A survey of Fault Attacks in Pairing Based Cryptography

Nadia El Mrabet, Jacques Jean-Alain Fournier, Louis Goubin, Ronan Lashermes

A survey of Fault Attacks in Pairing Based Cryptography

Nadia El Mrabet[1], Jacques J.A. Fournier[2], Louis Goubin[3], and Ronan Lashermes[2,3]

[1] LIASD - Université Paris 8
elmrabet@ai.univ-paris8.fr
[2] CEA-TechReg
ronan.lashermes@cea.fr, jacques.fournier@cea.fr
[3] UVSQ-PRiSM
louis.goubin@prism.uvsq.fr

Abstract. The latest implementations of pairings allow efficient schemes for Pairing Based Cryptography. These make the use of pairings suitable for small and constrained devices (smart phones, smart cards...) in addition to more powerful platforms. As for any cryptographic algorithm which may be deployed in insecure locations, these implementations must be secure against physical attacks, and in particular fault attacks. In this paper, we present the state-of-the-art of fault attacks against pairing algorithms, more precisely fault attacks against the Miller algorithm and the final exponentiation which are the two parts of a pairing calculation.[4]

Keywords: Pairing based cryptography, Miller's algorithm, fault attacks.

1 Introduction

In 1984, A. Shamir challenged the cryptography community to find a protocol based on the user identity [40]. This challenge was solved nearly twenty years later by D. Boneh and M. Franklin. In 2003, D. Boneh and M. Franklin created an identity-based encryption (IBE) scheme based on pairings [10]. The general scheme of an identity based encryption is described in [10] and several protocols based on pairings have been developed since [28]. A feature of Identity Based protocols is that a computation of a pairing involving the private key and the ciphertext is performed in order to decipher a message. A pairing is a bilinear map e taking as inputs two points P and Q of an elliptic curve. The pairing computation gives the result $e(P, Q)$. Several pairings have been described in the literature. The Weil and the Tate pairing were developed [42] without any considerations for the efficiency of the computation. Once pairings were used to construct protocols, cryptographers sought more efficient algorithms. In chronological order, the Duursma and Lee algorithm [15], the Eta [6], Ate, twisted Ate [25], optimal pairings [44] and pairing lattices [24] were discovered. Recently, a construction of pairing over a general abelian variety was proposed in [33]. The latest implementations results [2, 5, 11, 22, 38] of pairing computations are fast enough to consider the use of pairing based protocols in embedded devices. Consequently, it seems fair to wonder if pairing based protocols involving a secret are secure against physical attacks in general and fault attacks in particular. Side channel attacks have been analysed in [47] where they conclude that an efficient countermeasure would be to set the secret as the first parameter. In [30], Kim *et al.* analyse the effect of side channel attacks against pairings over binary fields. According to the recent work of Joux [27], pairings over binary fields are not secure. We focus here on fault attacks against pairings in fields with a large prime characteristic.

[4] "The final publication is available at Springer via http://dx.doi.org/10.1007/s12095-014-0114-5".

Since 2006, several fault attacks against pairings have been proposed. In this article, we will present what are in our opinion the most significant ones. For each attack, we assume that the pairing is used during an Identity Based Protocol. The secret point is stored into a smart card or an electronic device that can be attacked with fault attacks. The location of the secret is in practice not important. Indeed, the equations that leak information about the secret can provide information whether the secret is the first or the second parameter. Often, the attack is easier when the secret is the second parameter. That is why we consider the cases where the first parameter is the secret argument.

The article is organized as follows. We briefly recall the background necessary to understand pairings and IBE in Section 2. The first fault attack against a pairing was proposed by Page and Vercauteren [36] and is presented in Section 3.1. Then, we describe the adaptations of the previous attack against the Miller algorithm in Section 3.2. Whelan et Scott [46] highlighted the fact that pairings without a final exponentiation are more sensitive to a sign change fault attack. After that, El Mrabet [16] generalized the attack of Page and Vercauteren to the Miller algorithm used to compute all the recent optimizations of pairings. Another method is adopted in [3], based on instruction skips, and presented in Section 3.2. In [31], Lashermes *et al.* proposed a fault attack against the final exponentiation during a Tate-like pairing. Their attack is described in Section 4. In Section 5, we present the attack against the pairing defined over a general abelian variety. Finally, we conclude in Section 6.

2 Background on pairings

In this section, the definition and construction of a pairing is presented. We briefly recall the interest of pairing based cryptography and we present the security issues.

2.1 Short introduction to pairings

We consider pairings defined over an elliptic curve $E(\mathbb{F}_p)$, for p a large prime number. The point at infinity of E is denoted P_∞. In order to illustrate the attacks, we consider the short Weierstrass equation of E which is in Jacobian coordinates: $Y^2 = X^3 + aXZ^4 + bZ^6$, with a, $b \in \mathbb{F}_p$.

Remark 1. Pairings can be defined for $E(\mathbb{F}_q)$, with q a power of a prime number. In practice the small characteristics are not secure [27] so we do not take these cases into consideration. Today, a pairing is not constructed over $\mathbb{F}_q$, for q a power of a medium prime. As far as we know it has never been realized in practice but it could be interesting for efficiency reasons. Nevertheless, the attacks we describe here can be adapted to these cases.

Remark 2. The equation of E does not influence the scheme of the attacks described in this paper. We describe them considering the short Weierstrass equation, but the same attack can be adapted for any other kind of equation like Edwards [21] for instance. Furthermore, a recent work [33] proposes the computation of pairings over theta functions. We show in Section 5 that the pairing defined over theta functions are sensitive to fault attacks too. The attack is also independent from the choice of the coordinates. As the pairing is often more efficient in Jacobian coordinates, we choose to work with those. But the same attacks are effective for affine, projective or any other coordinates [17].

Let r be a prime number dividing $\mathrm{card}(E(\mathbb{F}_p))$. Let k be the smallest integer such that r divides $(p^k - 1)$. The integer k is called the embedding degree of E with respect to r. Let $\mathbb{G}_1 \subset E(\mathbb{F}_p)$, $\mathbb{G}_2 \subset E(\mathbb{F}_{p^k})$, $\mathbb{G}_3 \subset \mathbb{F}^*_{p^k}$, be three groups of order r.

Definition 1. *A pairing is a bilinear and non degenerate function:* $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$.

Initially, pairings were exclusively defined as mathematical functions, e.g. the Weil and the Tate pairings [42]. With the development of pairing based cryptography, researchers were aiming for an efficient implementation and this brought out the definition of the Duursma and Lee algorithm [15], the Eta [6], Ate, twisted Ate [25], optimal pairing [44] and pairing lattices [24] (in chronological order and from the less to the most efficient pairing). The Weil and the Tate pairings are constructed using the Miller algorithm [34], as for the Ate, twisted Ate, optimal pairing and pairing lattices. The Duursma and Lee algorithm and the Eta pairing are not based on the Miller algorithm. The most efficient pairings are constructed on the Tate model: Ate, twisted Ate, optimal pairing and pairing lattices. So we only recall here the definition of the reduced Tate pairing. A more complete definition of the Tate pairing can be found in [7, §IX.5].

Definition 2. *Let* $E(\mathbb{F}_p)$ *be an elliptic curve over the finite field* $\mathbb{F}_p$ *for* p *a prime number,* r *a divisor of card*$(E(\mathbb{F}_p))$*,* k *the embedding degree of* E *relatively to* r*. Let* $\mathbb{G}_1 = E(\mathbb{F}_p)[r]$*,* $\mathbb{G}_2 = E(\mathbb{F}_{p^k})/rE(\mathbb{F}_{p^k})$ *and* $\mathbb{G}_3 = \{\mu \in \mathbb{F}_{p^k} \text{ such that } \mu^r = 1\}$*. The reduced Tate pairing is defined as*

$$\begin{aligned} e_T : \mathbb{G}_1 \times \mathbb{G}_2 &\to \mathbb{G}_3, \\ (P, Q) &\to f_{r,P}(Q)^{\frac{p^k-1}{r}} \end{aligned}$$

where $f_{r,P}(Q)$ *is the Miller function defined by the divisor* $D = r(P) - (rP) - (r-1)(P_\infty)$.

