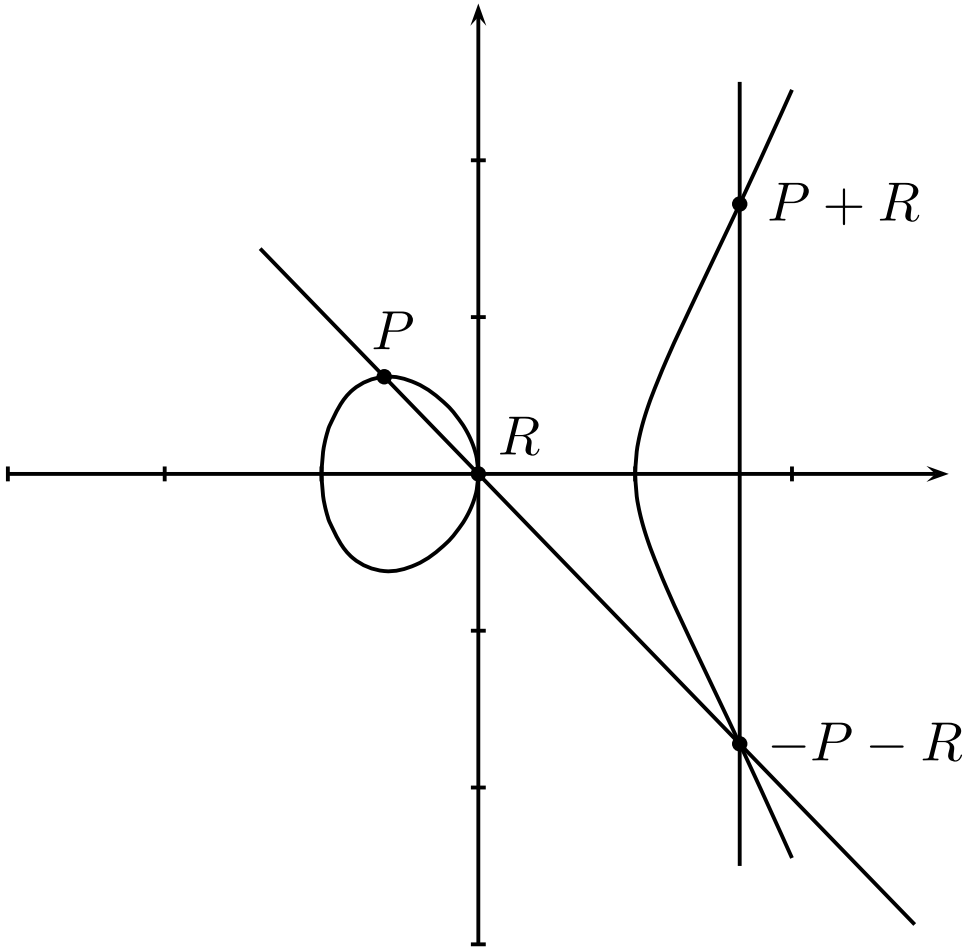
$$y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6, \quad \text{with } a_i \in \mathbb{F}_q$$

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 $E(\mathbb{F}_q)$

$q = 2^l$	binary curve	$y^2 + xy = x^3 + a_2x^2 + a_6$
$q = p \geq 5$	prime curve	$y^2 = x^3 + a_4x + a_6$

Group Law in $E(\mathbb{F}_q)$

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



Differential Power Analysis and Countermeasures

DPA consists of a statistical analysis of power consumption.

General idea: To be immunised against DPA, add a part of randomness.

There are lot of countermeasures, we consider two types

- Random base point representation

$$P \rightarrow \widetilde{P} \text{ such that } kP = k\widetilde{P}$$

- Random decomposition of the secret key.

$$k \rightarrow \widetilde{k} \text{ such that } kP = \widetilde{k}P$$

Gallant-Lambert-Vanstone Method

Let $E(\mathbb{F}_q)$ be an elliptic curve defined over $\mathbb{F}_q$, and P be a point of E of order n.
General idea: use a fast endomorphism Φ

- Let λ: $\Phi(P) = \lambda P$
- Decompose k as

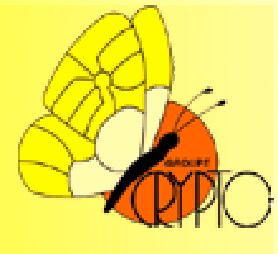
$$k = k_1 + \lambda k_2 \pmod{n} \text{ with } k_i = O(\sqrt{n})$$

- Compute $kP = k_1 P + k_2 \Phi(P)$ using elliptic Shamir algorithm

Key point: compute with half length of $\log_2 n$ instead of full length.

