Side Channel Attacks against Pairing over Theta Functions

Nadia EL MRABET[*]

LIASD - University Paris 8 - elmrabet@univ-paris8.fr

Abstract. In [17], Lubicz and Robert generalized the Tate pairing over any abelian variety and more precisely over Theta functions. The security of the new algorithms is an important issue for the use of practical cryptography. Side channel attacks are powerful attacks, using the leakage of information to reveal sensitive data. The pairings over elliptic curves were sensitive to side channel attacks. In this article, we study the weaknesses of the Tate pairing over Theta functions when submitted to side channel attacks.

Key words: pairing based cryptography, Theta function, side channel attacks, differential power analysis, fault attacks.

1 Introduction

Since they appeared in cryptography, the efficient computation of pairings is a very active area of research. Originally defined over elliptic curves in Weierstrass model [19], pairings have been computed in other models of elliptic curves (for example Edwards [13], Huff [14], Jacobi [5]). They have also been studied in different systems of coordinates such as affine [16], Jacobian, projective, Chudnovsky [3] or in original representation of finite fields RNS [2]. The main algorithm to compute pairings is the Miller algorithm [19]. It is based on a double and add scheme. Several works aimed to reduce the number of iterations of Miller's algorithm and to develop the notion of optimal pairings [12]. In both Optimal Pairings [21] and Pairings Lattices [11] the authors present methods to find the Miller algorithm with the smallest number of iterations. All these works deal with a computation of pairing over elliptic (or hyper elliptic) curves.

The latest improvement in the computation of pairing was the description of efficient pairing computation in a more general case for any algebraic variety; and in particular pairings over Theta functions. In [17], Lubicz and Robert generalize the notion of the Weil and the Tate pairings to any abelian variety. To do so, they made an explicit link between the Weil and the Tate pairings and the intersection pairing on the degree 1 homology of an abelian variety. The result is a general definition of pairings and they explicit the formulas for the case of level 2 and 4 Theta functions in order to obtain the most efficient algorithm, considering time and memory consumption. Their algorithm to compute pairing is based on a Montgomery Ladder's approach.

Each time new formulas for pairing are proposed, the security and implementation of the new algorithms are an important issue for the use in practical cryptography. As for every cryptographic protocol constructed nowadays, the size of groups involved in pairing computation are chosen to be large enough to avoid the discrete logarithm attack. Consequently,

[*] The author wishes to acknowledge support from French project ANR INS 2012 SIMPATIC.

pairing implementations are secured against mathematical attacks. Nevertheless, considering side channel attacks, we cannot predict if an algorithm is more or less secure than another given the representation of the groups. Weaknesses to side channel attacks of pairing based cryptography over elliptic curve have been highlighted [20, 22, 23, 7, 8]. Then, wondering if a pairing implemented in Theta function would be vulnerable to side channel attacks is an important issue for pairing based cryptography and this is the main objective of this contribution. The remaining of the article is organized as follows. The Section 2 is devoted to the definition of pairings over Theta functions. In Section 3 we describe the application of side channel attacks to pairing over Theta functions, we highlight the weaknesses of the pairing and provide countermeasures to secure the computation. We conclude in Section 4.

2 Pairings over Theta Function

This Section is a brief review of the results in [17]. We present the notations and background of Theta functions in Section 2.1. We give the definition of the Weil and the Tate pairings and of the algorithm to compute the Tate pairing in Section 2.2.

2.1 Background on Theta function

Let $\mathbb{H}_g$ be the g dimensional Siegel upper-half space which is the set of $g \times g$ symmetric matrices Ω whose imaginary part is positive definite. For $\Omega \in \mathbb{H}_g$, let $\Lambda_\Omega = \Omega\mathbb{Z}^g \times \mathbb{Z}^g$ the lattice of $\mathbb{C}^g$ defined by Ω. If A is an abelian variety of dimension g over the number field K with a principal polarization then A is analytically isomorphic to $\mathbb{C}^g/\Lambda_\Omega$. Let $\Pi : \mathbb{C} \to \mathbb{C}^g/\Lambda_\Omega = A$ be the canonical projection. The classical theory of Theta functions gives a lot of functions on $\mathbb{C}^g$ that are pseudo-periodic with respect to Λ_Ω and can be used as a projective coordinate system for A. For $a, b \in \mathbb{Q}^g$, the Theta function with rational characteristics (a, b) is an analytic function on $\mathbb{C}^g \times \mathbb{H}_g$ given by

$$\theta \begin{bmatrix} a \\ b \end{bmatrix} (z, \Omega) = \sum_{n \in \mathbb{Z}^g} \exp \left[\Pi i^t(n+a).\Omega.(n+a) + 2\Pi i^t(n+a).(z+b) \right],$$

where ^{t} represents the transpose of a vector.

In order to describe the pseudo-periodicity relations verified by the Theta function, we introduce a certain pairing on $\mathbb{C}^g$. We have that $\mathbb{C}^g$ is isomorphic to $\mathbb{R}^{2g}$ via the map

$$\begin{cases} \mathbb{R}^{2g} \longrightarrow \mathbb{C}^g \\ (x_1, x_2) \longrightarrow \Omega x_1 + x_2. \end{cases}$$

For α, $\beta \in \mathbb{R}^{2g}$, let $\alpha = (\alpha_1, \alpha_2)$ and $\beta = (\beta_1, \beta_2)$, we define $e_\Omega : \mathbb{R}^{2g} \to \mathbb{C}$ by $e_\Omega(\alpha, \beta) = \exp(2i\Pi(\alpha_1\beta_2 - \alpha_2\beta_1))$.

