Fault Attack against Miller's algorithm

Nadia EL MRABET

LIASD, University Paris 8,
2, rue de la liberté,
93256 St Denis, France
elmrabet@ai.univ-paris8.fr

Abstract. We complete the study of [23] and [27] about Miller's algorithm. Miller's algorithm is a central step to compute the Weil, Tate and Ate pairings. The aim of this article is to analyze the weakness of Miller's algorithm when it undergoes a fault attack. We prove that Miller's algorithm is vulnerable to a fault attack which is valid in all coordinate systems, through the resolution of a nonlinear system. We highlight the fact that putting the secret as the first argument of the pairing is not a countermeasure. This article is an extensed version of the article [15].

Key words: Miller's algorithm, Identity Based Cryptography, Fault Attack, Coordinates System.

1 Introduction

In 1984, A. Shamir challenged the cryptographer community to find a protocol based on the user's identity [25]. This challenge was issued almost ten years later by D. Boneh and M. Franklin. In 2003, D. Boneh and M. Franklin created an identity-based encryption scheme based on pairings [10]. The general scheme of an identity based encryption is described in [10]. The important point is that to decipher a message using an Identity Based Protocol, a computation of a pairing involving the private key and the message is done. 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. The secret key influences neither the execution time nor the number of iterations of the algorithm. Fault attack against pairing based cryptography were first developed three years ago ([23], [26] and [27]).

In [23], D. Page and F. Vercauteren introduce a fault attack against the Duursma and Lee algorithm. The fault attack consists in modifying the number of iterations of the algorithm. We complete this idea in order to apply it to Miller's algorithm, and we describe a way to realise this fault injection.

In [27], C. Whelan and M. Scott present a fault attack against the Weil and Eta pairings. They consider the case when exactly the last iteration is modified by a fault injection. They deduced that Miller's algorithm is not vulnerable to a fault attack, because the system obtained after the fault attack is nonlinear and then impossible to solve. In [26] they also concluded that if the secret is used as the first argument of the pairing computation, then it cannot be found. Contrary to their conclusion, we show that even if the secret is the first argument of the pairing, we can discover it with a fault attack, and solve the nonlinear system obtained after the fault attack on Miller's algorithm. Moreover, we generalise the fault attack to every iteration of the algorithm, not only the last one. Both articles considered affine coordinates. We show that in every coordinate systems, our attack will give us the result.

Our contribution is to generalise the fault attack to Miller's algorithm, not only for the last iteration, but for every possible iterations; and to demonstrate that for all the coordinate systems (affine, projective, Jacobian, and Edwards coordinates) a fault attack against Miller's algorithm can be done through the resolution of a nonlinear system [15]. This demonstration will be followed by discussion about the weakness to this fault attack of pairings based on Miller's algorithm. We show that the Weil pairing is directly sensitive to the fault attack described. Some methods to override the final exponentiation are given, and then, for a motivated attacker, the final exponentiation will no longer be a natural counter measure for the Tate and Ate pairings [12].

The outline of this article is as follow. First we will give a short introduction to pairing and to Miller's algorithm in Section 2. Section 3 presents our fault attack against Miller's algorithm, in Section 6 we analyse the vulnerability of pairings using Miller's algorithm as a central step in Jacobian coordinates. We present the attack in Edward's coordinates in Section 4. Finally, we give our conclusion in Section 7.

2 Pairings and Miller's algorithm

2.1 Short introduction to the pairing

We will consider pairings defined over an elliptic curve E over a finite field $\mathbb{F}_q$, for q a prime number. In the case where q is a power of a prime number, while the equations are a slightly different the same scheme can be applied. We describe the attack for calculations in Jacobian coordinates. The affine, projective and Edwards coordinates cases can be treated by the same way.

We will consider the Weierstrass elliptic curve in Jacobian coordinates : $Y^2 = X^3 + aXZ^4 + bZ^6$, with a and $b \in \mathbb{F}_q$. Let $l \in \mathbb{N}^*$, and k be the smallest integer such that l divides $(q^k - 1)$, k is called the embedding degree. Let $G_1 \subset E(\mathbb{F}_q)$, $G_2 \subset E(\mathbb{F}_{q^k})$, $G_3 \subset \mathbb{F}^*_{q^k}$, be three groups of order l.

Definition 1. *A pairing is a bilinear and non degenerate function:* $e : G_1 \times G_2 \to G_3$.

The most useful property in pairing based cryptography is bilinearity: $e([n]P, [m]Q) = e(P,Q)^{nm}$. Four different pairings are used in cryptography, and three of them are constructed in the same way. Miller's algorithm [22] is the central step for Weil, Tate and Ate pairings computations.

2.2 Miller's algorithm

The following description of Miller's algorithm is referenced in [13, chapter 16].

Miller's algorithm is the most important step for the Weil, Tate and Ate pairings computation. It is constructed like a double and add scheme using the construction of $[l]P$. Miller's algorithm is based on the notion of divisors. We only give here the essential elements for the pairing computation.

Miller's algorithm constructs the rational function F_P associated to the point P, where P is a generator of $G_1 \subset E(\mathbb{F}_q)$; and at the same time, it evaluates $F_P(Q)$ for a point $Q \in G_2 \subset E(\mathbb{F}_{q^k})$.

Algorithm 1: Miller(P, Q, l)

Data: $l = (l_n \dots l_0)$(radix 2 representation), $P \in G_1(\subset E(\mathbb{F}_q))$ and $Q \in G_2(\subset E(\mathbb{F}_{q^k}))$;
Result: $F_P(Q) \in G_3(\subset \mathbb{F}^*_{q^k})$;
$1 : T \leftarrow P$;
$2 : f_1 \leftarrow 1$;
$3 : f_2 \leftarrow 1$;
for $i = n-1$ **to** 0 **do**
  $4 : T \leftarrow [2]T$, where $T = (X, Y, Z)$ and $[2]T = (X_2, Y_2, Z_2)$;
  $5 : f_1 \longleftarrow {f_1}^2 \times h_1(Q)$, $h_1(x)$ is the equation of the tangent at the point T;
  if $l_i = 1$ **then**
    $6 : T \leftarrow T + P$;
    $7 : f_1 \longleftarrow f_1 \times h_2(Q)$, $h_2(x)$ is the equation of the line (PT);
  end
end
return f_1

Algorithm 2 is a simplified version of Miller's algorithm (see [6]). The original algorithm is given in Section A.1. Without loss of generality we can consider this simplified Miller's algorithm. We will see in Section 6.1 that the conclusions for the original algorithm are the same.

3 Fault Attack against Miller's algorithm

From here on, the secret key will be denoted P and the public parameter Q. We are going to describe a fault attack against Miller's algorithm. We assume that the algorithm is implemented on an electronic device (like a smart card). We restrict this study to the case where the secret is used as the first argument of the pairing. If the secret is used as the second argument, the same attack can easily be applied as it is explained in Section 3.3. Thus whatever the position of the secret point, we can recover it.

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, new technologies could allow for this attack [17].

