Elliptic Curves and Side-Channel Analysis

[Published in *ST Journal of System Research* **4**(1):283–306, 2003.]

Marc Joye

Gemplus Card International, Card Security Group
Parc d'Activités de Gémenos, B.P. 100, 13881 Gémenos Cedex, France
marc.joye@gemplus.com – http://www.geocities.com/MarcJoye/
http://www.gemplus.com/smart/

Abstract. Naive implementations of crypto-algorithms are susceptible to side-channel analysis. This paper surveys the known methods for preventing side-channel analysis in elliptic curve cryptosystems.

Keywords. Elliptic curve cryptography (ECC), side-channel analysis, timing attacks, simple power analysis (SPA), differential power analysis (DPA).

1 Introduction

Provable security becomes more and more popular in the cryptographic community. As exemplified by the NESSIE project [22], it is now common to see it as an attribute of a cryptosystem. Provable security is at the protocol level, a harder task may be to evaluate the security of a cryptosystem at the implementation level. Rather than considering a cryptosystem as a black-box, we may assume that some sensitive data can leak during the course of the execution of a (naively implemented) crypto-algorithm. A concrete example is given by the so-called *side-channel analysis* [14, 15].

Side-channel analysis is a powerful technique re-discovered by P. Kocher in 1996. The principle consists in monitoring some side-channel information like the running time [14], the power consumption [15], or the electromagnetic radiation [7, 23]. Next, from the monitored data, the attacker tries to deduce the inner-workings of the algorithm and thereby to retrieve some secret information. When there is a single measurement, the process is referred to as a *simple* side-channel analysis; and when there are several measurements handled together with statistical tools, the process is referred to as *differential* side-channel analysis.

This paper is aimed at studying the resistance of elliptic curve cryptosystems against those two classes of attacks. In particular, we survey the various strategies proposed so far to prevent side-channel attacks.

2 Elliptic Curve Cryptography

We start with a brief review of elliptic curve cryptography and refer the reader to the many excellent textbooks on the subject (e.g., [2]) for more detail.

An elliptic curve presents the mathematical structure of an additive group. What makes elliptic curves particularly attractive for cryptographic applications [13, 18] is that the discrete logarithm problem in elliptic curve groups is harder than in groups previously considered. As a result, with shorter key lengths, comparable levels of security can be attained.

An elliptic curve over a field $\mathbb{K}$ is formed by the point $\boldsymbol{O}$ 'at infinity' and the set of points $\boldsymbol{P} = (x, y) \in \mathbb{K} \times \mathbb{K}$ satisfying a (non-singular) Weierstraß equation

$$E_{/\mathbb{K}} : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \ .$$

The basic operation in elliptic curve cryptography is the *scalar multiplication*, that is, given a point $\boldsymbol{P} \in E(\mathbb{K})$, one has to compute $\boldsymbol{Q} = k\boldsymbol{P} := \boldsymbol{P} + \boldsymbol{P} + \cdots + \boldsymbol{P}$ (k times). The discrete logarithm problem consists in finding the value of k from the values of $\boldsymbol{P}$ and $\boldsymbol{Q} = k\boldsymbol{P}$.

3 Simple Side-Channel Analysis

A widely-used method for performing a scalar multiplication is the celebrated double-and-add method (i.e., the additive analogue of the square-and-multiply algorithm).

```
Input:   P, k = (1, k_{ℓ-2}, ..., k_0)_2
Output:  Q = kP
---------------------------------------
R0 ← P
for j = ℓ - 2 downto 0 do
  R0 ← 2R0
  if (k_j = 1) then R0 ← R0 + P
endfor
return R0
```

Fig. 1. Double-and-add method

As given in textbooks, the formulæ for doubling a point or for adding two (distinct) points on a Weierstraß elliptic curve are different. Therefore, a simple power analysis (i.e., a simple side-channel analysis using power consumption as side channel) will produce different power traces that may reveal the value of k in the double-and-add method, from the distinction between the two operations. There are basically three approaches to circumvent the leakage. This can be achieved by:

