Resistance of the Point randomisation countermeasure for pairings against side-channel attack

Damien Jauvart[1], Nadia El Mrabet[2], Jacques J.A. Fournier[3], and Louis Goubin[4]

[1] Aix-Marseille Université
163 Avenue de Luminy, 13009 Marseille, France
damien.jauvart@univ-amu.fr
[2] Mines Saint-Étienne, 880, route de Mimet, 13541 Gardanne Cedex, France
nadia.el-mrabet@emse.fr
[3] CEA LETI, 17 rue des Martyrs, 38054 Grenoble Cedex 9, France
jacques.fournier@cea.fr
[4] Laboratoire de Mathématiques de Versailles, UVSQ, CNRS, Université Paris-Saclay, 78035 Versailles, France
louis.goubin@uvsq.fr

Abstract. The field of Pairing Based Cryptography (PBC) has seen recent advances in the simplification of their calculations and in the implementation of original protocols for security and privacy. Like most cryptographic algorithms, PBC implementations on embedded devices are exposed to physical attacks such as side channel attacks. Such attacks which are able to recover the secret input used in some PBC-based schemes are our main focus in this paper. Various countermeasures have consequently been proposed. The present paper provides an updated review of the state of the art countermeasures against side channel attacks that target PBC implementations. We especially focus on a technique based on point blinding using randomization. We propose a collision based side-channel attack against an implementation embedding the point randomization countermeasure. It is, to the best of our knowledge, the first proposed attack against this countermeasure used in the PBC context. This raises questions about the validation of countermeasures for complex cryptographic schemes such as PBC. We also discuss about ways of thwarting our attack. This article is in part an extension of the paper [19] published at Secrypt 2017.

Keywords: Pairing-based cryptography, Miller's algorithm, side-channel attack, collision side-channel attack, countermeasures.

1 Introduction

Bilinear pairings are used in cryptography for various innovative protocols. In 2001, Boneh and Franklin published the Identity-Based Encryption (IBE) scheme based on Pairings [10]. The one-round tripartite key exchange [21] based on Pairings is another interesting practical use of such cryptographic primitives. Both protocols have been rewarded by the 2013 Gödel price for their advance in the cryptographic area.

Several studies have investigated about the vulnerability of PBC to side-channel attacks. The first papers to consider the security of pairings regarding side-channel attacks were mainly concerned with elliptic curves defined over small fields of characteristics 2 and 3 is Page *et al.* [34]. Although Joux [22] and Barbulescu [3] recently suggested that such fields should be avoided, some of those techniques intended for small characteristic fields can nevertheless be applied over large prime fields. At high level a pairing is a bilinear and not degenerate map denoted $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$ where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ are abelian groups of the same order l. In an IBE [10] scheme that uses pairings, a cipher is decrypted by the computation of a pairing between a **secret** point and another point that is part of the **input** cipher. In a nutshell, in the IBE [10] scheme the decryption step consists in deciphering the ciphertext constituted by the pair $\{U, V\}$ with $U \in \mathbb{G}_1$ and $V \in \{0, 1\}^n$ using the private key D. The entity needs to compute $e(D, U)$. Side-channel attacks against such a scenario aim at exploiting the interaction between the **known ciphertext point** and the **secret point** (which is part of the private secret key). A pairing calculation has a *double-and-add* structure, as is the case in Elliptic Curve Cryptography (ECC). However, with PBC the problem regarding side-channel attacks is different: the number of iterations and the scalar are known; and the secret is one of the arguments of the pairing. Consequently, side-channel attacks on PBC implementations are more likely to rely on Correlation Power Analysis-like (CPA) techniques to target the secret point (compared to using Simple Power Analysis-like (SPA) approaches to target the scalar in the double-and-add structure).

In this paper, we review various side-channel attacks used against PBC implementations and the associated countermeasures. We then focus on one of those countermeasures in order to explain and illustrate how to defeat it with an attack that has never been developed against PBC. The paper is organized as follows. The Section 2 recalls the bases on the pairings necessary for their comprehension for this article. We review related work concerning side-channel around PBC in the Section 3. The Section 4 provides an analysis of a countermeasures under our investigation and explains how this protection can be defeated. Then the Section 5 describes the practical experiments and results obtained when implementing this attack against a software pairing calculation running on a 32-bit platform. A conclusion is then proposed in the Section 6.

2 Pairings as a cryptographic application

In this section, we provide the concepts and notations that will be used throughout this paper. For a detailed explanation of pairings we refer the reader to [38].

2.1 Pairing description

Let q be a large prime number and E be an elliptic curve defined over $\mathbb{F}_q$. E can be written as

$$E = \{(x, y) \in \mathbb{F}_q \times \mathbb{F}_q | y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6\} \cup \{\mathcal{O}\}, \quad (1)$$

where $\mathcal{O}$ denotes the point at infinity: it is the identity element for the addition group law. The set of l*-torsion points of* E is $E[l] := \ker[l]$ (the set of points P in E such that $[l]P = \mathcal{O}$), the *rational torsion points* are given by $E(\mathbb{F}_q)[l] := E(\mathbb{F}_q) \cap E[l]$. The smallest positive integer k such that l divides $q^k - 1$ is called the embedding degree of $E(\mathbb{F}_q)$ with respect to l. As soon as $k > 1$, the group $E(\mathbb{F}_q)[l]$ is included into $E(\mathbb{F}_{q^k})$. Let $\mathbb{G}_1 = E(\mathbb{F}_q)[l] \cup E(\mathbb{F}_q)$, $\mathbb{G}_2 = E(\mathbb{F}_{q^k}) \cup E(\mathbb{F}_q)[l]$ and $\mathbb{G}_3$ be the group of l-roots of unity in $\mathbb{F}_{q^k}$.

1. Non-degeneracy: $\forall P \in \mathbb{G}_1 \setminus \{\mathcal{O}\} \quad \exists Q \in \mathbb{G}_2$ such that $e(P, Q) \neq 1$,

2. Bilinearity:

$$\begin{array}{r} e([a]P_1 + [b]P_2, Q) = e(P_1, Q)^a e(P_2, Q)^b, \text{for} a, b \in \mathbb{F}_l \\ e(P, [a]Q_1 + [b]Q_2) = e(P, Q_1)^a e(P, Q_2)^b \end{array}.$$

The above properties can be verified by using groups of points on elliptic curves for both abelian groups.

The Tate Pairing. The widely used Tate pairing [6, 13, 16, 37] takes as inputs two points P and Q such that $P \in E(\mathbb{F}_q)[l]$ and $Q \in E(\mathbb{F}_q)$ as provided in Equation 2 where μ_l is the group of the l-th roots of unity such that $\mu_l = \{\xi \in \mathbb{F}^\star_{q^k} | \xi^l = 1\}$. A final exponentiation $\frac{q^k-1}{l}$ is applied to the output $f_P(Q)$ in order to obtain a unique value of order l.

$$\begin{array}{rcll} \tau_l : E(\mathbb{F}_q)[l] \cup E(\mathbb{F}_q) \times E(\mathbb{F}_{q^k}) \setminus E(\mathbb{F}_q)[l] & \rightarrow \mathbb{F}^\star_{q^k} / \left(\mathbb{F}^\star_{q^k}\right)^l & \rightarrow & \mu_l \subset \mathbb{F}^\star_{q^k} \\ P, Q & \mapsto f_{l,P}(Q) & \mapsto & f_P(Q)^{\frac{q^k-1}{l}} \end{array} \quad (2)$$

where $f_{l,P}$ represents the Miller's function.

