Improved Side Channel Attacks on Pairing Based Cryptography

Johannes Blömer* Peter Günther† Gennadij Liske‡

University of Paderborn
Department of Computer Science
Germany

January 24, 2012

Abstract

Techniques from pairing based cryptography (PBC) are used in an increasing number of cryptographic schemes. With progress regarding efficient implementations, pairings also become interesting for applications on smart cards. With these applications the question of the vulnerability to side channel attacks (SCAs) arises. Several known invasive and non-invasive attacks against pairing algorithms only work if the second but not if the first argument of the pairing is the secret. In this paper we extend some of these attacks also to the case where the first argument is the secret. Hence we may conclude that positioning the secret as the first argument of the pairing does not improve the security against SCAs, as it sometimes has been suggested.

1 Introduction

Since the invention of the first fully functional identity based encryption (IBE) scheme [BF03], that was based on bilinear pairings, pairings have become an important tool in cryptography. Today numerous schemes such as attribute based encryption [GPSW06], short signatures [BLS04], and anonymous group signatures [BS04] make use of pairings as their building blocks. The adoption of pairings in cryptographic applications is followed by the request for efficient implementations. Over the past years research efforts led to pairings that have

*bloemer@upb.de

†peter.guenther@uni-paderborn.de. This author was supported by the German Federal Ministry of Education and Research (BMBF), grant 01IS10030C.

‡utyf@mail.upb.de

efficient implementations and can be implemented on resource constrained devices such as smart cards [SCA06]. It is well known that in this case, mathematical cryptanalysis is not sufficient. Instead the vulnerability to side channel attacks (SCAs) has to be evaluated as well.

Bilinear pairings are usually realized on groups of elliptic curves. Although pairing based cryptography (PBC) uses methods from elliptic curve cryptography (ECC), the vulnerability of PBC against SCAs is not well understood. In ECC based schemes, such as ECDSA, the secret is a scalar multiplier of a point on the curve. In PBC, the secret is usually a point on the curve. Often this point is an argument of the pairing. Therefore, the pairing itself is an interesting target for SCAs. But when it comes to PBC the effort that has been spent on the analysis of SCAs is much smaller than in the case of standart ECC. Nevertheless, there are some results that analyze the vulnerability of pairings to passive attacks as well as to active attacks [PV04, KTH$^+$06, WS06, WS07, PV06].

There is a variety of pairings that can be used for PBC, e.g., the Weil pairing, the Tate pairing, the eta pairing, and their variations. Obviously, SCAs depend heavily on the pairing and its specific implementation. Regarding non-invasive attacks, i.e., attacks that exploit timings, power consumption, or electro-magnetic radiance, in [KTH$^+$06] the authors have investigated attacks for implementations of the eta pairing on supersingular curves in characteristic 2. In [WS06], differential power analysis (DPA) based attacks on the Tate pairing were analyzed. The authors showed how to attack implementations of the Tate pairing if the second argument represents the secret. Bilinear pairings are not symmetric in their arguments. Hence, an attack on the second argument does not necessarily imply an attack on the first argument of the pairing. Furthermore, the authors of [WS06] conjectured that it may be more difficult to attack some implementations of the Tate pairing if the first argument of the pairing is the secret. In [MFN09] this problem has been addressed for the case where the first argument is represented in Jacobian coordinates. Here, a DPA of a modular multiplication and a DPA of a modular addition was required to succeed.

In this paper we show that the attack from [WS06] that is based on a DPA of the modular multiplication can be extended to the case where the first argument is the secret. But, contrary to [MFN09] our attack requires either a DPA of the multiplication *or* a DPA of the addition. For pairings that are defined on elliptic curves over finite fields, usually an extension field of the base fields is required. For our results, we assume that the first argument of the pairing is defined over the base field while the second argument is defined over the extension field. This setting is relevant for many efficient implementations [BKLS02]. Furthermore, we introduce a DPA on the modular addition and give evidence for its feasibility. Then we use this DPA to describe a possiblity to attack the pairing with the first argument beeing secret. Here, we require that the base field has large prime characteristic.

