What about Vulnerability to a Fault Attack of the Miller Algorithm during an Identity Based Protocol?

Nadia El Mrabet

What About Vulnerability to a Fault Attack of the Miller's Algorithm During an Identity Based Protocol?

Nadia El Mrabet

LIRMM Laboratory, I3M, CNRS, University Montpellier 2,
161, rue Ada, 34 392 Montpellier, France
elmrabet@lirmm.fr

Abstract. We complete the study of [16] and [20] about the Miller's algorithm. The Miller's algorithm is a central step to compute the Weil, Tate and Ate pairings. The aim of this article is to analyse the weakness of the Miller's algorithm when it undergoes a fault attack. We prove that the Miller's algorithm is vulnerable to a fault attack which is valid in all coordinate systems, through the resolution of a nonlinear system. We show that the final exponentiation is no longer a counter measure to this attack for the Tate and Ate pairings.

Keywords: Miller's algorithm, Identity Based Cryptography, Fault Attack.

1 Introduction

In 1984, A. Shamir challenged the cryptographer community to find a protocol based on the user's identity [18]. 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 [4]. The general scheme of an identity based encryption is described in [4]. 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 ([16], [19] and [20]).

In [16], 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 the Miller's algorithm, and we describe a way to realise this fault injection.

In [20], 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 deduce that the Miller's algorithm is not vulnerable

J.H. Park et al. (Eds.): ISA 2009, LNCS 5576, pp. 122–134, 2009.

to a fault attack, because the system obtained after the fault attack is nonlinear and then impossible to solve. In [19] they conclude 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 the 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 the 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 the Miller's algorithm can be done through the resolution of a nonlinear system. This demonstration will be followed by discussion about the weakness to this fault attack of pairings based on the 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 [6].

The outline of this article is as follow. First we will give a short introduction to pairing and to the Miller's algorithm in Section 2. Section 3 presents our fault attack against the Miller's algorithm, Section 4 analyses the vulnerability of pairings using the Miller's algorithm as a central step, finally, we give our conclusion in Section 5.

2 Pairings and the 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:* $e : G_1 \times G_2 \rightarrow G_3$.

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. The Miller's algorithm [15] is the central step for Weil, Tate and Ate pairings computations.

2.2 Miller's Algorithm

The following description of the Miller's algorithm is referenced in [7, chapter 16].

The 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 $[l]P$. The Miller's algorithm is based on the notion of divisors. We only give here the essential elements for the pairing computation.

The 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 1 is a simplified version of the Miller's algorithm (see [3]). 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 4.1 that the conclusions for the original algorithm are the same.

3 Fault Attack Against the 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 the 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 [10].

3.1 Description of the Fault Attack

We complete the scheme of attack described in [16] to use it against the Miller's algorithm. In [16] the attack consists in modifying the number of iterations. We complete the idea of [16] by giving a precise description of the attack, by computing the probability of finding suitable number of iterations and by adapting it to the 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 the 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 the 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 the Miller's algorithm. For example the disturbance can be induced by a laser [2]. Lasers are today thin enough to make this attack realistic [10]. 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 the Miller's algorithm, we give the probability to obtain such couple in Section 3.2.

3.2 The d^{th} Step

We execute the 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 1. 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:

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

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 1 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 [16] and [20], 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 the Miller's algorithm is efficient. We describe it for example in Jacobian coordinates. The difference between with the cases described in [16] and [20] 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 3.3 can be solve "by hand", without the resultant method described in Section 3.3. 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 [3].

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

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

As we make random modifications of l during the fault attack, we suppose that we stop the 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: $\boldsymbol{l_{d+1} = 0}$. We know the results of the d^{th} and $(d+1)^{th}$ iterations of the 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 the 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:

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

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 the 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:

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

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 $\mathbb{F}_{q^k}$.

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

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

$$\begin{cases} Y_jZ_j^3 = \lambda_2 & (1) \\ Z_j^2(X_j^2 - Z_j^4) = \lambda_1 & (2) \\ 3X_j(X_j^2 - Z_j^4) + 2Y_j^2 = \lambda_0 & (3) \end{cases}$$

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 [17], 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 $24 = 12 \times 2 \times 1$ 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: $\boldsymbol{l_{d+1} = 1}$. 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

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

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

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 $[j]P \in E(\mathbb{F}_q)$.

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

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.

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 $\mathbb{F}_{q^k}$.

4 Vulnerability of Pairings Based on the Miller's Algorithm

4.1 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 the 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 3.3. 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.

4.2 Tate and Ate Pairings

The Tate and Ate pairings are constructed on the same model, one execution of the 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 [20] 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 [8]. 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 the 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 [2] or the combination of the cipher instruction search attack realised by M. Khun and described in [2] 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 [1], 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 [13], or to use an asynchrone clock.

5 Conclusion

We have presented in this paper the vulnerability to a fault attack of the 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 the 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 the 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 the Miller's algorithm, and consequently to pairings based on the Miller's algorithm.

