Preventing Differential Analysis in GLV Elliptic Curve Scalar Multiplication

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

UCL Crypto Group
Place du Levant, 3. B-1348 Louvain-la-Neuve. Belgium
{ciet, jjq, sica}@dice.ucl.ac.be – http://www.dice.ucl.ac.be/crypto

Abstract. In [2], Gallant, Lambert and Vanstone proposed a very efficient algorithm to compute $Q = kP$ on elliptic curves having non-trivial efficiently computable endomorphisms. Cryptographic protocols are sensitive to implementations, indeed as shown in [6,7] information about the secret can be revealed analysing external leakage of the support, typically a smart card. Several software countermeasures have been proposed to protect the secret. However, speed computation is needed for practical use. In this paper, we propose a method to protect scalar multiplication on elliptic curves against Differential Analysis, that benefits from the speed of the Gallant, Lambert and Vanstone method. It can be viewed as a two-dimensional analogue of Coron's method [1] of randomising the exponent k. We propose two variants of this method (one linear and one affine), the second one slightly more effective, whereas the first one offers "two in one", combining point-blinding and exponent randomisation, which have hitherto been dealt separately. For instance, for at most a mere 37.5% (resp. 25%) computation speed loss on elliptic curves over fields with 160 (resp. 240) bits the computation of kP can take on 2^{40} different consumption patterns.

Keywords. *Public key cryptography, differential power analysis, elliptic curve cryptosystem, fast computation.*

1 Introduction

Since the paper of Kocher [6] on the timing attack in 1996 and the Kocher, Jaffe and Jun paper Differential Power Analysis (DPA) [7] in 1999, it is well known that non careful implementations can leak and that it is possible to recover the secret key using the information on which access is possible. For the particular case of elliptic curves, different methods has been proposed to prevent these attacks as [1,4,5,8,9]. Independently Gallant Lambert and Vanstone, proposed in [2] a new principle of computation using efficient endomorphism on certain elliptic curves. In the past, Solinas also proposed to use such endomorphisms but his method could only be applied for elliptic curve defined over binary fields, the

* Supported by the European Commission through the IST Programme under Contract IST-1999-12324, http://cryptonessie.org/.

B.S. Kaliski Jr. et al. (Eds.): CHES 2002, LNCS 2523, pp. 540–550, 2003.

endomorphism considered being the Frobenius. However, the motivation of the Gallant-Lambert-Vanstone (GLV for short) and Solinas methods are not exactly the same, in the first case "reduction" of secret multiplier is obtained and in the second new decomposition is the basis of speedup computation.

In this paper we propose a specific method that benefits from fast endomorphisms, allowing its use in the case of smart card or parallel implementation. The speedup obtained by using a particular endomorphism is not helpful in solving the discrete logarithm problem. Hence the curves on which our methods apply are not cryptographically weaker than a generic curve. The methods presented here do not immunise elliptic curve cryptosystems against simple power analysis. It is possible to combine several methods to prevent simple and differential power analysis, such as randomisation of the private multiplier or blinding the point (see [1]) or elliptic curve isomorphisms, field isomorphisms or extension fields (see [5]). Differential analysis is not only a theoretical attack but can be applied in practice to recover the secret key, analysing leakage.

This paper is organised as follows. In a first part, a countermeasure against differential analysis using randomisation is reviewed. Afterwards, the GLV method based on efficient endomorphisms to speed up computation is explained. Then our method to prevent elliptic curve cryptosystems from differential analysis is proposed. It is based on random sublattices, and after a theoretical discussion distinguishing two cases a practical application is given. At the same time considerations of extra computation time is taken into account and analysis of real prevention is discussed, by counting the number of different possible representations of the same multiplier k. Before concluding, an affine generalisation is introduced which from the implementation perspective is quite interesting because of the small modifications needed compared to the original GLV method. For all methods presented here, no extra routine are necessary to be implemented.

2 Differential Analysis and Previous Work on Randomised Endomorphisms

In [7], Kocher, Jaffe and Jun introduced the differential power analysis (DPA). In practice, a cryptosystem is not a black box, it can reveal a part of information about the secret. DPA consists in using side-channel information about the state of the machine which computes, typically a smart card. Differential power analysis uses power consumption and analyses such data statistically. We refer the reader to [1,5] for a description in case of elliptic curve cryptography.