1. inserting dummy instructions [5];
2. considering alternative parameterizations [10, 16, 1] or unifying the addition formulæ [3];
3. using algorithms that already behave 'regularly' [17, 21, 19, 3, 8, 6].

```
Input:   P, k = (1, k_{ℓ−2}, ..., k_0)_2
Output:  Q = kP
R_0 ← P
for j = ℓ − 2 downto 0 do
  R_0 ← 2R_0 ;  R_1 ← R_0 + P
  b ← k_j ;  R_0 ← R_b
endfor
return R_0
```

(a) Double-and-add *always* [5]

```
Input:   P, k = (1, k_{ℓ−2}, ..., k_0)_2
Output:  Q = kP
R_0 ← P;  R_1 ← 2P
for j = ℓ − 2 downto 0 do
  b ← k_j
  R_{1−b} ← R_0 + R_1 ;  R_b ← 2R_b
endfor
return R_0
```

(b) Montgomery ladder [20, 12]

```
Input:   P, k = (1, k_{ℓ−2}, ..., k_0)_2
Output:  Q = kP
R_0 ← 2P;  R_1 ← P;  j ← ℓ − 2
while (j ≥ 1) do
  b ← k_j ;  R_0 ← R_0 + R_b
  k_j ← 0;  j ← j + b − 1
endwhile
R_1 ← R_0 + P;  b ← k_0 ;  R_0 ← R_b
return R_0
```

(c) Double-and-add with multiplier rewriting [9]

Fig. 2. Regular scalar multiplication algorithms

The first and third approaches share the same idea: it consists in ultimately having an algorithm that behaves consistently and regularly whatever the processed data. In [5], Coron suggests to perform a dummy addition in the double-and-add method when the processed bit is '0' so that each iteration appears as a doubling followed by an addition (see Fig. 2-a). Another possibility is to use a scalar multiplication method that already behaves regularly, as is the case for the Montgomery ladder [20, 12] (see Fig. 2-b). The corresponding algorithm for elliptic curves over binary fields is detailed in [17] and in [21] over fields of large characteristic. The latter algorithm is however restricted to 'Montgomery' curves; see [3, 8, 6] for general Weierstraß elliptic curves.

The second approach for preventing simple side-channel analysis is to rewrite the addition formulæ so that the same formula can be used for doubling or adding points, indifferently. This is suggested by Brier and Joye in [3] where unified addition formulæ for Weierstraß elliptic curves are presented. In [16], Liardet and Smart propose to represent elliptic curves as the intersection of two quadric surfaces. Contrary to the Weierstraß parameterization, the classical addition formula on this parameterization is already valid for doubling or adding points [4]. However, for efficiency reasons, only elliptic curves with three points of order 2 (and thus whose order is multiple of 4) give rise to fast arithmetic. Consequently, for real-life applications, the technique is not available for general elliptic curves (see also [1] for several improvements). For elliptic curves whose order is a multiple of 3, one can use the Hessian parameterization. A trick for evaluating a doubling in terms of a general addition on a Hessian elliptic curve is described by Joye and Quisquater [10].

We note, however, that the double-and-add algorithm depicted in Fig. 1 cannot be used as is with unified addition formulæ. This algorithm is *not* regular because of the `if-then` instruction; a simple side-channel analysis may reveal sensitive data (although the analysis is at a smaller scale). One has to use a regular variant of the double-and-add algorithm. We quote one such variant from [9] (see Fig. 2-c).

4 Differential Side-Channel Analysis

Even if an algorithm is protected against side-channel analysis, it may succumb to the more sophisticated differential analysis [5]. Practically, we note however that very few elliptic curve cryptosystems are susceptible to such attacks as, usually, the input point is imposed by the system and the multiplier is an ephemeral parameter, varying at each execution.

