Failure of the Point Blinding Countermeasure Against Fault Attack in Pairing-Based Cryptography

Nadia El Mrabet, Emmanuel Fouotsa

Failure of the Point Blinding Countermeasure against Fault Attack in Pairing-Based Cryptography ⋆

Nadia EL MRABET[1] and Emmanuel FOUOTSA[2]

(1) LIASD − Université Paris 8 -France - elmrabet@ai.univ-paris8.fr
(1) SAS - CMP Gardanne - France - nadia.el-mrabet@emse.fr
(2) Dep of Mathematics, Higher Teacher's Training College, University of Bamenda - Cameroun
(2) LMNO − Université de Caen - France - emmanuelfouotsa@yahoo.fr

Abstract. Pairings are mathematical tools that have been proven to be very useful in the construction of many cryptographic protocols. Some of these protocols are suitable for implementation on power constrained devices such as smart cards or smartphone which are subject to side channel attacks. In this paper, we analyse the efficiency of the point blinding countermeasure in pairing based cryptography against side channel attacks. In particular,we show that this countermeasure does not protect Miller's algorithm for pairing computation against fault attack. We then give recommendation for a secure implementation of a pairing based protocol using the Miller algorithm.
Key words: Miller's algorithm, Identity Based Cryptography, Side Channel Attacks, Fault Attacks, Countermeasure.

1 Introduction

Pairings are bilinear maps defined on the group of rationals points of elliptic or hyper elliptic curves [36]. Nowadays, more and more protocols using pairings are proposed in the literature [10, 21, 6]. Among these protocols, only those constructed on the identity based model involve a secret which is one of the argument during the computation of a pairing. The implementation of a pairing based protocol is efficient enough to allow the use of pairing based cryptography on power constrained device such as smart cards and mobile phones [31, 22, 19]. Smart cards are by nature sensitive to side channel attacks. Side channel attacks are powerful attacks that use the implementation of a protocol to obtain information on the secret. They are divided into two families: invasive and non invasive attacks. Invasive attacks are based on the model of fault attacks. The execution of a protocol is disturbed, the result is then a faulty one and the analysis of this faulty result can provide information on the secret. In non invasive attacks, the information can be leaked by the time of execution, the electric consumption or the electromagnetic emission of the device. Several works have investigated the robustness of identity based cryptography to side channel attacks. They are mainly focused on fault attacks [27, 37, 11, 2]. Few works consider differential power analysis attack [27, 13, 5]. As the secret during an identity based protocol can be recovered by side channel attacks, several countermeasures were proposed. Those countermeasures are the same for invasive and non invasive attacks [14]. In [16], Ghosh, Mulhopadhyay and Chowdhury proposed an analysis of countermeasures to fault attack presented in [27]: the new point blinding method and the alliterating point blinding method. They concluded that the countermeasures are not sufficient and proposed new one. However,

⋆ This work was supported in part by the French ANR-12-INSE-0014 SIMPATIC Project. The second author is supported by The Simons Foundations through Pole of Research in Mathematics with applications to Information Security, Subsaharan Africa

their explanations on the non efficiency of the countermeasure are not convincing. Later, Park et al. [28] clearly exposed the weaknesses of the point blinding technique against fault attacks described by Page and Vercauteren [27].

In this article we analyze and extend the work in [16, 28] on the efficiency of the point blinding countermeasure in pairing based cryptography. Especially, we generalize the attack of Park et al. [28] and expose its failure to protect the Miller algorithm, main tool in pairing computation. As the most efficient pairings are constructed on the model of the Tate pairing, we focus on the Miller algorithm, used for the Tate pairing considering Weierstrass elliptic curve. Obviously, this analysis is the same for the (optimal) Ate, twisted Ate or pairing lattices; and for every model of elliptic curve or coordinates.

The rest of this paper is organized as follows: The Section 2 presents brief concepts on pairings that are useful to understand this work. In Section 3 we present side channel attacks with emphasis on fault attacks in pairing based cryptography. In Section 4 we explicitly demonstrate that the point blinding countermeasure fails to protect the Miller algorithm against fault attack. Finally we conclude the work in Section 6.

2 Background on pairings

In this section, we briefly recall basics on pairings and on the Miller algorithm [25], main tool for an efficient computation of pairings. Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$, with q a prime number or a power of a prime. The neutral element of the additive group law defined on the set of rational points of E is denoted P_∞. Let r be a large prime divisor of the group order $\sharp E(\mathbb{F}_q)$ and k the embedding degree of E with respect to r, i.e. the smallest integer k such that r divides $q^k - 1$. The integer k is also the smallest integer such that $E\left(\overline{\mathbb{F}_q}\right)[r] \subset E(\mathbb{F}_{q^k})$, where $E\left(\overline{\mathbb{F}_q}\right)[r] = \{P \in E\left(\overline{\mathbb{F}_q}\right) : [r]P = P_\infty\}$ with $[r]P = \underbrace{P + P + \ldots + P}_{r \text{ times}}$ and $\overline{\mathbb{F}_q}$ is the algebraic closure of $\mathbb{F}_q$.

In general , the sizes of r, q and k are dependent from the security level and the currently recommendations are at least $r > 2^{160}$ and $q^k > 2^{2024}$ [15]. The recent results for the discrete logarithm problem [20, 3] imply that the number q must be a large prime number. The security recommendations allow the choice of k to be a product of power of 2 and 3. A consequence of the fact that $k \equiv 0 \mod 2$ is the use of a twist representation for the point Q. This representation using a twisted elliptic curve allow the denominator elimination optimization [23].