The pseudo periodicity of θ is given by

$$\theta \begin{bmatrix} a \\ b \end{bmatrix} (z + \Omega.m + n, \Omega) = e_\Omega(\Omega a + b, \Omega m + n) \times e^{(-\Pi i^t m\Omega m - 2\Pi i^t mz)} \times \theta \begin{bmatrix} a \\ b \end{bmatrix} (z, \Omega).$$

A function f on $\mathbb{C}^g$ is Λ_Ω-Theta-periodic of level $l \in \mathbb{N}$ if for all $z \in \mathbb{C}^g$ and $m \in Z^g$, we have $f(z+m) = f(z)$, $f(z+\Omega.m) = \exp(-\Pi i l^t m.\Omega.m - 2\Pi i l^t z.m) f(z)$. For any $l \in \mathbb{N}^\star$, the set $H_{\Omega,l}$ of Λ_Ω-quasi-periodic functions of level l is a finite dimensional $\mathbb{C}$-vector space whose basis can be given by the Theta functions with characteristics $\left(\theta \begin{bmatrix} 0 \\ b/l \end{bmatrix} (z, l^{-1}.\Omega)\right)_{b\in[0,\dots,l-1]^g}$. If $l = k^2$, then an alternative basis of $H_{\Omega,l}$ is $\left(\theta \begin{bmatrix} a/k \\ b/k \end{bmatrix} (kz, \Omega)\right)_{a,b\in[0,\dots,k-1]^g}$.

Once the level $l \in \mathbb{N}$ is fixed, the following conventions are adopted $\mathbb{Z}(\bar{l}) = (\mathbb{Z}/l\mathbb{Z})^g$ and for a point $z_P \in \mathbb{C}^g$ and $i \in \mathbb{Z}(\bar{l})$ let $\theta_i(z_P) = \theta \begin{bmatrix} 0 \\ i/l \end{bmatrix} (z_P, \Omega/l)$. If $l = k^2$, for i, $j \in \mathbb{Z}(\overline{k})$, let $\theta_{i,j}(z_P) = \theta \begin{bmatrix} i/k \\ j/k \end{bmatrix} (k.z_P, \Omega)$.

Let $\widetilde{P}$ denote the element of $\mathbb{A}^{l^g}(\mathbb{C})$ with coordinates $\widetilde{P}_i = \theta_i(z_P)$. Let P be the associated point of A that will be be considered depending on the situation as embedded in $\mathbb{P}^{l^g-1}$ or as a point on the analytic variety $\mathbb{C}^g/\Lambda_\Omega$. For n, $l \in \mathbb{N}$, if n divides l then $\mathbb{Z}(\overline{n})$ will be considered as a subgroup of $\mathbb{Z}(\bar{l})$ via the morphism $x \to (l/n).x$. Let Ξ be the Theta divisor of level l on A, i.e. Ξ is the divisor of zero of $\left(\theta \begin{bmatrix} 0 \\ 0 \end{bmatrix} (z, l^{-1}\dot{\Omega})\right)$. There is an isogeny $\phi_l : A \to \widehat{A} = \mathrm{Pic}^0_A$, defined by $x \to \tau_x^* \Xi_l - \Xi_l$ where τ_x is the translation by x morphism on A. Let $A[l]$ be the kernel of ϕ_l. Let $K(A)$ be the function field of A and (f) be the divisor of a function $f \in K(A)$. We then present the definition of the Weil and theTate pairing.

2.2 Definition and computation of pairings over Theta function

The Weil pairing For $\Omega \in \mathbb{H}_g$, let $A = \mathbb{C}^g/\Lambda_\Omega$ be the complex abelian variety and denote by $\pi : \mathbb{C}^g \to A$ the natural projection. Let l be a positive integer and μ_l be the subgroup of $\mathbb{C}^\star$ of l^{th} roots of unity. For z_P, $z_Q \in \mathbb{C}^g$, let P, Q be the associated points of A. The Weil pairing is the map $e_W : A[l] \times A[l] \to \mu_l$, $(P, Q) \to e_\Omega(z_P, z_Q)^l$. The value $e_W(P, Q)$ does not depend on the choice of z_P and z_Q representing P and Q and e_W is a non-degenerate skew linear form. This pairing can be expressed using certain Theta functions.

Definition 1. *Let $\Omega \in \mathbb{H}_g$, $a, b \in \mathbb{Q}^g$, l be a positive integer and let z_P, $z_Q \in \mathbb{C}^g$ be such that $l.z_P = l.z_Q = 0 \mod \Lambda_\Omega$. Let $P = \pi(z_P)$ and $Q = \pi(z_Q)$. Let*

$$L(z_P, z_Q) = \frac{\theta \begin{bmatrix} a \\ b \end{bmatrix} (l.z_P + z_Q, \Omega)}{\theta \begin{bmatrix} a \\ b \end{bmatrix} (z_Q, \Omega)} \frac{\theta \begin{bmatrix} a \\ b \end{bmatrix} (0, \Omega)}{\theta \begin{bmatrix} a \\ b \end{bmatrix} (l.z_P, \Omega)},$$

$$R(z_P, z_Q) = \frac{\theta \begin{bmatrix} a \\ b \end{bmatrix} (l.z_Q + z_P, \Omega)}{\theta \begin{bmatrix} a \\ b \end{bmatrix} (z_P, \Omega)} \frac{\theta \begin{bmatrix} a \\ b \end{bmatrix} (0, \Omega)}{\theta \begin{bmatrix} a \\ b \end{bmatrix} (l.z_Q, \Omega)}.$$

If $L(z_P, z_Q)$ and $R(z_P, z_Q)$ are well defined and non null, then

$$e_W(P,Q) = L(z_P, z_Q)^{-1}.R(z_P, z_Q) = e_\Omega(z_P, z_Q)^l.$$

The algorithm to compute the Weil pairing is composed of four calls to the function ScalarMult.