The Barreto–Naehrig curves [5]. This family of curves is widely used to get efficient implementations of pairings. The pairing-friendly ordinary elliptic curves over a prime field $\mathbb{F}_q$ are defined by $E : y^2 = x^3 + b$ where $b \neq 0$. Their embedding degree is $k = 12$. The order of E is l, a prime number. The BN curves are parametrized with p and l as follows:

$$\begin{array}{l} p(t) = 36t^4 + 36t^3 + 24t^2 + 6t + 1, \\ l(t) = 36t^4 + 36t^3 + 18t^2 + 6t + 1, \end{array} \quad (3)$$

where $t \in \mathbb{Z}$ is chosen in order to get $p(t)$ coprime to $l(t)$ and large enough to guarantee an adequate security level. According to recent development on the resolution of discrete logarithm, the BN curves could be no longer the most accurate choice for an efficient pairing implementation. We demonstrate our work using BN curves as we had an existing efficient implementation. However, our approach is independent from the family.

The Miller's function The Miller's function $f_{l,P}$ is a rational function defined by its divisor: $Div(f_{l,P}) = l(P) - ([l-1]P) - (\mathcal{O})$. The computation of such a map is a well known problem [6] and an efficient way of computing such pairings was proposed as a recursive scheme by Miller [30]. Miller's algorithm, which works as the main calculation to compute a pairing, uses an iterative relation to construct the rational function $f_{l,P}$. The Miller's loop is given in Algorithm 1.

Algorithm 1: Miller's algorithm.

```
Data: $l = (l_{n-1} \dots l_0)_2$, $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$
Result: $f_{l,P}(Q) \in \mathbb{G}_3$
$T \leftarrow P$;
$f \leftarrow 1$;
for $i = n-1$ downto $0$ do
    $f \leftarrow f^2(l_{T,T}(Q))$;
    $T \leftarrow [2]T$;
    if $l_i == 1$ then
        $f \leftarrow f(l_{T,P}(Q))$;
        $T \leftarrow T + P$;
    end
end
return $f$;
```

In Algorithm 1:

1. $l_{T,T}(Q)$ is the equation of the tangent at T evaluated at point Q.
2. $l_{T,P}(Q)$ is the equation of the line through T and P evaluated at point Q.

These equations are optimized by using mixed system coordinates for the points' representations as suggested in [1, 2, 8, 26] and [33]:

1. P and Q are in affine coordinates.
2. T is in Jacobian coordinates, i.e. if $T = (x_T, y_T) = \left(\frac{X_T}{Z_T^2}, \frac{Y_T}{Z_T^3}\right)$ in affine coordinates then $T = (X_T : Y_T : Z_T)$ in Jacobian.

With this representation the tangent and line equation are shown in Equation 4.

$$\begin{array}{l} l_{T,T}(Q) = 2y_Q Y_T Z_T^3 - 2Y_T^2 - \left(3X_T^2 + aZ_T^4\right)\left(x_Q Z_T^2 - X_T\right) \\ l_{T,P}(Q) = \left(y_Q - y_P\right) Z_T \left(X_T - Z_T^2 x_P\right) - \left(Y_T - Z_T^3 y_P\right)\left(x_Q - x_P\right) \end{array} \quad (4)$$

In the following, our implementation is a Tate pairing over Barreto and Naehrig curves [5].

2.2 Application to Identity-Based Encryption

An IBE scheme can be used to simplify a widely known issue in public key cryptography: the key exchange [23]. A Public-Key Infrastructure (PKI) based on IBE is less complex and more scalable compared to classical schemes (with certifications).

In an IBE, the public key of a character **is** its identity. The associated private key can't be computed by this character, but generated by the Private Key Generator (PKG). Of course, the decryption should be possible only with the private key.

A simplified version of the scheme of Boneh-Franklin [10] work in four steps:

1. Setup. The PKG have to generate some public parameters for the pairings. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of order l such that $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a bilinear pairing. Let $P \in \mathbb{G}_1$ be a generator of $\mathbb{G}_1$. let $H_1 : \{0,1\}^\star \to \mathbb{G}_1^\star$ and $H_2 : \mathbb{G}_2 \to \{0,1\}^n$ be two cryptographic hash functions. Let $s \in \mathbb{Z}_r$ be random, s is their private key (a master key of the system). Let $P_{PUB} = [s]P$ be the global public key. The set of public parameters is

$$\{r, n, \mathbb{G}_1, \mathbb{G}_2, e, P, P_{PUB}, H_1, H_2\}.$$

2. Extraction. The extraction algorithm supplies the private key of a user. Let $ID =$ "Bob" $\in \{0,1\}^\star$ be the identity of a user Bob. The PKG hashes this string into $\mathbb{G}_1$ in order to obtain $Q_B = H_1(ID)$. Bob's private key is $d_B = [s]Q_B$ (computed and transmitted to Bob by the PKG).

3. Encryption. Alice wants to send a message $M \in \{0,1\}^n$ to Bob, she proceeds as follows:

1. She computes $Q_B = H_1($"Bob"$)$.
2. She randomly picks k.
3. She computes $g_B = e(Q_B, P_{PUB}) \in \mathbb{G}_2^\star$.
4. Then, she computes the ciphertext $C = \{[k]P, M \oplus H_2(g_B^k)\}$ and sends it to Bob.

4. Decryption. Bob wants to decrypt the ciphertext $C = \{U, V\}$ where $U \in \mathbb{G}_1, V \in \{0,1\}^n$, he proceeds as follows:

1. He computes $e(d_B, U)$ which is equal to $e([s]Q_B, [k]P) = e(Q_B, P)^{sk} = e(Q_B, [s]P)^k = e(Q_B, P_{PUB})^k = g_B^k$.
2. He gets the message $M = V \oplus H_2(g_B^k)$.

3 Related Work and Contributions

Differential Power Analysis (DPA) attacks have been first introduced by Kocher *et al.* in [28]. Since then, DPA-like techniques have been successfully used to attack implementations of most cryptographic algorithms.

3.1 Related Work in Side-Channel Attacks against Pairings

The first paper to investigate about the physical security of pairing algorithms was published in 2004. In this paper, Page and Vercauteren [34] simulated an attack on the Duursma–Lee Algorithm [12] which is used to compute Tate pairings using elliptic curves over finite fields of characteristic 3. The authors exposed the vulnerability of such pairings with respect to active (fault injections) and passive (side-channel observations) attacks. The authors also proposed two countermeasures to thwart side channel attacks.

The first countermeasure is based on the bilinearity of the pairing where, if a and b are two random values, then we have the equality: $e(P,Q) = e([a]P,[b]Q)^{1/ab}$. The second countermeasure, proposed in [34], works for cases where P is secret and a mask is added to the point Q as follows: select a random point $R \in \mathbb{G}_2$ and compute $e(P,Q+R)e(P,R)^{-1}$ instead of $e(P,Q)$, with different random values of R at every call to e.

The main inconvenience of these countermeasures is the computation overhead where two pairings are calculated instead of one.

Pan and Marnane [35] simulated a side-channel attack where they proposed a CPA based on a Hamming distance model to target a pairing over a base field of characteristic 2 over super-singular curves. The practical results obtained by Pan and Marnane can be used to assess the feasibility of using CPA to target pairings on an FPGA platform.

Kim *et al.* [24] also examined the security of pairings over binary fields. They addressed timing, SPA, and DPA attacks targeting arithmetic operations. In order to propose a more efficient countermeasure to protect Eta pairings, Kim *et al.* [24] implemented the third countermeasure proposed by Coron [11], which uses random projective coordinates. The randomization countermeasure proposed by Kim *et al.* adds just one step at the beginning. For greater efficiency, when P is secret, they randomized only the known input point Q. Its effect is "removed" during the final exponentiation.

This approach can be adapted to other pairing algorithms that are based on either small or large characteristic prime fields. This method is similar to the countermeasure suggested by Scott [37]. It consists in randomizing the Miller variable in Algorithm 1 by multiplying the operations 4 and 7 by a random $\lambda \in \mathbb{F}_q^\star$. The result is correct because the random element is eliminated through the final exponentiation.

