First Practical Side-Channel Attack to Defeat Point Randomization in Secure Implementations of Pairing-Based Cryptography

Damien Jauvart[1,2], Jacques J.A. Fournier[3] and Louis Goubin[2]
[1]*CEA Tech, Centre Microélectronique de Provence, 880 avenue de Mimet, 13541 Gardanne, France*
[2]*Laboratoire de Mathématiques de Versailles, UVSQ, CNRS, Université Paris-Saclay, 78035 Versailles, France*
[3]*CEA LETI, 17 rue des Martyrs, 38054 Grenoble Cedex 9, France*
{*damien.jauvart2, jacques.fournier*}*@cea.fr, louis.goubin@uvsq.fr*

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

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, which have been shown to recover the secret points used in some PBC-based schemes. 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/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 and this raises questions about the validation of countermeasures for complex cryptographic schemes such as PBC. We also discuss about ways of thwarting our attack.

1 INTRODUCTION

Bilinear pairings are used in cryptography for various innovative protocols. For example, in 2001, Boneh and Franklin published the Identity-Based Encryption (IBE) scheme based on Pairings (Boneh and Franklin, 2001). The one-round tripartite key exchange (Joux, 2004) based on Pairings is another interesting practical use of such cryptographic primitives.

Several studies have investigated about the vulnerability of Pairing-Based Cryptography (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. Although Joux (Joux et al., 2014) and Barbulescu (Barbulescu et al., 2015a) 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.

In an IBE (Boneh and Franklin, 2001) 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 (Boneh and Franklin, 2001) scheme the decryption step consists in deciphering the ciphertext $\{U,V\}$ 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 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 CPA[1]-like techniques to target the secret point (compared to using SPA[2]-like 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 and explain and illustrate how

[1]Correlation Power Analysis
[2]Simple Power Analysis

to defeat it. The paper is organized as follows. Section 2 recalls the basic definitions of and notations for pairings. We review related work in Section 3. Section 4 provides an analysis of one of those countermeasures which is based on point randomization and explains how this countermeasure can be defeated. Then 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 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 (Silverman, 2009).

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two abelian groups and $\mathbb{G}_3$ a multiplicative group of the same order. A pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$ with the following properties:

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

$$\begin{array}{l} e([a]P_1 + [b]P_2, Q) = e(P_1,Q)^a e(P_2,Q)^b \\ e(P,[a]Q_1 + [b]Q_2) = e(P,Q_1)^a e(P,Q_2)^b \end{array}.$$

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

Let E be an elliptic curve defined over $\mathbb{F}_q$. E can be written as

$$\begin{array}{rl} E = \{(x,y) \in \mathbb{F}_q \times \mathbb{F}_q | y^2 + a_1xy + a_3y \\ = x^3 + a_2x^2 + a_4x + a_6\} \cup \{O\}, \end{array} \quad (1)$$

where 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 $[l]P = O$), the *rational torsion points* are given by $E(\mathbb{F}_q)[l] := E(\mathbb{F}_q) \cap E[l]$. The group $E(\mathbb{F}_q)[l]$ contains a point of order 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.

The Tate Pairing. The widely used Tate pairing (Barreto et al., 2002; Eisenträger et al., 2004; Galbraith et al., 2002; Scott, 2005) 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.

The Barreto–Naehrig curves (Barreto and Naehrig, 2005). Such curves are 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:

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

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.

Miller's Algorithm. The computation of such a map is a well known problem (Barreto et al., 2002) and an efficient way of computing such pairings was proposed as a recursive scheme by Miller (Miller, 1986). Miller's algorithm, which works as the main calculation to compute a pairing, uses an iterative relation to find a rational function f_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: $f_P(Q) \in \mathbb{G}_3$

$T \leftarrow P$;
$f \leftarrow 1$;
for $i = n-1$ **downto** 0 **do**
  $f \leftarrow f^2 \frac{l_{T,T}(Q)}{v_{[2]T}(Q)}$;
  $T \leftarrow [2]T$;
  if $l_i == 1$ **then**
    $f \leftarrow f \frac{l_{T,P}(Q)}{v_{T+P}(Q)}$;
    $T \leftarrow T + P$;
  end
end
return f;

In Algorithm 1:

1. $l_{T,T}(Q)$ is the equation of the tangent at T evaluated at point Q.

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

2. $l_{T,P}(Q)$ is the equation of the line through T and P evaluated at point Q.
3. $v_R(Q)$ is the equation of the vertical line at R evaluated at Q.