With respect to invasive SCAs the first result in the context of PBC was presented in [PV06]. The authors attacked two algorithms for the Tate pairing. Later, the vulnerability of several algorithms for the Weil, Tate and eta pairings in presence of fault attacks was studied in [WS07]. The authors of [Mra09] analyze the vulnerability of Miller's algorithm against the same fault type as used in [PV06]. To apply these results to conrete pairing algorithms that are based on Miller's algorithm, stronger assumptions about the involved faults are necessary than in [PV06] and [WS07].

In this paper we are especially interested in the fault attacks from [WS07] on an algorithm for the eta pairing. This algorithm consists of a certain number of loop iterations. The authors analyzed the consequences of faults in different memory cells where intermediate values are stored. The two most promising attacks against the eta pairing assumed a random fault in specific memory cells during the execution of the last loop iteration. Both of these attacks can be used to recover the second argument of the pairing. However, if the secret is used as the first argument the results are more restricted. In this paper we show that the recovery of the first argument is not more complex. Furthermore we show that the restriction of the fault attacks of [WS07] to the last loop iteration is not necessary.

Altogether, we conclude that schemes where the first argument is the secret are not less vulnerable to non-invasive and invasive SCAs than schemes where the second argument is the secret.

The work is organized as follows. In Section 2, we introduce the necessary background on elliptic curve cryptography, pairing based cryptography, and non-invasive side channel analysis. In Section 3, we present one of our main results, namely a non-invase attack on an implementation of the Tate pairing when the first argument is secret. We do this by means of a DPA of the modular multiplication and of the modular addition. A DPA for the multiplication has been described in [WS06] that we use in a different manner. A DPA on the modular addition is first presented in this paper in Section 4. Furthermore, a countermeasure against our attack is presented. However, we give only heuristic arguments for its effectiveness. In Section 5 we show that the fault attacks of [WS07] are as powerful in the case the first argument of the pairing is the secret as in the case where the second argument is the secret.

2 Background

Before we will provide some background about ECC and especially PBC we will introduce the notation that we follow throughout this work.

2.1 Notation

By $\mathbb{F}_q$ we denote a field of size q. If q is prime we emphasize this by the notation $\mathbb{F}_p$. We always assume that elements of $\mathbb{F}_p$ are stored in the binary representation of their representatives in $\mathbb{Z}_p$. With $\mathbb{F}_{q^e}$ we denote an extension field of $\mathbb{F}_q$ with size q^e. Here, we assume that elements of $\mathbb{F}_{q^e}$ are represented as elements from the $\mathbb{F}_q$-vector field of dimension e. For $a \in \mathbb{F}_{q^e}$ we denote the i-th component of this vector as $a^{(i)}$. Hence, $a \in \mathbb{F}_{q^e}$ is represented as $a = \left(a^{(1)}, \ldots, a^{(e)}\right)$ with $a^{(i)} \in \mathbb{F}_q$.

2.2 Elliptic Curve Cryptography

Consider elliptic curves in general Weierstrass form

$$E : F(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 \text{ with } a_i \in \mathbb{F}_q. \quad (1)$$

We define the $\mathbb{F}_q$-rational points of an elliptic curve E:

$$E(\mathbb{F}_q) = \left\{(x, y) \in (\mathbb{F}_q)^2 \middle| F(x, y) = 0\right\} \cup \{\mathcal{O}\}.$$

Here, $\mathcal{O}$ denotes an additional point, called point at infinity.

It is well known that an additive group can be defined on $E(\mathbb{F}_q)$ with $\mathcal{O}$ as the neutral element [Sil09]. Additionally, we define the scalar multiplication as $aP = \sum_{i=1}^{a} P$ for $a \in \mathbb{N}$. The subgroup $E(\mathbb{F}_q)[a] \subseteq E(\mathbb{F}_q)$ is defined as the kernel of the multiplication by a, i.e., it contains all points of order dividing a. It is called the group of a-torsion points.