Theorem 1. *Suppose that n and l are relatively prime. For X, $Y \in A(\overline{K})$, denote by $\widetilde{X}$, $\widetilde{Y}$, $\widetilde{X+Y}$ any affine lifts of X, Y and $X+Y$. For $i \in \mathbb{Z}(\overline{n})$, let $\widetilde{X_i}$ be the i^{th} coordinate of $\widetilde{X}$. For $\in \mathbb{N}$ and $i \in \mathbb{Z}(\overline{n})$, let*

$$f_T(\widetilde{X}, \widetilde{Y}, \widetilde{X+Y}, \widetilde{0}, l, i) = \frac{\mathit{ScalarMult}(\widetilde{X+Y}, \widetilde{X}, \widetilde{Y}, \widetilde{0}, l)_i}{\mathit{ScalarMult}(\widetilde{X}, \widetilde{X}, \widetilde{0}, \widetilde{0}, l)_i} \frac{\widetilde{0}_i}{\widetilde{Y}_i}.$$

Then for P, $Q \in A[l]$ and $i \in Z(\overline{n})$, we have

$$e_W(P,Q)^n = f_T(\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}, \widetilde{0}, l, i)^{-1} f_T(\widetilde{Q}, \widetilde{P}, \widetilde{P+Q}, \widetilde{0}, l, i),$$

whenever the right hand side is well defined.

The Tate pairing For efficiency reasons, the pairing that will be implemented is the Tate pairing (or a variant of the Tate pairing) so we only consider the side channel attacks against the Tate pairing. Let K be a number field and suppose that A is defined over K. Recall that $l \in \mathbb{N}$ is the level of the Theta function and it is fixed once for all. In this section, we suppose that $\mu_l \subset K$ and that $A[l]$ is rational over K. Let $\overline{K}$ be the algebraic closure of K and $G = \mathrm{Gal}(\overline{K}/K)$. Let $\delta_1 : K^\star/K^{\star l} \to \mathrm{Hom}(G, \mu_l)$ (respectively $\delta_2 : A(K)/[l]A(K) \to \mathrm{Hom}(G, A[l])$) be the connecting morphism of the Galois cohomology long exact sequence associated to the Kummer exact sequence (respectively to the short exact sequence $0 \to A[l] \to A(\overline{K}) \to A(\overline{K}) \to 0$). There exists a bilinear application often referred to as the Tate pairing $e_T : A(K)/[l]A(K) \times A[l] \to K^\star/K^{\star l}$ such that for $(P,Q) \in A(K)/[l]A(K) \times A[l]$, $e_W(\delta_2(P), Q) = \delta_1(e_T(P,Q))$, where e_W is the Weil pairing over Theta functions.

Definition 2. *Let K be a number field and let A be a dimension g abelian variety over K. Let $\Omega \in \mathbb{H}_g$ be such that A is analytically isomorphic to $\mathbb{C}^g/\Lambda_\Omega$. Let a, $b \in \mathbb{Q}^g$ and l be a positive integer. Let $P \in A(K)/[l]A(K)$, $Q \in A[l](K)$ and z_P, $z_Q \in \mathbb{C}^g$ such that $\pi(z_P) = P$ and $\pi(z_Q) = Q$ where $\pi : \mathbb{C}^g \to A$ is the natural projection [1]. Suppose that z_P, z_Q and z_{P+Q} are chosen such that*

$$\frac{\theta\begin{bmatrix}0\\0\end{bmatrix}(z_P + z_Q, \Omega)}{\theta\begin{bmatrix}0\\0\end{bmatrix}(z_P, \Omega)} \frac{\theta\begin{bmatrix}0\\0\end{bmatrix}(0, \Omega)}{\theta\begin{bmatrix}0\\0\end{bmatrix}(z_Q, \Omega)} \in K^\star,$$

then

[1] By abuse of notation we use P, Q to denote the corresponding points of an algebraic and analytic model of A.

$$e_T(P,Q) = \frac{\theta \begin{bmatrix} 0 \\ 0 \end{bmatrix} (l.z_Q + z_P, \Omega)}{\theta \begin{bmatrix} 0 \\ 0 \end{bmatrix} (z_P, \Omega)} \frac{\theta \begin{bmatrix} 0 \\ 0 \end{bmatrix} (0, \Omega)}{\theta \begin{bmatrix} 0 \\ 0 \end{bmatrix} (l.z_Q, \Omega)}.$$

The algorithm for computation of pairings over Theta functions Let n, $l \in \mathbb{N}$ and assume that 2 divides n and that $\gcd(n,l) = 1$. Let A be an abelian variety over $\mathbb{C}$ with period matrix Ω. We represent A as a closed subvariety of $\mathbb{P}^{n^g-1}$ by the way of level n Theta functions and suppose that this embedding is defined over K. Let $\widetilde{A}$ be the pullback of A via the natural projection $\kappa : A^{n^g} \to \mathbb{P}^{n^g-1}$. For $P \in A$, let $\widetilde{P}$ be an affine lift of P that is a point of A^{n^g} such that $\kappa(\widetilde{P}) = P$. Important ingredients of the algorithm in [17] are the Riemann addition formulas. Suppose that the Theta null point $\widetilde{0} = (\theta_i(0))_{i \in \mathbb{Z}(\overline{n})}$ is known. From [17, Theorem 1], we can construct an algorithm that takes as input $\widetilde{P} = \left(\widetilde{P_i}\right)_{i \in \mathbb{Z}(\overline{n})}$, $\widetilde{Q} = \left(\widetilde{Q_i}\right)_{i \in \mathbb{Z}(\overline{n})}$ and $\widetilde{P-Q} = \left((\widetilde{P-Q})_i\right)_{i \in \mathbb{Z}(\overline{n})}$ and outputs $\widetilde{P+Q} = \left((\widetilde{P+Q})_i\right)_{i \in \mathbb{Z}(\overline{n})}$. Let $\widetilde{P+Q} = \mathrm{PseudoAdd}(\widetilde{P}, \widetilde{Q}, \widetilde{P-Q})$. Using the Riemann addition formulas, if $n = 4$, the projective point $P+Q$ can be recovered from P and Q. As a consequence, with the knowledge of $\widetilde{P}$, $\widetilde{Q}$ and $\widetilde{P-Q}$ there is a unique affine point $\widetilde{P+Q}$ above $P+Q$ that satisfies the addition formulas from [17, Theorem 1]. The result is extended in [17] for $n = 2$.