3.1 Description of the fault attack

We complete the scheme of attack described in [23] to use it against Miller's algorithm. In [23] the attack consists in modifying the number of iterations. We complete the idea of [23] by giving a precise description of the attack, by computing the probability of finding suitable number of iterations and by adapting it to Miller's algorithm case.

We assume that the pairing is used during an Identity Based Protocol, that the secret point P is introduced in a smart card or an electronic device as the first argument of the pairing. If the secret key is the second argument, then it is easier to find it, as it is explained in Section 3.3. The aim of the attack is to find P in the computation of $e(P, Q)$. We assume that we have as many public point Q as we want, and for each of them we can compute the pairing between the secret point P and the point Q. In order to find the secret P, we modify the number of iterations in Miller's 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. l) in Miller's algorithm. This step can be done by using reverse engineering procedures. In classical architecture, the counter is divided into small piece of 8 bits. We want to find the piece corresponding with the less significant bits of the counter. 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 Miller's algorithm. For example the disturbance can be induced by a laser [4]. Lasers are today thin enough to make this attack realistic [17]. Counting the clock cycles, we are able to know how many iterations the Miller loop has done. If the difference between the new number of iterations and the number of non modified iterations is smaller than 2^8, then we find the correct piece. If not, we repeat this manipulation until we find the piece of the counter corresponding to the less significant bits.

Once the less significant bits are found, we make several pairing computations and for each of them we modify the value of the counter. Each time, we record the value of the Miller loop and the number of iterations we made. The aim is to obtain a couple $(d, d+1)$ of two consecutive values, corresponding to d and $d+1$ iterations during Miller's algorithm, we give the probability to obtain such couple in Section 3.2.

3.2 The d^{th} step

We execute Miller's algorithm several times. For each execution we provoke a disturbance in order to modify the value of l, until we find the result of the d^{th} and $(d+1)^{th}$ iterations of Algorithm 2. We denote the two results by $F_{d,P}(Q)$ and $F_{d+1,P}(Q)$. To conclude the attack, we consider the ratio $\frac{F_{d+1,P}(Q)}{F_{d,P}(Q)^2}$. By identification in the basis of $\mathbb{F}_{q^k}$, we are lead to a system which can reveal the secret point P, which is described in Section 3.3.

The probability. The important point of this fault attack is that we can obtain two consecutive couples of iterations, after a realistic number of tests. The number of picks with two consecutive number is the complementary of the number of picks with no consecutive numbers. The number $B(n, N)$ of possible picks of n numbers among N integers with no consecutive number is given by the following recurrence formula:

$$\begin{cases} N \leq 0, n > 0, B(n, N) = 0, \\ \forall N, n = 0 B(n, N) = 1 \\ B(n, N) = \sum_{j=1}^{N} \sum_{k=1}^{n} B(n-k, j-2). \end{cases}$$

With this formula, we can compute the probability to obtain two consecutive numbers after n picks among N integers. This probability $P(n, N)$ is

$$P(n, N) = 1 - \frac{B(n, N)}{C_{n+N}^{n}}$$

The probability for obtaining two consecutive numbers is sufficiently large to make the attack possible. In fact, for an 8-bits architecture only 15 tests are needed to obtain a probability larger than one half, $P(15, 2^8) = 0.56$.

Finding j. After d iterations, if we consider that the algorithm 2 has calculated $[j]P$ then during the $(d+1)^{th}$ iteration, it calculates $[2j]P$ and considering the value of the $(d+1)^{th}$ bit of l, it either stops, or it calculates $[2j+1]P$. Q has order l,(as P and Q have the same order). By counting the number of clock cycles during the pairing calculation, we can find the number d of iterations. Then reading the binary decomposition of l gives us directly j. We consider that at the beginning $j = 1$, if $l_{n-1} = 0$ then $j \leftarrow 2j$, otherwise $j \leftarrow 2j+1$, and we continue, until we arrive at the $(n-1-d)^{th}$ bit of l. For example, let $l = 1000010000101$ in basis 2, and $d = 5$. At the fifth iteration $j = 65$.

3.3 Curve and equations

In [23] and [27], only the affine coordinates case is treated. In this case, a simple identification of the element in the basis of $\mathbb{F}_{q^k}$ gives the result. We demonstrate that for every coordinate systems, the fault attack against Miller's algorithm is efficient. We describe it for example in Jacobian coordinates. The difference between with the cases described in [23] and [27] is that we solve a nonlinear system.

The embedding degree. In order to simplify the equations, we consider case $k = 4$. As the important point of the method is the identification of the decomposition in the basis of $\mathbb{F}_{q^k}$, it is easily applicable when k is larger than 3. $k = 3$ is the minimal value of the embedding degree for which the system we obtain in Section 5.4 can be solve "by hand", without the resultant method described in Section 5.4. We use $k = 4$ in order to make the demonstration easier.

We denote $B = \{1, \xi, \sqrt{\nu}, \xi\sqrt{\nu}\}$ the basis of $\mathbb{F}_{q^k}$, this basis is constructed by a tower extensions. $P \in E(\mathbb{F}_q)$ is given in Jacobian coordinates, $P = (X_P, Y_P, Z_P)$ and the point $Q \in E(\mathbb{F}_{q^k})$ is in affine coordinates. As k is even, we can use a classical optimisation in pairing based cryptography which consists in using the twisted elliptic curve to write $Q = (x, y\sqrt{\nu})$, with $x,\ y$ and $\nu \in \mathbb{F}_{q^{k/2}}$ and $\sqrt{\nu} \in \mathbb{F}_{q^k}$, for more details we refer the reader to [6].

The equations of the function h_1 and h_2 in Miller's algorithm are the following:

$$\begin{cases} P = (X_P, Y_P, Z_P), \\ Q = (x, y\sqrt{\nu}) \\ T = (X, Y, Z) \\ h_1(x, y\sqrt{\nu}) = Z_3 Z^2 y\sqrt{\nu} - 2Y^2 - \\ \quad = 3(X - Z^2)(X + Z^2)(xZ^2 - X), \\ \text{with } Z_3 = 2YZ \text{ in step 5,} \\ h_2(x, y\sqrt{\nu}) = Z_3 y\sqrt{\nu} - (Y_P Z^3 - Y Z_P^3)x \\ \quad = -(X_p Y Z_p - X Y_P Z), \\ \text{with } Z_3 = Z Z_P (X_P Z^2 - X Z_P^2) \text{ in step 7.} \end{cases}$$

As we make random modifications of l during the fault attack, we suppose that we stop Miller's algorithm at its d^{th} step. Moreover, as the point P is of order l, it is sufficient to observe what happens for $d < l$, because:
$[j + \rho l]P = [j]P$ for $\rho \in \mathbb{N}$, so we consider $1 \leq d < l$.

Case 1: $l_{d+1} = 0$. We know the results of the d^{th} and $(d+1)^{th}$ iterations of Miller's algorithm, $F_{d,P}(Q)$ and $F_{d+1,P}(Q)$. We examine what happens during the $(d+1)^{th}$ iteration.