These equations can be optimized by using mixed system coordinates for the points' representations as suggested in (Aranha et al., 2011; Bajard and El Mrabet, 2007; Beuchat et al., 2010; Koblitz and Menezes, 2005) and (Naehrig et al., 2010):

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. These equations are presented without their denominator because they are elements of a strict subfield of $\mathbb{F}_{q^k}$ and therefore the final exponentiation sends these elements to 1 (the neutral element for multiplication).

In the following, our implementation is a Tate pairing over Barreto and Naehrig curves (Barreto and Naehrig, 2005).

2.1 Application to identity-based encryption

An IBE scheme can be used to simplify a widely known issue in public key cryptography: the key exchange. 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 le 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 (Boneh and Franklin, 2001) work in four steps: 1. Set-up; 2. Extraction; 3. Encryption; 4. Decryption.

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 $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear pairing. Let $P \in \mathbb{G}_1$ be a generator of $\mathbb{G}_1$. let $H_1 : \{0,1\}^\star \rightarrow \mathbb{G}_1^\star$ and $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$ be two cryptographic hash functions. Let $s \in \mathbb{Z}_r$ be a random 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

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

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 onto $\mathbb{G}_1$ 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).

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. And, she computes the ciphertext $C = \{[k]P, M \oplus H_2(g_B^k)\}$ and sends it to Bob.

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 (Kocher et al., 1999). 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.

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

In this paper, Page and Vercauteren (Page and Vercauteren, 2004) simulated an attack on the Duursma–Lee Algorithm (Duursma and Lee, 2003) 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, with $e([a]P,[b]Q)^{1/ab} = e(P,Q)$, then for each pairing computation, we take different values for a and b, and compute $e([a]P,[b]Q)^{1/ab}$. The second countermeasure, proposed in (Page and Vercauteren, 2004), 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 (Pan and Marnane, 2011) 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 supersingular 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.* (Kim et al., 2006) 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.* (Kim et al., 2006) implemented the third countermeasure proposed by Coron (Coron, 1999), 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 (Scott, 2005). It consists in randomizing the Miller variable in Algorithm 1 by multiplying the operations 4 and 7 by a random $\lambda \in \mathbb{F}_q$. 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 (Whelan and Scott, 2006) 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 point Q (second entry) as secret for the attack to work. The latter conclusion was refuted in (El Mrabet et al., 2009), 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.* (Ghosh and Roychowdhury, 2011). 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 (El Mrabet et al., 2009; Whelan and Scott, 2006) and (Blömer et al., 2013).

Moreover, Blömer *et al.* (Blömer et al., 2013) 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 (Page and Vercauteren, 2004) with no overhead.

The importance of implementing countermeasures is supported by the recent results of Unterluggauer and Wenger (Unterluggauer and Wenger, 2014) and Jauvart *et al.* (Jauvart et al., 2016), 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. 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, 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 in 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 (Schramm et al., 2003). 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 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 (Moradi et al., 2010).

The use of collisions against implementations of public key cryptographic algorithms have also been described. To achieve this, Fouque and Valette (Fouque and Valette, 2003) 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 (Bauer et al., 2013). 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 (Varchola et al., 2015), 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 (Hutter et al., 2009). 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 (Bauer et al., 2013) and (Fouque and Valette, 2003), 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 analyze one of these countermeasures: Miller's algorithm with randomized Jacobian coordinates. First, we show how collisions can be used to make this countermeasure fail. We introduce a first "straightforward" scheme to detect collisions and we show that this approach has its limits in practice. Then we adapt a refined method proposed in (Varchola et al., 2015) 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 The Miller's algorithm with randomized Jacobian coordinates

For performance reasons, the use of mixed affine-Jacobian coordinates has been often proposed in the literature (Aranha et al., 2011; Bajard and El Mrabet, 2007; Beuchat et al., 2010; Koblitz and Menezes, 2005) and (Naehrig et al., 2010). 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. $T \leftarrow (X_T : Y_T : Z_T)$.

The above steps 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 2. In the end, the mask λ is automatically removed in the final exponentiation as $\lambda^{\frac{q^k-1}{l}} = 1$.

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

Of course, since the recent results of Joux (Joux et al., 2014) and Barbulescu (Barbulescu et al., 2015a), 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.