In the end, these countermeasures only add few modular multiplications, which means a small overhead.

Whelan and Scott [41] studied pairings with different base field characteristics. They analyzed the arithmetic operations and concluded that the secret can be recovered by using a CPA. But the authors specified the need to have the point Q (second entry) as secret for the attack to work. The latter conclusion was refuted by El Mrabet in [14], which, to our best knowledge, is the first paper to present a concrete attack on Miller's algorithm with P (first input) as secret.

Another attack, this time on an FPGA platform, is proposed by Ghosh *et al.* [17]. They performed a bitwise DPA attack on an FPGA platform by measuring the power consumption leakages during the modular subtraction operations. To counteract this attack, the authors proposed a "low-cost" protection based on a rearrangement principle whose aim is to prevent interaction between a known value and a secret input as it happens in the calculations involved in the tangent or line evaluations. To achieve this, the authors proposed to rewrite the line equation to prevent the addition and/or subtraction operations between the known and secret data. They used the distributivity properties, i.e. if an instruction is $(k - y_1)y_2$ with k being the secret and y_1, y_2 being known integers, then the target operation is $(k - y_1)$. To avoid this, the authors proposed to rewrite it as $ky_2 - y_1y_2$. Indeed, this trick avoids the critical subtraction. However, this time this trick does not protect the modular multiplication and fails to protect against classical attack schemes as presented in [14, 41] and [9].

Moreover, Blömer *et al.* [9] studied DPA attacks by targeting modular addition and multiplication operations of finite field elements with large prime characteristics. Their paper describes an improved DPA for cases in which modular addition is targeted by combining information derived from manipulations of the least and most significant bits. In addition, the study provided simulation results to prove the feasibility of the attack. Furthermore, they propose a new countermeasure. In the reduced Tate pairing, the set of the second argument input is the equivalence class ${}^{E(\mathbb{F}_{q^k})}/_{lE(\mathbb{F}_{q^k})}$. If the random point T is chosen initially from $E(\mathbb{F}_{q^k})$ of order r, coprime to l, then $T + Q \sim Q$. Hence, $e(P, Q + T) = e(P, Q)$. This trick makes it possible to obtain a countermeasure as powerful as that of [34] with no overhead.

The importance of implementing countermeasures is supported by the recent results of Unterluggauer and Wenger [39] and Jauvart *et al.* [20], where attacks are presented in the real world environment. Indeed, Ate pairings implemented on Virtex-II FPGA, ARM Cortex-M0 and ARM Cortex-M3 have been broken efficiently with CPA attacks.

Despite all this existing literature on side-channel countermeasures for Pairings, to our best knowledge, none have actually tested or validated the efficiency of those countermeasures when considering analysis attack. In this paper we investigate about the level of protection provided by the randomization of coordinates which seems to be a classy and efficient countermeasure. To our best knowledge, except for fault attacks, no particular problem has been reported in the literature regarding this countermeasure applied to Pairings. But our analysis shows that this countermeasure can be defeated by a collision-based attack.

3.2 Collision based side channel attacks

The use of “collisions” as a means of exploiting side channel attacks is not something new in the literature. In this section we provide a quick review of the existing background in this very precise field before describing our approach and the differences with the existing state-of-the-art.

Collision attacks were first introduced in [36]. The main idea is to use the side-channel leakages to detect collisions in the encryption function, such collisions may appear internal to the function, in their attack it is not mandatory to observe collisions only at the output. Collisions inside the Data Encryption Standard (DES) can be detected using side-channels and exploited in order to retrieve the secret key used by the algorithm. This new class of attack was later used to circumvent countermeasures used in “secure” implementations of the Advanced Encryption Standard (AES) in [32].

The use of collisions against implementations of public key cryptographic algorithms have also been described. To achieve this, Fouque and Valette [15] use the following assumption: if two operations involve a common operand then the use of this common operand, can be detected using side-channels for attacking, in their example, the RSA exponent. More precisely, even if an adversary is not able to tell which computation is done by the device, he can at least detect when the device does the same operation twice. For example, if the device computes $2.A$ and $2.B$, the attacker is not able to guess the value of A nor B but he is able to check if $A = B$.

Similar work has been suggested by Bauer *et al.* in [7]. This time the target is a scalar multiplication over an elliptic curve. The assumption is still the same: the adversary can detect when two field multiplications have at least one operand in common. In a double-and-add algorithm the doubling step and the addition have a slight difference. One of them (depending on the curve representation) performs two modular multiplications with the same operand. Collision detection allows to distinguish between the doubling and the addition operations, from which the secret scalar can be deduced.

In [40], the authors target a protected Elliptic Curve Digital Signature Algorithm (ECDSA) implementation. One of the weak points of this protocol is the calculation of a modular multiplication between a known variable and the secret key. Thus a DPA is able to recover the key [18]. To counteract this attack, a trick consists in distributing a calculus to remove such critical operations. As an example, whenever the operation $mask(plaintext + public\_key \times secret\_key)$ must be computed, they propose to do $mask \times plaintext + (mask \times secret\_key)public\_key$ instead. The drawback revealed by Varchola *et al.* is that the additional calculation is between the known message and the temporary mask (which changes from one execution to another) while another calculation is made between this same mask and the secret. Thus, with the same assumption that in the previous cases [7] and [15], the collision detection will make it possible to discover whether the known (and controllable) message is equal to the secret key.

Our contribution adapts the principle of collision detection – based on the detecting when the same operand is used twice – to circumvent the randomization of Jacobian coordinates countermeasure used to protect pairing.

4 Security analysis of the countermeasure based on randomized Jacobian coordinates

The previously described countermeasures have been proposed without any theoretical security proofs, and to the best of our knowledge, no practical evidence has been provided neither. In this section, we first introduce the classical CPA against an unprotected pairing implementation. This will allow us to show how the countermeasure analysed in this paper plays an important role. Next, we will be able to compare the effectiveness of an attack on an unprotected and a counter-measured pairing to note the effectiveness of the countermeasure. The analysed contermesure is the Miller's algorithm with randomized Jacobian coordinates. We show how collisions can be used to make this countermeasure fail. We introduce a first "straight-forward" scheme to detect collisions and we show that this approach has its limits in practice. Then we adapt a refined method proposed in [40] for detecting collisions by implementing it on our target device and we show how practical results of how this collision-detection scheme defeats the point randomization countermeasure.

4.1 Classical side-channel attack against unprotected pairing implementation

For performance reasons, the use of mixed affine-Jacobian coordinates has been often proposed in the literature [1, 2, 8, 26] and [33]. In this case, at the beginning of the Miller's algorithm, the point P is assigned to T, with T expressed in Jacobian coordinates. This operation comprises the following steps:

1. $X_T \leftarrow x_P; \quad Y_T \leftarrow y_P; \quad Z_T \leftarrow 1,$
2. $T \leftarrow (X_T : Y_T : Z_T).$

We are looking for an operation that takes as operands: known data and secret data. Such operation appear during the first iteration of the Miller loop. More precisely, in the calculation of the tangent line, the evaluation of the tangent line is as follow:

$$2Y_T Z_T^3 y_Q - 2Y_T^2 - (3X_T^2 + aZ_T^4)(x_Q Z_T^2 - X_T), \tag{5}$$

where a is a parameter for the BN curves. The attacked coordinates are x_Q and y_Q, they are respectively involved in the multiplications $x_Q(Z_T^2)$ and $(2Y_T Z_T^3)y_Q$. The secret coordinates are attacked with a CPA against the modular multiplication. The coordinates of the point $T = (X_T : Y_T : Z_T)$ are known since they are initialized to those of the known point P.