The Miller function is computed through the Miller algorithm presented in Algorithm 1 and referenced in [12]. The Miller algorithm is constructed on the double and add scheme using the construction of rP. The Miller algorithm is based on the notion of divisors. We only give here the essential elements for the pairing computation.

The Miller algorithm constructs the rational function $f_{r,P}$ associated to the point P, where P is a generator of $\mathbb{G}_1$; and at the same time, it evaluates $f_{r,P}(Q)$ for a point $Q \in G_2 \subset E(\mathbb{F}_{p^k})$.

Algorithm 1: Miller(P, Q, r)

Data: $r = (r_n \dots r_0)$(radix 2 representation), $P \in \mathbb{G}_1 (\subset E(\mathbb{F}_p))$ and $Q \in \mathbb{G}_2 (\subset E(\mathbb{F}_{p^k}))$;
Result: $f_{r,P}(Q) \in \mathbb{G}_3 (\subset \mathbb{F}^*_{p^k})$;

```
T ← P ;
f ← 1 ;
for i = n − 1 to 0 do
    f ⟵ f^2 × h_1(Q), h_1(x, y) is the equation of the tangent at the point T;
    T ← [2]T;
    if r_i = 1 then
        f ⟵ f × h_2(Q), h_2(x, y) is the equation of the line (PT);
        T ← T + P ;
    end
end
return f
```

The Ate, twisted Ate, optimal pairing and pairing lattices are constructed on the model of the Tate pairing. They are composed of one Miller algorithm execution followed by a final exponentiation. They differ by their number of iterations and sometimes by the role of P and Q. More details can be found in [24, 25, 44].

The final exponentiation is used to ensure the uniqueness of the resulting value of two equal pairing computations (e.g. $e(P,[2]Q) = e([2]P,Q)$). The final exponentiation maps the result of the Miller algorithm into the group formed by the r^{th} roots of unity in $\mathbb{F}_{p^k}^*$. The exponentiation $f^{\frac{p^k-1}{r}}$ is often used [39].

The most useful property in pairing based cryptography is its bilinearity: $e([n]P,[m]Q) = e(P,Q)^{nm}$, with n and m integers. Pairings can be used to construct several protocols and in particular those allowing identity based cryptography [28]. During an identity based protocol, one of the argument of the pairing is secret, it can be the point P or the point Q. All the other parameters like the elliptic curve $E(\mathbb{F}_p)$, r, k the embedding degree and the implementation of the pairing are public. We present here existing fault attacks that allow the recovery of the secret.

2.2 Identity based cryptography

The first use of pairings was for the cryptanalysis of Elliptic Curves Cryptography: the Weil pairing shifts the discrete logarithm problem (DLP) from an elliptic curve to a finite field. After that the pairing was used to improve existing protocols as tri-partite Diffie Hellman key exchange [26], and to construct original protocol like identity based encryption [10, 7].

The aim of identity based encryption is that a person λ can use his own identity λ as a public key. His private key would be sent to him by a trusted authority T. This trusted authority will have all the private keys related to the identity based protocol. In this protocol, there is no need to maintain a public directory of all public keys linked to the identities. The general scheme of an identity based key exchange is the following.

The public data is an elliptic curve E over a finite field $\mathbb{F}_q$, a pairing e, and a hash function H, this hash function associates a point of $E(\mathbb{F}_q)$ to an identity: $H : \{Identity\} \rightarrow E(\mathbb{F}_q)$. We consider that two persons, Alice and Bob, want to have a common secret key in order to have a secure communication.

With the public data, Alice can compute $Q_B = H(Bob)$ the public key of Bob, and Bob can compute $Q_A = H(Alice)$ the public key of Alice.
Alice and Bob ask the trusted authority to receive their secret key. The secret key is a point of $E(\mathbb{F}_q)$.

The trusted authority chooses s, its own secret key, then it generates $P_A = [s]Q_A$ the secret key of Alice, and $P_B = [s]Q_B$ the secret key of Bob.

Then Alice (respectively Bob) can compute $e(P_A, Q_B)$ (resp. $e(Q_A, P_B)$). By bilinearity, Alice and Bob possess a shared key: $e(Q_A, Q_B)^s$. Indeed:

$$e([s]H(A), H(B)) = e(H(A), [s]H(B)) = e(H(A), H(B))^s$$

The particularity of identity based cryptography is that a potential spy knows the algorithm used, the number of iterations and the exponent. The secret is only one of the argument of the pairing. The secret key does not influence the execution time nor the number of iterations in the algorithm, which is different from RSA protocols.

In this simple protocol, the pairing computation involves the secret point and a public point. If the secret point is discovered by an attacker, he can then impersonate the target.

2.3 Fault attacks

The goal of a fault attack is to inject errors during the calculation of an algorithm in order to reveal sensitive data. At first these attacks required a very precise positioning and expensive apparatuses to be performed, but now even some cheap apparatuses allow to perform them [23]. The faults can be performed using a laser, an electromagnetic pulse, power or clock glitches [13, 14, 29].

The effect of a fault can be permanent, i.e. a modification of a value in memory, or transient, i.e. a modification of a data which is not stored into memory at one precise moment.

At the bit level, a fault can be a bit-flip if the value of a bit is complemented. Or it can be stuck-at (0 or 1) if the bit modification depends on its value.

The fault can not only modify the data manipulated but also modify the program execution. As an example in a microcontroller, if a fault occurs on the opcode and modifies it, the executed instruction will be modified. This method gives rise to what is called an instruction skip fault model where an instruction is skipped by modifying its opcode to a value representing an instruction without effect (*e.g.* NOP) or invalid.

3 Fault attacks against the Miller algorithm

In this section we present the existing attacks against the Miller algorithm. We describe in Section 3.1 an attack against the Duursma and Lee algorithm since it was the first attack against a pairing and, more importantly, all the following attacks are constructed on this scheme. Then in Section 3.2 are described the attacks against the Miller algorithm.

3.1 Attack against the Dursma and Lee algorithm

The Duursma and Lee algorithm is not constructed using the Miller algorithm. But it was the first implementation of a pairing to be attacked. The attack was developed by Page and Vercauteren in [36].

Duursma and Lee [15] define a pairing over hyperelliptic curves and in particular over super singular elliptic curves over finite fields of characteristic 3. For $\mathbb{F}_q$ with $q = 3^m$ and $k = 6$, suitable curves are defined by

$$E : y^2 = x^3 - x + b$$

with $b = \pm 1 \in \mathbb{F}_3$. If $\mathbb{F}_{q^3} = \mathbb{F}_q[\rho]/(\rho^3 - \rho - b)$ and $\mathbb{F}_{q^6} = \mathbb{F}_{q^3}[\sigma]/(\sigma^2 + 1)$. The distortion map $\phi : E(\mathbb{F}_q) \to E(\mathbb{F}_{q^6})$ is defined by $\phi(x, y) = (\rho - x, \sigma y)$. Then, with $\mathbb{G}_1 = \mathbb{G}_2 = E(\mathbb{F}_{3^m})$ and $\mathbb{G}_T = \mathbb{F}_{q^6}$, Algorithm 2 computes an admissible, symmetric pairing.

The attack developed by Page and Vercauteren in [36] consists in modifying the number of iterations during the Duursma and Lee algorithm. The hypotheses to perform the attack are that

- the two inputs parameters (points P and Q) are fixed, one is secret and the other public;
- the pairing implementation is public;
- two pairing computations are done, one valid and one faulty.

The analysis of the quotient of the two results gives information about the secret. Indeed, the quotient of the two results cancel terms that are not influenced by the fault. In a first time, Page and Vercauteren described how to recover the secret point if the final exponentiation is not performed (i.e. Line 9 of Algorithm 2). Then they explain how to reverse the final exponentiation for a complete attack.