Algorithm 2: 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**
  $f \leftarrow f^2 \frac{l_{T,T}(Q)}{v_{[2]T}(Q)}$;
  $T \leftarrow [2]T$;
  if $l_i == 1$ **then**
    $f \leftarrow f \frac{l_{T,P}(Q)}{v_{T+P}(Q)}$;
    $T \leftarrow T + P$;
  end
end
return f;

4.2 Detecting collisions in point-randomized Pairing calculations

Our attack is based on the following observation: in Algorithm 2 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 (Montgomery, 1985). 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.

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.

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

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 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_times_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:

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

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 3.

Algorithm 3: EM measurements acquisition procedure.

Data: *n_times_P*, the repetition number
Result: A data base of EM measurement $R \in \mathcal{M}_{256,n\_times\_P,2t}$, t is the traces length

for $j = 0$ **to** 255 **do**
  $x_j \leftarrow (0\ldots0 j_7 j_6 \ldots j_0)_2$; // $j = (j_7 \ldots j_0)_2$ in radix 2 representation
  for $i = 0$ **to** *n_times_P* $- 1$ **do**
    $\lambda \leftarrow$ random in $\{0,\ldots,2^{32}-1\}$;
    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 1.

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

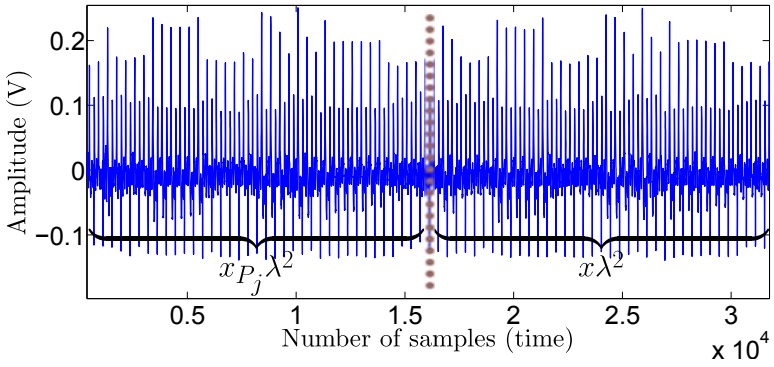

Figure 1: An example of electromagnetic emanation measurement.

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 4.2 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_times_P* repetitions.

We used this method with our EM measurements by using the correlation criterion and another one, named BCDC (Bounded Collisions Detection Criterion). As shown in (Diop et al., 2015), this criterion detection can be used instead of correlation. This criterion takes two traces C_1 and C_2, compares them by computing $\frac{1}{\sqrt{2}} \frac{\sigma(C_1 - C_2)}{\sigma(C_1)}$ and returns a value in $[0,1]$. A collision is detected if the value of BCDC is close to 0. The notation $\sigma(C_1)$ denotes the standard deviation of the leakage vector C_1 while $\sigma(C_1 - C_2)$ is the standard deviation of the difference $C_1 - C_2$.

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

The method does not provide convincing results

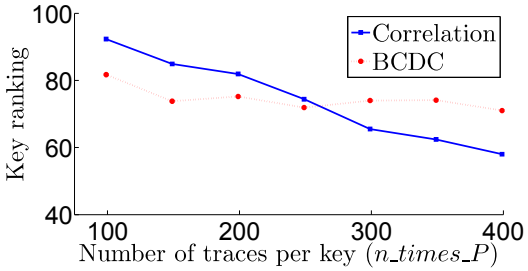

Figure 2: Results for a naive collision attack.

even if we use more EM measurements.

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:

$$\rho(C_1, C_2) = \frac{covariance(C_1, C_2)}{\sigma(C_1)\sigma(C_2)}. \quad (5)$$

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 3 and 4 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.

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

5.1.1 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 (Varchola et al., 2015).

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

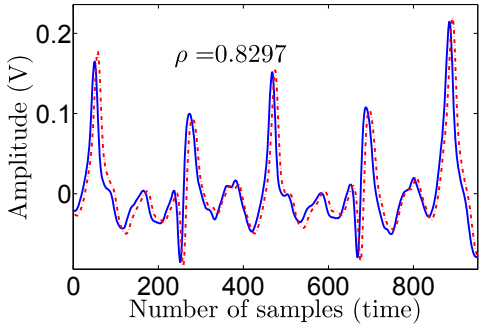

Figure 3: Traces without retouching.