2.3 Pairing Based Cryptography

The key element of pairing based cryptography is a bilinear map called pairing. There are different pairings known in the literature. The most important are the Weil pairing and the Tate pairing [BSS05]. Later in this section, we introduce two implementations of the Tate pairing. The first one is based on the Miller algorithm [Mil04]. The second one is given for characteristic 2 by the eta pairing [Pau07].

Next, we will define the Tate pairing and list some of its properties.

2.3.1 Definition of the Tate Pairing

In order to define the domain of the pairing, we need an additional parameter called embedding degree.

Definition 1. Let l be the order of a subgroup of $E(\mathbb{F}_q)$. Then the embedding degree, with respect to l and q, is defined as the smallest positive integer k such that $l|q^k - 1$.

The subgroup $E(\mathbb{F}_{q^k})[l]$ is isomorphic to the direct product $\mathbb{Z}_l \times \mathbb{Z}_l$. We will only consider the case when l is prime. Then $\mathbb{Z}_l$ is a field and hence $E(\mathbb{F}_{q^k})[l]$ is isomorphic to a vector space of dimension 2 over $\mathbb{Z}_l$ [BSS05]. As a consequence $E(\mathbb{F}_{q^k})[l]$ is generated by two linear independent elements.

Let $P \in E(\mathbb{F}_{q^k})[l]$ and D_Q be a divisor that is equivalent to $(Q) - (\mathcal{O})$. Note that the mapping from Q to D_Q is an isomorphism between the points on the elliptic curve and equivalence classes of divisors. With $f_{l,P}$ we denote the evaluation of a function with divisor $l\,(P) - l\,(\mathcal{O})$. For more background on divisors in the context of elliptic curves, see [Sil09]. The Tate pairing with final exponentiation [BSS05] is defined as:

$$\begin{array}{rccc} \mathrm{e}: & E(\mathbb{F}_{q^k})[l] \times E(\mathbb{F}_{q^k})/lE(\mathbb{F}_{q^k}) & \rightarrow & \mathbb{F}_{q^k}^* \\ & (P,Q) & \mapsto & f_{l,P}(D_Q)^{(p^k-1)/l}. \end{array} \tag{2}$$

The Tate pairing is a non-degenerate, bilinear map [BSS05]. To be interesting for cryptographic applications, the groups $E(\mathbb{F}_{q^k})[l]$ and $E(\mathbb{F}_{q^k})/lE(\mathbb{F}_{q^k})$ must satisfy additional properties, see for example [BF03].

2.3.2 Applications of Pairings in Cryptography

Pairing based cryptography allows the realization of many powerful schemes. Examples are identity based encryption [BF03], short signatures [BLS04], and attribute based encryption [GPSW06]. Here, we are interested in schemes where the secret key is one argument of the pairing. Take the identity based encryption scheme of [BF03] as an example. For this scheme, the secret decryption key is one argument of the pairing, while the other argument is a part of the ciphertext.

For pairing based schemes, their implementation usually offers some degree of freedom with respect to the arguments of the pairing. For example the secret key can be chosen as either the first or the second argument.

Another choice is the selection of the first argument $\mathbb{G}_1$ of the pairing. The restriction of $\mathbb{G}_1 = E(\mathbb{F}_q)[l]$ allows a more efficient implementation of the scheme. The reason is that field operations that are based on the first argument are limited to the smaller base field $\mathbb{F}_q$. Naturally, this is a common optimization often proposed in the literature [BKLS02]. In the following, we focus on implementations that make use of this optimization. To ensure non-degeneracy for embedding degree $k > 1$ the restriction of the first argument to $E(\mathbb{F}_q)[l]$ requires that the second argument $\mathbb{G}_2$ is selected such that $\mathbb{G}_2 \not\subseteq E(\mathbb{F}_q)[l]$ [BSS05, Lemma IX.8].

The size of the embedding degree k is another factor that influences the effectiveness of pairing based cryptographic schemes. For the pairing to be efficiently computable the embedding degree needs to be reasonable small. On the other hand, it must not be too small. The rationale behind this is that the pairing reduces the hardness of the discrete logarithm problem (DLOG) in $E(\mathbb{F}_{q^k})[l]$ to the hardness of DLOG in an order l subgroup of $\mathbb{F}_{q^k}$ [MOV93]. Embedding degrees of $k = 4, \ldots, 16$ result in the most efficient implementations for nowadays security parameters [FST10].