Attack against the modular multiplication. The context is the following: the targeted modular multiplication $a \times b \mod p$ is a Montgomery's CIOS Multiprecision Multiplication algorithm 2. It is a common choice for cryptographic applications due to its low cost, this choice does not affect the attack anyway. For more details about Montgomery multiplication implementation, we recomend to read [27].

Algorithm 2: CIOS Modular Montgomery Multiplication

Data: The modulus $p = (p_{\mathfrak{N}-1} \dots p_0)_{\overline{\mathfrak{W}}}$ coprime with $\overline{\mathfrak{W}}$, $\mathfrak{R} = \overline{\mathfrak{W}}^{\mathfrak{N}}$, p' such that $\mathfrak{R}\mathfrak{R}^{-1} - pp' = 1$ and two integers $a = (a_{\mathfrak{N}-1} \dots a_0)_{\overline{\mathfrak{W}}}$ and $b = (b_{\mathfrak{N}-1} \dots b_0)_{\overline{\mathfrak{W}}}$.

Result: The unique integer $c = (c_{\mathfrak{N}-1} \dots c_0)_{\overline{\mathfrak{W}}}$ such that $c = \textsc{Redc}(ab) = \left(ab\mathfrak{R}^{-1}\right) \mod p$.

```
1  $c \leftarrow 0$ ;
2  for $i = 0$ to $\mathfrak{N} - 1$ do
3      $u \leftarrow 0$ ;
4      for $j = 0$ to $\mathfrak{N} - 1$ do
5          $(uv) \leftarrow c_j + a_j b_i + u$ ;
6          $c_j \leftarrow v$ ;
7      end
8      $(uv) \leftarrow c_{\mathfrak{N}} + u$ ;
9      $c_{\mathfrak{N}} \leftarrow v$ ;
10     $c_{\mathfrak{N}+1} \leftarrow u$ ;
11     $m \leftarrow c_0 p'_0 \mod \overline{\mathfrak{W}}$ ;
12     $(uv) \leftarrow c_0 + m p_0$ ;
13     for $j = 1$ to $\mathfrak{N} - 1$ do
14         $(uv) \leftarrow c_j + m p_j + u$ ;
15         $c_{j-1} \leftarrow v$ ;
16     end
17     $(uv) \leftarrow c_{\mathfrak{N}} + u$ ;
18     $c_{\mathfrak{N}-1} \leftarrow v$ ;
19     $c_{\mathfrak{N}} \leftarrow c_{\mathfrak{N}+1} + u$ ;
20 end
21 if $(c_{\mathfrak{N}-1} \dots c_0)_{\overline{\mathfrak{W}}} < p$ then
22     $c \leftarrow (c_{\mathfrak{N}-1} \dots c_0)_{\overline{\mathfrak{W}}}$ ;
23 else
24     $c \leftarrow (c_{\mathfrak{N}-1} \dots c_0)_{\overline{\mathfrak{W}}} - p$ ;
25 end
26 return $c$ ;
```

The operation which involves known and unknown data is the following (line 5):

$$(uv) \leftarrow \underbrace{c_j}_{\text{carry}} + \underbrace{a_j b_i}_{\text{word integers}} + \underbrace{u}_{\text{register}} . \tag{6}$$

The 64-bit registers (uv) are affected into the new variable. In this new variable appear the product $a_j b_i$, two interesting machine words. One of them (a_j) is a word of the secret integer, i.e. derived from the point Q. While b_i is a word from the known integer P inputs in the Miller algorithm. The 32-bit of the secrete word are recover byte by byte. Thus, the known word is over 32 bits several leakage models are possible, the paper of Jauvart *et al.* [20] show some difference and select one of them. Figure 1 summarizes the choice of the 16-bit model.

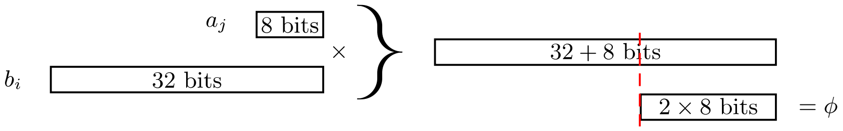

Fig. 1. Word machine leakage model

The aim of correlation power analysis is to compare, by a correlation calculus, the side-channel leakages and the hypothetical intermediate state during the multiplication procedure. The side-channel leakages are the measurement traces. The intermediate state is computed between the known input (b_i) and all of the $2^8 = 256$ key hypothesis for the least significant byte. This intermediate calculus is mapped in intermediate state by the well know Hamming weight leakage model. The key whose calculation of correlation between the derived intermediate state and the side-channel traces which maximizes the correlation is retained as the candidate key.

Afterward, the others 3 bytes are attacked in the same way.

4.2 The Miller's algorithm with randomized Jacobian coordinates

Such protection is convenient to be applied in the Miller's algorithm using the mixed affine-Jacobian coordinates and relies on the homogeneity of Jacobian coordinates. The above steps in Section 4.1 are replaced by the proposed countermeasure, for which the input point P is **known**, and Q is the **secret** point:

1. $\lambda \in \mathbb{F}_q^\star$ is randomly generated,
2. $X_T \leftarrow x_P \lambda^2; \quad Y_T \leftarrow y_P \lambda^3; \quad Z_T \leftarrow \lambda,$
3. $T \leftarrow (X_T : Y_T : Z_T).$

The full Miller algorithm that integrates this countermeasure is given in Algorithm 3.

All attacks against pairings proposed so far are DPA/CPA-like approaches that target arithmetic operations such as modular additions or multiplications between a known (public) value and a secret (key) one. Our attack scheme is

Algorithm 3: Miller's algorithm with randomization of Jacobian coordinates.

```
Data: $l = (l_{n-1} \dots l_0)$ radix 2 representation, $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$
Result: $f_P(D_Q) \in \mathbb{G}_3$
$\lambda \in \mathbb{F}_q^\star$ is randomly generated ;
$X_T \leftarrow x_P \lambda^2$;
$Y_T \leftarrow y_P \lambda^3$ ;
$Z_T \leftarrow \lambda$;
$f \leftarrow 1$;
for $i = n-1$ downto $0$ do
    $f \leftarrow f^2(l_{T,T}(Q))$;
    $T \leftarrow [2]T$;
    if $l_i == 1$ then
        $f \leftarrow f(l_{T,P}(Q))$;
        $T \leftarrow T + P$;
    end
end
return $f$;
```

different as it exploits collisions which may appear during the same execution of a pairing.

Of course, since the recent results of Joux [22] and Barbulescu [3], pairings in small characteristic based field are no longer recommended. Nevertheless, the proposed countermeasures in such fields can also be used in other fields with a very small overhead.

4.3 Detecting collisions in point-randomized Pairing calculations

Our attack is based on the following observation: in Algorithm 3 the mask is applied once to the public parameter and at least once to the secret input. During the first iteration of Miller's loop, the tangent evaluation calculates $(xZ_T^2 - X_T)$, which is in fact $(x\lambda^2 - x_P\lambda^2)$, for which $x\lambda^2$ is computed in the tangent evaluation and $x_P\lambda^2$ is computed in the randomization step.

Thus, if the known input x_P is equal (or "partially equal") to the secret x, then the EM traces are expected to be similar. The data x, x_p are long precision integers, for instance 256-bit integers, and then it is impossible to test all the 2^{256} values for x. However, the targeted operations work on "word reprensentations" of those integers, like for example when implementing the Montgomey multiplication [31]. So, we can consider only one word of each of those integers. Even with this remark, the words are still too long, for instance, a 256-bit integer can be stored in 8 words of 32 bits in an 32-bit architecture. Then "partially equal" denotes the equality of a part of the word such as the least significant byte.