Let us just mention that we consider two generic types of countermeasures to protect the computation of a multiple kP of P lying on an elliptic curve.

- blinding the base point: replace P by a random $\widetilde{P}$ such that $kP = k\widetilde{P}$,
- randomised secret exponent: replace k by a random $\tilde{k}$ such that $kP = \tilde{k}P$.

In all currently proposed countermeasures, only one consists of a modified efficient scalar multiplication technique based on randomised endomorphisms.

Joye and Tymen first presented in [5, Section 5] a randomised endomorphism to prevent DPA for elliptic curve implementation, based on previous work of Solinas [12]. Let us consider a Koblitz curve E defined over $\mathbb{F}_{2^n}$ with $y^2 + xy = x^3 + ax^2 + 1$ as equation, and $a \in \{0, 1\}$. Letting $\tau : (x, y) \mapsto (x^2, y^2)$ the Frobenius endomorphism, which satisfies the equation $u^2 - (-1)^{1-a}u + 2 = 0$. The ring $\mathbb{Z}[\tau]$ is an euclidian domain with the norm $\mathbf{N}(.)$ defined as $\mathbf{N}(r + \tau s) = r^2 + (-1)^{1-a}rs + 2s^2$. The key point of their countermeasure with randomised endomorphisms [5, Section 5] is to use the following property. Let $\rho \in \mathbb{Z}[\tau]$. If $k_1 \equiv k_2 \pmod{\rho(\tau^n - 1)}$, then $\forall P \in E$, $k_1 P = k_2 P$. Thus, they choose randomly $\rho \in \mathbb{Z}[\tau]$ such that $\mathbf{N}(\rho) < 2^{40}$, then compute $\kappa' = k \pmod{\rho(\tau^n - 1)}$, they decompose $\kappa' = \sum_i k'_i \tau^i$ using NAF algorithm [3], and compute $Q = kP$ as $\sum_i k'_i \tau^i(P)$.

This method only applies in characteristic two and for ABC curves. We propose hereafter a new method valid in any characteristic. After recalling the GLV method which constitutes the base of our algorithms in the next section, we develop a first variant of our DPA-resistant algorithm in two versions (cases 1 and 2). These variants combine the two countermeasure types expressed above into a unique algorithm. Thereafter, Section 7 will explore another variant of a DPA-resistant algorithm based on the GLV method, which achieves better performance over similar proved security. Here we only randomise the exponent k. On the other hand the implementation code contains only slight modifications of its deterministic version.

3 The Gallant-Lambert-Vanstone Computation Method

In this part, we briefly summarize the GLV computation method [2]. Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ and P be a point of this curve with order a large prime n (say $\#E(\mathbb{F}_q)/n \leq 4$). Let us consider Φ a non-trivial endomorphism of E defined over $\mathbb{F}_q$ and $X^2 + rX + s$ its characteristic polynomial.

By the Hasse bound, since n is large, $\Phi(P) = \lambda P$ for some $\lambda \in [1, n-1]$. Indeed, there is only one copy of $\mathbb{Z}/n$ inside $E(\mathbb{F}_q)$ and $\Phi(P)$ has also order dividing n. We can easily exclude the case where $\lambda = 0$ which is exceptional (for instance in the examples we have $n \nmid s$, by the Hasse bound). In all cases, λ is obtained as a root of $X^2 + rX + s$ modulo n. The GLV algorithm decomposes k as $k \equiv k_1 + k_2\lambda \pmod{n}$ where $k_i = O(\sqrt{n})$ for $i = 1, 2$. We quickly describe this construction. Let $f : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}/n$ denote the homomorphism defined by $(i, j) \mapsto (i + j\lambda) \pmod{n}$. The goal is to find a small vector u such that $f(u) = k$. As $f((k, 0)) = k$, the problem is reduced to finding two linearly independent vectors v_1 and v_2 of small length (say $O(\sqrt{n})$) such that $f(v_1) = f(v_2) = 0$ and to decompose $(k, 0)$ in this basis with coefficients in $\mathbb{Q}$ and then rounding off $(k, 0)$ to the nearest vector v which is a linear combination of v_1 and v_2 with coefficient in $\mathbb{Z}$. Finally, u is chosen as $u = (k, 0) - v$. The problem of finding v_1 and v_2 is solved in [2] using the extended Euclidean algorithm (see also [10,11] for alternative methods).