Algorithm 2: The Duursma-Lee pairing algorithm.

Input : $P = (x_P, y_P) \in \mathbb{G}_1$ and $Q = (x_Q, y_Q) \in \mathbb{G}_2$.
Output: $e(P, Q) \in \mathbb{G}_3$.

1 $f \leftarrow 1$;
2 **for** $\underline{i = 1 \textbf{ upto } m}$ **do**
3 $\quad x_P \leftarrow x_P^3$, $y_P \leftarrow y_P^3$;
4 $\quad \mu \leftarrow x_P + x_Q + b$;
5 $\quad \lambda \leftarrow -y_P y_Q \sigma - \mu^2$;
6 $\quad g \leftarrow \lambda - \mu\rho - \rho^2$;
7 $\quad f \leftarrow f \cdot g$;
8 $\quad x_Q \leftarrow x_Q^{1/3}$, $y_Q \leftarrow y_Q^{1/3}$;
9 **end**
10 **return** $\underline{f^{q^3-1}}$;

Attack without the final exponentiation Let $P = (x_P, y_P)$ be the secret input during the pairing computation and let $Q = (x_Q, y_Q)$ be selected by the attacker. We consider the Duursma and Lee algorithm without the final exponentiation (Line 9).

Let $\overline{e}[\Delta]$ be the execution of Algorithm 2 where the fault replaces the loop bound m (in Line 2) with Δ. Then the result of the Duursma and Lee algorithm without the final exponentiation instead of being a product of polynomials of the form

$$\prod_{i=1}^{m} \left[(-y_P^{3^i} \cdot y_2^{3^{m-i+1}} \sigma - (x_P^{3^i} + x_2^{3^{m-i+1}} + b)^2) - (x_P^{3^i} + x_2^{3^{m-i+1}} + b)\rho - \rho^2 \right]$$

is a product of the form

$$\prod_{i=1}^{\Delta} \left[(-y_P^{3^i} \cdot y_2^{3^{m-i+1}} \sigma - (x_P^{3^i} + x_2^{3^{m-i+1}} + b)^2) - (x_P^{3^i} + x_2^{3^{m-i+1}} + b)\rho - \rho^2 \right]$$

for a random integer Δ.

If $\Delta = m + 1$, then recovering the secret point P is easy. We have two results

$$R_1 = \overline{e}[m](P, Q)$$
$$R_2 = \overline{e}[m + 1](P, Q)$$

where R_1 is correct and R_2 is faulty. Let $g_{(i)}$ be the i-th factor of a product produced by the algorithm. The quotient of the two results produces a single factor

$$g_{(m+1)} = (-y_P^{3^{m+1}} \cdot y_2 \sigma - (x_P^{3^{m+1}} + x_2 + b)^2) - (x_P^{3^{m+1}} + x_2 + b)\rho - \rho^2.$$

Given that $\forall z \in \mathbb{F}_q, z^{3^m} = z$, the attacker can easily extract x_P or y_P based on the knowledge of x_Q and y_Q.

In practice, the faulty result Δ cannot be forced to $m+1$. It is more realistic to assume that the fault gives $\Delta = m \pm \tau$ for a random unknown integer τ. As a consequence, the attacker compute two results

$$R_1 = \overline{e}[m \pm \tau](P, Q)$$
$$R_2 = \overline{e}[m \pm \tau + 1](P, Q),$$

and once again, considering the quotient, the attacker obtains a single term $g_{(m \pm \tau + 1)}$.

In order to apply the same approach, the attacker should discover the exact value of τ. Indeed, this value is needed to correct the powers of x_P, y_P, x_Q and y_Q. As the implementation of Duursma and Lee algorithm is supposed public, the number of operations performed during the faulty execution leaks the value of τ. Then the attack consists in several faulty executions of Algorithm 2 until we find two results R_1 and R_2 satisfying the requirements. The probability to obtain two values R_1 and R_2 after a realistic number of tests was computed in [16].
The probability to obtain two consecutive numbers after n picks among N integers is

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

where

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

For instance, 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.

Reversing the final exponentiation The attack described above is efficient without the final exponentiation. But since the final exponentiation is a part of the Duursma and Lee algorithm, Page and Vercauteren present a method to reverse it. The problem is that given the result $R = e(P, Q)$ the attacker want to recover S, the value obtained Line 7 of Algorithm 2 before the final exponentiation (i.e. $R = S^{q^3-1}$). Given R, the value of S is only determined up to a non zero factor in $\mathbb{F}_{q^3}$. Indeed, the Fermat little theorem implies that $\forall c \in \mathbb{F}_{q^3} \setminus \{0\},\ c^{q^3-1} = 1$. Furthermore, for one solution S of the equation $X^{q^3-1} - R = 0$, all the other solutions are of the form cS, for $c \in \mathbb{F}_{q^3} \setminus \{0\}$. At first sight, the attacker would not be able to choose the correct value S among the $q^3 - 1$ possibilities. However, given the description of the attack, the attacker does not need to reverse the powering of a full factor, but only a single factor with a special form:

$$R = \frac{R_2}{R_1} = \frac{\overline{e}[m \pm \tau + 1](P, Q)}{\overline{e}[m \pm \tau](P, Q)} = g_{(m \pm \tau + 1)}^{q^3-1}.$$

We want to recover $g_{(m \pm \tau + 1)}$, in order to find the coordinates of the secret point x_P and y_P.
In order to solve this problem, Page and Vercauteren split it in two

1. a method to compute one valid root of $R = g^{q^3-1}$ for some factor g, and
2. a method to derive the correct value of g from among all possible solutions.

The first problem is solved throughout the method of Lidl and Niederreiter [32] to compute roots of the linear operator $X^{q^3} - R \cdot X$ on the vector space $\mathbb{F}_{q^6}/\mathbb{F}_{q^3}$. They use a matrix representation of the problem to find all the solution of the equation $X^{q^3-1} - R = 0$. Then in order to find the correct root among the $q^3 - 1$ possibilities, Page and Vercauteren use the specific form of the factors in the product. Indeed, the terms $\rho\sigma$ and $\rho^2\sigma$ do not appear in the correct value and this gives a linear system of equations providing the solution. As the method to reverse the final exponentiation is specific to the Duursma and Lee algorithm, we do not give the equations. They are presented with examples in [20, 36].

3.2 Attacks against the Miller algorithm

A specific sign change attack The first attack against the Miller algorithm was developed by Whelan and Scott [46]. They use the same approach than the attack against Duursma and Lee. They compute two pairing values, one correct and one faulty. However the fault is no longer on the Miller loop bound but into the Miller variable. Whelan and Scott analyse several pairings and study the success of the attack whether the secret is the point P or Q. They consider the case of the Eta pairing [6]. This pairing is defined over super singular curves for small characteristics. Considering the recent result on the discrete logarithm problem [27] and the fact that the attack is based on the scheme of Page and Vercauteren attack, we do not describe it. Whelan and Scott target the Weil pairing. First they try to describe a general fault model: any fault is injected during any iteration of the Miller algorithm. The attacker needs to solve a non linear system and they conclude that it could not be done. So they consider a more specific attack: a sign change fault attack (a single sign bit is flipped [9]). They consider that the attacker modifies the sign of one of the coordinates of the point P or Q. This attack is the most efficient when exactly the last iteration of the Miller algorithm is corrupted. They consider the ratio between a valid and a faulty executions of the Weil pairing and, using the equations, they obtain a linear system in the coordinates of the secret point. In this case, the attack is successful. If the fault is injected earlier in the Miller algorithm, the analysis is more complex, as several square and cubic roots have to be computed, but possible. Then they consider the Tate pairing. As the Tate pairing is also constructed using the Miller algorithm, the attack described for the Weil pairing should be efficient. However, due to the complex final exponentiation they conclude that the Tate pairing is efficiently protected against the sign change fault they propose.