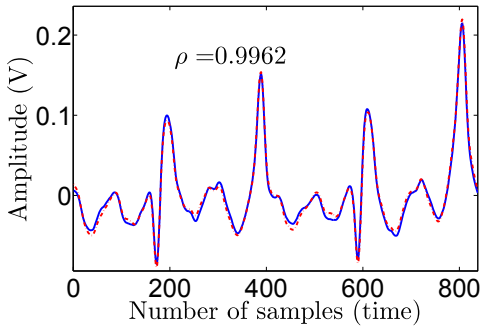

Figure 4: Shifted traces on 7 points sample.

(where each point is a temporal instant within each trace).

Figure 5 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 5 only shows three traces ($n\_times\_P = 3$).

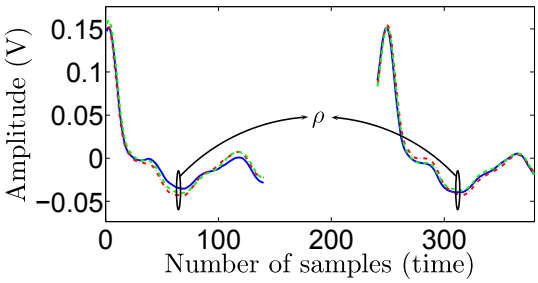

Figure 5: Vertical correlation collision principle.

The trick proposed by Varchola *et al.* (Varchola et al., 2015) to avoid the synchronization problem is to select 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 $R_j \in \mathcal{M}_{n\_times\_P,2t}$ as we have seen in Algorithm 3. The result R is like the matrix in Equation 6.

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 7.

To illustrate the general shape of such a "correlation trace" we refer the reader to Figure 6. For this example we make a toy example with $n\_times\_P = 100$.

As in classical side-channel attacks, the highest correlation allows to identify the most probable key (the thickest blue curve in Figure 7, with $n\_times\_P = 400$).

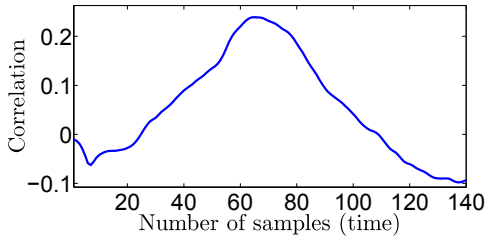

Figure 6: Vertical correlation collision toy example.

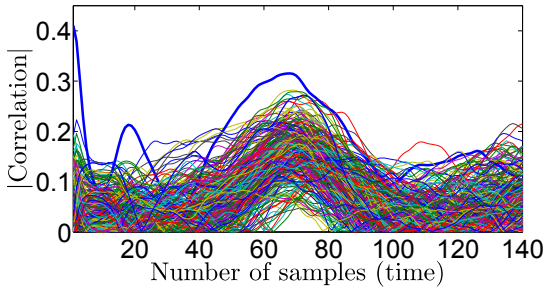

Figure 7: Vertical correlation collision.

5.1.2 Practical results using the advanced collision detection

In (Varchola et al., 2015), 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).

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 (Varchola et al., 2015), 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

$$R_j = \begin{pmatrix} C_{1,1}^{(1)} & C_{1,2}^{(1)} & \dots & C_{1,t}^{(1)} & C_{2,1}^{(1)} & \dots & C_{2,t}^{(1)} \\ C_{1,1}^{(2)} & \dots & & C_{1,t}^{(2)} & C_{2,1}^{(2)} & \dots & C_{2,t}^{(2)} \\ \vdots & & & \vdots & \vdots & & \vdots \\ C_{1,1}^{(n\_times\_P)} & \dots & & C_{1,t}^{(n\_times\_P)} & C_{2,1}^{(n\_times\_P)} & \dots & C_{2,t}^{(n\_times\_P)} \end{pmatrix}. \quad (6)$$

$$corr_j(i) = \rho\left(\begin{pmatrix} C_{1,t_{\text{interest}}}^{(1)} \\ \vdots \\ C_{1,t_{\text{interest}}}^{(n\_times\_P)} \end{pmatrix}, \begin{pmatrix} C_{2,i}^{(1)} \\ \vdots \\ C_{2,i}^{(n\_times\_P)} \end{pmatrix}\right), \forall i = 1, \dots, t \quad (7)$$

Table 1: Difference between target and set-up.