Definition of a twisted elliptic curve. We explain here the concept of twist of elliptic curve in the context of Weierstrass elliptic curve. This will help us to understand the choice of the coordinates of points in Section 4. The quadratic twist of the elliptic curve $E : y^2 = x^3 + ax + b$ over $\mathbb{F}_{p^k}$ is the elliptic curve $\tilde{E} : \frac{1}{\nu}y^2 = x^3 + ax + b$ where $\{1, \nu\}$ is a basis of $\mathbb{F}_{q^k}$ as $\mathbb{F}_{q^{k/2}}$ vector space. The two curves are isomorphic via

$$\begin{aligned} \psi : \tilde{E}(\mathbb{F}_{q^{k/2}}) &\longrightarrow E(\mathbb{F}_{q^k}) \\ (x, y) &\longmapsto (x, y\sqrt{\nu}). \end{aligned}$$

This isomorphism is particularly useful since it enables to take the point $Q \in E(\mathbb{F}_{q^k})$ in the following manner $Q = \psi(Q')$ where $Q' = (x_Q, y_Q)$ with $x_Q, y_Q \in \mathbb{F}_{q^{k/2}}$. This ensures an

efficient computation since many computations will be consequently done instead in the subfield $\mathbb{F}_{q^{k/2}}$ and more interestingly, it enables to avoid the inversions in the Miller algorithm. This elimination is the denominator elimination [23].

Indeed, if $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ are two points of the elliptic curve in Weierstrass form $E : y^2 = x^3 + ax + b$ then the function h_{P_1,P_2} with divisor

$$\mathrm{Div}(h_{P_1,P_2}) = (P_1) + (P_2) - (P_1 + P_2) - (P_\infty),$$

is $h_{P_1,P_2} = \frac{\ell_{P_1,P_2}}{v_{P_1+P_2}}$ where ℓ_{P_1,P_2} is the straight line defining P_1+P_2 and $v_{P_1+P_2}$ is the corresponding vertical line passing through $P_1 + P_2$. Explicitly, we have

$$h_{P_1,P_2}(x, y) = \frac{y - \lambda x - \alpha}{x - x_3},$$

where x_3 is the first coordinate of $P_1 + P_2$ and $\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$ if $P_1 \neq P_2$, $\lambda = \dfrac{3x_1^2 + a}{2y_1}$ if $P_1 = P_2$ and $\alpha = y_1 - \lambda x_1$.

In the particular case of doubling ($P_1 = P_2$), a straightforward computation gives, after changing to Jacobian coordinates ($x_1 = \frac{X_1}{Z_1^2}$, $y_1 = \frac{Y_1}{Z_1^3}$)

$$h_{P_1,P_1}(Q) = h_{P_1,P_2}(x_Q, y_Q\sqrt{\nu}) = \frac{2Y_1Z_1^3y_Q\sqrt{\nu} - 2Y_1^2 - (3X_1^2 + aZ_1^4)(x_QZ_1^2 - X_1)}{2Y_1Z_1^3(x_Q - x_3)},$$

We then remark that the denominator of the previous expression is an element of $\mathbb{F}_{q^{k/2}}$ and consequently will be equal to 1 during the final exponentiation. So the main expression that will be used in the Miller algorithm is:

$$h_{P_1,P_1}(Q) = h_{P_1,P_2}(x_Q, y_Q\sqrt{\nu}) = 2Y_1Z_1^3y_Q\sqrt{\nu} - 2Y_1^2 - (3X_1^2 + aZ_1^4)(x_QZ_1^2 - X_1) \qquad (1)$$

The expression given by equation 1 is used in algorithms 1 and 2 and will be particularly useful in Section 4 to illustrate our attack.

The Tate pairing. Consider a point $P \in E(\mathbb{F}_q)[r]$, the principal divisor $D = r(P) - r(P_\infty)$ and a function $f_{r,P}$ with divisor Div $(f_{r,P}) = D$. Let $Q \in E(\mathbb{F}_{q^k})[r]/E(\mathbb{F}_q)$ and μ_r be the group of r-th roots of unity in $\mathbb{F}_{q^k}^*$. The reduced Tate pairing e_r is a bilinear and non degenerate map defined as

$$\begin{aligned} e_r : E(\mathbb{F}_q)[m] \times E(\mathbb{F}_{q^k})[r] &\to \mu_m \\ (P, Q) &\mapsto f_{r,P}(Q)^{\frac{q^k-1}{r}} \end{aligned}$$

The value $f_{r,P}(Q)$ can be determined efficiently using Miller's algorithm [25].

More information on pairings can be found in [9]. In order to obtain the result of the Tate pairing, the output of Miller's algorithm must be raised to the power $\frac{q^k-1}{r}$, this operation is called the final exponentiation.

Algorithm 1: Miller's Algorithm

Input : $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r]$, $m = (1, m_{n-2}, ...m_1, m_0)_2$.
Output: $f_{m,P}(Q)$

1: Set $f \leftarrow 1$ and $T \leftarrow P$
2: **For** $i = n - 2$ **down to** 0 **do**
3: $f \leftarrow f^2 \cdot h_{T,T}(Q)$, with $h_{T,T}$ the Equation (1) of the tangent to E at point T
4: $T \leftarrow 2T$
5: **if** $m_i = 1$ **then**
6: $f \leftarrow f \cdot h_{T,P}(Q)$, with $h_{T,P}$ the equation of the line (PT)
7: $T \leftarrow T + P$
8: **end if**
9: **end for**
10: **return** f