GLV Method Description

- $E/\mathbb{F}_p$, such that $\#E(\mathbb{F}_p) = hn$, with $h \leq 4$ and P a point of the curve of prime order n
- Φ a nontrivial endomorphism defined over $\mathbb{F}_p$, $X^2 + rX + s$ its characteristic polynomial, $\Delta = r^2 - 4s < 0$
- $\Phi(P) = \lambda P$ for some $\lambda \in [0, n-1]$ by Hasse's bound (λ is a root of $X^2 + rX + s$ modulo n)
- Let $$\begin{aligned} f : \mathbb{Z} \times \mathbb{Z} &\rightarrow \mathbb{Z}/n \\ (i, j) &\mapsto i + \lambda j \pmod{n} \end{aligned}$$



GLV Method Description (cont.)

- Let $v_1, v_2 \in \ker f$ two linearly independent vectors with $\max(|v_1|, |v_2|) = O(\sqrt{n})$
- Write $(k, 0) = \beta_1 v_1 + \beta_2 v_2$ with $\beta_i \in \mathbb{Q}$
- Define b_i as the nearest integer to β_i and let $v = b_1 v_1 + b_2 v_2$
- $v \in \ker f$ and $(k_1, k_2) = u \stackrel{\text{def}}{=} (k, 0) - v$ satisfies $|u| = \max(|k_1|, |k_2|) = O(\sqrt{n})$ and
- we get $k = k_1 + k_2 \lambda \pmod{n}$ or equivalently $kP = k_1 P + k_2 \Phi(P)$

Curve Examples

Specifications	Example 1	Example 2
$\mathbb{F}_p$ with $p > 3$ and	$p \equiv 1 \pmod 4$	$p \equiv 1 \pmod 3$
Curve equation	$y^2 = x^3 + ax$	$y^2 = x^3 + b$
$\Phi(x, y) =$	$(-x, \sqrt[4]{1}y)$	$(\sqrt[3]{1}x, y)$
Characteristic polynomial of Φ	$X^2 + 1$	$X^2 + X + 1$
$\mathbb{Z}[\Phi]$ maximal?	✓	✓
$\mathbb{Z}[\Phi]$ principal?	✓	✓

Curve Examples

Specifications	Example 3	Example 4
$\mathbb{F}_p$ with $p > 3$ and	$\left(\frac{-7}{p}\right) = 1$	$\left(\frac{-2}{p}\right) = 1$
Curve equation	$y^2 = x^3 - \frac{3}{4}x^2 - 2x - 1$	$y^2 = 4x^3 - 30x - 28$
$\Phi(x, y) =$	$\left(\frac{x^2-\xi}{\xi^2(x-a)}, \frac{y(x^2-2ax+\xi)}{\xi^3(x-a)^2}\right)$	$\star$
Characteristic polynomial of Φ	$X^2 - X + 2$	$X^2 + 2$
$\mathbb{Z}[\Phi]$ maximal?	✓	✓
$\mathbb{Z}[\Phi]$ principal?	✓	✓

Randomising the GLV Decomposition

Let

$$A = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \quad 0 \leq \alpha, \beta, \gamma, \delta \leq R$$

and $\alpha\delta - \beta\gamma \neq 0$. Define

$$\Phi_0 = \alpha + \beta\Phi \qquad \Phi_0(P) = \lambda_0 P$$
$$\Phi_1 = \gamma + \delta\Phi \qquad \Phi_1(P) = \lambda_1 P$$

Idea: Replace GLV map f by

$$\begin{aligned} f' \colon \mathbb{Z} \times \mathbb{Z} &\rightarrow \mathbb{Z}/n \\ (i, j) &\mapsto i\lambda_0 + j\lambda_1 \pmod{n} \ . \end{aligned}$$

Description of Randomised GLV Method

We want to write $kP = k'_1\Phi_0(P) + k'_2\Phi_1(P)$ for some $k'_1, k'_2 = O(\sqrt{n})$. Use same strategy as in the original GLV method, replacing f by f'.