A general fault attack In [16], El Mrabet considers a fault attack based on the Page and Vercauteren attack [36]. The fault consists in modifying the number of iterations during the execution of the Miller algorithm. As the Miller algorithm is the central step for the Weil, the Tate, Ate, twisted Ate, optimal pairings and pairing lattices, the fault model is valuable for a wide class of pairings. However, the attack targets only the Miller algorithm, the final exponentiation is not reversed cryptanalytically and the author assume that another attack could annihilate it. In Section 4 we describe a recent attack that reverse the final exponentiation. We describe here the general attack against the Miller algorithm. The difficulty of the attack relates to the resolution of a non linear system.

El Mrabet considers that the number of iterations in the Miller algorithm is modified by a fault attack and denotes τ the new number of iterations. The value of τ is random but can be determined afterwards if the attacker knows the number of iterations, by monitoring the timing of the computation for example. The goal is to obtain a pair of results, $F_{\tau,P}(Q)$ and $F_{\tau+1,P}(Q)$, of two executions of the Miller algorithm. As in the attack on the Duursma and Lee algorithm, we consider the ratio $\frac{F_{\tau+1,P}(Q)}{F_{\tau,P}(Q)^2}$. Then an identification in the basis of $\mathbb{F}_{p^k}$ leads to a system which reveals the secret point.

Without loss of generality, we describe the attack when the embedding degree of the curve is $k = 4$. This allows the description of the equation. As the important point of the method is the identification of the decomposition in the basis of $\mathbb{F}_{p^k}$, it is easily applicable when k is larger than 3. Indeed, $k = 3$ is the minimal value of the embedding degree for which the system obtained can be solved. At the τ-th step, the Miller algorithm calculates $[j]P$. During the $(\tau + 1)^{th}$ iteration,

it calculates $[2j]P$ and considering the value of the $(\tau+1)^{th}$ bit of $\log_2(r)$, it either stops at this moment, or it calculates $[2j+1]P$.

Let $B = \{1, \xi, \sqrt{\nu}, \xi\sqrt{\nu}\}$ be the basis of $\mathbb{F}_{p^k}$, this basis is constructed using tower extensions. The point $P \in E(\mathbb{F}_p)$ is given in Jacobian coordinates, $P = (X_P, Y_P, Z_P)$ and the point $Q \in E(\mathbb{F}_{p^k})$ 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 $\nu \in \mathbb{F}_{p^{k/2}}$ and $\sqrt{\nu} \in \mathbb{F}_{p^k}$[4]. We will consider here only the case where $r_{\tau+1} = 0$. The case where $r_{\tau+1} = 1$ can be treated similarly and is described in [16]. The non linear system in the case $r_{\tau+1} = 1$ is a bit more complex and must be solved using the discriminant theory.

When $r_{\tau+1} = 0$, we have that $F_{\tau+1,P}(Q) = (F_{\tau,P}(Q))^2 \times h_1(Q)$, $[j]P = (X_j, Y_j, Z_j)$, where j is obtained by reading the τ first bits of r and $T = [2j]P = (X_{2j}, Y_{2j}, Z_{2j})$.

Using the equation of h_1, we obtain the following equality:

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

$$\left(Z_{2j}Z_j^2 y\sqrt{\nu} - 2Y_j^2 - 3(X_j - Z_j^2)(X_j + Z_j^2)(xZ_j^2 - X_j)\right).$$

Considering that the secret is the point P, we know j, τ, the coordinates of Q. The Miller algorithm gives us $F_{\tau+1,P}(Q)$ and $F_{\tau,P}(Q)$. We calculate the ratio $R = \frac{F_{\tau+1,P}(Q)}{(F_{\tau,P}(Q))^2}$. Using the theoretical form of R and its decomposition in the base B, by identification we can obtain after simplification the following system:

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

where we know the three values λ_0, λ_1 and λ_2.

The resolution[16] of this non linear system gives the following equation:

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

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 $[j]P$. Once we have the coordinates of $[j]P$, to find the possible points P, we have to find j' the inverse of j modulo r, 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 the Miller 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 $r_{\tau+1} = 0$. The case of $r_{\tau+1} = 1$ leads also to a non linear system that can be solved using a Grobner basis.

Remark 3. We present the attack in Jacobian coordinates. As the attack is not dependent on the system of coordinates, it will be successful for other systems. In [16], the affine, projective and Edwards coordinates are also treated. In the paper [45], the authors consider Hessian coordinates.

Remark 4. We describe the attack with the secret point being P. If the secret is the point Q the attack is also valid, we just obtain an easier system to solve.

The attack against the Miller algorithm is efficient. A model of the attack was implemented in [37]. It is fair to wonder if this attack can be applied to a complete pairing. As the Weil pairing consists in two applications of the Miller algorithm, the Weil pairing is sensitive to this attack. For the Tate-like pairings (Ate, twisted Ate,...) the final exponentiation must be cancelled for the attack to be efficient. As the result of the Miller algorithm has no particular form, it seems difficult to cryptanalytically reverse the final exponentiation. As far as we know it has not be done yet. El Mrabet cites several works in microelectronics that would give the result of the Miller algorithm during a Tate-like pairing computation: for example the scan attack [48] or the under voltage technique [1]. We describe in Section 4 a recent fault attack against the final exponentiation.

Attack against the *if* instruction In [3], the authors propose a new fault model as well as an implementation of their fault attack.

The if skip fault model In the Miller algorithm, the addition step is performed or not according to the bits of r. This decision is usually implemented with an *if* instruction. If an attacker is able to skip an *if* instruction he can avoid the addition step.

This fault model has several advantages. It can target the last iteration only of the Miller algorithm and as a consequence only one fault injection is required to find the value $h_2(Q)$. This is better than altering the counter value, where the attacker had to perform fault attacks until he find the faulty results for two consecutive iterations. Then it is not as easy to develop a countermeasure against it as for an attack on the loop counter. In the latter case, it is enough to check the number of iterations that the chip executed. In the *if* skip case, the number of addition steps is highly dependant on the l value and can vary even if the security level of the parameters do not.

Recovery of $h_2(Q)$ Let $F_P(Q) = f^2 \cdot h_1(Q) \cdot h_2(Q)$ be the result of the (correct) Miller algorithm expressed with the variables of the last iteration. As we are describing the attack for the Tate pairing, the function h_1 is the tangent at the current point $T = \frac{(r-1)}{2}P$ and h_2 is the equation of the vertical line passing through the points P and $T = (r-1)P$. For any other Tate-like pairing, we should substitute to the previous equation r by ρ which is the integer that gives the number of iteration of the Miller algorithm.

If an attacker skip the *if* instruction in the last iteration, he obtains the value $F_P(Q)^* = f^2 \cdot h_1(Q)$.

With a faulty result and a correct one, he can then compute the ratio

$$\frac{F_P(Q)}{F_P(Q)^*} = \frac{f^2 \cdot h_1(Q) \cdot h_2(Q)}{f^2 \cdot h_1(Q)} = h_2(Q). \tag{1}$$

Finding the secret with $h_2(Q)$ With the value $h_2(Q)$, the attacker still has to find the secret (the point P in our case). The following computations are done for the Tate pairing in particular. In this case the value r is the order of the groups used in the pairing. As a consequence, in the last iteration, the equation $T = -P$ holds.

In affine coordinates in the last iteration, with an embedding degree 2, $h_2(Q) = x_Q - x_P$ since $T = -P$: the line is the vertical passing through P. So knowing the value $h_2(Q)$ the attacker can find x_P with x_Q known. Using the elliptic curve equation, two candidates are possible for the y_P value. By trying the two possible input points in the Miller algorithm, he can find y_P with the comparison of these two Miller results and the correct one.