Fig. 1. The Miller algorithm

We call a Tate-like pairing any pairing constructed on the following model: an execution of the Miller algorithm followed by a final exponentiation. Every Tate-like pairing was an improvement of the previous. The ate pairing [18] was an improvement of the Tate pairing [29], the twisted ate pairing [18] an improvement of the ate pairing, the notion of optimal pairings [35] an improvement of the ate and twisted ate pairing and finally the pairing lattices [17] another way to deal with optimal pairings. The algorithmic difference between the Tate pairing and a Tate-like pairing is principally the number of iterations, sometimes it could also be the role playing by P and Q. In Algorithm 1, we describe the Miller algorithm. In order to keep our explanations general , the number of iterations in the Miller algorithm is indexed over m. The integer m would be r for the Tate pairing, or smaller than r for a Tate-like pairing. We describe the attack considering that we are computing a pairing using $f_{m,P}(Q)$, for m the integer giving the number of iterations of the pairing. Obviously, the discussion can be straightforward adapted for the computation of $f_{m,Q}(P)$.

Obviously, the system of coordinates influences the equations of the Miller algorithm, but if the attack is efficient over one model of elliptic curve for one system of coordinates, then the same attack will be efficient over any other model of elliptic curve and considering any other system of coordinates.

3 Side channel attacks on Pairing-Based cryptography and Countermeasures

In this section we briefly recall and describe existing side channel attacks and countermeasures in the context of pairing-based cryptography. Especially, we analyse the point blinding countermeasure presented in [27] and its weakness exposed in [28].

3.1 Background on side channel attacks

The first analysis of side channel attacks against a pairing was proposed by Page and Vercauteren [27]. They attack the Duursma and Lee algorithm used to compute a pairing over super singular elliptic curves. Page and Vercauteren described a new fault attack model and mention without development the differential power analysis against pairings. The fault model consists in the modification of the number of iterations of an algorithm. The fault attack was

adapted by further works on the Miller algorithm [37, 11, 2]. Whelan et Scott [37] highlighted the fact that pairings without a final exponentiation are more sensitive to a sign change fault attack. They analyzed the Weil, the Tate and Eta pairing. They used a simplified version of Page and Vercauteren attack. After that, El Mrabet [11] generalized the attack of Page and Vercauteren to the Miller algorithm used to compute all the recent optimizations of pairings. El Mrabet considered only the Miller algorithm and did not take into account the final exponentiation. The target of El Mrabet's attack is the loop counter in the Miller algorithm. The final exponentiation was attacked by Lasherme et al. [24]. They used three faults to inverse the final exponentiation of the Tate pairing, which is the same for Ate and twisted ate pairing. Recently, an attack against a whole pairing, i.e. the Miller algorithm together with the final exponentiation, was published by Blömer et al. in [4]. The attack consists in modifying the clock of the device and as a consequence, the device returns intermediary results that allow to recover the secret. Few works consider differential power analysis. In [13] El Mrabet et al. highlight the fact that without protection the Miller algorithm is sensitive to a differential power analysis attack. Their work was recalled in [5]. In practice, the efficiency of side channel attacks does not lay on the choice of the characteristic, neither on the choice of the elliptic curve, nor on the choice of the coordinates. To each attack, several countermeasures were proposed. The countermeasures rely on the bilinearity of pairings, or on the homogeneity of the coordinates [14].

3.2 Description of Fault Attack

In an Identity Based Encryption scheme [6], one argument of the pairing is secret. So fault attacks can be performed to reveal the secret. We describe the attack against the Miller algorithm. As stated in the introduction, fault attack on pairing algorithm tries to corrupt the loop bound (which is $\log(m)$) of the Miller algorithm. The attacker injects fault repetitively in such a way that he can obtain two consecutive loop bounds $\log(m-s)$ and $\log(m-s)+1$ and the corresponding pairings $e_{m-s}(P,Q)$ and $e_{m-s+1}(P,Q)$, for a certain integer s. It has been shown in [11] that it is possible to obtain such consecutive integers in a finite number of fault injections.

The clock glitch attack described in [4] highlights the fact that in practice a modification of the glitch can make the device stop and return intermediary results, such as internal results of Miller's algorithm. In order to explain how the attacker can obtain the secret point from the erroneous pairings $e_{m-s}(P,Q)$ and $e_{m-s+1}(P,Q)$ we consider the two following situations.

First situation: Excluding the final exponentiation. Instead of obtaining the values $e_{m-s}(P,Q)$ and $e_{m-s+1}(P,Q)$ after the final exponentiation, the attacker tries to get the final values obtained after $\log(m-s)$ and $\log(m-s)+1$ iterations, just before the final exponentiation. A method to obtain those intermediary values is the use of a clock glitch attack [4]. We denote these values by $f_{m-s,P}(Q)$ and $f_{m-s+1,P}(Q)$. Depending of the last bit corresponding to each iteration, we have four possibilities for the expression of $f_{m-s,P}(Q)$ and $f_{m-s+1,P}(Q)$.

Without lost of general ity, we can consider the case when

$$f_{m-s+1,P}(Q) = (f_{m-s,P}(Q))^2 \times h_{[j]P,[j]P}(Q),$$

with j the integer composed by the $log_2(m-s)$ most significant bits of m.

Consequently, the attacker knows

$$S = \frac{f_{m-s+1,P}}{f^2_{m-s,P}}(Q) = h_{[2j]P,[2j]P}(Q).$$

The trick of the attacker is now to use the representation of S and $h_{[2j]P,[2j]P}(Q) \in \mathbb{F}_{q^k}$ in a basis of $\mathbb{F}_{q^k}/\mathbb{F}_q$ in order to obtain by identification, a system of linear or non-linear equations. The resolution of this system leads to the obtention of the coordinates of the secret point. A successful such attack has been mounted against the Miller algorithm [12]. We briefly recall the attack and refer to [11] for a complete description of this attack.