At the step 4 of Miller's algorithm we calculate $[2j]P = (X_{2j}, Y_{2j}, Z_{2j})$ and store the result in the variable T. The coordinates of $[2j]P$ are given by the following formula:

$$\begin{cases} X_{2j} = -8X_j Y_j^2 + 9(X_j - Z_j^2)^2 (X_j + Z_j^2)^2, \\ Y_{2j} = 3(X_j - Z_j^2)(X_j + Z_j^2) \times \\ \quad = (4X_j Y_j^2 - X_2) - 8Y_j^4, \\ Z_{2j} = 2Y_j Z_j. \end{cases}$$

where we denote $[j]P = (X_j, Y_j, Z_j)$.

Step 5 then gives:

$$F_{d+1,P}(Q) = (F_{d,P}(Q))^2 \times$$
$$\left(Z_{2j} Z_j^2 y\sqrt{\nu} - 2Y_j^2 - 3(X_j - Z_j^2)(X_j + Z_j^2)(xZ_j^2 - X_j)\right).$$

As we suppose that $l_{d+1} = 0$, the additional step is not done. The return result of Miller's algorithm is $F_{d+1,P}(Q)$. We dispose of $F_{d,P}(Q)$, $F_{d+1,P}(Q)$ and the point $Q = (x, y\sqrt{\nu})$, with x and $y \in \mathbb{F}_{q^2}$. Recall that the coordinates of Q can be freely chosen.
We can calculate the value $R \in \mathbb{F}_{q^k}^*$ of the ratio $\frac{F_{d+1,P}(Q)}{(F_{d,P}(Q))^2}$,

$$R = R_3 \xi\sqrt{\nu} + R_2\sqrt{\nu} + R_1\xi + R_0,$$

where $R_3,\ R_2,\ R_1,\ R_0 \in \mathbb{F}_q$.

Moreover, we know the theoretical form of R in the basis $B = \{1, \xi, \sqrt{\nu}, \xi\sqrt{\nu}\}$ which depends of coordinates of $[j]P$ and Q:

$$R = 2Y_j Z_j^3 y\sqrt{\nu} - 3Z_j^2 (X_j^2 - Z_j^4)x - 3X_j(X_j^2 - Z_j^4) - 2Y_j^2.$$

As the point $Q = (x, y\sqrt{\nu})$ is known, we know the decomposition of $x,\ y \in \mathbb{F}_{q^{k/2}}$, $x = x_0 + x_1\xi$, $y = y_0 + y_1\xi$, where $(1, \xi)$ defines the basis of $\mathbb{F}_{q^{k/2}}$, and the value of x_0, x_1, y_0, y_1. Furthermore, X_j, Y_j, and Z_j are in $\mathbb{F}_q$.

Consequently, with the exact value of R in $\mathbb{F}_{q^k}$, the coordinates of point Q and the theoretical expression of R depending on the coordinates of P and Q, we obtain the following system of equations in $\mathbb{F}_q$, by identification in the basis of $\mathbb{F}_{q^k}$.

$$\begin{cases} 2Y_jZ_j^3y_1 = R_3, \\ 2Y_jZ_j^3y_0 = R_2, \\ (-3Z_j^2(X_j^2 - Z_j^4))x_1 = R_1, \\ (-3Z_j^2(X_j^2 - Z_j^4))x_0 - 3X_j(X_j^2 - Z_j^4) - 2Y_j^2 = R_0. \end{cases}$$

This system can be simplified to the following (where we know value of $\lambda_{0,1,2}$):

$$\begin{cases} Y_jZ_j^3 = \lambda_2 \\ Z_j^2(X_j^2 - Z_j^4) = \lambda_1 \\ 3X_j(X_j^2 - Z_j^4) + 2Y_j^2 = \lambda_0 \end{cases}$$

This nonlinear system can be solve by the following way. Equation (1) gives Y_j as a function of Z_j, then equation (2) gives $3(X_j^2 - Z_j^4)$ as a function of Z_j. Substituting this expression in equation (3) gives X_j as a function of Z_j, substituting this expression of X_j in equation (2), we obtain a degree 12 equation in Z_j:

$$(\lambda_0^2 - 9\lambda_1^2)Z^{12} - (4\lambda_0\lambda_2^2 + 9\lambda_1^3)Z^6 + 4\lambda_1^4 = 0$$

This equation in Z_j admits by construction the point P as a solution. As the degree is even, this equation admits automatically at least an other solution, and at worst 12 solutions. We can use the function `factorff` in PariGP, a software for mathematical computation [24], to obtain the factorization of the equation in Z_j in $\mathbb{F}_q$, and consequently the solutions of this equation. Using equation (2) we can express X_j in Z_j, and the first equation gives Y_j. Solving the equation in Z_j, we find at most $24 = 12\times2\times1$ possible triplets (X_j, Y_j, Z_j) for the coordinates of the point $[j]P$. In practice we find at most eight possible solutions for Z_j, one example is given in Annex B. Once we have the coordinates of $[j]P$, to find the possible points P, we have to find j' the inverse of j modulo l, and then calculate $[j'][j]P = [j'j]P = P$. Using the elliptic curve equation, we eliminate triplets that do not lie on E. Then we just have to perform Miller's algorithm with the remaining points and compare with the result obtained with the secret point P. So we recover the secret point P, in the case where $l_{d+1} = 0$.

Case 2: $l_{d+1} = 1$. In this case, the $(d+1)^{th}$ iteration involves the addition in the Miller's algorithm. The doubling step is exactly the same, for the addition step, we have to consider $[2j+1]P = (X_{2j+1}, Y_{2j+1}, Z_{2j+1})$ knowing that $[j]P = (X_j, Y_j, Z_j)$, $[2j]P = (X_{2j}, Y_{2j}, Z_{2j})$ and $P = (X_P, Y_P, Z_P)$.

As we have that

$$h_2(X, Y) = Z_{2j+1}y\sqrt{\nu} - (Y_PZ_{2j}^3 - Y_{2j}Z_P^3)x - (X_PY_{2j}Z_P - X_{2j}Y_PZ_{2j}),$$

only the coordinate Z_{2j+1} appears in Step 7 of algorithm 2, and $Z_{2j+1} = Z_PZ_{2j}(X_PZ_{2j}^2 - X_{2j}Z_P^2)$.