Another technique to improve the performance of pairing based cryptography is to use a twist E' of E [JN09]. It allows to transfer some operations from $E(\mathbb{F}_{q^k})$ to $\mathbb{G}_2' \subset E'(\mathbb{F}_{q^d})$ with $d < k$. For the computation of the pairing, elements from $\mathbb{G}_2'$ are then mapped back to $E(\mathbb{F}_{q^k})$.

2.3.3 Implementation of the Tate Pairing for General Characteristics

In this section we recall how the Tate pairing from (2) can be computed efficiently for finite fields $\mathbb{F}_q$. As motivated above, we will restrict ourselves to the case where $P \in E(\mathbb{F}_q)[l]$ instead of considering the general case $P \in E(\mathbb{F}_{q^k})[l]$. In this specific situation it is sufficient to evaluate $f_{l,P}$ at point Q instead of divisor D_Q. The (reduced) Tate pairing is then defined as in [BLS03]:

$$\mathrm{e}(P,Q) = f_{l,P}(D_Q)^{(q^k-1)/l} = f_{l,P}(Q)^{(q^k-1)/l} \text{ for } P \in E(\mathbb{F}_q)[l].$$

In general, there is no closed form for $f_{l,P}$ that can be efficiently computed. But V. Miller introduced an algorithm that iteratively constructs the result of $f_{l,P}$ at Q [Mil04]. For constant embedding degree it has polynomial running time in q. For the details of the algorithm see Algorithm 1.

Algorithm 1 Miller Algorithm for evaluating a function $f_{l,P}$ with divisor $(f_P) = l\,(P) - l\,(\mathcal{O})$ at Q. The function $l_{U,V}(Q)$ is defined in (4).

Require: $P \in E[l](\mathbb{F}_{q^k}), Q = (x_Q, y_Q) \in E(\mathbb{F}_{q^k})$, binary representation $l = (l_{t-1} \dots l_0)$
Ensure: $f_{l,P}(Q)$ where $(f_{l,P}) = l\,(P) - l\,(\mathcal{O})$
1: **procedure** $f_{l,P}(Q)$
2: $\quad f \leftarrow 1, R \leftarrow P$
3: $\quad$ **for** $j \leftarrow 1, \dots, t-1$ **do**
4: $\quad\quad f \leftarrow f^2 \cdot g_{R,R}(x_Q, y_Q)/g_{2R,-2R}(x_Q, y_Q)$
5: $\quad\quad R \leftarrow 2R$
6: $\quad\quad$ **if** $l_{t-1-j} = 1$ **then**
7: $\quad\quad\quad f \leftarrow f \cdot g_{R,P}(x_Q, y_Q)/g_{R+P,-(R+P)}(x_Q, y_Q)$
8: $\quad\quad\quad R \leftarrow R + P$
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: $\quad$ **return** f
12: **end procedure**

During the execution of the Miller algorithm, the value of $f_{l,P}(Q)$ is iteratively constructed (see Line 4 and Line 7 of Algorithm 1). Therefore, we define the value of R in Line 4 of Algorithm 1 in iteration j as $R_j = L_j P$ with

$$L_j = \sum_{i=t-j}^{t-1} l_i 2^{i-(t-j)}. \tag{3}$$

That is, L_j represents the j most significant bits (MSBs) of l. In iteration j of Miller's algorithm, intermediate functions f_{l,R_j} with divisor $l\,(R_j) - l\,(\mathcal{O})$ are evaluated at Q. To do this, functions of the form

$$g_{U,V}(x,y) = y - y_U + \lambda_{U,V}(x_U - x). \tag{4}$$

with $U, V \in E(\mathbb{F}_{q^k})$ are computed. The parameter $\lambda_{U,V} \in \mathbb{F}_{q^k}$ is defined as in the addition of $V = (x_V, y_V)$ and $U = (x_U, y_U)$:

$$\lambda_{U,V} = \begin{cases} \frac{y_V - y_U}{x_V - x_U} & x_U \neq x_V \\ \frac{3x_U^2 + 2a_2 x_U + a_4 - a_1 y_U}{2y_U + a_1 x_U + a_3} & x_U = x_V. \end{cases} \tag{5}$$

The SCA we will introduce in Section 3 is based on the computation of these functions.