In order to exploit this observation, our proposed attack scheme is the following. We assume that there exist 2^8 points P_j such that the 8 LSBs (Least Significant Bits) of the x coordinate cover all 2^8 possibilities.

$$\begin{aligned} x_{P_0} &= (\star\star\cdots\star\,\mathbf{00000000})_2 \\ x_{P_1} &= (\star\star\cdots\star\,\mathbf{00000001})_2 \\ &\vdots \\ x_{P_{255}} &= (\star\star\cdots\star\,\mathbf{11111111})_2 \end{aligned}$$

We then perform a pairing between each P_j and the secret point Q. The λ_j value is chosen at random. For each EM trace, it is necessary to focus on two critical moments: the computations of $x_{P_j}\lambda_j^2$ and $x\lambda_j^2$.

For each of the resulting "pairs of traces", we need to evaluate the similarities between the two signals. These similarities can be estimated through cross correlations for example. The maximum correlation coefficient then yields a candidate for the 8 LSBs of the secret x.

Averaging is necessary in order to reduce the effect of noise on the attack. Obviously, due to the randomness of λ, it is not recommended to average the acquired traces. However, for a fixed input x_{P_j} and x, we computed the cross correlations between traces for $x_{P_j}\lambda^2$ and $x\lambda^2$ computations. We thereby obtain $c_{P_j}^{(0)}$, and we repeat the process with other unknown λ. We then collect some $c_{P_j}^{(n)}$ coefficients for each key hypothesis and subsequently compute an average correlation coefficient for all hypothetical keys. This number n is further denoted n_P.

We subsequently repeat the method with other values of P_j covering another portion of the data, and the secret is fully recovered.

5 Practical implementation of the collision attack against point randomization

In this section, we report the practical results obtained when implementing our collision attack. The experiments were carried in two stages. A first stage consisted in testing the feasibility of detecting collisions, at word level, on our 32-bit target device. In the second stage, we implemented our attack on a Pairings implementation integrating Jacobian point randomisation countermeasure.

5.1 Preliminary characterization of collision detection on our target device

The targeted device is an ARM Cortex M3 processor working on 32-bit registers. We implemented the representative target operations over 32-bit integers:

- $x_{P_j} \times \lambda^2$,
- $x \times \lambda^2$.

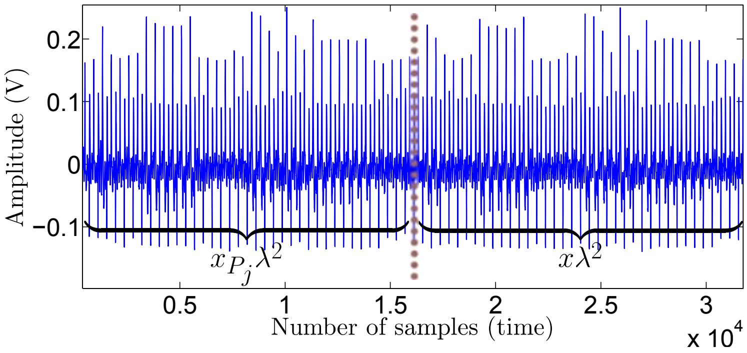

Fig. 2. An example of electromagnetic emanation measurement. © [19]

As source of side channel information, we use the ElectroMagnetic (EM) waves emitted by the chip during the targeted calculations. This technique does not need any depackaging of the chip and allowed to have "local" measurements when precisely positioned on top of the die. The electromagnetic emanation (EM) measurements were done using a Langer EMV-Technik LF-U 5 probe equipped with a Langer Amplifier PA303 BNC (30dB). The curves were collected using a Lecroy WaveRunner 640Zi oscilloscope. The acquisition frequency of the oscilloscope is 10^9 samples per second. The EM measurements acquisition is done as in Algorithm 4.

Algorithm 4: EM measurements acquisition procedure.

```
Data: $n_P$, the repetition number
Result: A data base of EM measurement $R \in \mathcal{M}_{256,n_P,2t}$, $t$ is the traces length
for $j = 0$ to 255 do
    $x_j \leftarrow (0 \dots 0 j_7 j_6 \dots j_0)_2$;   // $j = (j_7 \dots j_0)_2$ in radix 2 representation
    for $i = 0$ to $n_P - 1$ do
        $\lambda \leftarrow$ random in $\{0, \dots, 2^{32} - 1\}$;
        Execute the routine: computation of $x_j \times \lambda^2$, $x \times \lambda^2$;
        Store the EM measurement in $R[j, i]$
    end
end
return $R$;
```

As a result, in one EM measurement there are two multiplications. An example of such a trace is given in Figure 2.

The choice of EM leakages source is justified by the fact that the device under test is not appropriate for acquiring power consumption. Indeed, the device has many power sources and grounds, so if we want to keep the power consumption, the choices of them is not so simple, and can be a combination of several sources/grounds. The other reason is linked to the practical equipment to make the power consumption attack. A resistance should be placed on the source or on the ground, then there is a risk of damaging the circuit. The EM equipment is just a probe to place over the integrated circuit. Furthermore, in the case of our device, it is not necessary to depackage to integrated circuit, then there is no dangerous manipulation of the circuit.

At the end of Section 4.3 we introduced the theoretical technique to distinguish the good key when the correlation coefficient is used to detect collisions. This naive method consists in comparing two traces by cross correlation for each couple $x_{P_j} \times \lambda^2$ and $x \times \lambda^2$, and computes a coefficient for each key hypothesis (denoted by x_{P_j}) by averaging the correlation over the n_P repetitions.

We used this method with our EM measurements by using the correlation criterion. The attack succeeds if the maximal value for the collision detection criterion is reached when $x_{P_j} = x$. Then we can classify the key candidates (on 8 bits) according to their criterion values. The keys are now ranked from the most to the least probable, the position of the correct secret key is called the "key ranking". The key ranking is a value between 1 and 256, it is worth 1 if the attack succeeds in recovering the secret's least significant byte.

Figure 3 shows that the key ranking slightly decreases with the number of traces used for the attack.

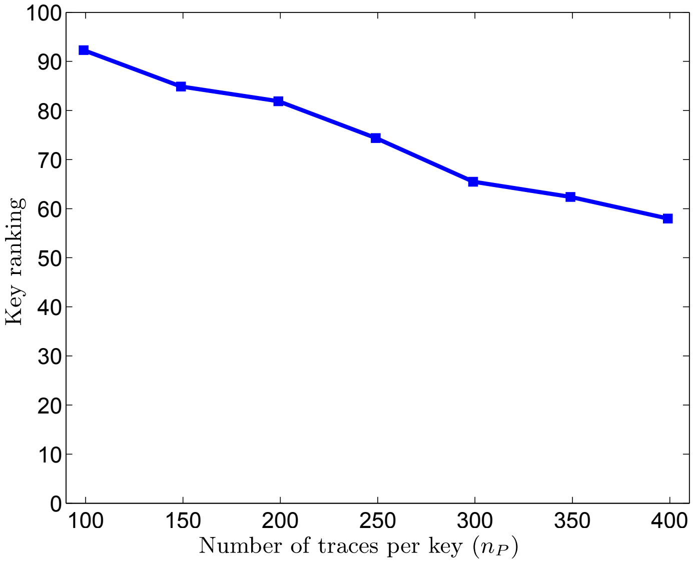

Fig. 3. Results for a naive collision attack.

The method does not provide convincing results with the $256 \times \underbrace{300}_{n_P}$ traces. However, the shape of the curve in the Figure 3 is decreasing. If the decrease is approximatively linear, the good secret key will begin to be distinguishable from $n_P \approx 1000$ traces. Nevertheless, this number is high, we will see in the following paragraph how we managed to decrease this value.