At the $(d+1)^{th}$ iteration we have to calculate:

$$F_{d+1,P}(Q) = (F_{d,P}(Q))^2 \times h_1(Q)h_2(Q).$$

This time, the unknown values are X_j, Y_j, Z_j and X_P, Y_P, Z_P in the ratio $R = h_1(Q)h_2(Q)$. With the value of R and Q, and the theoretical expression of R, by identification we obtain four equations in the six unknown value. The elliptic curve equation will give us two others equation, as P and $[j]P \in E(\mathbb{F}_q)$.

$$\begin{cases} W_1(X_P, Y_P, Z_P, X_j, Y_j, Z_j) = \lambda_1, \\ W_2(X_P, Y_P, Z_P, X_j, Y_j, Z_j) = \lambda_2, \\ W_3(X_P, Y_P, Z_P, X_j, Y_j, Z_j) = \lambda_3, \\ W_4(X_P, Y_P, Z_P, X_j, Y_j, Z_j) = \lambda_4, \\ Y_P^2 - X_P^3 + 3X_P Z_P^4 - bZ_P^6 = 0 \\ Y_j^2 - X_j^3 + 3X_j Z_j^4 - bZ_j^6 = 0 \end{cases}$$

Where, $W_{\{1,2,3,4\}}()$ is a polynomial and $\lambda_{\{1,2,3,4\}} \in \mathbb{F}_q$. We get then a slightly more difficult system to solve, but giving us the coordinates of P directly, as coordinates of P are solution of the system. We can use the resultant method to find the coordinates of the point P. Considering two polynomials $S_1(X, Y)$ and $S_2(X, Y)$, if they are seen as polynomials in X with coefficients in $\mathbb{F}_q[Y]$, then the resultant of S_1 and S_1 is a polynomial in Y whose roots are solution of the system composed with $S_1(X, Y)$ and $S_2(X, Y)$. A succession of resultant will give an equation in only one unknown value. Experiments show that this equation is of degree 48, but this equation have at most 8 solutions. We can use the function `polresultant` in PariGP to compute the resultant.

Other coordinates To not overload the explanation, we consider only the case $l_{d+1} = 0$, the other case can be done exactly the same way.

Affine coordinates system has been studied in [23] and in [27]. The authors consider the case $k = 2$. With our method, we can find the secret point even if $k = 1$. The ratio of two consecutive iterations in Miller's algorithm will be as following: $R = \alpha x_j + \beta y_j + \gamma$, considering the elliptic curve equation we obtain a system:

$$\begin{cases} \alpha x_j + \beta y_j + \gamma = R \\ y_j^2 = x_j^3 + a x_P + b \end{cases}$$

Where, α, β and γ are known constants, and a, b define the elliptic curve equation. The first equation gives y_j in fonction of x_j. Injecting this equality in equation (5) gives a degree 4 equation in x_j. We know that this equation admits at least one solution, namely the value x_j. Consequently as it is an even equation, two solutions, and at worst 4 solutions. Once we have the possible values for x_j, we can find y_j. We obtain at most 8 couple of possible solutions, trying each one in a Miller computation will give the correct one. So for all the value of k, the fault attack can recover the secret point P.

The projective coordinates system has not been studied in literature. We just apply the same method as in the Jacobian case, we need k to be larger than 3 to find 3 equations in the 3 unknown coordinates. Considering the ratio R, we obtain a nonlinear system in coordinates of $[j]P$. The system is the following:

$$\begin{cases} Z_j^2 = \lambda_0 \\ Z_j(3X_j + aZ^2) = \lambda_1 \\ X(3X_j + aZ^2) + YZ = \lambda_3 \end{cases}$$

This system is quite easy to solve, we find Z_j from the first equation, two possible values. Then equation (7) give 2 possible values for X_j. Finally equation (8) gives 4 possible values for Y_j. As a result we obtain at most 16 triplets of possible solutions. We can also find the secret P, if $k = 2$, but we have to use the resultant method with the two equations obtained by identification in the base of $\mathbb{F}_{q^2}$, and the elliptic curve equation.

When the secret is the second argument of the pairing. If the point Q is secret during the pairing computation, all the system written above are linear in Q coordinates, so it can be recover very easily, by identification in the base of $\mathbb{F}_{q^k}$.

In [23], D. Page and F. Vercauteren introduce a fault attack against the particular case of the Duursma and Lee algorithm. The fault attack consists in modifying the number of iterations of the algorithm. This idea was completed in [15] in application to the Miller algorithm in Weierstrass coordinates and describe above. In [26] the authors conclude that if the secret is used as the first argument of the pairing computation, then it can not be found. This countermeasure is not one, as concluded in [15] and demonstrated. This three articles consider the case of Weierstrass coordinates. Recently, Edwards coordinates were introduced for computing pairings [7, 9, 18, 5].

Edwards curves became interesting for elliptic curve cryptography when it was proved by Bernstein and Lange in [8] that they provide addition and doubling formulas faster than all addition formulas known at that time. The advantage of Edwards coordinates is that the addition law can be complete and thus the exponentiation in Edwards coordinates is naturally protected against Side Channel Attacks.

Our contribution is to find out if Pairing Based Cryptography in Edwards coordinates is protected against fault attack. We show that a fault attack against the Miller algorithm in Edwards coordinates can be done through the resolution of a non linear system. In the paper [1] the authors introduced a fault attack against pairing in Edwards coordinates. The aim of the attack is to modify the least significant bit of l (the order of points) to 0, and to change the value of the number of iterations to 1. This kind of attack is very difficult to lead. Indeed this kind of fault attack implies that we can modify a register in the device forcing it to take a chosen value, which is a very strong assumption for a fault attack [2].

4 Background on Edwards curves

4.1 Definition and properties

Edwards showed in [16] that every elliptic curve E defined over an algebraic number field is birationally equivalent over some extension of that field to a curve given by the equation:

$$x^2 + y^2 = c^2(1 + x^2y^2). \tag{1}$$

In this paper, we use the notion of Twisted Edwards curves denoted $E_{a,d}$ and defined over a field $\mathbb{F}_q$, where q is a power of prime different from 2 :

$$E_{a,d} := \{(x, y) \in \mathbb{F}_q^2 \ \text{ such that } \ ax^2 + y^2 = 1 + dx^2y^2\}$$

They were introduced by Bernstein et al in [9] as a generalization of Edwards curves.

On a twisted Edwards curve, we consider the following addition law:

$$(x_1, y_1), (x_2, y_2) \to \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2}\right). \tag{2}$$

The neutral element of this addition law is $O = (0, 1)$. For every point $P = (x, y)$ the opposite element is $-P = (-x, y)$.

In [8], it was shown that this addition law is *complete* when d is not a square. This means it is defined for all pairs of input points on the Edwards curve with no exceptions for doubling, neutral element etc.

In the following sections we use projective coordinates. A projective point (X, Y, Z) satisfying $(aX^2 + Y^2)Z^2 = Z^4 + dX^2Y^2$ and $Z \neq 0$ corresponds to the affine point $(X/Z, Y/Z)$ on the curve $ax^2 + y^2 = 1 + dx^2y^2$. The Edwards curve has two points at infinity $(0 : 1 : 0)$ and $(1 : 0 : 0)$. The fastest formulas for computing pairings over Edwards curves are given in [5].

5 Pairings over Edwards curves

For efficiency reasons, we restrict the domain of the Tate pairing to a product of cyclic subgroups of order r on the elliptic curve. In general, the point P can be chosen such that $\langle P\rangle$ is the unique subgroup of order

r in $E(\mathbb{F}_q)$. In order to get a non-degenerate pairing, we take Q a point of order r, $Q \in E(\mathbb{F}_{q^k}) \backslash E(\mathbb{F}_q)$. Moreover, if the embedding degree is even, it was shown that the subgroup $\langle Q \rangle \subset E(\mathbb{F}_{q^k})$ can be taken so that the x-coordinates of all its points lie in $\mathbb{F}_{q^{k/2}}$ and the y-coordinates are products of elements of $\mathbb{F}_{q^{k/2}}$ with $\sqrt{\alpha}$, where α is a non square in $\mathbb{F}_{q^{k/2}}$ and $\sqrt{\alpha}$ is in $\mathbb{F}_{q^k}$ (see [19, 5] for details).