The result in Jacobian coordinates is slightly different. The equations are computed with an embedding degree 4 and the basis $B = \{1, \xi, \sqrt{\nu}, \xi\sqrt{\nu}\}$. The point P has Jacobian coordinates (x_P, y_P, z_P) and Q has coordinates $(x_Q, y_Q\sqrt{\nu})$.

In the last iteration, the simplified value $h_2(Q)$ is $h_2(Q) = z_P^2 x_Q - x_P$. When the attacker computes the ratio $R = \frac{F_P(Q)}{F_P(Q)^*}$, he finds a value which can be decomposed on the basis B.

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

The decomposition of $h_2(Q)$ on the basis B yields the system

$$R_1 = z_P^2 x_{Q_1} \tag{2}$$
$$R_2 = z_P^2 x_{Q_0} - x_P, \tag{3}$$

where $x_Q = x_{Q_0} + x_{Q_1}\xi$.

Since Q is known to the attacker, this system can be solved to provide the values z_P^2 and then x_P. There are 4 possible candidates for the point P which have to be verified by comparing with the correct result of the Miller algorithm.

Remark 5. In the case of other pairings (Ate,...), the same attack can be applied. The main difference is that we find a point multiple of P: λP for a public integer λ. Indeed, we consider that except the secret point, every detail of the implementation is public.

An implementation of the attack The authors of this attack [3] implemented their attack on a chip, an ATmega128L, with a laser fault injection. They demonstrated the feasibility of the *if* instruction skip on a dummy algorithm mimicking the structure of the Miller algorithm. After locating the right spot for the laser fault injection, they were able to successfully skip an *if* instruction.

The *if* instruction skip has two big advantages. It easily target a specific iteration in the Miller algorithm. It is possible to combine it with another instruction skip in the final exponentiation in order to realise a full attack on the pairing computation algorithm. But this later possibility is yet to be proven experimentally.

3.3 Countermeasures

Several countermeasures can be implemented to prevent a fault attack. They are referred in [20], we briefly recall them here. We can preventively use randomization of the inputs in order to prevent any leakage of information or detect any alteration of the circuit and then abort the pairing computation.

In order to detect any alteration of the computation we can

a) use fault resilient counters to avoid attacks focused on changing the Miller loop bound [35, Section 5.3],
b) implement the algorithm to perform a random number of iterations greater than the correct one [21, Section 4].
c) check intermediate results during the computation: verify that the points are still on the elliptic curve, compare the last point T with $(r-1)P$ [20],
d) duplicate the computation using bilinerarity: $R_1 = e(P, Q)$, $R_2 = e(aP, bQ)$ and check if $R_2 = R_1^{ab}$ [20],

Two realizations of fault attacks are based on the perturbations of the iteration of Miller's algorithm. As a consequence, the countermeasure a) should always be implemented. The countermeasure b) alone does not seem accurate for an efficient implementation of pairings. Indeed, it induces extra computation that do not improve so much the security. The countermeasure c) would be interesting for instance, in the case of a Goubin attack, but as far as we know the Goubin attack has not been developed in the context of pairing based cryptography. No attack used the alteration of the input of a pairing. Consequently this countermeasure seems useless in the context of pairings. The countermeasure d) is the strongest with the drawback of a double pairing computation.

The randomization and blinding methods are both based on the bilinearity of pairings:

α) use the homogeneity property of Jacobian and projective coordinates to represent the point P,
β) use the homogeneity property of Jacobian and projective coordinates to represent the point Q (with a modification of the equations in the Miller algorithm),
γ) randomize the inputs points using a random field element and modify the pairing algorithm in order to cancel out the effects [41],
δ) choose integers a and b such that $ab = 1 \mod (r)$ and compute $e(P, Q) = e(aP, bQ)$ [36],
Ω) choose a random point R such that $S = e(P, R)^{-1}$ is defined and compute $e(P, Q) = e(P, Q + R)S$,

The two blinding methods α and β are the lighter one, only 3 multiplications in the finite field $\mathbb{F}_p$ or 3 multiplications between an element of $\mathbb{F}_p$ and an element of $\mathbb{F}_{p^k}$. Unfortunately, they are not sufficient to prevent a fault attack. Indeed in [19] the authors demonstrate that the blinding of the coordinates using their homogeneity is not sufficient to protect a pairing against fault attacks. The countermeasure γ implies a modification of the Miller algorithm and it could be tricky to find efficient and secure equations. The two countermeasures δ and Ω seem to be the strongest. The method Ω has the drawback of requiring two pairing executions but could also prevent any alteration of the computation. As a consequence, we think that in order to assure a secure implementation of pairings, the countermeasure Ω should be considered.

4 A fault attack against the final exponentiation

The main difficulty faced by fault attacks on the pairing is the final exponentiation. Even if efficient schemes are able to reverse the Miller algorithm, they still require the attacker to have access to the result of the Miller algorithm, correct or faulty.

Several possibilities have been proposed to access these values. First, for some exponents (e.g. $q^3 - 1$), it is possible to reverse the final exponentiation by using the structure of the Miller result as shown in [36]. A more implementation dependant approach has been proposed in [16] where the authors propose to realise a scan chain attack or to override completely the final exponentiation to directly read the result of the Miller algorithm.

Despite considered previously unrealistic, multiple fault injections during one execution of an algorithm seem to be more and more feasible with some new results in this direction [8, 43]. This new possibility opens the door to a new scheme where two fault attacks are combined: one to reverse the final exponentiation, one to reverse the Miller algorithm.

Until recently, the final exponentiation was thought to be an efficient countermeasure against the fault attacks on the Miller algorithm since it is mathematically impossible to find the unique preimage of the exponentiation and thus the result of the Miller loop. However in [31], the authors propose a fault attack to reverse the final exponentiation.

4.1 Description of the attack

They chose the case were the embedding degree $k = 2d$ is even and they attack the final exponentiation algorithm proposed in [39].

The exponent is $\frac{p^k-1}{r}$ and can be decomposed as $\frac{p^k-1}{r} = (p^d-1)\cdot\frac{p^d+1}{r}$. If the result of the Miller algorithm is noted f, we choose the following notation: $f_2 = f^{p^d-1}$ and $f_3 = f_2^{\frac{p^d+1}{r}}$ (f_3 is the pairing result observed at the end of the computation). Since $f \in \mathbb{F}^*_{p^k}$, f, f_2 and f_3 satisfy the relations

$$f^{p^k-1} = 1 \ ; \ f_2^{p^d+1} = 1 \ ; \ f_3^r = 1. \tag{4}$$

These relations shows that these intermediary values belongs to the groups noted $f_2 \in \mu_{p^d+1}$ and $f_3 \in \mu_r$.

Let $\mathbb{F}_{p^k} = \mathbb{F}_{p^d}[w]/(w^2 - v)$ be the construction rule for the $\mathbb{F}_{p^k}$ extension field. v is a quadratic nonresidue in $\mathbb{F}_{p^d}$ and is a public parameter.

Let $f_2 = g_2 + h_2 \cdot w$ with $g_2, h_2 \in \mathbb{F}_{p^d}$. Then $f_2^{p^d+1} = 1$ implies $g_2^2 - v \cdot h_2^2 = 1$.

4.2 First fault

But this equation holds because $f_2 \in \mu_{p^d+1}$. Now if an attacker injects a fault of value $e \in \mathbb{F}_{p^d}$ such that the faulty value f_2^* equals

$$f_2^* = f_2 + e \notin \mu_{p^d+1}. \tag{5}$$

It is possible to write the fault effect as

$$f_2^* = (g_2 + e) + h_2 \cdot w. \tag{6}$$

And the value $(f_2^*)^{p^d+1}$ can be computed by the attacker since he can measure the value f_3^* and r:

$$(f_2^*)^{p^d+1} = (f_3^*)^r \in \mathbb{F}_{p^d}. \tag{7}$$

Moreover,

$$\begin{aligned}(f_2^*)^{p^d+1} &= (g_2 + e)^2 - v \cdot h_2^2 \\ &= 1 + 2 \cdot e \cdot g_2 + e^2.\end{aligned}$$

If the attacker knows the error value e, he can computes

$$g_2 = \frac{(f_3^*)^l - 1 - e^2}{2 \cdot e}. \tag{8}$$

And deduces the two candidates for h_2

$$h_2^+ = \sqrt{\frac{g_2^2 - 1}{v}} \ ; \ h_2^- = -\sqrt{\frac{g_2^2 - 1}{v}}. \tag{9}$$

With one fault, the attacker found the intermediary value f_2 by checking the two candidates and comparing $(f_2^+)^{\frac{p^d+1}{r}}$ and $(f_2^-)^{\frac{p^d+1}{r}}$ with f_3.