Finally kP is computed more efficiently than previous existing methods by calculating first $\Phi(P)$, decomposing $k \equiv k_1 + k_2\lambda \pmod{n}$ with $\max(|k_1|, |k_2|) = O(\sqrt{n})$, and computing $k_1 P + k_2 \Phi(P)$ using elliptic Straus-Shamir multiplication described in [3] or in [13].

4 Elliptic Scalar Multiplication Using Sub-lattices and the GLV Method

4.1 Retrieving New v_1 and v_2

Let Φ be a non trivial endomorphism defined over $\mathbb{F}_q$ such that $\Phi^2 + r\Phi + s = 0$, as in the GLV method. We want to describe a way to mask scalar multiplication using a GLV decomposition so that it becomes immune to DPA. The idea is to use a random sub-lattice $\mathcal{L} = A(\mathbb{Z} \times \mathbb{Z})$, or rather a random matrix A with coefficients in some range (here $[0, R]$ for an given integer R, to be chosen later). We will denote $\Delta = \det A$. The linear map A will often be viewed as a matrix $A = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$ with respect to the canonical basis. Later, we will also translate $\mathbb{Z} \times \mathbb{Z} \cong \mathbb{Z}[\Phi]$ via the isomorphism which sends the canonical basis to $\{1, \Phi\}$. The matrix A will be therefore also viewed as a linear transformation of $\mathbb{Z}[\Phi]$, sending $\{1, \Phi\}$ respectively to $\{\Phi_0, \Phi_1\}$.

Consider the GLV homomorphism:

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

For a sublattice $\mathcal{L} = A(\mathbb{Z} \times \mathbb{Z}) \subset \mathbb{Z} \times \mathbb{Z}$, we denote $v_1^{(\mathcal{L})}$ and $v_2^{(\mathcal{L})}$ two linearly independent vectors in the kernel of $(\mathbb{Z} \times \mathbb{Z})|_{\mathcal{L}} \to \mathbb{Z}/n$ which is the restriction of f to $\mathcal{L}$. We also require that $v_1^{(\mathcal{L})}$ and $v_2^{(\mathcal{L})}$ have rectangle norm $O(\sqrt{n})$.[1]

Such vectors can be computed from the traditional GLV vectors v_1 and v_2. Indeed, note that the index of $\mathcal{L}$ inside $\mathbb{Z} \times \mathbb{Z}$ is

$$(\mathbb{Z} \times \mathbb{Z} : \mathcal{L}) = |\Delta| \ ,$$

and this is also the order of $(\mathbb{Z} \times \mathbb{Z})/\mathcal{L}$. Therefore we may set[2]

$$v_i^{(\mathcal{L})} = \Delta v_i \in (\ker f) \cap \mathcal{L} \ . \tag{1}$$

[1] The rectangle norm of (x, y) is by definition $\max(|x|, |y|)$. We denote it by $|(x, y)|$.

[2] The vectors $v_i^{(\mathcal{L})}$ generate a sublattice of index $|\Delta|$ inside $(\ker f) \cap \mathcal{L}$. In practice (see Appendix A) we use different vectors v_i' than those defined by (3). However we can only prove a bound on the length of the GLV decomposition using these vectors, so in the future performance loss could still be lowered theoretically.

4.2 Decomposition of k

We want to write $kP = k'_1\Phi_0(P) + k'_2\Phi_1(P)$ for some $k'_1, k'_2 = O(\sqrt{n})$. We use the same strategy as in the original GLV method. Indeed as before there exist $\lambda_0, \lambda_1 \in [0, n-1]$ such that $\Phi_0(P) = \lambda_0 P$ and $\Phi_1(P) = \lambda_1 P$. We are aiming therefore at a decomposition of the form

$$k \equiv k'_1\lambda_0 + k'_2\lambda_1 \pmod{n}, \quad \text{with } k'_1, k'_2 = O(\sqrt{n}) \ . \tag{2}$$

Note that $\lambda_0 \equiv \alpha + \beta\lambda \pmod{n}$ and $\lambda_1 \equiv \gamma + \delta\lambda \pmod{n}$. Let us consider the modified GLV map