Chaining the algorithm PseudoAdd in a classical Montgomery Ladder yields an algorithm that takes as inputs $\widetilde{Q}$, $\widetilde{P+Q}$, $\widetilde{P}$, $\widetilde{0}$ and an integer l and outputs $\widetilde{P+lQ}$.

Let $\widetilde{P+lQ} = \mathrm{ScalarMult}(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P}, \widetilde{0}, l)$. In particular, $l\widetilde{P} = \mathrm{ScalarMult}(\widetilde{P}, \widetilde{P}, \widetilde{0}, \widetilde{0}, l)$. The output of ScalarMult is independent on the particular chain of PseudoAdd calls it uses.

Theorem 2. *Suppose that n and l are relatively prime. For X, $Y \in A(\overline{K})$, denote by $\widetilde{X}$, $\widetilde{Y}$, $\widetilde{X+Y}$ any affine lifts of X, Y and $X+Y$. For $i \in \mathbb{Z}(\overline{n})$, let $\widetilde{X_i}$ be the i^{th} coordinate of $\widetilde{X}$. For $\in \mathbb{N}$ and $i \in \mathbb{Z}(\overline{n})$, let*

$$f_T(\widetilde{X}, \widetilde{Y}, \widetilde{X+Y}, \widetilde{0}, l, i) = \frac{\mathit{ScalarMult}(\widetilde{X+Y}, \widetilde{X}, \widetilde{Y}, \widetilde{0}, l)_i}{\mathit{ScalarMult}(\widetilde{X}, \widetilde{X}, \widetilde{0}, \widetilde{0}, l)_i} \frac{\widetilde{0}_i}{\widetilde{Y}_i}.$$

Then, for $P \in A(K)/[l]A(K)$, $Q \in A[l]$, if we suppose that $\widetilde{0}$, $\widetilde{P}$, $\widetilde{Q}$ and $\widetilde{P+Q}$ are affine lifts of 0, P, Q and $P+Q$ with coordinates in K, then we have for $i \in \mathbb{Z}(\overline{n})$,

$$e_T(P,Q)^n = f_T(\widetilde{Q}, \widetilde{P}, \widetilde{P+Q}, \widetilde{0}, l, i),$$

whenever the right hand side is well defined.

For example, let E be an elliptic curve defined by $\Omega \in \mathbb{H}_1$ and $\Omega' = \Omega/2$. Put

$$a = \theta \begin{bmatrix} 0 \\ 0 \end{bmatrix} (0, \Omega'); \quad b = \begin{bmatrix} 0 \\ 1/2 \end{bmatrix} (0, \Omega'); \quad \mathcal{A} = \theta \begin{bmatrix} 0 \\ 0 \end{bmatrix} (0, 2\Omega'); \quad \mathcal{B} = \theta \begin{bmatrix} 1/2 \\ 0 \end{bmatrix} (0, 2\Omega').$$

The algorithm ScalarMult is composed by a doubling algorithm and a differential addition algorithm given in Figure 1.

```
Doubling Algorithm
Input: A point $P = (x_P : z_P)$.
Output: The double $2P = (x_{2P} : z_{2P})$
1. $x_0 = (x_P^2 + z_P^2)^2$
2. $z_0 = \frac{\mathcal{A}^2}{\mathcal{B}^2}(x_P^2 - z_P^2)^2$
3. $x_{2P} = x_0 + z_0$
4. $z_{2P} = \frac{a}{b}(x_0 - z_0)$
5. Return $(x_{2P} : z_{2P})$
```

```
Differential Addition Algorithm
Input: Two points $P = (x_P : z_P)$ and
$Q = (x_Q, z_Q)$ on $E$, $R = (x_R : z_R) = P - Q$,
with $x_R z_R \neq 0$.
Output: The point $P + Q = (x_{P+Q} : z_{P+Q})$
1. $x_0 = (x_P^2 + z_P^2)(x_Q^2 + z_Q^2)$
2. $z_0 = \frac{\mathcal{A}^2}{\mathcal{B}^2}(x_P^2 - z_P^2)(x_Q^2 - z_Q^2)$
3. $x_{P+Q} = (x_0 + z_0)/x_R$
4. $z_{P+Q} = (x_0 - z_0)/z_R$
5. Return $(x_{P+Q} : z_{P+Q})$
```

Fig. 1. Doubling and Differential Addition Algorithms

3 Side Chanel Attacks against the Tate pairing over Theta function

3.1 Side channel attacks in pairing based cryptography

The general scheme of an identity based encryption is recalled in [9]. The important point is that to decipher a message using an Identity Based Protocol, a computation of a pairing between a private key and a public message is performed. Side channel attacks are powerful attacks using the leakage of information during the execution of a cryptographic protocol. As soon as the algorithm involves a computation between a secret and a public data, side channel attacks can be applied in order to reveal the secret, or information about the secret. The particularity of identity based cryptography is that an attacker can know the algorithm used, the number of iterations and the exponent. The secret is only one of the arguments of the pairing. We describe here two attacks, namely the Differential Power Analysis (DPA) and the fault attack. There are other side channel attacks, but the popular ones are either a generalization of the DPA (DEMA, CPA) or fault attacks.

3.2 The possible targets

If we compare the efficiency of the Tate and of the Weil pairings, the former is more efficient than the later at least for the security levels considering today, when pairings are computed using a Miller's algorithm. In the case of Theta functions, the algorithmic complexity of the Tate pairing consists in two applications of the function ScalarMult, while the Weil pairing consists in four applications of this function described in Appendix 4. It is quite evident that the Tate pairing over Theta function will always be more efficient than the Weil pairing over the Theta function. So we study only the weakness of the Tate pairing considering side channel attacks. Nevertheless, the attacks described for the Tate pairing can easily be adapted to the Weil pairing. As a consequence the countermeasure proposed here must be considered also for the implementation of the Weil pairing.