1. Compute $\lambda_0^{-1} \in [0, n-1]$ such that $\lambda_0\lambda_0^{-1} \equiv 1 \pmod{n}$ with an application of the extended Euclidean algorithm.
2. Compute $\lambda' = \lambda_1\lambda_0^{-1} \pmod{n}$.
3. Find v'_1 and v'_2 in $\ker f'$ by applying the original GLV algorithm, replacing λ by λ'.

Description of Randomised GLV Method (cont.)

4. Express $(k\lambda_0^{-1}, 0) = \beta_1 v_1' + \beta_2 v_2'$, where $\beta_i \in \mathbb{Q}$.
5. Let $b_i = \lceil \beta_i \rfloor$ and $v' = b_1 v_1' + b_2 v_2'$.
6. Compute $(k_1', k_2') = (k\lambda_0^{-1}, 0) - v'$.
7. Compute kP as $k_1'\Phi_0(P) + k_2'\Phi_1(P)$ with the elliptic Straus-Shamir (Solinas) algorithm.

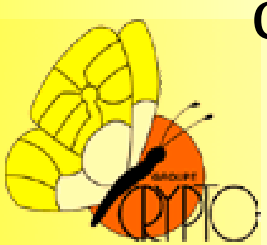
How much do we shuffle?

We show the following result

If $R < \sqrt{n}/\sqrt{1 + |r| + s}$ *then each matrix* A *gives a different consumption pattern. Hence for a choice of* R *there are about* R^4 *different (in terms of measured power) ways to compute* kP *with the randomised GLV method.*

Proof: Let $$A = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \quad B = \begin{pmatrix} \epsilon & \zeta \\ \eta & \theta \end{pmatrix}$$

two different non-degenerate matrices with integral coefficients in $[0, R]$, producing Φ_0, Φ_1 (resp. Ψ_0, Ψ_1).



Proof of No Collision

Same measured consumption means

$$kP = k'_1\Phi_0(P) + k'_2\Phi_1(P) = \tilde{k}'_1\Psi_0(P) + \tilde{k}'_2\Psi_1(P)$$

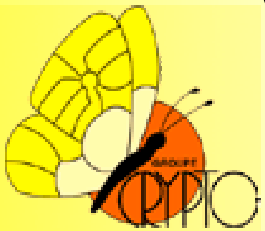
$$k'_i = \tilde{k}'_i \quad \text{and}$$

$$\Phi_i(P) = \Psi_i(P) \quad i = 1, 2$$

The last condition is equivalent to

$$\alpha + \beta\lambda + c_1 n = \epsilon + \zeta\lambda \quad \text{and} \quad \gamma + \delta\lambda + c_2 n = \eta + \theta\lambda$$

with $c_1, c_2 \in \mathbb{Z}$.



Proof of No Collision (cont.)

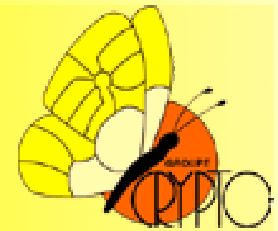
This implies

$$(\alpha - \epsilon) + (\beta - \zeta)\lambda \equiv 0 \pmod{n}$$

so that $(\alpha - \epsilon, \beta - \zeta) \in \ker f$. By a result of Sica, Ciet and Quisquater [SAC 2002] if $(x, y) \in \ker f - \{(0, 0)\}$ we have

$$\max(|x|, |y|) > \sqrt{n}/\sqrt{1 + |r| + s} > R$$

contradiction, hence $\alpha = \epsilon$, $\beta = \zeta$ and similarly $\gamma = \eta$, $\delta = \theta$. ■