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

Observe that f' is a linear map which can be written in matrix form as

$$\begin{pmatrix} i \\ j \end{pmatrix} \mapsto (i,j)\begin{pmatrix} \lambda_0 \\ \lambda_1 \end{pmatrix} = (i,j)A\begin{pmatrix} 1 \\ \lambda \end{pmatrix} \pmod{n} \ .$$

Hence, denoting by A^{T} the transpose of A, we can write $f' = f \circ \ell$, with

$$\begin{aligned} \ell \colon \mathbb{Z} \times \mathbb{Z} &\to \mathbb{Z} \times \mathbb{Z} \\ \begin{pmatrix} i \\ j \end{pmatrix} &\mapsto \begin{pmatrix} i' \\ j' \end{pmatrix} = A^{\mathrm{T}}\begin{pmatrix} i \\ j \end{pmatrix} \ . \end{aligned}$$

Let $\mathcal{L}' = \ell(\mathbb{Z} \times \mathbb{Z})$, $\mathcal{K}' = \ker f'$ and $\mathcal{K} = \ker f$. Note that Δ is also the index of $\mathcal{L}'$ inside $\mathbb{Z} \times \mathbb{Z}$. Therefore two short vectors v'_1 and v'_2 of $\mathcal{K}'$ are given by

$$\begin{aligned} v'_i &= \pm\big(A^{\mathrm{T}}\big)^{-1} v_i^{(\mathcal{L}')} \quad i = 1, 2 \\ &= \hat{A}^{\mathrm{T}} v_i \quad \text{by (1)} \ , \end{aligned} \tag{3}$$

where $\hat{A} = \begin{pmatrix} \delta & -\gamma \\ -\beta & \alpha \end{pmatrix}$ is the adjoint matrix of A.

Since $|v_i| \leq \sqrt{1 + |r| + s}\sqrt{n}$ by [11, Theorem 1], if $\alpha, \beta, \gamma, \delta \in [0, R]$ we get

$$|v'_i| \leq 2R\,|v_i| \leq 2R\sqrt{1 + |r| + s}\sqrt{n} \ . \tag{4}$$

To find k'_1 and k'_2 satisfying (2), we proceed as in the original GLV method with a small difference. We need first to find a vector w such that $f'(w) = k \pmod{n}$. In the original method, we could take $w = (k, 0)$. Here we define $\lambda_0^{-1} \in [0, n-1]$ such that $\lambda_0\lambda_0^{-1} \equiv 1 \pmod{n}$ (we have $\lambda_0 \not\equiv 0 \pmod{n}$, see the discussion following (7)). Finding λ_0^{-1} amounts to an application of the extended Euclidean algorithm to n, λ_0. Then we have $f'(k\lambda_0^{-1}, 0) = k \pmod{n}$. From this point everything else is identical to the original method by replacing $(k, 0)$ with $(k\lambda_0^{-1}, 0)$. At the end we get that

$$\max(|k'_1|, |k'_2|) \leq 2R \max(|k_1|, |k_2|) \leq 2R\sqrt{1 + |r| + s}\sqrt{n} \ . \tag{5}$$

4.3 The Case of Small Determinant

The bound of (5) can be considerably sharpened when Δ is small. We shall not dwell on all cases, but shall consider only the case $\Delta = 1$. Let us draw some generic observations from the last section. Indeed from the relation $f' = f \circ \ell$ it is immediate to see that $\ell(\mathcal{K}') \subset \mathcal{K}$. On the other hand clearly

$$\hat{A}^{\mathrm{T}}(\mathcal{K}) \subset \mathcal{K}' \ ,$$

which implies

$$\Delta\mathcal{K} \subset A^{\mathrm{T}}\mathcal{K}' \subset \mathcal{K} \ .$$

Hence if $\Delta = 1$ one gets equalities in all the above inclusions. In particular, $\mathcal{K}' = \mathcal{K}$ and this implies that one may take $(v_1', v_2') = (v_1, v_2)$ in the GLV sublattice decomposition, so that in (5) the parameter R is no longer there.

In the appendix, we give full details of the modified GLV algorithms. In the following, we measure the protection offered by this method against DPA and its performance slow-down.