Assume that the double-and-add method is implemented with one of the regular variants given in Fig. 2. Let $k = (k_{\ell-1}, \dots, k_0)_2$ be the binary expansion of multiplier k. Suppose that an attacker already knows the highest bits, $k_{\ell-1}, \dots, k_{j+1}$, of k. Then, he guesses that the next bit k_j is equal to '1'. He randomly chooses several points $\boldsymbol{P_1}, \dots, \boldsymbol{P_t}$ and computes $\boldsymbol{Q_r} = (\sum_{i=j}^{\ell-1} k_i 2^i)\boldsymbol{P_r}$ for $1 \le r \le t$. Using a boolean selection function g, he prepares two sets: the first set, $\mathcal{S}_{\texttt{true}}$, contains the points $\boldsymbol{P_r}$ such that $g(\boldsymbol{Q_r}) = \texttt{true}$ and the second set, $\mathcal{S}_{\texttt{false}}$, contains those such that $g(\boldsymbol{Q_r}) = \texttt{false}$ (a candidate for the selection function may, for example, be the value of a given bit in the representation of $\boldsymbol{Q_r}$). Let $\mathcal{C}(r)$ denote the side-channel information associated to the computation of $k\boldsymbol{P_r}$ by the cryptographic device (e.g., the power consumption). If the guess $k_j = 1$ is incorrect then the difference

$$\langle \mathcal{C}(r) \rangle_{\substack{1 \le r \le t \\ \boldsymbol{P_r} \in \mathcal{S}_{\texttt{true}}}} - \langle \mathcal{C}(r) \rangle_{\substack{1 \le r \le t \\ \boldsymbol{P_r} \in \mathcal{S}_{\texttt{false}}}}$$

will be ≈ 0 as the two sets appear as two random (i.e., uncorrelated) sets; otherwise the guess is correct. Once k_j is known, the remaining bits, $k_{j-1}, \dots, k_0$, are recovered recursively, in the same way.

In order to thwart the above differential side-channel analysis, one has to randomize the inputs of the crypto-algorithm so that the attacker is no longer able to prepare two sets of points with a selection function. Several methods are available, we list some of them:

1. randomizing the base-point $\boldsymbol{P}$:
   - by point blinding [5]: compute $\boldsymbol{Q} = k\boldsymbol{P}$ as $\boldsymbol{Q} = k(\boldsymbol{P} + \boldsymbol{R}) - k\boldsymbol{R}$ for a random point $\boldsymbol{R}$;
   - with randomized projective coordinates [5]: in projective coordinates, $(X : Y : Z)$ and $(rX : rY : rZ)$ with $r \neq 0$ represent the same point. So for a random r, if $\boldsymbol{P} = (x_0, y_0)$, $\boldsymbol{Q}$ is computed as $\boldsymbol{Q} = k(rx_0 : ry_0 : r)$;
   - with randomized elliptic curve isomorphisms [11]: if ϕ denotes a random isomorphism between $E(\mathbb{K})$ and $E'(\mathbb{K})$, then one computes $\boldsymbol{Q}$ as $\boldsymbol{Q} = \phi^{-1}\Big(k\big(\phi(\boldsymbol{P})\big)\Big)$;
   - with randomized field isomorphisms [11]: if ϕ is a random isomorphism between $\mathbb{K}$ and $\mathbb{K}'$, then $\boldsymbol{Q}$ can be computed as above. We refer the reader to the original paper ([11]) for a concrete realization over binary fields $\mathbb{K}$;
2. randomizing the multiplier k:
   - by multiplier blinding [5]: if $n = \mathrm{ord}_E(\boldsymbol{P})$ denotes the order of $\boldsymbol{P} \in E(\mathbb{K})$, then $\boldsymbol{Q}$ is computed as $\boldsymbol{Q} = (k + r\,n)\boldsymbol{P}$ for a random r. Alternatively, one can replace n by the order of the elliptic curve, $\#E(\mathbb{K})$;
   - by randomized multiplier recoding [11]: this technique applies to Koblitz curves over $GF(2^m)$. Let $\tau : (x, y) \mapsto (x^2, y^2)$ represent the Frobenius endomorphism. Considering k as an element of $\mathbb{Z}[\tau] \subseteq \mathrm{End}(E)$, one chooses a random $\rho \in \mathbb{Z}[\tau]$, evaluates the τ-NAF expansion of $\kappa := k \bmod \rho(\tau^m - 1)$, $\kappa = \sum_i \kappa_i 2^i$ with $\kappa_i \in \{-1, 0, 1\}$, and computes $\boldsymbol{Q}$ as $\boldsymbol{Q} = \sum_i \kappa_i\, \tau^i(\boldsymbol{P})$.

All these techniques are of independent interest and can of course be combined to better fulfill the needs of a particular application. Moreover, it is easy to derive variants thereof; the idea being to randomize the execution of the crypto-algorithm.