4.3 Second fault

At this step, the attacker knows f_3 the correct result of the pairing computation and the intermediary value f_2. Let $f = g + h \cdot w$, $f^{-1} = g' + h' \cdot w$ and $f_2 = f^{p^d-1}$. Then we note K the ratio

$$K = \frac{g_2 - 1}{v \cdot h_2} = \frac{h'}{g'} = -\frac{h}{g}. \tag{10}$$

In order to recover f, the attacker creates a new fault $e_2 \in \mathbb{F}_{p^d}$ during the inversion in the exponentiation by exponent $p^d - 1$.

Then

$$f_2 = f^{p^d-1} = \bar{f} \cdot f^{-1} \text{ and } f_2^* = \bar{f} \cdot (f^{-1} + e_2). \tag{11}$$

Let Δ_{f_2} be the difference $\Delta_{f_2} = f_2^* - f_2 = \bar{f} \cdot e_2$. Since $e_2 \in \mathbb{F}_{p^d}$, Δ_{f_2} can be written $\Delta_{f_2} = \Delta_{g_2} + \Delta_{h_2} \cdot w$ with $\Delta_{g_2} = e_2 \cdot g$ and $\Delta_{h_2} = -e_2 \cdot h$.

As f_2^* is not in μ_{p^d+1} with high probability, the attacker can compute $(f_2^*)^{p^d+1} = (f_3^*)^r \in \mathbb{F}_{p^d}$.

Here

$$\begin{aligned}(f_3^*)^{p^d+1} &= (g_2 + \Delta_{g_2})^2 - v \cdot (h_2 + \Delta_{h_2})^2 \\ &= (g_2 + e_2 \cdot g)^2 - v \cdot (h_2 - e_2 \cdot h)^2.\end{aligned}$$

Using the relation $h = -g \cdot K$, we obtain

$$g^2 \cdot e_2^2 \cdot (1 - v \cdot K^2) + g \cdot 2 \cdot e_2 \cdot (g_2 - v \cdot K \cdot h_2) + 1 - (f_3^*)^r = 0. \tag{12}$$

This quadratic equation provides two solutions for g, each one giving only one possibility thanks to K. The attacker has two candidates for f if he knows e_2.

If he does not exactly know the fault values but is able to have a limited number of guesses, he can still find f_2 easily. But in order to find f he will have to inject more faults similar to the second one in order to uniquely determine f.

As a conclusion, with a minimum of two separate faults during two executions (plus one correct execution) of the pairing computation, the attacker is able to reverse the final exponentiation.

A notable fact about this fault attack is that it can be achieved with instruction skip faults. As a consequence it is possible to combine it with a fault on the Miller algorithm, if the attacker can inject two faults in the same execution, in order to achieve a full pairing fault attack.

A major disadvantage to this attack, making it easy to counter, is that the attacker must be able to observe $f_3^* = (f_2^*)^{\frac{p^d+1}{r}}$. But often, since $f_2 \in \mu_{p^d+1}$ is called a unitary element, it is possible to speed up the final exponentiation computation by replacing the inversions in the computation of f_3 by conjugations (which is equivalent to an inversion for unitary elements). As a consequence, the attacker cannot observe f_3^* in this case and he cannot realise the attack.

5 Fault attack against pairings over Theta functions

The latest improvement in the computation of pairings was the description of efficient pairing computations in a more general case for any algebraic variety; and in particular for pairings over Theta functions. In [33], Lubicz and Robert generalize the notion of the Weil and the Tate pairings to any abelian variety. To do so, they made an explicit link between the Weil and the Tate pairings

and the intersection pairing on the degree 1 homology of an abelian variety. The result is a general definition of pairings and they explicit the formulas for the case of level 2 and 4 Theta functions in order to obtain the most efficient algorithm, considering time and memory consumption. Their algorithm to compute a pairing is based on a Montgomery Ladder approach. In [18], El Mrabet scrutinizes the pairings over Theta functions and proposes a successful fault attack. We describe here the attack. We will not present the theory about Theta functions, we refer to [33] for a detailed approach. We will only present the algorithm for the computation of pairings over Theta functions and the attack.

The Tate pairing over Theta functions For efficiency reasons, the pairing that will be implemented is the Tate pairing (or a variant of the Tate pairing) so we only consider the side channel attacks against the Tate pairing.

Let n, $l \in \mathbb{N}$ with n even and assume that $\gcd(n, l) = 1$. Let A be an abelian variety over $\mathbb{C}$ with period matrix Ω. We represent A as a closed subvariety of $\mathbb{P}^{n^g-1}$ by the way of level n Theta functions and suppose that this embedding is defined over K. Let $\widetilde{A}$ be the pullback of A via the natural projection $\kappa : A^{n^g} \to \mathbb{P}^{n^g-1}$. For $P \in A$, let $\widetilde{P}$ be an affine lift of P that is a point of A^{n^g} such that $\kappa(\widetilde{P}) = P$. Important ingredients of the algorithm in [33] are the Riemann addition formulas. Suppose that the Theta null point $\widetilde{0} = (\theta_i(0))_{i\in\mathbb{Z}(\overline{n})}$ is known.

Theorem 1. *[33, Theorem 1] Suppose that n and l are relatively primes. For $X, Y \in A(\overline{K})$, denoted by $\widetilde{X}$, $\widetilde{Y}$, $\widetilde{X+Y}$ any affine lifts of X, Y and $X+Y$. For $i \in \mathbb{Z}(\overline{n})$, let $\widetilde{X_i}$ be the i^{th} coordinate of $\widetilde{X}$. For $\in \mathbb{N}$ and $i \in \mathbb{Z}(\overline{n})$, let*

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

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

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

with $e_T(.,.)$ representing the Tate pairing whenever the right hand side is well defined.

The algorithm ScalarMult is composed of a doubling algorithm and a differential addition algorithm given in Figure 1, where a, b, $\mathcal{A}$ and $\mathcal{B}$ are constants depending on the Theta functions.

If we compare the efficiency of the Tate and of the Weil pairings, the former is more efficient than the later at least for the security levels considered today, when pairings are computed using a Miller algorithm. In the case of Theta functions, the algorithmic complexity of the Tate pairing consists in two applications of the function ScalarMult, while the Weil pairing consists in four applications of this function described in [33]. It is quite evident that the Tate pairing over Theta functions will always be more efficient than the Weil pairing over the Theta functions. So we study only the weaknesses of the Tate pairing against a fault attack. Nevertheless, the attacks described for the Tate pairing can easily be adapted to the Weil pairing. As a consequence the countermeasure proposed here must be considered also for the implementation of the Weil pairing.

The Tate pairing is composed of two applications of ScalarMult. First of all, we focus on a fault attack against one application of ScalarMult and after that we will consider an attack against the

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

Fig. 1. Doubling and Differential Addition Algorithms

Tate pairing. The same argument can provide the result of a fault attack against the Weil pairing, or any optimization of the Tate pairing namely Ate, twisted Ate or optimal pairings. The function ScalarMult is a Montgomery Ladder composed by the Doubling and Differential Addition algorithms at each step. When the secret is the exponent this algorithm is an efficient countermeasure to side channel attacks. In the case of pairing based cryptography, the secret is not the exponent but one of the parameters of the Montgomery Ladder algorithm. Consequently, the analysis considering side channel attacks against the Montgomery Ladder for the classical use in cryptography (efficient exponentiation) is no more available.