The Tate pairing is composed of two applications of ScalarMult. First of all, we focus on side channel attacks against one application of ScalarMult and after that we will consider side channel attacks against the Tate pairing. The same argument can provide the result of side channel attacks against the Weil pairing, or any optimizations of the Tate pairing namely Ate, twisted Ate or optimal pairings. The function ScalarMult is a Montgomery Ladder composed by the Doubling and Differential Addition algorithms at each step. When the secret is the exponent this algorithm is an efficient countermeasure to side channel attacks. In the case of pairing based cryptography, the secret is not the exponent but one of the parameters of the Mongtomery Ladder algorithm. Consequently, the analysis considering side channel attacks against Montgomery's Ladder for the classical use in cryptography (efficient exponentiation) is no more available. We analyze the weaknesses of the algorithm to compute pairing using Theta functions. We will focus on the DPA and on the fault attack. The consideration of the DPA includes also the consideration of the Correlation Power Analysis (CPA) and the Differential Electromagnetic Attack (DEMA). Indeed the DEMA works exactly like the DPA and the CPA is an improvement of the DPA.

3.3 Differential Power Analysis Attack and generalization

In order to simplify the explanation, we describe here only the differential power analysis (DPA) attack. As the concept is the same for all differential attacks, we include in the same family the differential power analysis (DPA) and the differential electromagnetic attack (DEMA) [4]. Further on, correlation attack [22] is just a form of DPA using the particular side-channel distinguisher i.e. Pearson correlation.

We now introduce some theoretical issues that allow the reader to understand the principle underlying the DPA attack, more details can be found in [15, 18]. We consider the output of a gate whose state depends on both the plain text to be ciphered (primary inputs) and the secret key. It is called the target node. We consider now a sequence of input patterns $P_0, P_1, \ldots, P_n$ that generate the transitions $T_1(P_0 \to P_1), T_2(P_1 \to P_2), \ldots, T_n(P_{n-1} \to P_n)$ on the circuit primary inputs. A logic simulation of the circuit while monitoring the target node allows classifying these input transitions in two sets, according to a guess on the key:

- P_A, composed by the transitions that make the target node to commute from 0 to 1 and therefore that make the target gate to consume current;
- P_B, composed by the transitions that do not lead the target gate to participate to the power consumed by the circuit (i.e., transitions from 0 to 0, 1 to 1 and 1 to 0 on the target node).

Figure 2 represents the power consumption of the device when stimulated by numerous input vectors. We assume here that the guess on the secret key is correct. In other word, the simulation is performed with the key actually used in the circuit from which power consumptions are collected. Each rectangle represents the total power consumed by the circuit when a new vector is applied to the inputs. In this figure and just for clarity of explanation, the power consumption is represented by a rectangle corresponding to the average of the consumption over the transition time. The set of transitions on the circuit inputs is splitted in the two sets: in the left part there are the P_A transitions and the related consumptions while in the right part there are the P_B transitions and their corresponding consumptions. A part of the power consumption related to the transitions belonging to P_A is due to the power consumed by the target gate (shaded rectangles). Obviously, the commutation from 0 to 1 of non-target nodes also contributes to the power consumption of the circuit but input transitions that lead to such commutations are assumed to be evenly distributed to sets P_A and P_B. If a large number of transitions are considered, mean consumptions related to sets P_A and P_B are almost equal, except for the contribution of the target node. In other words, since the two sets are classified in such a way that the set P_A always leads to a component of power consumption that is not present in the set P_B, the difference between the two mean powers computed from set P_A and set P_B must show a noticeable difference.

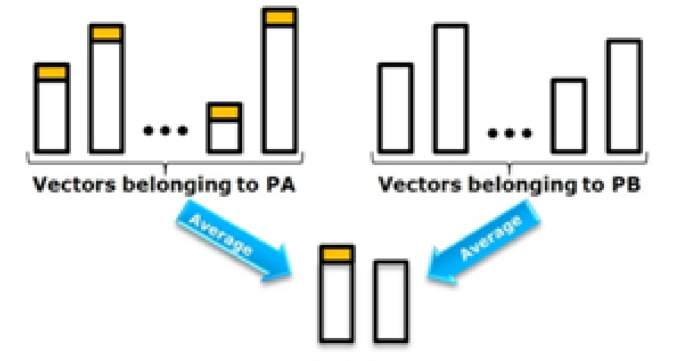

Fig. 2. Power consumption after pattern partitioning

During a DPA attack, the target node is chosen in such a way that it depends on a small part of the key only, so that all the key guesses can be considered.

For each key guess, the two sets P_A and P_B are created according to the results of the logic simulation and the key guess under evaluation. The power mean values are calculated for each set using the simulated power traces of the circuit under attack for each transition. Finally, the differences of the mean values of the two sets are calculated. When the key guess is correct (and only in this case), P_A actually includes the input transitions that lead to a transition 0 to 1 on the target node while P_B does not include any of these transitions. The difference between the mean power obtained from P_A and P_B can be observed in this case. On the contrary, when the curves are classed in P_A or P_B independently from the actual value of the secret key, the two average curves do not present any noticeable difference. The classification process is illustrated in Figure 3 where K_x is assumed to be the correct key, the one actually used during ciphering.