We recall that the point Q is public, the point P is secret and R is random in $E(\mathbb{F}_{q^k})$. For efficiency reasons, the embedding degree k is smooth and at least divisible by 2, or 4 or for the best cases by 6. A smooth integer is a number that admits a factorisation into small prime numbers. This condition on k enables efficient computation of pairings and the denominator elimination thanks to the twist of the elliptic curve. A consequence is that the points Q and R are seen as images of points belonging to the twist. The coordinates of R are composed by at most k values in $\mathbb{F}_{q^{k/d}}$, where d is the degree of the twist. The point P could be given in affine, projective or Jacobian coordinates. The choice will depend on the most efficient computation for the pairing. Whatever the choice is, the coordinates of point P will always count as 2 unknown values X_P and Y_P. This is obvious if P is given in affine coordinates. If P is given in projective or Jacobian coordinates, P would be characterized and gives improvement of the pairing computations by 3 unknown values X_P, Y_P and Z_P. But, using the homogeneity of projective and Jacobian coordinates, we could consider that the point P is in fact X'_P, Y'_P and 1. Indeed, we know that for $Z \neq 0$ in projective coordinates $(X, Y, Z) \cong (X/Z, Y/Z, 1)$ and in Jacobian coordinates $(X, Y, Z) \cong (X/Z^2, Y/Z^3, 1)$.

Putting all together one obtains a system of $k + 2$ polynomial equations in $k + 2$ unknown values. This system admits solutions as it is derived from a constructive algorithm. The points P and R are defined by construction. So, we can use the Gröbner basis [8] for instance to solve the system and find the coordinates of the point P. If the secret is the point Q, the attack is easier and successful [11].

Second situation: Including the final exponentiation. In this situation we consider the values $e_{m-s}(P, Q)$ and $e_{m-s+1}(P, Q)$ obtained after the final exponentiation. Then

$$\frac{e_{m-s+1}}{e^2_{m-s}} = \left[h_{[2j]P,[2j]P}(Q)\right]^{\frac{(q^k-1)}{r}}$$

The aim here is, since it has been easy to obtain $e_{m-s}(P, Q)$ and $e_{m-s+1}(P, Q)$ contrary to situation 1, to reverse the exponent $\frac{(q^k-1)}{r}$, such that an application of the method in situation 1 may lead to the obtaining of the secret. In secured pairing based protocols, it has been shown that the exponent $\frac{(q^k-1)}{r}$ is difficult to reverse mathematically [30, 24]. So the attack in this situation requires a fault model that would neutralize the final exponentiation, which is possible experimentally. One possibility can be to combine two fault models to neutralize the final exponentiation. For instance use a fault attack to reduce the number of iterations as in [11] and a fault attack to reverse the exponentiation as in [24]. Another way would be to use a fault model that modifies the time of execution as modification of the glitch or under voltage attack [4].

Remark 1. In the case of super singular elliptic curves, the final exponentiation can be reversed by mathematical considerations, the form of the exponent combined with a sparse decomposition in the basis of $\mathbb{F}_{p^k}$ allow this operation [27]. This is specific to pairings over supersingular elliptic curves and cannot be applied to ordinary elliptic curves.

3.3 The Point Blinding countermeasure and weaknesses

In [16], Ghosh, Mulhopadhyay and Chowdhury proposed an analysis of countermeasures to fault attack presented in [27]. They analyze what they called the new point blinding technique:

$$e(P,Q) = e([x]P, [y]Q) \text{ for random } x, y \text{ such that } xy \equiv 1 \mod (r)$$

and the altering traditional point blinding:

$$e(P,Q) = \frac{e(P, Q+R)}{e(P,R)},$$

for R a random point in $E(\mathbb{F}_q)$ such that the pairings $e(P,Q)$ and $e(P,R)$ are defined. They conclude that these two countermeasures are not sufficient against the fault attack described in [27]. However their analysis was not convincing. Concerning the new point blinding method, they claim that the intermediary steps of a pairing computation are bilinear which is not the case. The ratio obtained in the attack depends on the coordinates of the points $[x]P$ and $[y]Q$, with x and y unknown to the attacker. They do not explain how they can recover the value of the secret point used during the pairing computation. Concerning the altering traditional point blinding method, their analysis was not clear enough. In [16] the explanation did not take into account the randomness induced by the point R. We demonstrate in the next section that this countermeasure is not efficient with a precise approach and we develop the corresponding equation.

In [28] Park et al. exposed the weaknesses of the point blinding technique against fault attacks of Page and Vercauteren [27]. They presented an attack where they omit the last iteration of the Duursma and Lee algorithm. We generalize their approach to the Miller algorithm and for every iteration not only the last one.

4 Attack against the point blinding countermeasure during Miller's algorithm

In this section, we first explain how the Miller algorithm can be implemented with the point blinding technic. As far as we know, this is the first time that an algorithm is proposed for the implementation of this counter measure. The aim of point blinding method is to add randomness to the known entry of the pairing computation. Indeed, a side channel attack is successful principally because the attacker knows the value of data combined with the secret. The point blinding countermeasure is made to blind the knowledge of the attacker. As the point R is random, the point $Q + R$ is also random. This countermeasure is considered as sufficient to prevent any side channel attack against a pairing implementation.

We then show how this countermeasure does not really protect the algorithm against fault attack.