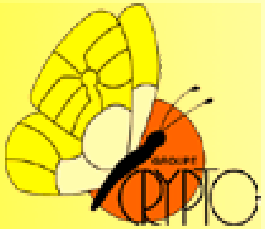


Performance Estimates

Three parts of extra computation can be distinguished:

1. computation of $\Phi_0(P) = \lambda_0 P$, $\Phi_1(P) = \lambda_1 P$
2. computation of v_1' and v_2'
3. computation of $kP = k_1'\lambda_0 P + k_2'\lambda_1 P$ with respect to the original $kP = k_1 P + k_2 \lambda P$

By the Randomised GLV Algorithm, Step 2 as well as decomposing $kP = k_1'\lambda_0 P + k_2'\lambda_1 P$ once knowing λ_0, λ_1 takes almost the same time as through the traditional GLV algorithm



Performance Estimates (cont.)

The vectors $v'_1, v'_2 \in \ker f'$ *satisfy*

$$\max(|v'_1|, |v'_2|) \leq 2R \max(|v_1|, |v_2|)$$

In particular

$$\max_k(|k'_1|, |k'_2|) \leq 2R \max_k(|k_1|, |k_2|)$$

We use this result in evaluating extra computation cost in step 3

Performance Estimates (cont.)

To compute $\Phi_0(P)$, $\Phi_1(P)$ and $k'_1\Phi_0(P) + k'_2\Phi_1(P)$ we use the elliptic Straus-Shamir method of Solinas.

It is known that to compute $aP + bQ$ its average cost for $l = \max(\log_2 |a|, \log_2 |b|)$ is l doublings and $l/2$ curve additions.

We can now estimate the performance loss

Performance Results

The cost of the Randomised GLV Algorithm with respect to a traditional unprotected GLV Algorithm is augmented by

$$\frac{600 \log_2 R}{\log_2 n}\%$$

Example: Choosing $R = 2^{10}$ get a 37.5% slowdown on a 160-bit curve, gaining 2^{40} different consumption patterns

Affine Generalisation

Idea: Use affine map $x \mapsto Ax + \rho$ instead of linear map. We illustrate the case $A = \mathrm{Id}$.

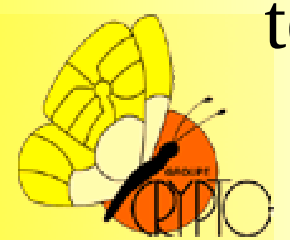
1. Randomly choose $\rho_1, \rho_2 \in [1, R]$
2. Compute v_1 and v_2 by the GLV algorithm
3. Decompose k as $(k, 0) = \beta_1 v_1 + \beta_2 v_2$
4. Let $b'_i = \lceil \beta_i \rfloor - \rho_i$ for $i \in \{1, 2\}$
5. Let $(k'_1, k'_2) = u' = (k, 0) - v'$ where $v' = b'_1 v_1 + b'_2 v_2$
6. Compute $kP = k'_1 P + k'_2 \Phi(P)$, with the elliptic Straus-Shamir (Solinas) algorithm

Results on Affine Version

The Affine Randomised GLV algorithm performs 33.3% better than its pure linear counterpart for the same collision probability. However, the points P and $\Phi(P)$ appear in the computation of kP.

The cost of the Affine Randomised GLV Algorithm with respect to a traditional unprotected GLV Algorithm is augmented by $200 \log_2 R / \log_2 n \%$

Example: Choosing $R = 2^{20}$ get a 25% slowdown on a 160-bit curve, gaining 2^{40} different consumption patterns



Conclusion

- Introduced a new method to protect GLV algorithm against DPA by introducing different ways to compute kP:

  Two countermeasures in one:

  $(P, \Phi(P)) \rightarrow (\Phi_0(P), \Phi_1(P))$ and $k = k_1 + \lambda k_2 \rightarrow \lambda_0 k'_1 + \lambda_1 k'_2$

- Complete analysis of collision events
- Very effective: in practice, only 37% (resp. 25%) speed loss on 160-bit (resp. 240-bit) curve
- If not important to leak P, can use affine version, even faster (slowdown only 25%, resp. 16.7%)