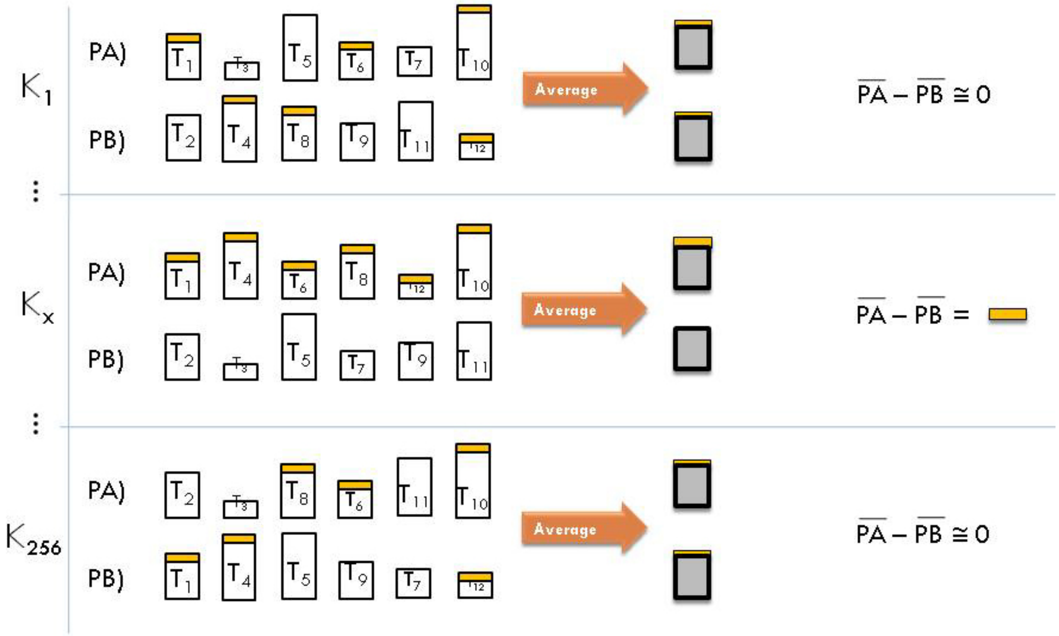

Fig. 3. Pattern classification for several key guesses

3.4 DPA attack

The computations sensitive to the DPA attack are the one involving the coordinates of the points P and Q. As a consequence, the DPA attack could only be done in the addition step. Indeed, the doubling step does not involve any computation between the coordinates of P and Q, the operations are multiplications by constant.

Without protection, the DPA attack is a threat against the addition step, whenever the secret is (point P or Q). According to the algorithm of ScalarMult, the argument Q of the pairing is in fact a multiple of the point Q and the point P is fixed.

The target of the DPA attack is the computation of x_0 and z_0 in the differential addition algorithm. In order to compute x_0, we have to perform a multiplication between $(x_P^2 + z_P^2)$ and $(x_Q^2 + z_Q^2)$. Suppose that the point P is public and that Q is secret, we know the value of $(x_P^2 + z_P^2)$ and the value $(x_Q^2 + z_Q^2)$ is secret. We perform the DPA attack against the multiplication $(x_P^2 + z_P^2) \times (x_Q^2 + z_Q^2)$. Assuming that the multiplication is implemented using the Schoolbook method, the guesses on the value $(x_Q^2 + z_Q^2)$ can be done by words of 32 or 64 bits and begin by the less significant bits. The result of the DPA attack against x_0 is $(x_Q^2 + z_Q^2)$.

During the computation of x_0 the Differential Addition Algorithm would give us the value $\widetilde{x_Q}^2 + \widetilde{z_Q}^2$. In parallel, another DPA attack during the computation of z_0 would give the value $\widetilde{x_Q}^2 - \widetilde{z_Q}^2$. Once we have these two values, it is easy to extract $\widetilde{x_Q}^2$ and $\widetilde{z_Q}^2$, which gives 4 possible couples for the coordinates of point Q.

As the Differential Addition Algorithm is symmetric in the coordinates of P and Q, the same attack is efficient if the point P is secret and Q is public.

Classical countermeasures presented in the case of pairing over elliptic curves can easily be adapted for the computation of pairings over Theta functions [8]. A native countermeasure is the homogeneity of the projective coordinates. Indeed, the point $P = (x_P : z_P)$ is also the point $(\lambda x_P : \lambda z_Q)$, for λ a non zero integer and $Q = (\beta x_Q, \beta y_Q)$, for β a non zero integer. The main hypothesis of a DPA attack is that the secret is the same for several executions of the algorithm. So, if we modify the coordinates of the secret for each execution the DPA attack can no longer be performed.

An efficient countermeasure consists in multiplying the coordinates of P by a random non zero integer λ and the coordinates Q by a random β for every iteration of the ScalarMult algorithm. This countermeasure is a good protection against any differential attack (power or electromagnetic) and consequently a protection for the Tate (and the Weil) pairing.

3.5 Description of the fault attack

The goal of a fault injection attack is to provoke mistakes during the calculation of an algorithm, for example by modifying the internal memory, in order to reveal sensitive data. This attack needs a very precise positioning and an expensive apparatus to be performed. Nevertheless, current technologies could allow for this attack [10]. the faults can be performed using a laser or electromagnetic emissions [4].

We follow the scheme of attack described in [20] and completed in [6]. We assume that the pairing is used during an Identity Based Protocol, the secret point is introduced in a smart card or an electronic device and is a parameter of the pairing. In order to find the secret, we modify the number of iterations in the Tate pairing algorithm by the following way.

First of all, we have to find the flip-flops belonging to the counter of the number of iterations (i.e. $\log_2(s)$) in the Tate pairing algorithm. This step can be done by using reverse engineering procedures. In classical architecture, the counter is divided into small pieces of 32 or 64 bits (according to the size of a word). To find it, we make one normal execution of the algorithm, without any fault. Then we choose one piece of the counter and provoke disturbances in order to modify it and consequently the number of iterations of the algorithm. For example the disturbance can be induced by a laser [1]. Nowadays lasers are thin enough to make this attack realistic [10]. Counting the clock cycles, we are able to know how many iterations the Tate pairing loop has done. Each time, we record the value of the pairing loop and the number of iterations we made.

3.6 Fault attack

State of the art The principle of fault attacks in pairing based cryptography consists to force the algorithm to stop by reducing the number of iterations and by finding the results of two consecutive iterations τ and $\tau+1$. The results of these two executions give equations that allow to find the secret. In the case of pairings over Theta function, the fault attack consists in finding one the coordinates involved during the computation of ScalarMult$(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P})$. The ScalarMult algorithm is composed by the doubling and differential addition algorithms, the result of of ScalarMult are the coordinates of $\widetilde{P+lQ}$. The fault attack consists in reducing the number of iteration of ScalarMult. To do so, we can use a laser or electromagnetic emissions to locally modify the register storing l. The target of this attack is then a smart card or a FPGA. Let τ be the reduced number of iteration performed by ScalarMult. In practice τ can be recovered using the number of clock cycles made by the algorithm. Indeed, we know the binary decomposition of l, we are then able to find when the algorithm stops and how many iterations were done. Let j be the integer composed by the τ most significant bits of l, which is public. The fault attack for pairing over Theta function is easier than the classical fault attack in pairing based cryptography. We need only one fault and the result of this faulty execution to find the secret involved in the ScalarMult algorithm

The result of the pairing is the coordinates of the point $\widetilde{P+lQ}$. We can suppose that we obtain one of the two coordinates, for example the coordinate z. With the z coordinate of the result, we are able to recover the secret argument of the pairing computations.