4.1 Implementation of the countermeasure

We discuss here the possible ways to implement the Miller algorithm using the point blinding countermeasure: $e(P,Q) = \frac{e(P,Q+R)}{e(P,R)}$.

Case 1: We consider that the secret is the point $P \in E(\mathbb{F}_q)$. The point $Q \in E(F_{q^k})$ is public. The countermeasure consists in adding randomness to the point Q, expecting that it would be then impossible to perform the fault attack. The randomness is the choice of a point R such that the pairings $e(P,R)$ and $e(P,Q+R)$ are defined.

In practice, for optimization reason, k is smooth. In order to simplify the explanation, we consider that $k \equiv 0 \mod 2$. The point Q is represented as the image of a point Q' belonging to the twisted elliptic curve E' of E and defined over $\mathbb{F}_{q^{k/2}}$. The coordinates of Q are $Q = (x_Q, y_Q\sqrt{\nu})$, for a quadratic twist. If another twist is used, the scenario is the same, but the equation must be adapted in consequence.

The device is implemented to compute $\frac{e(P,Q+R)}{e(P,R)}$. For efficiency reasons, as these two pairing computations are performed during the scalar multiplication of the point P, the two computations $e(P,Q+R)$ and $e(P,R)$ would be done in parallel. In order to compute only one exponentiation on the elliptic curve. The inversion in the field $\mathbb{F}_{q^k}$ and the final exponentiation are expensive operations. So, once obtained the results $f_{m,P}(Q+R)$ and $f_{m,P}(R)$, it will be more efficient to perform the inversion followed by the final exponentiation instead of two final exponentiations followed by an inversion. In practice, the discussion about inverting the final exponentiation is the same for the altering point blinding countermeasure and the classical Miller algorithm recalled in Section 3.2. Given these efficiency considerations, the Miller algorithm that would be used for the point blinding countermeasure would likely to be as presented in Algorithm 2. For clarity of explanations, we add the inversion at the end of Miller algorithm (step 14), it could be performed outside the Miller algorithm and that would not change our discussion.

Algorithm 1: Miller's Algorithm with the point blinding countermeasure
Input : $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r] \setminus E(\mathbb{F}_q)[r]$, $m = (1, m_{n-2}, ...m_1, m_0)_2$.
Output: $\frac{f_{m,P}(Q+R)}{f_{m,P}(R)}$

1: Choose R randomly in $E(\mathbb{F}_{q^k})[r] \setminus E(\mathbb{F}_q)[r]$
2: If $R = -Q$, go to 1.
3: Set $f \leftarrow 1$, $g \leftarrow 1$ and $T \leftarrow P$
4: **For** $i = n-2$ **down to** 0 **do**
5: $f \leftarrow f^2 \cdot h_{T,T}(Q+R)$
6: $g \leftarrow g^2 \cdot h_{T,T}(R)$
7: $T \leftarrow 2T$
8: **if** $m_i = 1$ **then**
9: $f \leftarrow f \cdot h_{T,P}(Q+R)$
10: $g \leftarrow g \cdot h_{T,P}(R)$
11: $T \leftarrow T + P$
12: **end if**
13: **end for**
14: **return** $\frac{f}{g}$

Fig. 2. The modified Miller algorithm

Case 2: We consider that the point $P \in E(\mathbb{F}_q)$ is public and the secret is the point $Q \in E(F_{q^k})$. The randomness, considering the point blinding countermeasure would be added to the point P. The device would be implemented in order to compute $\frac{e(P+R,Q)}{e(R,Q)}$. The implementations of the two Miller algorithms would then be done either in parallel or consecutively. The choice would highly depend on the target for the implementation. On a multiple processor device the parallel solution would be preferred. On a constrained device, as a smart card, the computation would be done one after the other, or delegated to a more powerful device. Considering this hypothesis we do not try to give a general way to perform the computation. Indeed, either the same counter will be used and if it is modified once, it will be for the two computations. Either two counters will be used and then two faults would be necessary to modify them. The case of a delegation of the computation would require a whole article. We do not describe it here.

4.2 Description of the attacks

We describe here the fault attack against the Miller algorithm implemented using the point blinding countermeasure $e(P,Q) = \frac{e(P,Q+R)}{e(P,R)}$.

Case 1: when the secret is the point P. We consider that the secret is the point P, we can freely choose the point Q and the randomness is the point R such that the pairings $e(P,R)$ and $e(P,Q+R)$ are defined. The device is implemented to compute $\frac{e(P,Q+R)}{e(P,R)}$ using the modified Miller algorithm described in the Algorithm 2.

The target of the fault attack is the counter given the number of iterations in the modified Miller algorithm. The aim of the fault is to reduce the number of iterations performed during the execution of the Miller algorithm. For instance, the fault can be induced by a laser [1, 34] or a modification of the glitch [4]. The probability to obtain two shortened Miller algorithms with consecutive number of iterations is high enough to made this hypothesis realistic [11]. So, we suppose that we have obtained the results of the modified Miller algorithm after the m'^{th} and the $(m'+1)^{th}$ iterations, for m' an integer smaller than m the original number of iterations. We exactly know what happens during the $(m'+1)^{th}$ iteration.