5 On the Protection Offered by the Randomised GLV Method against DPA

We have seen in the previous section how the choice of a random A with coefficients in $[0, R]$ affects the length of the vector $u = (k_1, k_2)$. A more detailed performance analysis of the analogue of the GLV algorithm in this case will be carried out in the next section. Here we will evaluate the number of different decompositions of kP offered by different choices of A with coefficients less than R.

In order to evaluate this quantity, we note that the number of matrices A such that $\Delta \neq 0$ is around R^4 and the number of those with $\Delta = \pm 1$ is around R^2.

The starting point of the analysis is the remark that DPA as described in Section 2 cannot work if the GLV decomposition is randomised as before, because the power consumption pattern is also randomised for a single exponent k. This will be quantified by the analysis of (6) and (7) below.

We are concerned with two points, both of which add entropy and enhance the security of our system. Let A be the previous matrix and $B = \begin{pmatrix} \epsilon & \zeta \\ \eta & \theta \end{pmatrix} \neq A$ another such matrix, with coefficients in $[0, R]$. Call

$$\begin{pmatrix} \Psi_0 \\ \Psi_1 \end{pmatrix} = B \begin{pmatrix} 1 \\ \Phi \end{pmatrix} \ .$$

We want to avoid a situation where

$$kP = k_1'\Phi_0(P) + k_2'\Phi_1(P) = \tilde{k}_1'\Psi_0(P) + \tilde{k}_2'\Psi_1(P),$$

$$k_i' = \tilde{k}_i' \quad \text{and} \tag{6}$$

$$\Phi_i(P) = \Psi_i(P) \quad i = 1, 2 \ . \tag{7}$$

Let us analyse first (7). This condition is equivalent to

$$\alpha + \beta\lambda + c_1 n = \epsilon + \zeta\lambda \quad \text{and} \quad \gamma + \delta\lambda + c_2 n = \eta + \theta\lambda \ , \tag{8}$$

with $c_1, c_2 \in \mathbb{Z}$. We want to show that one must have $c_1 = c_2 = 0$ as soon as for instance

$$R < \sqrt{n}/\sqrt{1 + |r| + s} \ . \tag{9}$$

We analyse the left-hand inequality, the other one being analogous. One has

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

so that $(\alpha - \epsilon, \beta - \zeta) \in \ker f$. But for $(x, y) \in \ker f - \{(0, 0)\}$ it is known by the proof of [11, Theorem 1] that

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

by (9), hence this forces $(x, y) = (0, 0)$, thus proving our claim. Therefore, given A, there is no other B satisfying (6) and (7).

Case 1: $\Delta \neq 0$. Since the total number of matrices A is of order R^4, the probability that two decompositions of kP match is less than $1/R^4$.

Case 2: $\Delta = \pm 1$. In this case the total number of A's is around R^2, so the probability that two decompositions of kP match is less than $1/R^2$.

6 On the Additional Computation Cost of the Randomised GLV Method

We now measure the extra computation cost of this method with respect to the plain GLV method. Three parts of extra computation can be distinguished:

1. computation of $\Phi_0(P)$, $\Phi_1(P)$,
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 $kP = k_1 P + k_2\lambda P$.

We analyse these points.

The decomposition of k into $k'_1\lambda_0 + k'_2\lambda_1$ through the GLV method can be applied as described at the end of Section 4.2. The increase in computation is to invert λ_0 modulo n, and it can be neglected in comparison with the global computation of the elliptic curve scalar multiplication.

The computation of v'_1 and v'_2 by (3) is also fast and can be neglected. This step does not apply in Case 2 ($\Delta = \pm 1$).