In this approach the comparison is horizontal. Indeed, the EM measurements C_1 and C_2 are sampled over t points, the returned coefficient is the cross correlation computed with the Pearson coefficient:

$$\rho\,(C_1, C_2) = \frac{covariance(C_1, C_2)}{\sigma(C_1)\sigma(C_2)}. \tag{7}$$

The main drawback of this method is the need to perfectly align the traces as the correlation coefficient largely depends on the adjustment of the traces' position. The toy example in Figures 4 and 5 of EM measurement show the great dependency between the coefficient correlation and the alignment of traces. This small demonstration and our practical experiences convinced us to use another collisions detection techniques.

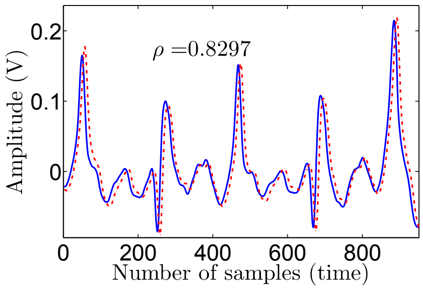

Fig. 4. Traces without retouching. © [19]

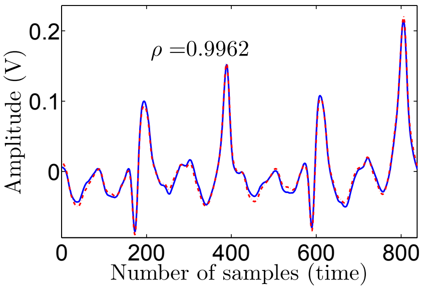

Fig. 5. Shifted traces on 7 points sample. © [19]

Due those lukewarm results we investigate another technique for detecting collisions.

Advanced collisions detection The aim this time is to detect if there exists a link in the EM measurements during the two targeted multiplications using "vertical correlations" as initially proposed in [40].

Instead of comparing the traces between each other and giving a coefficient that indicates whether there is a collision, it is a point-to-point comparison (where each point is a temporal instant within each trace).

Figure 6 illustrates this principle. The left pattern corresponds to the multiplication $x_{P_j} \times \lambda^2$ and the other one corresponds to the operation $x \times \lambda^2$. For the sake of have "clear" pictures, Figure 6 only shows three traces ($n_P = 3$).

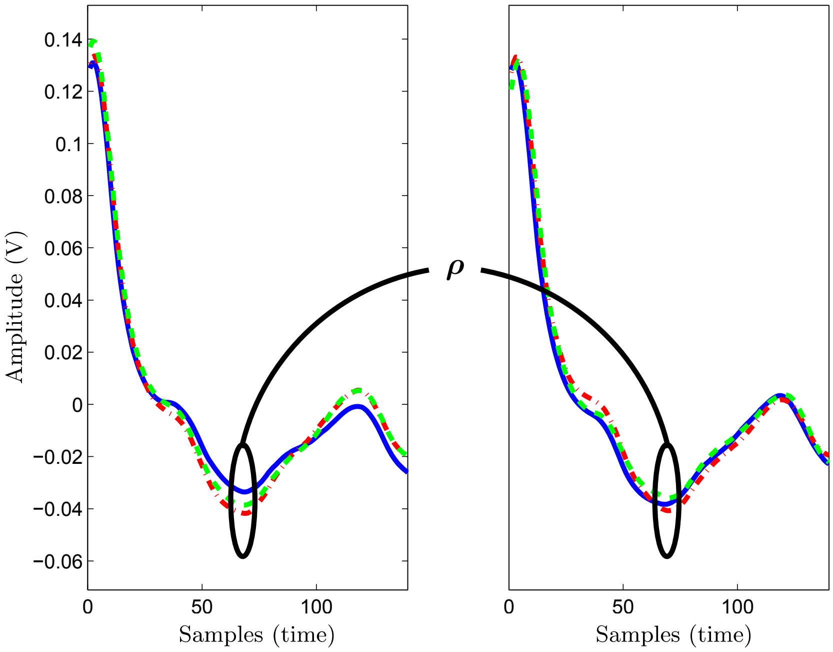

Fig. 6. Vertical correlation collision principle.

The trick proposed by Varchola *et al.* [40] to avoid the synchronization problem consists in the selection of a time instant in the first multiplication, a window in the second one and drag the single vector on the window to compute t correlations.

More precisely, for a fixed known input x_{P_j}, the collected EM measurements are $R_j \in \mathcal{M}_{n_P,2t}$ as we have seen in Algorithm 4. The result R is like the matrix in Equation 8.

$$R_j = \begin{pmatrix} C^{(1)}_{1,1} & C^{(1)}_{1,2} \dots & C^{(1)}_{1,t} & C^{(1)}_{2,1} & \dots & C^{(1)}_{2,t} \\ C^{(2)}_{1,1} & \dots & C^{(2)}_{1,t} & C^{(2)}_{2,1} & \dots & C^{(2)}_{2,t} \\ \vdots & & \vdots & \vdots & & \vdots \\ C^{(n_P)}_{1,1} & \dots & C^{(n_P)}_{1,t} & C^{(n_P)}_{2,1} & \dots & C^{(n_P)}_{2,t} \end{pmatrix}. \tag{8}$$

From there, the attacker builds a "correlation trace" $corr_j \in \mathcal{M}_{1,t}$ for a chosen time instant t_{interset} with $corr_j$ defined in Equation 9.

$$corr_j(i) = \rho\left(\begin{pmatrix} C^{(1)}_{1,t_{\text{interest}}} \\ \vdots \\ C^{(n_P)}_{1,t_{\text{interest}}} \end{pmatrix}, \begin{pmatrix} C^{(1)}_{2,i} \\ \vdots \\ C^{(n_P)}_{2,i} \end{pmatrix}\right), \forall i = 1, \dots, t \tag{9}$$

To illustrate the general shape of such a "correlation trace" we refer the reader to Figure 7. For this example we make a toy example with $n_P = 100$.

As in classical side-channel attacks, the highest correlation allows to identify the most probable key (the thickest blue curve in Figure 8, with $n_P = 400$).

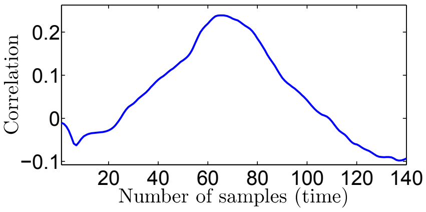

Fig. 7. Vertical correlation collision toy example. © [19]

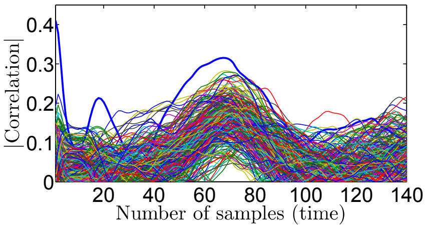

Fig. 8. Vertical correlation collision. © [19]

Practical results using the advanced collision detection In [40], the collision attack's success is supported by practical results. Their target is an 8-bit hardware implementation on an FPGA, their board was designed especially for the purpose of side-channel analyses. So, our experimental works are different in several points (see Table 1).

Table 1. Difference between target and set-up.

Settings	**Varchola *et al.* [40]**	**Our case**
Device	FPGA	Microcontroller
Architecture size	8-bit	32-bit
Clock frequency	16.384 MHz	24 MHz
Sampling frequency	20 Gsps	10 Gsps
Side-channel source	Power	EM

The main difference is of course the target of evaluation, hardware in their case and software in ours. Another important difference comes from the architecture as the attack targets the 8 least significant bits but the manipulated data are on 32 bits with the device producing leakages related to the 32-bit manipulated data. Therefore, unlike [40], our 256 keys hypothesis do not cover all the possible sought secret value.