Let f'_m and g'_m denote the results stored in f and g at the m'^{th} iteration, let m_i be the value of the corresponding bit. Then, in order to express $f_{m'+1}$ and $g_{m'+1}$ we must consider two possibilities, either the m_i is 0, or 1.
If $m_i = 0$, then $f_{m'+1} = f'^2_m \times h_{T,T}(Q+R)$ and $g_{m'+1} = g^2_m \times h_{T,T}(R)$, with $T = [1m_{n-1} \dots m_{i+1}m_i]P$.
If $m_i = 1$ then $f_{m'+1} = \left(f'^2_m \times h_{T,T}(Q+R)\right) \times h_{2T,P}(Q+R)$ and $g_{m'+1} = \left(g'^2_m \times h_{T,T}(R)\right) \times h_{2T,P}(R)$. The attacker will receive the two values $\frac{f'_m}{g'_m}$ and $\frac{f_{m'+1}}{g_{m'+1}}$ in $\mathbb{F}_{q^k}$. We could be tempted to follow the scheme of the attacks described in [27, 11], i.e. compute the exact value in $\mathbb{F}_{q^k}$ of the ratio $\frac{\frac{f_{m'+1}}{g_{m'+1}}}{\left(\frac{f'_m}{g'_m}\right)^2}$, use its theoretical decomposition (if $m_i = 0$ it is $\frac{h_{T,T}(Q+R)}{h_{T,T}(R)}$ or if $m_i = 1$ it is $\frac{h_{T,T}(Q+R) \times h_{2T,P}(Q+R)}{h_{T,T}(R) \times h_{2T,P}(R)}$) and after use the identification in the basis of $\mathbb{F}_{q^k}$ in order to obtain k equations depending on the coordinates or P, Q and R. The equation of the elliptic curve gives two more equations as P and R are on the curve.

But be careful! The point R is randomly chosen at each execution of the Algorithm 2. So in practice, we obtain $\frac{f'_m}{g'_m}(P,Q,R_1)$ and $\frac{f_{m'+1}}{g_{m'+1}}(P,Q,R_2)$, for R_1 and R_2 two random points in

$E(\mathbb{F}_{q^k})[r] \setminus E(\mathbb{F}_q)[r]$. In this case, the theoretical decomposition of the ratio $\frac{\frac{f_{m'+1}}{g_{m'+1}}(P,Q,R_2)}{\frac{f'^2_m}{g'^2_m}(P,Q,R_1)}$ would not admit any simplification and the previous description inspired from [27, 11] is no longer possible. We have to describe a more painful and awful attack.

In this attack, we need only one faulty result $\frac{f'_m}{g'_m}(P,Q,R)$, for P secret, Q chosen and R random. After one iteration of the Miller algorithm, assuming that the corresponding bits of m are 0, we have $f_1 = h_{P,P}(Q+R)$ and $g_1 = h_{P,P}(R)$. After two iterations, $f_2 = h_{[2]P,[2]P}(Q+R) \times (h_{P,P}(Q+R))^2$ and $g_2 = h_{[2]P,[2]P}(R) \times (h_{P,P}(R))^2$. We can express the equation of $h_{P,P}$ and $h_{[2]P,[2]P}$ in terms of the coordinates of P. The evaluation of these functions at the points $Q+R$ and R will give a polynomial expression in the coordinates of P and R.

The theoretical description of the coordinates of R will admit a decomposition in the basis of $\mathbb{F}_{q^k}$. If we are able to obtain the result of the Miller algorithm after m' iterations (denoted $\lambda_0 + \lambda_1\sqrt{\nu}$, with λ_0 and $\lambda_1 \in \mathbb{F}_{q^{k/2}}$), we have on one hand the theoretical description and on an other the value in $\mathbb{F}_{q^k}$ of this description:

$$\frac{f_{m'}(P,Q,R)}{g_{m'}(P,Q,R)} = \lambda_0 + \lambda_1\sqrt{\nu}. \tag{2}$$

We know the value of λ_0, λ_1 and the theoretical description of $f_{m'}(P,Q,R)$ and $g_{m'}(P,R)$. Exactly like at the end of the attack described in [11], by identification in the basis of $\mathbb{F}_{p^k}$, we obtain a system of k polynomial equations with coordinates of P and R as unknown. The degree of the polynomial depends on the number of iterations. That is why an important step of the attack is to minimize the number of iterations that are executed by the Miller algorithm.

As illustration we have for one iteration, $f_{m'}(P,Q,R) = h_{P,P}(Q+R)$ and $g_{m'}(P,R) = h_{P,P}(R)$. The equation 2 gives $h_{P,P}(Q+R) = (\lambda_0 + \lambda_1\sqrt{\nu}) \times h_{P,P}(R)$ which is a degree 3 polynomial in X_P, a degree 2 in Y_P, a degree 6 for Z_P and a degree 1 polynomial in x_R. We give the equations of $h_{P,P}(Q+R)$ and $h_{P,P}(R)$ (see section 2 for details) in order to illustrate an idea of the system.

$$\begin{aligned}
P &= (X_P, Y_P, Z_P), X_P, Y_P, Z_P \in \mathbb{F}_q \\
Q+R &= (x_{Q+R}, y_{Q+R}\sqrt{\nu}), x_{Q+R}, y_{Q+R} \in \mathbb{F}_{q^{k/2}} \\
R &= (x_R, y_R\sqrt{\nu}), x_R, y_R \in \mathbb{F}_{q^{k/2}} \\
h_{P,P}(Q+R) &= 2Y_P Z_P^3 y_{Q+R}\sqrt{\nu} - 2Y_P^2 - (3X_P^2 + aZ_P^4)(x_{Q+R}Z_P^2 - X_P) \\
h_{P,P}(R) &= 2Y_P Z_P^3 y_R\sqrt{\nu} - 2Y_P^2 - (3X_P^2 + aZ_P^4)(x_R Z_P^2 - X_P)
\end{aligned}$$

The equation of the elliptic curve in P and R gives us 2 more equations and we still have $k+2$ unknown values in $\mathbb{F}_q$. To conclude the attack, we will use the Gröbner basis. In order to ensure the fact that the solution will be in $\mathbb{F}_q$, we have to add the equation $\xi^p \equiv \xi \mod p$ for each unknown value. We therefore obtain a system of $2k+2$ polynomial equations for k+2 unknown values. The Gröbner basis is the perfect tool for solving this system, that admits solutions by construction.