Settings	(Varchola et al., 2015)	**Our case**
Device	FPGA	Microcontroller
Architecture size	8-bit	32-bit
Clock frequency	16.384 MHz	50 MHz
Sampling frequency	20 Gsps	10 Gsps
side-channel	Power	EM

time? Our attack is a chosen ciphertext attack because the x_{P_j} have a particular shape, indeed, $x_{P_j} = (00\dots0j_7j_6\dots j_0)_2$

In our experimentation we chose $n\_times\_P = 400$ and hence we have the attack results presented in Figure 8 (green squares). 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.

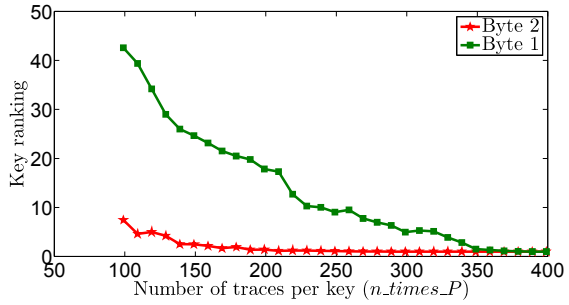

Figure 8: Key ranking for the attack against the least significant byte and the following byte.

The attack recovers, the 8-bit key when the number of trace per key n_times_P is close to 400 (i.e. a total of EM measurement close to $n\_times\_P \times 2^8 = 400 \times 256 = 102400$). 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).

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 $(\widehat{x_7}\widehat{x_6}\dots\widehat{x_0})_2 = \widehat{x}$ be the 8 least significant bits recovered by the attack, then the chosen ciphertext is $x_{P_j} = (00\dots0j_7j_6\dots j_0\widehat{x_7}\widehat{x_6}\dots\widehat{x_0})_2$. 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 8 (red stars).

It shows that the attack is easier as soon as the least significant bits are known. With only $n\_times\_P = 300$, 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.

5.2.1 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 5.1.2 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 non-protected version, recent results (Jauvart et al., 2016) 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 (Kim and Barbulescu, 2015; Barbulescu et al., 2015b; Menezes et al., 2016) 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 Mobi-Trust and by a French DGA-MRIS scholarship.

REFERENCES

Aranha, D. F., Karabina, K., Longa, P., Gebotys, C. H., and López, J. (2011). Faster Explicit Formulas for Computing Pairings over Ordinary Curves. In *EUROCRYPT*, pages 48–68. Springer.

Bajard, J. and El Mrabet, N. (2007). Pairing in cryptography: an arithmetic point of view. *Advanced Signal Processing Algorithms, Architectures, and Implementations*.

Barbulescu, R., Gaudry, P., Guillevic, A., and Morain, F. (2015a). Improving NFS for the discrete logarithm problem in non-prime finite fields. In *EUROCRYPT*, pages 129–155. Springer.

Barbulescu, R., Gaudry, P., and Kleinjung, T. (2015b). The Tower Number Field Sieve. In Iwata, T. and Cheon, J. H., editors, *ASIACRYPT 2015*, volume 9453, pages 31–58. Springer.

Barreto, P., Kim, H., Lynn, B., and Scott, M. (2002). Efficient algorithms for pairing-based cryptosystems. In *CRYPTO 2002*, pages 354–396. Springer.

Barreto, P. S. L. M. and Naehrig, M. (2005). Pairing-Friendly Elliptic Curves of Prime Order. SAC'05, pages 319–331, Berlin, Heidelberg. Springer-Verlag.

Bauer, A., Jaulmes, E., Prouff, E., and Wild, J. (2013). Horizontal Collision Correlation Attack on Elliptic Curves. SAC'13, pages 553–570. Springer.

Beuchat, J.-L., González-Díaz, J. E., Mitsunari, S., Okamoto, E., Rodríguez-Henríquez, F., and Teruya, T. (2010). High-speed software implementation of the optimal ate pairing over barreto–naehrig curves. In *ICPBC*, pages 21–39. Springer.

Blömer, J., Günther, P., and Liske, G. (2013). Improved Side Channel Attacks on Pairing Based Cryptography. *COSADE*, 7864:154–168.

Boneh, D. and Franklin, M. (2001). Identity-Based Encryption from the Weil Pairing. In *Advances in Cryptology - CRYPTO 2001*, volume 32, pages 213–229. Springer.

Coron, J. (1999). Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. *CHES*, pages 292 – 302.

Diop, I., Liardet, P.-Y., Linge, Y., and Maurine, P. (2015). Collision based attacks in practice. In *DSD*, pages 367–374. IEEE.

Duursma, I. and Lee, H. (2003). Tate Pairing Implementation for Hyperelliptic Curves $y^2 = x^p - x + d$. *ASIACRYPT*, 4:111–123.