One model of fault attacks in pairing based cryptography consists in forcing the algorithm to stop early by reducing the number of iterations and by finding the results of two consecutive iterations τ and $\tau + 1$. The results of these two executions give equations that allow to find the secret. In the case of pairings over Theta functions, the fault attack consists in finding one of the coordinates involved during the computation of $\text{ScalarMult}(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P})$. The ScalarMult algorithm is composed by the doubling and differential addition algorithms, the results of ScalarMult are the coordinates of $\widetilde{P + lQ}$. The fault attack consists in reducing the number of iteration of ScalarMult. The fault attack for a pairing over Theta functions is easier than the classical fault attack in pairing based cryptography. We need only one fault and the result of this faulty execution to find the secret involved in the ScalarMult algorithm.

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

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

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

$$z_1 = \left[(x_j^2 + z_j^2)(x_P^2 + z_P^2) - \frac{\mathcal{A}^2}{\mathcal{B}^2}(x_j^2 - z_j^2)(x_P^2 - z_P^2)\right]\frac{1}{\overline{z}}, \tag{13}$$

where

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

- $\mathcal{A}$ and $\mathcal{B}$ are constants.

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

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

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

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

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

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

As the coordinates of P are such that $x_P z_P \neq 0$, we can consider that $\alpha = \frac{1}{z_P}$ and write the equation $\boldsymbol{\lambda} = \boldsymbol{\beta}((x'_P)^2 + 1) + \boldsymbol{\gamma}((x'_P)^2 - 1)$, which leads to $(x'_P)^2 = \frac{\boldsymbol{\lambda}-\boldsymbol{\beta}+\boldsymbol{\gamma}}{\boldsymbol{\beta}-\boldsymbol{\gamma}}$.

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

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

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

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

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

The attacks described above for ScalarMult can be directly adapted to the Tate pairing (and also to the Weil pairing). For efficiency reasons, the computation of $\text{ScalarMult}(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P}, l)_i$ and $\text{ScalarMult}(\widetilde{Q}, \widetilde{Q}, \widetilde{0}, l)_i$ would certainly be implemented in parallel. As a consequence, the fault attack forces the algorithm to stop after the same number of iterations and the result $\text{ScalarMult}(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P}, j)_i$ and $\text{ScalarMult}(\widetilde{Q}, \widetilde{Q}, \widetilde{0}, j)_i$, for the same integer j. For both cases, the secret being either P or Q, the homogeneity of projective coordinates is a trapdoor that gives information about the secret. Let P be the secret point and Q be public, then the coordinate $\widetilde{P}_i$ is also secret, but the homogeneity of the projective coordinates allows us to consider that for example the z coordinate is set to 1, exactly like in the attack described above. We just have to be careful and set the same coordinate to 1 in both calls to ScalarMult, the z one for example. The Equation (13) would give a slightly different system but linear and easily solvable. The method is the same if the point Q is secret.

Countermeasure for the fault attack Considering that the fault attack uses the homogeneity of the coordinates, this latter property cannot be used as a countermeasure. A solution would be to use the bilinearity of the pairing [20]. Indeed, if we compute the Tate pairing between the points P and Q, the bilinearity induces that

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

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

6 Conclusion

We presented in this paper the vulnerability to fault attacks of pairing algorithms when used in an Identity Based Protocol. The first attack against Duursma and Lee algorithm target the number of iterations. The final exponentiation in this case can be reversed using cryptanalytic equations. The most efficient pairings are constructed on the Tate model: an execution of the Miller algorithm followed by a final exponentiation. The Miller algorithm and the final exponentiation were separately analysed with respect to fault attacks. The Miller algorithm was attacked by a modification of the number of iterations and by the corruption of the *if* condition during the last iteration. The final exponentiation was attacked using two "independent" errors in the computation.

Table 1. Summary of the presented attacks. $P(n,N)$ is the probability to obtain two consecutive numbers after n picks among N integers (cf Section 3.1).

Attack name	Target	Attack path	Fault model	Number of faults required (+ correct execution)
Page and Vercauteren [36]	Duursma and Lee algorithm	Loop counter	Data modification	$n \vert P(n,N) > 0.5$ (+1)
Whelan and Scott [46]	Miller algorithm	Sign change	Bit-flip	1 (+1)
El Mrabet [16]	Miller algorithm	Loop counter	Data modification	$n \vert P(n,N) > 0.5$ (+1)
Bae *et al.* [3]	Miller algorithm	If skip	Instruction skip	1 (+1)
Lashermes *et al.* [31]	Final exponentiation	Group change	Data modification	2+ (+1)
El Mrabet [18]	Pairing on Theta functions	Loop counter	Data modification	1

For once it would be interesting to validate all those fault attack schemes on practical implementations running on a embedded chip. In [19], the authors demonstrate the feasibility of two models of fault attacks against the Miller algorithm. They consider the controlled add and the loop skip fault models. During the controlled add attack the authors target experimentally one addition at several moments to recover the secret used during the pairing computation. The loop skip model realizes in practice the theoretical attack developed in [46]. These two attacks focus only on the Miller algorithm. Further work is necessary to develop an attack against a whole Tate-like pairing, including the Miller algorithm and the final exponentiation. For example in [8], the authors achieve this with clock glitches. For that purpose they perform two fault injections during one pairing execution. The first fault allows them to exit the Miller loop after the first iteration. The second fault is used to skip entirely the final exponentiation. Skipping the final exponentiation is possible only if the method is not inlined (without compiler optimizations). In the other case a fault attack on this algorithm (such as the one presented in Section 4) would be needed as precised by the authors.

We also highlight the fact that a more general pairing constructed over an algebraic variety is sensitive to fault attacks. As a conclusion, we can say that the fault attack is a threat against an identity based protocol and consequently any implementation of pairings should be protected against physical attacks. We describe existing countermeasures to fault attacks. For now the strongest countermeasures are the one related to the bilinearity of pairings, with the drawback of a double pairing computation.