5 Conclusions

Side-channel analysis is now well understood by implementors and efficient countermeasures are known. This paper surveyed various ways for protecting elliptic curve cryptosystems against both simple and differential side-channel analysis.

References

1. Olivier Billet and Marc Joye. The Jacobi model of an elliptic curve and side-channel analysis. Cryptology ePrint Archive, Report 2002/125, IACR, August 2002. Available at URL `http://eprint.iacr.org/2002/125/`.
2. Ian Blake, Gadiel Seroussi, and Nigel Smart. *Elliptic curves in cryptography*, volume 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1999.

3. Éric Brier and Marc Joye. Weierstraß elliptic curves and side-channel attacks. In D. Naccache, editor, *Public Key Cryptography*, volume 2274 of *Lecture Notes in Computer Science*, pages 335–345. Springer-Verlag, 2002.
4. D.V. Chudnovsky and G.V. Chudnovsky. Sequences of numbers generated by addition in formal groups and new primality and factorization tests. *Advances in Applied Mathematics*, 7:385–434, 1986/87.
5. Jean-Sébastien Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems (CHES '99)*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302. Springer-Verlag, 1999.
6. Wieland Fischer, Christophe Giraud, Erik Woodward Knudsen, and Jean-Pierre Seifert. Parallel scalar multiplication on general elliptic curves over $\mathbb{F}_p$ hedged against non-differential side-channel attacks. Cryptology ePrint Archive, Report 2002/007, IACR, January 2002. Available at URL `http://eprint.iacr.org/2002/007/`.
7. Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Ç.K. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer-Verlag, 2001.
8. Tetsuya Izu and Tsuyoshi Takagi. A fast parallel elliptic curve multiplication resistant against side channel attacks. In D. Naccache and P. Paillier, editors, *Public Key Cryptography*, volume 2274 of *Lecture Notes in Computer Science*, pages 280–296. Springer-Verlag, 2002.
9. Marc Joye. Recovering the lost efficiency of exponentiation algorithms on smart cards. *Electronics Letters*, 38(19), 2002.
10. Marc Joye and Jean-Jacques Quisquater. Hessian elliptic curves and side-channel attacks. In Ç.K. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 402–410. Springer-Verlag, 2001.
11. Marc Joye and Christophe Tymen. Protections against differential analysis for elliptic curve cryptography: An algebraic approach. In Ç.K. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 377–390. Springer-Verlag, 2001.
12. Marc Joye and Sung-Ming Yen. The Montgomery powering ladder. In B.S. Kaliski Jr., Ç.K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2002*, Lecture Notes in Computer Science. Springer-Verlag, To appear.
13. Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.
14. Paul Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer-Verlag, 1996.
15. Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In M. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer-Verlag, 1999.
16. Pierre-Yvan Liardet and Nigel P. Smart. Preventing SPA/DPA in ECC systems using the Jacobi form. In Ç.K. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 391–401. Springer-Verlag, 2001.

17. Julio López and Ricardo Dahab. Fast multiplication on elliptic curves over $GF(2^m)$ without precomputation. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems*, volume 1717 of *Lecture Notes in Computer Science*, pages 316–327. Springer-Verlag, 1999.
18. Victor S. Miller. Use of elliptic curves in cryptography. In H.C. Williams, editor, *Advances in Cryptology – CRYPTO '85*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer-Verlag, 1986.
19. Bodo Möller. Securing elliptic curve point multiplication against side-channel attacks. In G.I. Davida and Y. Frankel, editors, *Information Security*, volume 2200 of *Lecture Notes in Computer Science*, pages 324–334. Springer-Verlag, 2001.
20. Peter L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of Computation*, 48(177):243–264, January 1987.
21. Katsuyuki Okeya and Kouichi Sakurai. Power analysis breaks elliptic curve cryptosystems even secure against the timing attack. In B. Roy and E. Okamoto, editors, *Progress in Cryptology – INDOCRYPT 2000*, volume 1977 of *Lecture Notes in Computer Science*, pages 178–190. Springer-Verlag, 2000.
22. Bart Preneel. NESSIE: New European schemes for signatures, integrity, and encryption. In *this issue*.
23. Jean-Jacques Quisquater. Electromagnetic attacks. In *this issue*.