Eisenträger, K., Lauter, K., and Montgomery, P. L. (2004). Improved weil and tate pairings for elliptic and hyperelliptic curves. In *International Algorithmic Number Theory Symposium*, pages 169–183. Springer.

El Mrabet, N., Di Natale, G., Flottes, and Lise, M. (2009). A Practical Differential Power Analysis Attack Against the Miller Algorithm. *PRIME*, pages 308–311.

Fouque, P.-A. and Valette, F. (2003). The Doubling Attack – Why Upwards Is Better Than Downwards. In *CHES*, pages 269–280. Springer.

Galbraith, S., Harrison, K., and Soldera, D. (2002). Implementing the Tate Pairing. In *Algorithmic Number Theory*, pages 324–337. Springer.

Ghosh, S. and Roychowdhury, D. (2011). Security of prime field pairing cryptoprocessor against differential power attack. In *Security Aspects in Information Technology*, volume 7011 LNCS, pages 16–29. Springer.

Hutter, M., Medwed, M., Hein, D., and Wolkerstorfer, J. (2009). Attacking ECDSA-Enabled RFID devices. *Applied Cryptography and Network Security*, pages 519–534.

Jauvart, D., Fournier, J. J.-A., El Mrabet, N., and Goubin, L. (2016). Improving Side-Channel Attacks against Pairing-Based Cryptography. In *Risks and Security of Internet and Systems*. Springer.

Joux, A. (2004). A one round protocol for tripartite Diffie-Hellman. *Journal of Cryptology*.

Joux, A., Odlyzko, A., and Pierrot, C. (2014). The Past, evolving Present and Future of Discrete Logarithm. In *Open Problems in Mathematics and Computational Science*, pages 1–23. Springer.

Kim, T. and Barbulescu, R. (2015). Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case. Cryptology ePrint Archive.

Kim, T. H., Takagi, T., Han, D.-G., Kim, H. W., and Lim, J. (2006). Side Channel Attacks and Countermeasures on Pairing Based Cryptosystems over Binary Fields. *Cryptology and Network Security*, pages 168–181.

Koblitz, N. and Menezes, A. (2005). Pairing-based cryptography at high security levels. *Cryptography and Coding*, 3796 LNCS:13–36.

Kocher, P., Jaffe, J., and Jun, B. (1999). Differential power analysis. *Advances in Cryptology - CRYPTO'99*, pages 1–10.

Menezes, A., Sarkar, P., and Singh, S. (2016). Challenges with Assessing the Impact of NFS Advances on the Security of Pairing-based Cryptography. Cryptology ePrint Archive.

Miller, V. (1986). Use of elliptic curves in cryptography. *CRYPTO '85*, 218:417–426.

Montgomery, P. L. (1985). Modular multiplication without trial division. In *Mathematics of Computation*, volume 44, pages 519–519.

Moradi, A., Mischke, O., and Eisenbarth, T. (2010). Correlation-Enhanced Power Analysis Collision Attack. In *CHES*, pages 125–139. Springer.

Naehrig, M., Niederhagen, R., and Schwabe, P. (2010). New software speed records for cryptographic pairings. In *LATINCRYPT*, pages 109–123. Springer.

Page, D. and Vercauteren, F. (2004). Fault and Side-Channel Attacks on Pairing Based Cryptography. *IEEE Transactions on Computers*.

Pan, W. and Marnane, W. (2011). A correlation power analysis attack against Tate pairing on FPGA. *Reconfigurable Computing: Architectures, Tools and Applications*.

Schramm, K., Wollinger, T., and Paar, C. (2003). A New Class of Collision Attacks and Its Application to DES. In *Fast Software Encryption*, pages 206–222. Springer.

Scott, M. (2005). Computing the Tate pairing. *CT-RSA*, pages 293–304.

Silverman, J. H. (2009). *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, 2nd edition.

Unterluggauer, T. and Wenger, E. (2014). Practical Attack on Bilinear Pairings to Disclose the Secrets of Embedded Devices. *ARES*, pages 69–77.

Varchola, M., Drutarovsky, M., Repka, M., and Zajac, P. (2015). Side channel attack on multiprecision multiplier used in protected ECDSA implementation. In *ReConFig*, pages 1–6.

Whelan, C. and Scott, M. (2006). Side Channel Analysis of Practical Pairing Implementations: Which Path Is More Secure? *VIETCRYPT 2006*, pages 99–114.