Obviously for a greater number of iterations, by hand it is difficult to develop the theoretical expression without any mistake. We do not describe it even for one iteration. Fortunately, we have mathematical softwares that can help us, like PariGP[33], Sage [32], Magma [7] or Maple [26].

If we consider that each iteration raise the degree of the polynomials in the system by a power of 2, than after μ iterations, the degree of the polynomial would be 2^μ in the coordinates of P. In practice, the evaluation of the degree is more complex. The degree of $h_{P,P}(R)$ is 6 in Z_P. After 2 iterations, the degree of $g_{2,P}(R)$ will be at the most $6 \times 2 + 6$ for Z_P and 3 for the coordinates of R. (The degree of g and f are the same, we choose to describe it for g for clarity. The degree of f depends on the coordinates of $Q + R$.) For n iterations, $n > 2$, we can estimate the degree of the polynomial with the formulas:

$$\begin{aligned} \deg(n, Z_P) &= 2 \times \deg(n-1, Z_P) + 6^{n-2} \times 13 \\ \deg(n, R) &= 2 \times \deg(n-1, R) + 1, \end{aligned}$$

where $\deg(n, Z_P)$ represents the degree of the polynomial system after n iterations in the unknown value Z_P and $\deg(n, R)$ is the degree in the coordinates of R. The degree of the polynomial for X_P and Y_P is smaller than the degree for Z_P.

The interesting question is how many iterations can we deal with? What would be the maximum degree of the polynomial system that can be solved by Gröbner basis in a reasonable time? We refer to [8] for more details on Gröbner basis.

Case 2: when the secret is the point Q. We consider that the secret is the point Q, we can freely choose the point P, the randomness is the point R such that the pairings $e(P+R, Q)$ and $e(R, Q)$ are defined. The device is implemented to compute $\frac{e(P+R,Q)}{e(R,Q)}$ using a modified version of the Miller algorithm. If the same counter is used to perform the computation it would be modified once and used for both computations. If two counters are used, as in [34] we need two faults to modify the counters. After that, the scheme of the attack is the same. Once we obtain the intermediate results $\frac{f'_m}{g'_m}(P, R, Q)$, for P public, R random and Q secret. The theoretical expression of R, $P+R$ and $h_{T,T}(Q)$ depending on the coordinates of P, R and Q combined with the value of $\frac{f'_m}{g'_m}(P, R, Q)$ will give a polynomial system in the unknown coordinates of R and Q. This polynomial system would be solved using the Gröbner basis.

5 Acknoledgments

This work was supported by the French ANR-12-INSE-0014 SIMPATIC Project financed by the Agence National de Recherche (France). We would like to thank the anonymous reviewers for their numerous suggestions and remarks which have enables us to substantially improve the paper.

6 Conclusion

In this paper we analysed the efficiency of the point blinding countermeasure in pairing based cryptography considering fault attacks in Miller's algorithm. We describe a theoretical fault attack. We highlighted the fact that the point blinding countermeasure alone is not a protection in the case of pairing based cryptography. Whenever the secret is the first or the second parameter, a fault attack gives the coordinates of the secret.

In our opinion, we believe that the only way to provide a secure implementation of the pairing relies on the discrete logarithm problem. The computation of $e(P,Q)$, should be $e([a]P,[b]Q)$, with a and b integers such that $ab \equiv 1 \mod r$. Of course, the computation of $[a]P$ and $[b]Q$ should be secured.