Suppose that we can recover the coordinate z of the point $\widetilde{P+jQ}$, for $j < l$. As the points P and Q are of order l by construction, the result of the pairing itself cannot give us information. That is why we need to provoke a fault reducing the number of iterations of the ScalarMult algorithm.

Let $z_1 = z_{P+jQ}$, where j is a known integer. The equation of z_1 is the following

$$z_1 = \left[(x_j^2 + z_j^2)(x_P^2 + z_P^2) - \frac{\mathcal{A}^2}{\mathcal{B}^2}(x_j^2 - z_j^2)(x_P^2 - z_P^2) \right] \frac{1}{\overline{z}}, \qquad (1)$$

where

- $P = (x_P, z_P) = (\overline{x}, \overline{z})$ (with the notations introduced above)
- $(j-1)Q = (x_j, z_j)$
- $P + jQ = (x_1, z_1)$
- $\mathcal{A}$ and $\mathcal{B}$ are constants.

We first describe the attack of the algorithm ScalarMult, before considering the fault attack against the whole Tate pairing algorithm.

If the secret is the point $\boldsymbol{P}$ Suppose that the point P is secret. The fault attack provide us z_1, the values A, B, x_j and z_j are public. All together, they verify the equation

$$\boldsymbol{\lambda} z_P = \boldsymbol{\beta}(x_P^2 + z_P^2) + \boldsymbol{\gamma}(x_P^2 - z_P^2),$$

where the data in bold $(\boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ are known. The coordinates x_P and z_P are the values we are looking for.

The point P is given in projective coordinates, this equality is correct for any representative of the point P, i.e. for any $\alpha \neq 0$ we have that

$$\boldsymbol{\lambda}(\alpha z_P) = \boldsymbol{\beta}((\alpha x_P)^2 + (\alpha z_P)^2) + \boldsymbol{\gamma}((\alpha x_P)^2 - (\alpha z_P)^2).$$

As the coordinates of P are such that $x_P z_P \neq 0$, we can consider that $\alpha = \frac{1}{z_P}$ and write the equation

$$\boldsymbol{\lambda} = \boldsymbol{\beta}((x'_P)^2 + 1) + \boldsymbol{\gamma}((x'_P)^2 - 1),$$

which leads to

$$(x'_P)^2 = \frac{\boldsymbol{\lambda} - \boldsymbol{\beta} + \boldsymbol{\gamma}}{\boldsymbol{\beta} - \boldsymbol{\gamma}}.$$

Up to the sign, we find one coordinate of a representative of the point P and from that point we can find the secret.

If the secret is the point $\boldsymbol{Q}$ The formulas are symmetric in the coordinates of P and jQ. Following the same scheme, we obtain z_1 for j not equal to the order of Q and that gives the coordinates of a representative of jQ, knowing j. To find the coordinates of Q, we just have to compute the inverse of j mod (l) and after that we can recover the coordinates of the point Q.

The condition to perform the fault attack when Q is secret is to stop the computation before $j = l$, as Q is a point of order l. This is a simplification of the fault attack against the pairing considering Miller's algorithm, because we only need one faulty execution of ScalarMult.

Considering the computation of the Tate pairing Recall that the algorithm to compute the Tate pairing is

$$e_T = \frac{\mathrm{ScalarMult}(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P}, l)_i}{\mathrm{ScalarMult}(\widetilde{Q}, \widetilde{Q}, \widetilde{0}, l)_i} \frac{\widetilde{0}_i}{\widetilde{P}_i}.$$

The attacks described above for ScalarMult can be directly adapted to the Tate pairing (and also to the Weil pairing). For efficiency reasons, the computation of ScalarMult$(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P}, l)_i$ and ScalarMult$(\widetilde{Q}, \widetilde{Q}, \widetilde{0}, l)_i$ would certainly be implemented in parallel. As a consequence, the fault attack forces the algorithm to stop after the same number of iterations and the result ScalarMult$(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P}, j)_i$ and ScalarMult$(\widetilde{Q}, \widetilde{Q}, \widetilde{0}, j)_i$, for the same integer j. For both cases, either P secret or either Q, the homogeneity of projective coordinates is a trapdoor that gives information about the secret. Let P be the secret point and Q be public, then the coordinate $\widetilde{P}_i$ is also secret, but the homogeneity of the projective coordinates allows us to consider that for example the z coordinate is set to 1, exactly like in the attack described above. We just have to be careful to set the same coordinate to 1 in both calls to ScalarMult, the z one for example. The Equation (1) would give a slightly different system but linear and easily solvable. The method is the same if the point Q is secret.

Countermeasure to the fault attack Considering that the fault attack uses the homogeneity of the coordinates, the countermeasure to the DPA attack is clearly not sufficient. We have to present another countermeasure and this countermeasure must protect the pairing algorithm from the fault and the DPA attacks. So, we have to modify the coordinates of the point P and Q for every pairing computation. A solution would be to use the bilinearity of the pairing [8]. Indeed, if we compute the Tate pairing between the points P and Q, the bilinearity induces that

$$e_T(P,Q) = e_T\left(\delta P, (\delta^{-1} \mod (l))Q\right),$$

for a non zero integer δ. The cost of this countermeasure consists in two exponentiations over the variety $A(K)$.