Hence the question is the following: the traces are from 32-bit manipulated data, will the leakages be sufficiently meaningful to target only 8 bits at a time? Our attack is a chosen plaintext attack because the x_{P_j} have a particular shape, indeed, $x_{P_j} = (00\ldots0\boldsymbol{j_7j_6\ldots j_0})_2$

In our experimentation we chose $n_P = 400$ and hence we have the attack results presented in Figure 9. Each score is obtained by averaging the results for 100 attacks with different traces. These figures show the ranking of the correct key (8 least significant bits) among the 256 possible ones. The guess is correct when the rank equals to one. This ranking decreases with the number of traces in a significant way.

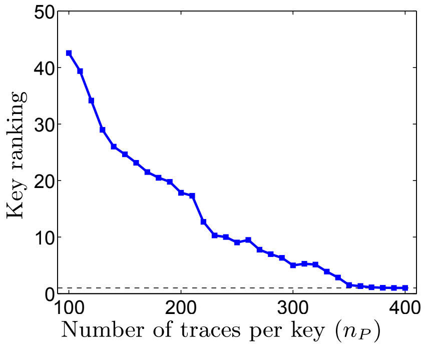

Fig. 9. Byte 1. © [19]

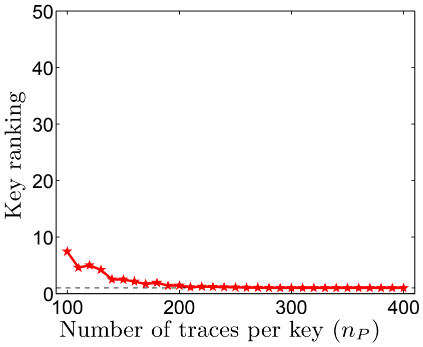

Fig. 10. Byte 2. © [19]

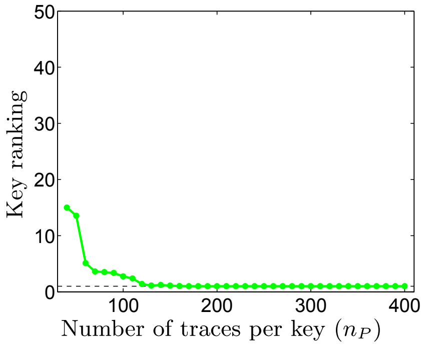

Fig. 11. Byte 3.

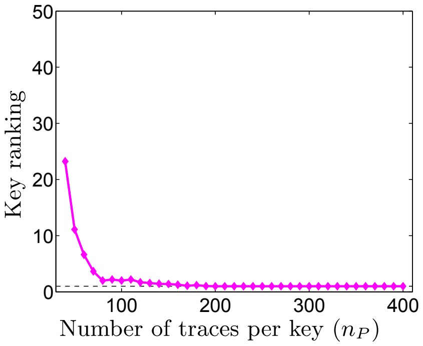

Fig. 12. Byte 4.

The attack recovers, the 8-bit key when the number of trace per key n_P is close to 400 (i.e. a total of EM measurement close to $n_P \times 2^8 = 400 \times 256 = 102400$). The secret key can be easily discriminated in a small set of candidates, resulting a huge loss of entropy.

5.2 Recovering the full 256 secret bits integer with collisions

In order to recover the full secret point during the Tate pairing calculation that we implemented and that we run on our 32-bit platform, we applied the method described previously to the other bytes, within the same 32-bit word first, and then for the other 32-bit words, in order to recover the full 256-bit secret integer. The used BN curves to implement the Tate pairing are set for $t = 3FC0100000000000$ (in hexadecimal).

Afterwards, the 8 least significant bits are fixed to the recovered byte and we pursue with the following byte. The attack on the other bytes is very similar to what has been described so far in the “advanced” technique. The known inputs x_{P_j} are different: they first “integrate” the 8 least significant bits recovered by the first step of the attack as carried in the previous section. Let $(\widehat{x_7}\widehat{x_6}\ldots\widehat{x_0})_2 = \widehat{x}$ be the 8 least significant bits recovered by the attack, then the chosen ciphertext is $x_{P_j} = (00\ldots0\boldsymbol{j_7j_6\ldots j_0}\widehat{x_7}\widehat{x_6}\ldots\widehat{x_0})_2$. Now, when j will have the same value as the secret, there will be a collision not only on 8 bits but also on 16 bits. When the 16 bits manipulated in the multiplications will be the same, the collision will be easier to detect than when there were only 8 identical bits. Practical results are provided in Figure 10.

It shows that the attack is easier as soon as the least significant bits are known. With only $n_P = 300$, the attack succeeds.

In the 32-bit word two bytes are still unknown. The same attack method allows us to recover these 32 secret bits. To attack the other words of the integer is not more complicated, everything relies on the proper understanding of the multiplication algorithm. Practical results are provided in Figure 11 and Figure 12. Efficiency of these attacks is similar when the target is the byte 2. That is about $n_P = 300$.

The cost of carrying the attack Our targeted Pairings implementations involve 256-bit length integer arithmetic. That is, since there are 8 words of 32-bit integer, then the previous attack needs to be performed 8 times. But, the messages (x_P) are chosen, so we can construct such x_P to recover the 8 words at the same time. It is like a parallel process :

1. Setting the messages $x_{P_j} = [X_{j,7}, X_{j,6}, \ldots, X_{j,0}]$ with the $X_{j,i}$ 32-bit word which are the x_{P_j} of section 9 and capture the side-channel leakages.
2. For each $i = 0, \ldots, 7$ make the attack to recover the 8 LSBs of each X_i.
3. Start again with the second least significant bits of X_i.

Thus, the attack to find the 256 bits does not require 8 times more traces than the one we presented to recover 32 bits. There are 8 independent attacks, but not 8 times more traces.

Thus, the number of traces required to break the x_Q coordinates of the secret input is:

$$E \simeq 2^8\,(400 + 3 \times 300) \simeq 3.5 \times 10^5$$

To compare with the attack against the unprotected version, recent results [20] show an attack with an averages of 200 traces to recover one word, so $8 \times 200 =$

1600 for the entire 256-bits secret integer. Then the countermeasure constrains the attacker to achieve 200 times more power measurements. In our experiment, one trace is acquired in an average of 0.4 second. Thus, the collision attack on the protected implementation take 2 days to recover the secret x_Q.

6 Conclusion

Several recent publications have addressed side-channel attacks against pairing-based cryptography. The present paper provides an overview of the different SCA schemes and a description of the countermeasures proposed to circumvent such attacks. To the best of our knowledge, these countermeasures have been proposed without any (theoretical or practical) security proofs. Our investigation thus constitutes the first critical analysis of the efficiency of one of these countermeasures. We have shown that the countermeasure based on point randomisation can be defeated using a collision-based side-channel attack. We also propose a method for improving the number of required curves for our attack. We have validated the feasibility of our attack against a pairing calculation which has been protected using this countermeasure.

At this stage we can therefore recommend to also randomize the secret at the beginning of the pairing. The randomization of Jacobian coordinates of the secret implies the non-reportion of the operation between the mask and the secret, and thus our comment on the collision is no longer valid. Moreover, to set the secret in Jacobian coordinates implies that the equations of the tangent and of the line are no longer in mixed coordinates (Equation 4). Then the generated overhead is eight modular multiplications (three in the computation of the tangent and five in the line).

Our analysis highlights the difficulty in devising countermeasures that protect implementations of complex cryptographic functions such as pairings against physical attacks. For such algorithms, tools should be developed to test whether the randomness properties that are initially introduced into a pairing calculation are sufficiently propagated across the entire computation, at least for as long as the secret point is still involved in the calculation.

Finally, recent papers [25, 4, 29] show new improvements in the algorithms used to solve discrete logarithm, in particular over BN curves. Those latter developments only require the redefinition of the Pairings' parameters and key sizes, in which case, in our opinion, our attack scenario would still hold as our attack is independent of the choice of the curve.