References

1. Abraham, D.G., Dolan, G.M., Double, G.P., Stevens, J.V.: Transaction Security System. IBM Systems Journal 30, 206–229 (1991)
2. Anderson, R., Kuhn, M.: Tamper Resistance – a Cautionary Note. In: The Second USENIX Workshop on Electronic Commerce Proceedings, Okland, California, pp. 1–11 (1996)

3. Bajard, J.C., El Mrabet, N.: Pairing in cryptography: an arithmetic point de view. In: Advanced Signal Processing Algorithms, Architectures, and Implementations XVI, part of SPIE (August 2007)
4. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
5. Brier, E., Joye, M.: Point multiplication on elliptic curves through isogenies. In: Fossorier, M.P.C., Høholdt, T., Poli, A. (eds.) AAECC 2003. LNCS, vol. 2643, pp. 43–50. Springer, Heidelberg (2003)
6. Boneh, D., DeMillo, R., Lipton, R.: On the importance of checking cryptographic protocols faults. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 37–51. Springer, Heidelberg (1997)
7. Cohen, H., Frey, G. (eds.): Handbook of elliptic and hyperelliptic curve cryptography. Discrete Math. Appl. Chapman & Hall/CRC, Boca Raton (2006)
8. Yang, B., Wu, K., Karri, R.: Scan Based Side Channel Attack on Dedicated Hardware Implementation of Data Encryption Standard. In: Test Conference 2004, proceedings ITC 2004, pp. 339–344 (2004)
9. Edwards, H.: A normal Form for Elliptic Curve. Bulletin of the American Mathematical Society 44(3) (2007)
10. Habing, D.H.: The Use of Lasers to Simulate Radiation-Induced Transients in Semiconductor Devices and Circuits. IEEE Transactions On Nuclear Science 39, 1647–1653 (1992)
11. Ionica, S., Joux, A.: Faster Pairing Computation on Edwards Curves. To appear at Indocrypt 2008 conference (2008)
12. Koblitz, N., Menezes, A.J.: Pairing-based cryptography at high security levels. In: Smart, N.P. (ed.) Cryptography and Coding 2005. LNCS, vol. 3796, pp. 13–36. Springer, Heidelberg (2005)
13. Macwilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes II. North-Holland Mathematical Library, vol. 16. North-Holland, Amsterdam (1998)
14. 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
15. Miller, V.: The Weil pairing and its efficient calculation. Journal of Cryptology 17, 235–261 (2004)
16. Dan, P., Frederik, V.: Fault and Side Channel Attacks on Pairing based Cryptography. IEEE Transactions on Computers 55(9), 1075–1080 (2006)
17. PARI/GP, version 2.1.7, Bordeaux (2005), http://pari.math.u-bordeaux.fr/
18. Shamir, A.: Identity Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
19. Whelan, C., Scott, M.: Side Channel Analysis of Practical Pairing Implementation: Which Path is More Secure? In: Nguyên, P.Q. (ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 99–114. Springer, Heidelberg (2006)
20. Whelan, C., Scott, M.: The Importance of the Final Exponentiation in Pairings When Considering Fault Attacks. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 225–246. Springer, Heidelberg (2007)

A Pairing Algorithm

A.1 Original Miller's Algorithm

Algorithm 2: 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
1   $T \leftarrow [2]T$
    $f_1 \longleftarrow {f_1}^2 \times h_1(Q)$
    $f_2 \longleftarrow {f_2}^2 \times h_2(Q)$ (where $Div(\frac{h_1}{h_2}) = 2(T) - ([2]T) - P_\infty)$
2   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 [17].

$k = 4$
$p = 6802412203485154774779492598941919023699396553915045681512070169946616890505876170525361872297 49$ (319 bit)

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

$card(E(Fp)) = 68024122034851547747794925989419190236993965539033817094583612321760641102231722226473506156 4936$ (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 $a^4 = 2$

$Q = [a^2, 10051291662999945753408354793254190036729474358269220626436332075306485504199426631197157348 8636 * a]$
We stop the Miller's algorithm at the 46th iteration.

The ratio R is:
33725958646808068348839953904622987479597324232233
9094577672485334431934756557508827
480079490557 $\times a^2 +$ 62475206273985700946754583669539512
7071983321507188174321543153770228940196002139337802972603156$\times a+$2904662950
149156985601567743940464818069284748735166316768106
92056674915620683567856541417846103

Written down the equations we obtain the following system:

$$\begin{cases} Y_jZ_j^3 = \lambda_2 = 52642153715028659889670329848 \\ \qquad 3149985967580207398544590133171776 \\ 285079049014186714839235255813297 \\ 3Z_j^2(X_j^2 - Z_j^4) = \lambda_1 = 47514830941754363936962131134 \\ \qquad 6136013833460391400891264127160029 \\ 381884835668719747434801612007813 \\ 3X_j(X_j^2 - Z_j^4) - 2Y_j^2 = \lambda_0 = 389774925333599778917792485 \\ \qquad 50014542056301118051798793647439 \\ 6324937986773429904049195994769383646 \\ \end{cases}$$

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

The function `factorff(f(Z),p)` in PariGP gives six different solutions in Z.
$[Mod($1660728175872055618509152840075099142307465465155823285225045
17934359493296306626253218820537301$,p),$
$Mod($18612943962395238829049904990
175006578277838877750622865464857
922468703654633140744590802
2796608$,p), Mod($3280389631373575273365349259849319
223564147200984160109740
53919835615159207949583353409343895840$,p), Mod($3522022
572111579501414143339092
599800135249352930885571771530971590465298426380336991
26843333909$,p), Mod($494111
78072456308918745020999244183658716126661399833949655
843776997465250425620960662
8164433141$,p), Mod($51416840276130991562703397588668198
813919310887592223962870249
9060302195754280990799317366692448$,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.