4 Conclusion

We analyze the weaknesses of the pairings over Theta function with respect to side channel attacks. We consider the differential power analysis and the fault attack. The scheme of the differential power analysis embraces the differential electromagnetic attack and the correlation power analysis. The ScalarMult algorithm is sensitive to the DPA attack, but the homogeneity of the projective coordinates provides a native countermeasure. Unfortunately, the homogeneity is a trapdoor for the fault attack. The fault attack against pairing over Theta functions is easier than in the case of pairings using the Miller's algorithm. We only need one fault to recover the secret. As the homogeneity of the coordinates is no longer a countermeasure, we present an alternative countermeasure. This countermeasure relies on the bilinearity of pairings and is efficient for all side channel attacks.

Acknowledgment

The author wishes to thank the anonymous referees for their helpful remarks and comments.

References

1. Ross Anderson and Markus Kuhn. Tamper resistance: a cautionary note. In *WOEC'96: Proceedings of the 2nd conference on Proceedings of the Second USENIX Workshop on Electronic Commerce*, pages 1–11, Berkeley, CA, USA, 1996. USENIX Association.
2. Ray C. C. Cheung, Sylvain Duquesne, Junfeng Fan, Nicolas Guillermin, Ingrid Verbauwhede, and Gavin Xiaoxu Yao. Fpga implementation of pairings using residue number system and lazy reduction. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 421–441. Springer, 2011.
3. Craig Costello, Tanja Lange, and Michael Naehrig. Faster pairing computations on curves with high-degree twists. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 224–242. Springer, 2010.
4. Elke De Mulder, Sidika Bernard Örs, Bart Preneel, and Ingrid Verbauwhede. Differential power and electromagnetic attacks on a FPGA implementation of elliptic curve cryptosystems. *Comput. Electr. Eng.*, 33(5-6):367–382, 2007.

5. Sylvain Duquesne and Emmanuel Fouotsa. Tate pairing computation on jacobi's elliptic curves. In Michel Abdalla and Tanja Lange, editors, *Pairing-Based Cryptography - Pairing 2012*, volume 7708 of *Lecture Notes in Computer Science*, pages 254–269. Springer, 2012.
6. Nadia El Mrabet. What about vulnerability to a fault attack of the miller's algorithm during an identity based protocol? In Jong Hyuk Park, Hsiao-Hwa Chen, Mohammed Atiquzzaman, Changhoon Lee, Tai-Hoon Kim, and Sang-Soo Yeo, editors, *Advances in Information Security and Assurance ISA 2009*, volume 5576 of *Lecture Notes in Computer Science*, pages 122–134. Springer, 2009.
7. Nadia El Mrabet, Giorgio Di Natale, and Marie-Lise Flottes. A practical differential power analysis attack against the miller algorithm. In *PRIME 2009 - 5th Conference on Ph.D. Research in Microelectronics and Electronics, Circuits and Systems Magazine*, IEEE Xplore, 2009.
8. Nadia El Mrabet, Dan Page, and Frederik Vercauteren. Fault attacks on pairing based cryptography: A state of the art. In *Fault Analysis in Cryptography*, Information Security and Cryptography, pages 221–236. Springer, Mark Joye and Michael Tunstall edition, 2012.
9. Steven Galbraith. *Pairings in Advances in Elliptic Curve Cryptography, F. Blake and G. Seroussi and N. Smart editors.* London Mathematical Society Lecture Note Series (No. 317),Cambridge University Press, 2005.
10. Donald Habing. The use of lasers to simulate radiation-induced transients in semiconductor devices and circuits. In *IEEE Transactions On Nuclear Science*, volume 39, pages 1647–1653, 1992.
11. Florian Hess. Pairing Lattices. In Steven Galbraith and Kenny Peterson, editors, *Pairing 2008*, volume 5209 of *Lectures Notes in Computer Science*, pages 18–38, Berlin, Heidelberg, 2008. Springer-Verlag.
12. Florian Hess, Nigel Smart, and Frederik Vercauteren. The Eta Pairing Revisited. In *IEEE Transactions on Information Theory*, volume 52, pages 4595–4602, 2006.
13. Sorina Ionica and Antoine Joux. Another approach to pairing computation in Edwards coordinates. In *INDOCRYPT '08: Proceedings of the 9th International Conference on Cryptology in India*, pages 400–413, Berlin, Heidelberg, 2008. Springer-Verlag.
14. Marc Joye, Mehdi Tibouchi, and Damien Vergnaud. Huff's model for elliptic curves. In Guillaume Hanrot, François Morain, and Emmanuel Thomé, editors, *ANTS 2010*, volume 6197 of *Lecture Notes in Computer Science*, pages 234–250. Springer, 2010.
15. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.
16. Kristin Lauter, Peter L. Montgomery, and Michael Naehrig. An analysis of affine coordinates for pairing computation. In Marc Joye, Atsuko Miyaji, and Akira Otsuka, editors, *Pairing*, volume 6487 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2010.
17. David Lubicz and Damien Robert. Efficient pairing computation with theta functions. In *Algorithmic Number Theory, 9th International Symposium, ANTS-IX, Nancy, France, July 19-23, 2010. Proceedings*, volume 6197 of *Lecture Notes in Computer Science*, pages 251–269. Springer, 2010.
18. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *DPA book.* Graz University of Technology, 2007.
19. Victor Miller. The Weil pairing, and its efficient calculation. In *Jounal of Cryptology*, volume 17, pages 235–261, Secaucus, NJ, USA, 2004. Springer-Verlag New York, Inc.
20. Dan Page and Frederik Vercauteren. A fault attack on pairing-based cryptography. volume 55, pages 1075–1080, 2006.
21. Frederik Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, 2010.
22. Claire Whelan and Michael Scott. Side channel analysis of practical pairing implementation: Which path is more secure? In *VietCrypt 2006*, Lecture Notes in Computer Science, 2006.
23. Claire Whelan and Michael Scott. The importance of the final exponentiation in pairings when considering fault attacks. In *Pairing-Based Cryptography in Pairing 2007*, volume 4575 of *Lecture Notes in Computer Science*, pages 225–246, 2007.