Acknowledgements

This work was supported in part by the EUREKA Catrene programme under contract CAT208 MobiTrust and by a French DGA-MRIS scholarship.

References

1. Diego F Aranha, Koray Karabina, Patrick Longa, Catherine H Gebotys, and Julio López. Faster Explicit Formulas for Computing Pairings over Ordinary Curves. In *EUROCRYPT*, pages 48–68. Springer, 2011.
2. J.C. Bajard and Nadia El Mrabet. Pairing in cryptography: an arithmetic point of view. *Proceedings of SPIE: ASPAAI*, 2007.
3. Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. Improving NFS for the discrete logarithm problem in non-prime finite fields. In *EUROCRYPT*, pages 129–155. Springer, 2015.
4. Razvan Barbulescu, Pierrick Gaudry, and Thorsten Kleinjung. The Tower Number Field Sieve. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015*, volume 9453, pages 31–58. Springer, November 2015.
5. Paulo S. L. M. Barreto and Michael Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. SAC'05, pages 319–331, Berlin, Heidelberg, 2005. Springer-Verlag.
6. PSLM Barreto, HY Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In *CRYPTO 2002*, pages 354–396. Springer, 2002.
7. Aurélie Bauer, Eliane Jaulmes, Emmanuel Prouff, and Justine Wild. Horizontal Collision Correlation Attack on Elliptic Curves. SAC'13. Springer, 2013.
8. Jean-Luc Beuchat, Jorge E González-Díaz, Shigeo Mitsunari, Eiji Okamoto, Francisco Rodríguez-Henríquez, and Tadanori Teruya. High-speed software implementation of the optimal ate pairing over barreto–naehrig curves. In *ICPBC*, pages 21–39. Springer, 2010.
9. Johannes Blömer, Peter Günther, and Gennadij Liske. Improved Side Channel Attacks on Pairing Based Cryptography. *COSADE*, 7864:154–168, 2013.
10. Dan Boneh and Matthew Franklin. Identity-Based Encryption from the Weil Pairing. In *Advances in Cryptology - CRYPTO 2001*, volume 32, pages 213–229. Springer, 2001.
11. JS Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. *CHES*, pages 292 – 302, 1999.
12. Iwan Duursma and HS Lee. Tate Pairing Implementation for Hyperelliptic Curves $y^2 = x^p - x + d$. *ASIACRYPT*, 4:111–123, 2003.
13. Kirsten Eisenträger, Kristin Lauter, and Peter L Montgomery. Improved weil and tate pairings for elliptic and hyperelliptic curves. In *International Algorithmic Number Theory Symposium*, pages 169–183. Springer, 2004.
14. Nadia El Mrabet, Giorgio Di Natale, Flottes, and Marie Lise. A Practical Differential Power Analysis Attack Against the Miller Algorithm. *PRIME*, pages 308–311, jul 2009.
15. Pierre-Alain Fouque and Frédéric Valette. The Doubling Attack – Why Upwards Is Better Than Downwards. In *CHES*, pages 269–280. Springer, 2003.
16. StevenD. Galbraith, Keith Harrison, and David Soldera. Implementing the Tate Pairing. In *Algorithmic Number Theory*, pages 324–337. Springer, 2002.
17. Santosh Ghosh and Dipanwita Roychowdhury. Security of prime field pairing cryptoprocessor against differential power attack. In *Security Aspects in Information Technology*, volume 7011 LNCS, pages 16–29. Springer, 2011.
18. Michael Hutter, Marcel Medwed, Daniel Hein, and Johannes Wolkerstorfer. Attacking ECDSA-Enabled RFID devices. *Applied Cryptography and Network Security*, pages 519–534, 2009.
19. Damien Jauvart, Jacques J. A. Fournier, and Louis Goubin. First practical side-channel attack to defeat point randomization in secure implementations of pairing-based cryptography. In *Proceedings of the 14th International Joint Conference on*

e-Business and Telecommunications - Volume 6: SECRYPT, (ICETE 2017), pages 104–115. INSTICC, SciTePress, 2017.

20. Damien Jauvart, Jacques Jean-Alain Fournier, Nadia El Mrabet, and Louis Goubin. Improving Side-Channel Attacks against Pairing-Based Cryptography. In *Risks and Security of Internet and Systems*. Springer, 2016.
21. Antoine Joux. A one round protocol for tripartite Diffie-Hellman. *Journal of Cryptology*, 2004.
22. Antoine Joux, Andrew Odlyzko, and Cécile Pierrot. The Past, evolving Present and Future of Discrete Logarithm. In *Open Problems in Mathematics and Computational Science*, pages 1–23. Springer, 2014.
23. Marc Joye and Gregory Neven (Eds). *Identity-Based Cryptography.* IOS Press, 2008.
24. Tae Hyun Kim, Tsuyoshi Takagi, Dong-Guk Han, Ho Won Kim, and Jongin Lim. Side Channel Attacks and Countermeasures on Pairing Based Cryptosystems over Binary Fields. *Cryptology and Network Security*, pages 168–181, 2006.
25. Taechan Kim and Razvan Barbulescu. Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case. Cryptology ePrint Archive, 2015.
26. Neal Koblitz and Alfred Menezes. Pairing-based cryptography at high security levels. *Cryptography and Coding*, 3796 LNCS:13–36, 2005.
27. C Kaya Koc, Tolga Acar, and Burton S Kaliski. Analyzing and comparing montgomery multiplication algorithms. *IEEE micro*, 16(3):26–33, 1996.
28. Paul Kocher, J Jaffe, and Benjamin Jun. Differential power analysis. *Advances in Cryptology - CRYPTO'99*, pages 1–10, 1999.
29. Alfred Menezes, Palash Sarkar, and Shashank Singh. Challenges with Assessing the Impact of NFS Advances on the Security of Pairing-based Cryptography. Cryptology ePrint Archive, 2016.
30. Victor Miller. Use of elliptic curves in cryptography. *CRYPTO '85*, 218, 1986.
31. Peter L. Montgomery. Modular multiplication without trial division. In *Mathematics of Computation*, volume 44, pages 519–519, 1985.
32. Amir Moradi, Oliver Mischke, and Thomas Eisenbarth. Correlation-Enhanced Power Analysis Collision Attack. In *CHES*, pages 125–139. Springer, 2010.
33. Michael Naehrig, Ruben Niederhagen, and Peter Schwabe. New software speed records for cryptographic pairings. In *LATINCRYPT*. Springer, 2010.
34. Dan Page and Frederik Vercauteren. Fault and Side-Channel Attacks on Pairing Based Cryptography. *IEEE Transactions on Computers*, 2004.
35. Weibo Pan and WP Marnane. A correlation power analysis attack against Tate pairing on FPGA. *Reconfigurable Computing: Architectures, Tools and Applications*, 2011.
36. Kai Schramm, Thomas Wollinger, and Christof Paar. A New Class of Collision Attacks and Its Application to DES. In *FSE*, pages 206–222. Springer, 2003.
37. Michael Scott. Computing the Tate pairing. *CT-RSA*, pages 293–304, 2005.
38. Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, 2nd edition, 2009.
39. Thomas Unterluggauer and Erich Wenger. Practical Attack on Bilinear Pairings to Disclose the Secrets of Embedded Devices. *ARES*, pages 69–77, 2014.
40. M. Varchola, M. Drutarovsky, M. Repka, and P. Zajac. Side channel attack on multiprecision multiplier used in protected ECDSA implementation. In *ReConFig*, pages 1–6, Dec 2015.
41. Claire Whelan and Mike Scott. Side Channel Analysis of Practical Pairing Implementations: Which Path Is More Secure? *VIETCRYPT 2006*, pages 99–114, 2006.