The same kind of considerations apply to Edwards curves and Twisted Edwards curves [5]. Using the trick of [19] the point $Q \in E(\mathbb{F}_{q^k})$ is written $(X_Q\sqrt{\alpha}; Y_Q; Z_Q)$ using a twist of degree 2. The element X_Q, Y_Q, Z_Q and α are in $\mathbb{F}_{q^{k/2}}$ and $\sqrt{\alpha} \in \mathbb{F}_{q^k}$. The point P is written (X, Y, Z) with X, Y and $Z \in \mathbb{F}_q$. In the following algorithm we used the denominator elimination trick [19].

Algorithm 2: Miller (P, Q, s)

Data: $s = (s_n \dots s_0)$(binary decomposition), $P \in \mathbb{G}_1$ $Q \in \mathbb{G}_2$;
Result: $f_{s,P}(Q) \in \mathbb{G}_3$;
$T \leftarrow P$, $f \leftarrow 1$, ;
for $i = n-1$ **to** 0 **do**
  $T \leftarrow [2]T$ and $f \longleftarrow f^2 \times g_d(Q)$;
  if $s_i = 1$ **then**
    $T \leftarrow T + P$ and $f \longleftarrow f \times g_a(Q)$;
  end
end
return $f = f_{s,P}(Q) \in \mathbb{F}^*_{q^k}$

Fig. 1. Miller's algorithm

The equation of the function g_d and g_a are described in the following Sections.

Doubling step We now take a look into the details of the computation of a Miller iteration. The doubling step is done for each iteration of Miller's algorithm. We note $T = (X_1, Y_1, Z_1)$. Following [5] the doubling formulas for $2T = (X_3, Y_3, Z_3)$ are:

$$\begin{aligned} X_3 &= (2X_1Y_1)(2Z_1^2 - aX_1^2 - Y_1^2), \\ Y_3 &= Y_1^4 - a^2X_1^4, \\ Z_3 &= (aA_1^2 + Y_1^2)(2Z_1^2 - aA_1^2 - Y_1^2). \end{aligned}$$

The function g_d has the following equations:

$$g_d(Q) = c_{Z^2}\eta'\sqrt{\alpha} + c_{XY}y_0 + c_{XZ}$$

where

$$\begin{aligned} \eta' &= \frac{Z_Q + Y_Q}{X_Q} \text{ and } y_0 = \frac{Y_Q}{Z_Q}, \\ c_{Z^2} &= X_1(2Y_1^2 - 2Y_1Z_1), \\ c_{XY} &= 2Z_1(Z_1^2 - aX_1^2 - Y_1Z_1), \\ c_{XZ} &= Y_1(2aX_1^2 - 2Y_1Z_1). \end{aligned}$$

Addition step This step is done only when the current bit of s is equal to 1. We note $T = (X_1, Y_1, Z_1)$ and $P = (X_P, Y_P, Z_P)$. Following [5] the addition formulas for $T + P = (X_3, Y_3, Z_3)$ in extended Edwards form are:

$$\begin{aligned} T_1 &= \frac{X_1 Y_1}{Z_1} \text{ and } T_P = \frac{X_P Y_P}{Z_P}, \\ X_3 &= (T_1 Z_P + T_P Z_1)(X_1 Y_P - X_P Y_1), \\ Y_3 &= (T_1 Z_P + T_P Z_1)(Y_1 Y_P + a X_1 X_P), \\ Z_3 &= (X_1 Y_P - X_P Y_1)(Y_1 Y_P + a X_1 X_P). \end{aligned}$$

The function g_a has the following equations:

$$g_a(Q) = c_{Z^2} \eta' \sqrt{\alpha} + c_{XY} y_0 + c_{XZ}$$

where

$$\begin{aligned} \eta' &= \frac{Z_Q + Y_Q}{X_Q} \text{ and } y_0 = \frac{Y_Q}{Z_Q}, \\ c_{Z^2} &= X_1 X_P (Y_1 Z_P - Y_P Z_1), \\ c_{XY} &= Z_1 Z_P (X_1 Z_P - Z_1 X_P + X_1 Y_P - Y_1 X_P) \\ c_{XZ} &= X_P Y_P Z_1^2 - X_1 Y_1 Z_P^2 + Y_1 Y_P (X_P Z_1 - X_1 Z_P). \end{aligned}$$

5.1 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, new technologies could allow for this attack [17].

We follow the scheme of attack described in [23],completed in [15] and describe in Section 3. 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 Miller's algorithm. The aim is to obtain a couple $(\tau, \tau + 1)$ of two consecutive values, corresponding to τ and $\tau + 1$ iterations during Miller's algorithm.

We denote the two results by $F_{\tau,P}(Q)$ and $F_{\tau+1,P}(Q)$. To conclude the attack, we consider the ratio $\frac{F_{\tau+1,P}(Q)}{F_{\tau,P}(Q)^2}$. By identification in the basis of $\mathbb{F}_{q^k}$, we are lead to a system which can reveal the secret point, which is described in Section 5.4.

The probability for obtaining two consecutive numbers is sufficiently large to make the attack possible [15]. In fact, for an 8-bits architecture only 15 tests are needed to obtain a probability larger than one half, $P(15, 2^8) = 0.56$, and only 28 for a probability larger than 0.9.

5.2 The τ^{th} step

We execute the Miller algorithm several times. For each execution we provoke disturbance in order to modify the value of $\log_2(s)$, until we find the result of the algorithm execution for two consecutive iterations, the τ^{th} and $(\tau + 1)^{th}$ iterations of algorithm 2. We denote the two results by $F_{\tau,P}(Q)$ and $F_{\tau+1,P}(Q)$.

After τ iterations, the algorithm 2 will have calculated $[j]P$. During the $(\tau + 1)^{th}$ iteration, it calculates $[2j]P$ and considering the value of the $(\tau + 1)^{th}$ bit of $\log_2(s)$, it either stops at this moment, or it calculates $[2j + 1]P$. In order to simplify the equations, we consider $k = 4$, but the method described can be generalised for $k \geq 4$. We denote $B = \{1, \gamma, \sqrt{\alpha}, \gamma\sqrt{\alpha}\}$ the basis used for written the elements of $\mathbb{F}_{q^k}$, this basis is constructed by a tower extensions [6].

5.3 Finding j

We know $\log_2(s)$, the order of the point Q,(as P and Q have the same order). By counting the number of clock cycles during the pairing calculation, we can find the number τ of iterations. Then reading the binary decomposition of $\log_2(s)$ gives us directly j. We consider that at the beginning $j = 1$, if $s_{n-1} = 0$ then $j \leftarrow 2j$, else $j \leftarrow 2j + 1$, and we go on, until we arrive to the $(n - 1 - \tau)^{th}$ bit of s. For example, let $s = 1000010000101$ in basis 2, and $\tau = 5$, at the first iteration we compute $[2]P$, at the second, as $s_{n-1} = 0$ we only make the doubling, so we calculate $[4]P$, it is the same thing for the second, third and fourth step so we have $[32]P$ in T.