Secondly, the elliptic Straus-Shamir method of Solinas [13] can be used to compute $\Phi_0(P)$, $\Phi_1(P)$ and $k'_1\Phi_0(P) + k'_2\Phi_1(P)$. It is known that its average

computation cost $C(l)$ for $l = \max(\log_2 k'_1, \log_2 k'_2)$ is l doublings and $l/2$ curve additions to obtain $k'_1\Phi_0(P) + k'_2\Phi_1(P)$. Therefore, using (5), the cost of computing $k'_1\Phi_0(P) + k'_2\Phi_1(P)$ is augmented by

$$\begin{cases} 300\,\dfrac{C(\log_2 R)}{C\,(\log_2 \sqrt{n})}\% \approx \dfrac{300\log_2 R}{\log_2 \sqrt{n}}\% & \text{in Case 1,} \\ 200\,\dfrac{C(\log_2 R)}{C\,(\log_2 \sqrt{n})}\% \approx \dfrac{200\log_2 R}{\log_2 \sqrt{n}}\% & \text{in Case 2.} \end{cases} \tag{10}$$

since $\log_2 k_i \approx (\log n)/2$. By letting

$$R < n^{1/16} \qquad \text{which implies (9)} \ , \tag{11}$$

one gets for instance that the augmentation cost is less than 37.5%. This is the case, for instance, if n has 160 bits and $R = 2^{10}$.

In Case 2, one has to double the size of R to achieve the same security, but performance is better than Case 1 for the same R, hence the 50% increase in computation cost.

7 An Affine Generalisation

One can in fact easily generalise our ideas to an affine setting, where instead of masking the GLV decomposition with a linear map $x \mapsto Ax$, one uses an affine map $x \mapsto Ax+\rho$. We present a special case of this affine method, in which $A = \mathrm{Id}$ and only ρ is randomised. It can be implemented with very small modifications of the code. The idea is to randomise the part "**Finding** v" of [2].

The vector $(k, 0)$ breaks down as $(k, 0) = \beta_1 v_1 + \beta_2 v_2$, with $\beta_1, \beta_2 \in \mathbb{Q}$ using [2, Lemma 1]. Let $R > 0$, $\rho_1, \rho_2 \in [0, R]$ two random integers and $b'_i = \lceil \beta_i \rfloor - \rho_i$ for $i \in \{1, 2\}$, where $\lceil \cdot \rfloor$ means the nearest integer. The vector $u' = (k'_1, k'_2)$, constructed as $u' = (k, 0) - v'$ where $v' = b'_1 v_1 + b'_2 v_2$, has norm at most $c \max(|v_1|, |v_2|)$ with $c \leq 2R + 1$. Furthermore, by construction we have $kP = k'_1 P + k'_2 \Phi(P)$.

With our method, we have a very easy control between the additional running time and the expected security, indeed for a typical field elliptic curve with a 160-bit number n, $|v_1|$ and $|v_2|$ are 80-bit numbers and we have a 25% increase in computation compared to the plain GLV method when $R = 2^{20}$.

On the other hand, security here can easily be justified, since the difference between the modified vector u' and the original one u is $\rho_1 v_1 + \rho_2 v_2$, which is a different vector for each choice of ρ_1, ρ_2. Hence the probability that (6) holds is R^{-2}.

8 Conclusion

In this paper, we propose to modify the Gallant-Lambert-Vanstone computation method to prevent differential analysis, in a way which benefits from the computational speedup of the method. The class of curves where this computation

is possible is the same as for the GLV method. Our method can be viewed as a two-dimensional generalisation of Coron's [1].

We have distinguished two cases of possible linear randomisations. In the second one the class of random matrices is more restricted and security is slightly reduced, however, this class offers the advantage of not having to redo the precomputation stage to find the vectors v_i' which we take to be the original v_i of the plain GLV method.

Thus, for instance, a single computation of kP can assume a randomly chosen consumption pattern, the probability that two of these matching being less than 2^{-40} independently of the chosen elliptic curve. With this security threshold, on a 160-bit elliptic curve, performance is only at most 37.5% slower than using the original GLV method (50% when using unimodular matrices).

In these two cases, we mask both the exponent k and the points P, $\Phi(P)$.

These considerations can be carried through to affine randomisations, where we presented the simplest example with collision probability 2^{-40} and 25% additional running time on a 160-bit elliptic curve. This variant can be considered as a generalisation of Coron's first countermeasure of randomising the private exponent [1, Section 5].

Acknowledgements. The authors would like to thank Marc Joye for useful comments on the preliminary versions of this paper, and François Koeune for answering some questions on side-channel cryptanalysis. We are also grateful to the anonymous referees for their valuable remarks.