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. Diego F. Aranha, Jean-Luc Beuchat, Jérémie Detrey, and Nicolas Estibals. Optimal eta pairing on supersingular genus-2 binary hyperelliptic curves. In Orr Dunkelman, editor, CT-RSA, volume 7178 of Lecture Notes in Computer Science, pages 98–115. Springer, 2012.
3. 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.
4. Jean-Claude Bajard and Nadia El Mrabet. Pairing in cryptography: an arithmetic point de view. In Advanced Signal Processing Algorithms, Architectures, and Implementations XVI. part of SPIE, 2007.
5. A Barenghi, G. Bertoni, L. Breveglieri, and G. Pelosi. A fpga coprocessor for the cryptographic tate pairing over fp. In Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on, pages 112–119, April 2008.
6. Paulo S. L. M. Barreto, Steven D. Galbraith, Colm O'Eigeartaigh, and Michael Scott. Efficient pairing computation on supersingular abelian varieties. Des. Codes Cryptography, 42(3):239–271, 2007.
7. Ian F. Blake, Gagiel Seroussi, Nigel Smart, and J. W. S. Cassels. Advances in Elliptic Curve Cryptography (London Mathematical Society Lecture Note Series). Cambridge University Press, New York, NY, USA, 2005.
8. J. Blömer, R. Gomes da Silva, P. Günther, J. Krämer, and J.-P. Seifert. A practical second-order fault attack against a real-world pairing implementation. In Fault Diagnosis and Tolerance in Cryptography (FDTC), 2014 Workshop on, Sept 2014.
9. Johannes Blomer, Martin Otto, and Jean-Pierre Seifert. Sign change fault attacks on elliptic curve cryptosystems. In Luca Breveglieri, Israel Koren, David Naccache, and Jean-Pierre Seifert, editors, FDTC, volume 4236 of Lecture Notes in Computer Science, pages 36–52. Springer, 2006.
10. Dan Boneh and Matthew Franklin. Identity-Based Encryption from the Weil pairing. SIAM J. of Computing, 32(3):586–615, 2003.
11. RayC.C. Cheung, Sylvain Duquesne, Junfeng Fan, Nicolas Guillermin, Ingrid Verbauwhede, and GavinXiaoxu Yao. Fpga implementation of pairings using residue number system and lazy reduction. In Bart Preneel and Tsuyoshi Takagi, editors, Cryptographic Hardware and Embedded Systems ? CHES 2011, volume 6917 of Lecture Notes in Computer Science, pages 421–441. Springer Berlin Heidelberg, 2011.
12. Henri Cohen and Gerhard Frey, editors. Handbook of elliptic and hyperelliptic curve cryptography. Discrete Math. Appl., Chapman & Hall/CRC, 2006.
13. Elke De Mulder, Sidika Bernard Örs, Bart Preneel, and Ingrid Verbauwhede. Differential power and electromagnetic attacks on a FPGA implementation of elliptic curve cryptosystems. Comput. Electr. Eng., 33(5-6):367–382, 2007.
14. Amine Dehbaoui, Jean-Max Dutertre, B. Robisson, and Assia Tria. Electromagnetic Transient Faults Injection on a Hardware and a Software Implementations of AES. In FDTC, pages 7–15. IEEE, 2012.
15. Iwan M. Duursma and Hyang-Sook Lee. Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$. In Chi-Sung Laih, editor, ASIACRYPT, volume 2894 of Lecture Notes in Computer Science, pages 111–123. Springer, 2003.
16. Nadia El Mrabet. What about Vulnerability to a Fault Attack of the Miller's algorithm During an Identity Based Protocol? In Advances in Information Security and Assurance, volume 5576 of LNCS, pages 122–134. Springer, 2009.
17. Nadia El Mrabet. Fault attack against miller's algorithm. IACR Cryptology ePrint Archive, 2011:709, 2011.
18. Nadia El Mrabet. Side channel attacks against pairing over theta functions. In Traian Muntean, Dimitrios Poulakis, and Robert Rolland, editors, CAI, volume 8080 of Lecture Notes in Computer Science, pages 132–146. Springer, 2013.

19. Nadia El Mrabet, Jacques J.A. Fournier, Louis Goubin, Ronan Lashermes, and Marie Paindavoine. Practical validation of several fault attacks against the miller algorithm. In Fault Diagnosis and Tolerance in Cryptography (FDTC), 2014 Workshop on, Sept 2014.
20. 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.
21. Santosh Ghosh, Debdeep Mukhopadhyay, and Dipanwita Roy Chowdhury. Fault attack and countermeasures on pairing based cryptography. International Journal of Network Security (IJNS), 12(1):26–33, 2011.
22. Gurleen Grewal, Reza Azarderakhsh, Patrick Longa, Shi Hu, and David Jao. Efficient implementation of bilinear pairings on arm processors. In Lars R. Knudsen and Huapeng Wu, editors, Selected Areas in Cryptography, volume 7707 of Lecture Notes in Computer Science, pages 149–165. Springer, 2012.
23. Donald Habing. The use of lasers to simulate radiation-induced transients in semiconductor devices and circuits. In IEEE Transactions On Nuclear Science, volume 39, pages 1647–1653, 1992.
24. 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.
25. Florian Hess, Nigel Smart, and Frederik Vercauteren. The Eta Pairing Revisited. In IEEE Transactions on Information Theory, volume 52, pages 4595–4602, 2006.
26. Antoine Joux. A one round protocol for tripartite diffie–hellman. In Wieb Bosma, editor, Algorithmic Number Theory, volume 1838 of Lecture Notes in Computer Science, pages 385–393. Springer Berlin Heidelberg, 2000.
27. Antoine Joux. A new index calculus algorithm with complexity l(1/4+o(1)) in very small characteristic. IACR Cryptology ePrint Archive, 2013:95, 2013.
28. Marc Joye and Gregory Neven. Identity-based Cryptography. Cryptology and information security series. IOS Press, 2009.
29. Chong Hee Kim and J-J Quisquater. Faults, injection methods, and fault attacks. Design & Test of Computers, IEEE, 24(6):544–545, 2007.
30. TaeHyun Kim, Tsuyoshi Takagi, Dong-Guk Han, HoWon Kim, and Jongin Lim. Side channel attacks and countermeasures on pairing based cryptosystems over binary fields. In David Pointcheval, Yi Mu, and Kefei Chen, editors, Cryptology and Network Security, volume 4301 of Lecture Notes in Computer Science, pages 168–181. Springer Berlin Heidelberg, 2006.
31. 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, Cryptographic Hardware and Embedded Systems - CHES 2013, volume 8086 of Lecture Notes in Computer Science, pages 365–382. Springer Berlin Heidelberg, 2013.
32. Rudolf Lidl and Harald Niederreiter. Finite Fields. Number vol. 20,ptie. 1 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1997.
33. David Lubicz and Damien Robert. Efficient pairing computation with theta functions. In Algorithmic Number Theory, 9th International Symposium, ANTS-IX, Nancy, France, July 19-23, 2010. Proceedings, volume 6197 of Lecture Notes in Computer Science, pages 251–269. Springer, 2010.
34. Victor Miller. The weil pairing and its efficient calculation. Journal of Cryptology, 17:235–261, 2004.
35. Erdinc Ozturk, Gunnar Gaubatz, and Berk Sunar. Tate pairing with strong fault resiliency. In Proceedings of the Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC '07, pages 103–111, Washington, DC, USA, 2007. IEEE Computer Society.
36. Dan Page and Frederik Vercauteren. A Fault Attack on Pairing-Based Cryptography. Computers, IEEE Transactions on, 55(9):1075 –1080, sept. 2006.
37. Jea Hoon Park, Gyo Yong Sohn, and Sang Jae Moon. A simplifying method of fault attacks on pairing computation. IEICE transactions on fundamentals of Electronics, Communications and Computer Sciences, E94-A(6):1473–1475, 2011.
38. Michael Scott. On the efficient implementation of pairing-based protocols. In Liqun Chen, editor, IMA Int. Conf., volume 7089 of Lecture Notes in Computer Science, pages 296–308. Springer, 2011.
39. 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.
40. Adi Shamir. Identity-based cryptosystems and signature schemes. In Proceedings of CRYPTO 84 on Advances in cryptology, pages 47–53, New York, NY, USA, 1984. Springer-Verlag New York, Inc.
41. Masaaki Shirase, Tsuyoshi Takagi, and Eiji Okamoto. An efficient countermeasure against side channel attacks for pairing computation. In Liqun Chen, Yi Mu, and Willy Susilo, editors, Information Security Practice and Experience, volume 4991 of Lecture Notes in Computer Science, pages 290–303. Springer Berlin Heidelberg, 2008.

42. Joseph H. Silverman. The Arithmetic of Elliptic Curves. Graduate Texts in Mathematics. Springer, 2009.
43. 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.
44. Frederik Vercauteren. Optimal pairings. IEEE Trans. Inf. Theor., 56(1):455–461, January 2010.
45. Jiang Weng, Yunqi Dou, and Chuangui Ma. Fault attacks against the miller algorithm in hessian coordinates. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, Inscrypt, volume 7537 of Lecture Notes in Computer Science, pages 102–112. Springer, 2011.
46. 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.
47. Claire Whelan and Mike Scott. Side channel analysis of practical pairing implementations: Which path is more secure? In PhongQ. Nguyen, editor, Progress in Cryptology - VIETCRYPT 2006, volume 4341 of Lecture Notes in Computer Science, pages 99–114. Springer Berlin Heidelberg, 2006.
48. Bo Yang, Kaijie Wu, and Ramesh Karri. Scan based side channel attack on dedicated hardware implementation of data encryption standard. In Test Conference 2004, proceedings ITC 2004, pages 339 – 344, 2004.