At the fifth iteration, $s_{n-6} = 1$, then we make the doubling and the addition, so $j = 2 \times 32 + 1$, i.e. $j = 65$.

5.4 Curve and equations

In [23, 27, 15], only the affine coordinates case is treated. In [23, 27], a simple identification of the element in the basis of $\mathbb{F}_{q^k}$ gives the result. Here, the difference between these cases and Edwards coordinates is that we solve a nonlinear system.

Using the equation of the pairing calculation proposed in Section 5, we find a nonlinear system of k equations using the equality $g(Q) = R$, where $g(Q)$ defines the equation of update of f during Miller's algorithm. This system is solvable with the resultant method. To solve the system in Edwards coordinates we need k to be greater than 2.

The embedding degree. In order to simplify the equations, we consider case $k = 4$. As the important point of the method is the identification of the decomposition in the basis of $\mathbb{F}_{q^k}$, it is easily applicable when k is larger than 2.

We denote $B = \{1, \gamma, \sqrt{\alpha}, \gamma\sqrt{\alpha}\}$ the basis of $\mathbb{F}_{q^k}$, constructed by a tower extensions. The point $P \in E(\mathbb{F}_q)$ is given in Jacobian coordinates, $P = (X_P, Y_P, Z_P)$ and the point $Q \in E(\mathbb{F}_{q^k})$ also. As k is even, we can use a classical optimisation in pairing based cryptography which consists in using the twisted elliptic curve to write $Q = (X_Q\sqrt{\alpha}; Y_Q; Z_Q)$, with $X_Q,\ Y_Q,\ Z_Q$ and $\alpha \in \mathbb{F}_{q^2}$ and $\sqrt{\alpha} \in \mathbb{F}_{q^4}$, as described in Section 5.

Case 1: $s_{\tau+1} = 0$. We know the results of the τ^{th} and $(\tau + 1)^{th}$ iterations of Miller's algorithm, $F_{\tau,P}(Q)$ and $F_{\tau+1,P}(Q)$. We examine what happens during the $(\tau + 1)^{th}$ iteration.

The doubling step gives:

$$F_{\tau+1,P}(Q) = (F_{\tau,P}(Q))^2 \times g_d(Q)$$

As we suppose that $s_{\tau+1} = 0$, the additional step is not done. The return result of Miller's algorithm is $F_{\tau+1,P}(Q) = (F_{\tau,P}(Q))^2\, g_d(Q)$. We dispose of $F_{\tau,P}(Q)$, $F_{\tau+1,P}(Q)$ and the point $Q = (X_Q\sqrt{\alpha}; Y_Q; Z_Q)$, with X_Q, Y_Q and $Z_Q \in \mathbb{F}_{q^2}$. Recall that the coordinates of Q can be freely chosen and that we describe the attack for $k = 4$, this can easily be generalised for $k > 4$.
We can calculate the value $R \in \mathbb{F}^*_{q^k}$ of the ratio $\frac{F_{\tau+1,P}(Q)}{(F_{\tau,P}(Q))^2}$,

$$R = R_3\gamma\sqrt{\alpha} + R_2\sqrt{\alpha} + R_1\gamma + R_0,$$

where $R_3,\ R_2,\ R_1,\ R_0 \in \mathbb{F}_q$.

Moreover, we know the theoretical form of R in the basis $B = \{1, \gamma, \sqrt{\alpha}, \gamma\sqrt{\alpha}\}$ which depends of coordinates of jP and Q:

$$R = g_d(Q) = c_{Z^2}\eta'\sqrt{\alpha} + c_{XY}y_0 + c_{XZ},$$

where the c_{Z^2}, C_{XY}, c_{XZ} are in $\mathbb{F}_q$ and $\eta',\ y_0 \in \mathbb{F}_{q^2}$.

When the secret is the first argument
This position was presented as a counter measure to SCA in [26]. We know the point Q, thus the value of η' and $y_0 \in \mathbb{F}_{q^2}$ and their decomposition in $\mathbb{F}_{q^2}$, $\eta' = \eta'_0 + \eta'_1\gamma$, $y_0 = y_{00} + y_{01}\gamma$, where $(1, \gamma)$ defines the basis of $\mathbb{F}_{q^2}$. The elements c_{Z^2}, c_{XY} and c_{XZ} are in $\mathbb{F}_q$. Using the equality :

$$c_{Z^2}(\eta'_0 + \eta'_1\gamma)\sqrt{\alpha} + c_{XY}(y_{00} + y_{01}\gamma) + c_{XZ} = R_0 + R_1\gamma + R_2\sqrt{\alpha} + R_3\gamma\sqrt{\alpha}$$

by identification in the basis of $\mathbb{F}_{q^k}$, we obtain, after simplification, the following system of equations in $\mathbb{F}_q$:

$$\begin{cases} c_{XZ} = \lambda_2 \\ c_{XY} = \lambda_1 \\ c_{Z^2} = \lambda_0 \end{cases}$$

The value λ_0, λ_1 and λ_2 are known. With the resultant method we recover the coordinates of the secret point P. An example is given in the appendix.
When the secret is the second argument

We know the point P, thus the value of c_{Z^2}, c_{XY} and $c_{XZ} \in \mathbb{F}_q$. Using the equality :

$$c_{Z^2}(\eta'_0 + \eta'_1\gamma)\sqrt{\alpha} + c_{XY}(y_{00} + y_{01}\gamma) + c_{XZ} = R_0 + R_1\gamma + R_2\sqrt{\alpha} + R_3\gamma\sqrt{\alpha}$$

By identification in the basis of $\mathbb{F}_{q^k}$, we can recover the value η' and y_0, and thus the coordinate of the point Q.

$$\begin{cases} \eta'_0 = \frac{R_2}{c_{Z^2}} \text{ and } \eta'_1 = \frac{R_3}{c_{Z^2}}, \\ y_{00} = \frac{R_0 - c_{XZ}}{c_{XY}} \text{ and } y_{01} = \frac{R_1}{c_{XY}} \end{cases}$$

Indeed, once we have $y_0 \left(= \frac{Y_Q}{Z_Q}\right)$, using the elliptic curve we can find the value of $x_0 \left(= \frac{X_Q}{Z_Q}\right)$, and the coordinates of point Q.

Case 2: $s_{\tau+1} = 1$. In this case, the $(\tau + 1)^{th}$ iteration involves the addition in Miller's algorithm.

Thus, at the $(\tau+1)^{th}$ iteration, Miller's algorithm compute $F_{\tau+1,P}(Q) = (F_{\tau,P}(Q))^2 g_d(Q)g_a(Q)$. We could repeat the scheme of the previous case, and thanks the resolution of a non linear system, we can recover the secret point, whatever its position is. TO obtain the system, we juste have to develop the product $g_d(Q)g_a(Q)$. Using the polynomial reduction for the base of $\mathbb{F}_{p^{k/2}}$ and $\mathbb{F}_{p^k}$, we find the system by identification in this basis.