References

1. J-S. Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. In Ç. K. Koç and C.Paar, editors, *Advances in Cryptology - Proceedings of CHES 1999*, volume 1717 of *Lecture Note in Computer Science*, pages 292–302. Springer, 1999.
2. R. P. Gallant, J. L. Lambert, and S. A. Vanstone. Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms. In J. Kilian, editor, *Advances in Cryptology - Proceedings of CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 190–200. Springer, 2001.
3. D. M. Gordon. A Survey of Fast Exponentiation Methods. *Journal of Algorithms*, 27(1):129–146, 1998.
4. M. Joye and J-J. Quisquater. Hessian Elliptic Curves and Side-Channel Attacks. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Advances in Cryptology - Proceedings CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 402–410. Springer, 2001.
5. M. Joye and C. Tymen. Protections against Differential Analysis for Elliptic Curve Cryptography -An Algebraic Approach-. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Advances in Cryptology - Proceedings CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 377–390. Springer, 2001.
6. P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In N. Koblitz, editor, *Advances in Cryptology - Proceedings of CRYPTO 1996*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

7. P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In M. Wiener, editor, *Advances in Cryptology - Proceedings of CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.
8. P.-Y. Liardet and N.P. Smart. Preventing SPA/DPA in ECC Systems using the Jacobi Form. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Advances in Cryptography - Proceedings CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 391–401. Springer, 2001.
9. B. Möller. Securing Elliptic Curve Point Multiplication against Side-Channel Attacks. In G.I. Davida and Y. Frankel, editors, *Advances in Cryptology - Proceedings of ISC 2001*, volume 2200 of *Lecture Note in Computer Science*, pages 324–334. Springer, 2001.
10. Y-H. Park, S. Jeong, C. Kim, and J. Lim. An Alternate Decomposition of an Integer for Faster Point Multiplication on Certain Elliptic Curves. In D. Naccache and P. Paillier, editors, *Advances in Cryptology - Proceedings of PKC 2002*, volume 2274 of *Lecture Notes in Computer Science*, pages 323–334. Springer, 2002.
11. F. Sica, M. Ciet, and J-J. Quisquater. Analysis of the Gallant-Lambert-Vanstone Method based on Efficient Endomorphisms: Elliptic and Hyperelliptic Curves. In H. Heys and K. Nyberg, editors, *Advances in Cryptology - Proceedings of SAC 2002*, Lecture Notes in Computer Science. Springer, 2002. To appear.
12. J. A. Solinas. An Improved Algorithm for Arithmetic on a Family of Elliptic Curves. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - Proceedings of CRYPTO 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 357–371. Springer, 1997.
13. J.A. Solinas. Low-Weight Binary Representations for Pairs of Integers. Technical Report CORR 2001-41, CACR, Available at: `www.cacr.math.uwaterloo.ca/techreports/2001/corr2001-41.ps`, 2001.

A Algorithm for Elliptic Scalar Multiplication Using Sub-lattices

1. Randomly choose $\alpha, \beta, \gamma, \delta \in [1, R]$ such that $\alpha\delta - \beta\gamma \neq 0$ in Case 1 (resp. $= \pm 1$ in Case 2).
2. Compute $\Phi_0(P)$ and $\Phi_1(P)$ with Solinas' algorithm as
$$\begin{pmatrix} \Phi_0(P) \\ \Phi_1(P) \end{pmatrix} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} 1 \\ \Phi(P) \end{pmatrix} .$$
3. 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.
4. Compute $\lambda' = \lambda_1 \lambda_0^{-1} \pmod{n}$.
5. Find v_1' and v_2' in $\mathbb{Z} \times \mathbb{Z}$ by applying the original GLV algorithm, replacing λ by λ' (resp. $v_i' = v_i$ in Case 2).
6. Express $(k\lambda_0^{-1}, 0) = \beta_1 v_1' + \beta_2 v_2'$, where $\beta_i \in \mathbb{Q}$.
7. Let $b_i = \lceil \beta_i \rfloor$ and $v' = b_1 v_1' + b_2 v_2'$.
8. Compute $(k_1', k_2') = (k\lambda_0^{-1}, 0) - v'$.
9. Compute kP as $k_1' \Phi_0(P) + k_2' \Phi_1(P)$ with the elliptic Straus-Shamir (Solinas) algorithm.

B Algorithm for Affine Generalisation

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