References

1. Ross Anderson and Marcus Kuhn. Tamper resistance – a cautionary note. In *The Second USENIX Workshop on Electronic Commerce Proceedings*, pages 1–11, 1996.
2. Kiseok Bae, Sangjae Moon, and Jaecheol Ha. Instruction fault attack on the Miller algorithm in a pairing-based cryptosystem. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*, pages 167–174, July 2013.
3. Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT'14*, volume 8441 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2014.
4. Johannes Blömer, Ricardo Gomes da Silva, Peter Günther, Juliane Krämer, and Jean-Pierre Seifert. A practical second-order fault attack against a real-world pairing implementation. In *Proceedings of Fault Tolerance and Diagnosis in Cryptography(FDTC)*, 2014. To appear. Updated version at http://eprint.iacr.org/2014/543.
5. Johannes Blömer, Peter Günther, and Gennadij Liske. Improved side channel attacks on pairing based cryptography. In *Constructive Side-Channel Analysis and Secure Design*, pages 154–168. Springer Berlin Heidelberg, 2013. The final publication is availableat link.springer.com: http://link.springer.com/chapter/10.1007
6. Dan Boneh and Matthew Franklin. Identity-Based Encryption from the Weil pairing. *SIAM J. of Computing*, 32(3):586–615, 2003.
7. J. Bosma, W. Cannon and C. Playout. The Magma algebra system I. the user language. *J. Symbolic Comput.*, vol. 24(3-4), pp. 235-265, 1997.
8. Bruno Buchberger. An algorithm form finding the basis elements of the residue class ring of a zero dimensional polynomial ideal (phd thesis 1965). In elsevier, editor, *Journal of Symbolic Computation*, volume 41, pages 475 – 511. elsevier, 2006.
9. Henri Cohen and Gerhard Frey, editors. *Handbook of elliptic and hyperelliptic curve cryptography*. Discrete Math. Appl., Chapman & Hall/CRC, 2006.
10. R. Dutta, R. Barua, and P. Sarkar. Pairing-based cryptography : A survey. *Cryptology ePrint Archive*, Report 2004/064, 2004.
11. Nadia El Mrabet. What about vulnerability to a fault attack of the Miller algorithm during an Identity Based Protocol? In *Advances in Information Security and Assurance*, volume 5576 of *LNCS*, pages 122–134. Springer, 2009.
12. Nadia El Mrabet. Fault attack against Miller's algorithm. *IACR Cryptology ePrint Archive*, 2011:709, 2011.
13. Nadia El Mrabet, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre, and Jean-Claude Bajard. Differential Power Analysis against the Miller algorithm. Technical report, August 2008. Published in Prime 2009, IEEE Xplore.
14. Nadia El Mrabet, Dan Page, and Frederik Vercauteren. Fault attacks on pairing-based cryptography. In Marc Joye and Michael Tunstall, editors, *Fault Analysis in Cryptography*, Information Security and Cryptography, pages 221–236. Springer Berlin Heidelberg, 2012.
15. David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *J. Cryptology*, 23(2):224–280, 2010.
16. Dipanwita Roy Chowdhury Ghosh Santosh, Mukhopadhyay Debdeep. Fault attack and countermeasures on pairing based cryptography. *International Journal of Network Security*, 12(1):21—28, 2011.
17. Florian Hess. Pairing lattices. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing*, volume 5209 of *Lecture Notes in Computer Science*, pages 18–38. Springer, 2008.
18. Florian Hess, Nigel Smart, and Frederik Vercauteren. The Eta Pairing Revisited. In *IEEE Transactions on Information Theory*, volume 52, pages 4595–4602, 2006.
19. Tadashi Iyama, Shinsaku Kiyomoto, Kazuhide Fukushima, Toshiaki Tanaka, and Tsuyoshi Takagi. Efficient implementation of pairing on brew mobile phones. In *Advances in Information and Computer Security*, pages 326–336. Springer, 2010.

20. Antoine Joux. A new index calculus algorithm with complexity $l(1/4+o(1))$ in small characteristic. 8282:355–379, 2013.
21. Marc Joye and Gregory Neven. *Identity-based Cryptography*. Cryptology and information security series. IOS Press, 2009.
22. Y. Kawahara, T. Takagi, and E. Okamoto. Efficient implementation of Tate pairing on a mobile phone using java. In *Computational Intelligence and Security, 2006 International Conference on*, volume 2, pages 1247–1252, Nov 2006.
23. Neal Koblitz and Alfred Menezes. Pairing-based cryptography at high security levels. In Nigel P. Smart, editor, *Cryptography and Coding, 10th IMA International Conference, Cirencester, UK, December 19-21, 2005, Proceedings*, pages 13–36. Springer, 2005.
24. Ronan Lashermes, Jacques Fournier, and Louis Goubin. Inverting the final exponentiation of Tate pairings on ordinary elliptic curves using faults. In Guido Bertoni and Jean-Sébastien Coron, editors, *CHES*, volume 8086 of *Lecture Notes in Computer Science*, pages 365–382. Springer, 2013.
25. Victor Miller. The Weil pairing and its efficient calculation. *Journal of Cryptology*, 17:235–261, 2004.
26. Michael B. Monagan, Keith O. Geddes, K. Michael Heal, George Labahn, Stefan M. Vorkoetter, James McCarron, and Paul DeMarco. *Maple 10 Programming Guide*. Maplesoft, Waterloo ON, Canada, 2005.
27. Dan Page and Frederik Vercauteren. A fault attack on Pairing-Based Cryptography. *Computers, IEEE Transactions on*, 55(9):1075 –1080, sept. 2006.
28. Jea Park, Gyo Sohn, and Sang Moon. Fault attack on a point blinding countermeasure of pairing algorithms. *ETRI Journal*, 33(6), 2011.
29. Michael Scott. Computing the Tate pairing. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 293–304. Springer Berlin Heidelberg, 2005.
30. Michael Scott, Naomie Benger, Manuel Charlemagne, Luis Dominguez, and Ezekiel Kachisa. On the Final Exponentiation for Calculating Pairings on Ordinary Elliptic Curves. In *Pairing-Based Cryptography Pairing 2009*, volume 5671 of *LNCS*, pages 78–88. Springer, 2009.
31. Michael Scott, Neil Costigan, and Wesam Abdulwahab. Implementing cryptographic pairings on smartcards. 4249:134–147, 2006.
32. W. Stein. Sage mathematics software (version 4.8). *The Sage Group*, 2012. http://www.sagemath.org.
33. The PARI Group, Bordeaux. *PARI/GP, version* `2.7.0`, 2014. available from http://pari.math.u-bordeaux.fr/.
34. Elena Trichina and Roman Korkikyan. Multi fault laser attacks on protected CRT-RSA. In *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2010 Workshop on*, pages 75–86. IEEE, 2010.
35. Frederik Vercauteren. Optimal pairings. *IEEE Trans. Inf. Theor.*, 56(1):455–461, January 2010.
36. L.C. Washington. Elliptic curves, number theory and cryptography. *Discrete Math .Aplli, Chapman and Hall*, 2008.
37. Claire Whelan and Michael Scott. The importance of the final exponentiation in pairings when considering Fault Attacks. In *Pairing-Based Cryptography – Pairing 2007*, volume 4575 of *LNCS*, pages 225–246. Springer, 2007.