6 Vulnerability of pairings based on Miller's algorithm

6.1 Weil pairing

The Weil pairing is directly sensitive to the attack, as it is composed of two Miller's algorithm executions.

Indeed, the Weil pairing is defined as $e_W(P, Q) = \frac{F_P(Q)}{F_Q(P)}$.
We consider that the same modified l is used for the Miller Lite and the Full Miller part. We can apply the attack described above, we describe it with the simplified version of Miller's algorithm, the equations with the original Miller's algorithm A.1 are similar.

Let H_1 and H_2 be the equations used in the steps 5 and 7, in the Full Miller part. For example, $H_1(P)$ is the equation of the tangent at point T in the Full Miller's algorithm, and at this moment $T = [2j]Q$.

The ratio R between the result of two consecutive iterations is then $\frac{h_1(Q)}{H_1(P)} = R$, the system obtained after the identification of the element in the basis of $\mathbb{F}_{q^k}$ is composed of 4 equations with 6 unknown values. Using the elliptic curve equation it can be solved with the resultant method exactly as in Section 5.4. If the original algorithm is employed, the ratio R becomes: $\frac{h_1(Q)H_2(P)}{h_2(Q)H_1(P)}$, and the same method can be applied.

6.2 Tate and Ate pairings

The Tate and Ate pairings are constructed on the same model, one execution of Miller's algorithm plus a final exponentiation, for example the Tate pairing is $e_T(P,Q) = (F_P(Q))^{\frac{q^k-1}{l}}$. The first difficulty in attacking these two pairings with our scheme is to find a $(\frac{q^k-1}{l})^{th}$ root of the result.
The conclusion in [27] was that the final exponentiation is a natural countermeasure to the fault attacks. However, several method exist in literature in the microelectronic community to read the intermediary result during a computation on a smart card, or to override the final exponentiation.

We describe one of them, the scan attack against smart card, presented by D. Ellie and R. Karri in [14]. This scan attack consists of reading the intermediary state in the smart card. All smart cards contains an access, the scan chains, for testing the chip, which allows for this scan attack. The method of a scan attack is to scan out the internal state in test mode. This scanning gives us all the intermediary states of the smart card. So if the computation are stopped exactly before the exponentiation, a scan attack can give the result of Miller's algorithm.

Other attacks to override the final exponentiation exist, they are quitte difficult to realise but not unrealistic. For exemple, the under voltage technique [4] or the combination of the cipher instruction search attack realised by M. Khun and described in [4] which consists in recognizing enciphered instructions from their effect and the use of a focused ion beam workstation to access the EEPROM. A taxonomy of attackers has been done in [3], to realise the fault attack describe above, we consider that we were a class II attacker (knowledge insider). In order to perform the scan, under voltage and cipher instruction search, the attacker must be a class III, i.e. a funded organisation. Some material counter measures exist to prevent the modification of the memory by light or electromagnetic emissions, e.g. a shield. It is also possible to add a Hamming code at the end of the register to detect the fault [20], or to use an asynchrone clock.

7 Conclusion

We have presented in this paper the vulnerability to a fault attack of Miller's algorithm when it is used in an Identity Based Protocol. The attack consists in modifying the internal counter of an electronic device to provoke shorter iterations of the algorithm, we consider all the possible iterations. We describe precisely the way to realise this fault attack. We give the probability of obtaining two consecutive iterations, and we find out that a small number of tests are needed to find two consecutive results.

We consider the case when the point P, the first argument of Miller's algorithm, is secret. The result of the fault attack is a nonlinear system, whose variables are coordinates of P and Q. We describe the method to solve this nonlinear system. If the secret is the second point Q, our scheme is also applicable and the nonlinear system becomes a linear system, which is easier to solve. Thus, whatever the position of the secret point, our fault attack will recover it. Moreover, we have described the resolution in Jacobian coordinates, but the scheme is the same in affine, projective and Edwards coordinates and we explain how to solve it.

Then, we have analised the weakness to this fault attack of pairing based on Miller's algorithm. The Weil pairing is directly sensitive to this attack. The Tate and Ate pairings present a final exponentiation which previously protect them against this fault attack. We introduce attacks used for a while in the microelectronic community to override the final exponentiation in the Tate and Ate pairings. The scan attack, the under voltage attack and the cipher instruction search are three different attacks which allow the attacker to get the result of the Miller iteration before the final exponentiation.

As a conclusion, we can say that the fault attack is a threat against Miller's algorithm, and consequently to pairings based on Miller's algorithm.

References

1. Santosh Ghosh, Debdeep Mukhopadhyay and Dipanwita Roy Chowdhury: Fault Attack, Countermeasures on Pairing Based Cryptography. In International Journal Network Security, p 21-28, 2011.
2. Johannes Blmer, Jean-Pierre Seifert: Fault Based Cryptanalysis of the Advanced Encryption Standard (AES). Financial Cryptography 2003: p 162-181.
3. Abraham D.G., Dolan G.M., Double G.P. and Stevens J.V.: Transaction Security System. IBM Systems Journal, vol 30, p 206 – 229, 1991.
4. Anderson R. and Kuhn M.: Tamper Resistance – a Cautionary Note. The Second USENIX Workshop on Electronic Commerce Proceedings, p 1–11, Okland, California 1996.
5. C. Arne, T. Lange, M. Naehrig and C. Ritzenhaler: *Faster Pairing Computation of the Tate pairing*, Cryptology ePrint Archive, Report 2009/155, http://eprint.iacr.org/2009/155,2009.
6. J.C.Bajard and N.El Mrabet: Pairing in cryptography: an arithmetic point de view. Advanced Signal Processing Algorithms, Architectures, and Implementations XVI, part of SPIE, August 2007.
7. D. J. Bernstein and T. Lange: *Performance evaluation of a new side channel resistant coordinate system for elliptic curves*, cr.yp.to/antiforgery/newelliptic-20070410.pdf, 2007.
8. D. J. Bernstein and T. Lange: *Faster additions and doubling on elliptic curves*, ASIACRYPT 2007, vol 4833, p 29–50, Springer-Verlag, 2007.
9. D. J. Bernstein, P. Birkner, M. Joye, T. Lange and C. Peters: *Twisted Edwards curves*, Afrycacrypt 2008, p 389–405,Springer-Verlag.
10. Boneh D., Franklin M.: Identity-based encryption from the Weil pairing. Extended abstract in Crypto 2001, LNCS 2139, pp. 213-229, 2001
11. Brier E., Joye M.: Point multiplication on elliptic curves through isogenies. AAECC 2003, LNCS., vol. 2643, 2003, p. 43–50.
12. Boneh D., DeMillo R. and Lipton R.: On the importance of checking cryptographic protocols faults. Advances in Cryptology Eurocrypt 1997, Lecture Notes in Comput. Sci., vol. 1233, Springer-Verlag, Berlin, 1997, p. 37–51.
13. Cohen, H., Frey, G. (editors): Handbook of elliptic and hyperelliptic curve cryptography. Discrete Math. Appl., Chapman & Hall/CRC (2006)
14. Yang B., Wu K. and Karri R.: Scan Based Side Channel Attack on Dedicated Hardware Implementation of Data Encryption Standard. Test Conference 2004, proceedings ITC 2004, p. 339 – 344.
15. N. El Mrabet, *What about Vulnerability to a Fault Attack of the Miller's Algorithm During an Identity Based Protocol?*, Advances in Information Security and Assurance 2009, LNCS 5576, p 122–134, Springer-Verlag.
16. Edwards H.: A normal Form for Elliptic Curve. Bulletin of the American Mathematical Society Vol. 44, Number 3, July 2007.
17. Habing D.H: The Use of Lasers to Simulate Radiation-Induced Transients in Semiconductor Devices and Circuits. IEEE Transactions On Nuclear Science, vol.39, p. 1647-1653, 1992.
18. Ionica S. and Joux A.: *Another Approach to Pairing Computation in Edwards Coordinates*, INDOCRYPT '08: Proceedings of the 9th International Conference on Cryptology in India, Springer-Verlag
19. Koblitz N., Menezes A. J.: Pairing-based cryptography at high security levels. Proceedings of the Tenth IMA International Conference on Cryptography and Coding, Springer-Verlag, LNCS 3796, 2005, p. 13-36;
20. Macwilliams F.J. and Sloane N.J.A.: The Theory of Error-Correcting Codes II. North-Holland Mathematical Library, vol. 16, North-Holland, Amsterdam, 1998.
21. Menezes A.: An introduction to pairing-based cryptography. Notes from lectures given in Santander, Spain, 2005, http://www.cacr.math.uwaterloo.ca/ ajmeneze/publications/pairings.pdf
22. Miller V.: The Weil pairing and its efficient calculation. Journal of Cryptology, 17 (2004), p. 235-261.
23. Page Dan and Vercauteren Frederik: Fault and Side Channel Attacks on Pairing based Cryptography. IEEE Transactions on Computers, vol. 55, no. 9, pp. 1075-1080, Sept., 2006.
24. PARI/GP, version `2.1.7`, Bordeaux, 2005, http://pari.math.u-bordeaux.fr/

25. Shamir A.: Identity Based Cryptosystems and Signature Schemes. Advances in Cryptology Crypto '84, LNCS, Vol. 196, pp 47-53, 1984
26. Whelan C. and Scott M.: Side Channel Analysis of Practical Pairing Implementation: Which Path is More Secure ?. Lecture Notes in Computer Science, Volume 4341 Springer-Verlag ed.VietCrypt 2006, 25-SEP-06 - 28-SEP-06, Hanoi, Vietnam, 99 - 114.
27. Whelan C. and Scott M.: The Importance of the Final exponentiation in Pairings when considering Fault Attacks. Lecture Notes in Computer Science, Volume 4575/2007 Springer-Verlag ed. Pairing07, Tokyo.

A Pairing algorithm

A.1 Original Miller's algorithm

Algorithm 3: Miller(P, Q, n)

Data: $n = (n_l \dots n_0)$(radix 2 representation), $P \in G_1(\subset E(\mathbb{F}_p))$ and $Q \in G_2(\subset E(\mathbb{F}_{p^k}))$;
Result: $F_P(Q) \in G_3(\subset \mathbb{F}^*_{p^k})$;
$T \leftarrow P$;
$f_1 \leftarrow 1$;
$f_2 \leftarrow 1$;
for $i = l - 1$ **to** 0 **do**
    $T \leftarrow [2]T$;
    $f_1 \longleftarrow {f_1}^2 \times h_1(Q)$;
    $f_2 \longleftarrow {f_2}^2 \times h_2(Q)$ (where $Div(\frac{h_1}{h_2}) = 2(T) - ([2]T) - P_\infty$);
    if $n_i = 1$ **then**
        $T \leftarrow T \oplus P$;
        $f_1 \longleftarrow f_1 \times h_1(Q)$;
        $f_2 \longleftarrow f_2 \times h_2(Q)$ (where $Div(\frac{h_1}{h_2}) = (T) + D_P - ((T) \oplus D_P) - P_\infty$);
    end
end
return $\frac{f_1}{f_2}$

B Example

We compute this exemple using PariGP [24].

$k = 4$

$p = 680241220348515477477949259894191902369939655391504568151207016994661689050587617052536187229749$ (319 bit)

$E : Y^2 = X^3 + 3XZ^4$

$card(E(Fp)) = 680241220348515477477949259894191902369939655390338170945836123217606411022317222264735061564936$ (319 bit)

$l = 1166397205370893777055276948271688598347500051217$ (160 bit)

$P = [12, 48, 2]$

To construct $\mathbb{F}_{q^k}$, we use the element $a \in \mathbb{F}_{q^k}$ such that $a^4 = 2$

$Q = [a^2, 100512916629999457534083547932541900367294743582692206264363320753064$
$8550419942663119715734888636 * a]$

We stop Miller's algorithm at the 46^{th} iteration.

The ratio R is:
$337259586468080683488399539046229874795973242322339094577672485334431934756557508827480079490557 \times a^2 + 624752062739857009467545836695395127071983321507188174321543153770228940196002139337802972603156 \times a + 290466295014915698560156774394046481806928474873516631676810692056674915620683567856541417846103$

Written down the equations we obtain the following system:

$$\begin{cases} Y_j Z_j^3 = \lambda_2 = 526421537150286598896703298483149985967580207398544590133171776285079049014186714839235255813297 \\ 3Z_j^2(X_j^2 - Z_j^4) = \lambda_1 = 475148309417543639369621311346136013833460391400891264127160029381884835668719747434801612007813 \\ 3X_j(X_j^2 - Z_j^4) - 2Y_j^2 = \lambda_0 = 3897749253335997789177924855001454205630111805179879364743963249379867734299040491959947693836 46 \end{cases}$$

$$(\lambda_0^2 - 9\lambda_1^2)Z^{12} - (4\lambda_0\lambda_2^2 + 9\lambda_1^3)Z^6 + 4\lambda_1^4 = 0$$

The function `factorff(f(Z),p)` in PariGP gives six different solutions in Z.

$[Mod(16607281758720556185091528400750991423074654651558232852250451793435949329630662 6253218820537301, p), Mod(18612943962395238829049904990175006578277838877750622865464857 92246870365463314074459080227966 08, p), Mod(32803896313735752733653492598493192235641472009841601097405391983561515920794958335340934389584 0, p), Mod(35220225721115795014141433390925998001352493529308855717715309715904652984263803369912684333390 9, p), Mod(49411178072456308918745020999244183658716126661399833949655843776997465250425620960662816443314 1, p), Mod(51416840276130991562703397588668198813919310887592223962870249906030219575428099079931736669244 8, p)]$

Among all the possible triplet the six are on the elliptic curve. We find the inverse modulo p of 46 and compute the six possibilities for P. Then we just have to perform six Miller's algorithms and compare with the result obtained with the secret point P.