Further note that since l is a prime, we can recover P efficiently from R_j:

$$P = \left(L_j^{-1} \mod l\right) R_j. \tag{6}$$

2.3.4 The Eta Pairing for Characteristic 2

The eta pairing was defined in [Pau07] for different types of algebraic curves. In [Ste07] the authors introduced an efficient method and an algorithm to compute this pairing for supersingular elliptic curves.

Algorithm 2 Algorithm to compute the eta pairing without final exponentiation

Require: $P = (x_P, y_P), Q = (x_Q, y_Q) \in E(\mathbb{F}_{2^m})[l]$
Ensure: $\eta(P, Q)$

```
g ← 1, v ← 1, T ← P
for j ← m − 1 to 0 do
    λ_j ← x_T^2 + 1
    g_j ← (y_Q + y_T + λ_j(x_Q + x_T + 1), λ_j + x_Q + 1, λ_j + x_Q, 0)
    g ← g^2 · g_j
    T ← 2T
    v_j ← (x_Q + x_T + 1, 1, 1, 0)
    v ← v^2 · v_j
end for
return g/v
```

C. Whelan and M. Scott [WS07] presented a slightly modified version of this algorithm for characteristic 2, which is given in Algorithm 2 (we refer to [Lis11] for detailed description and the proof of correctness of this algorithm). In Section 5 we will consider and generalize the fault attacks against Algorithm 2 presented in [WS07].

2.4 Non-Invasive SCA

In this section we define our model of a passive side channel and we give a short introduction into DPA.

2.4.1 Our Model of a Side Channel

Let $\mathcal{A}_s$ be a cryptographic, possibly randomized algorithm that is parametrized by a secret s. We assume that the implementation of the algorithm is known to an attacker. Further we assume that the attacker can trigger the execution of $\mathcal{A}_s$ and that he has physical access to the device that executes $\mathcal{A}_s$.

A side channel is a measure that depends on the data that are processed by an algorithm. Examples for such a measure are execution time, power consumption, or electromagnetic radiance. We regard a side channel as a combination of an intermediate state $z_{s,\xi}$ and a leakage oracle. Here, $z_{s,\xi}$ represents an internal state of $\mathcal{A}_s$ that depends on the input ξ and the secret s. We only consider the case where the internal state $z_{s,\xi}$ is not randomized by $\mathcal{A}_s$. Hence, it is a function of ξ and s denoted by $z_{s,\xi} = h(s, \xi)$.

In presence of a side channel an attacker has the possibility to query a leakage oracle. In our model the leakage consists of two parts. One part is linearly related to the Hamming weight of its input (see Hamming-Weight Model in [MOP07]). With $\mathrm{HW}(x)$ we denote the Hamming weight of x. The second part N is an additive Gaussian random variable with mean μ and variance σ^2. This part represents activity of the device that is independent of s and ξ. On input ξ, a query to the leakage oracle will return the value $o_{s,\xi} = \alpha\,\mathrm{HW}(z_{s,\xi}) + n$. The factor α is a constant that covers gain and polarity of the channel. Here n is distributed according to N. This model based on the Hamming weight might over-simplify the reality of, e.g., CMOS devices [MOP07]. But we merely use it to obtain an indication for the feasibility of our attack.

2.4.2 Differential Power Analysis

Assume the attacker made q queries to the leakage oracle with inputs $(\xi_1, \dots, \xi_q)$ to obtain the sequence $\left(o_{s,\xi_1}, \dots, o_{s,\xi_q}\right)$ of samples. The task of the attacker is then to use this sequence to recover s.

A DPA exploits that a variation of the input of the cryptographic algorithm $\mathcal{A}_s$ causes a variation of the distribution of $o_{s,\xi}$. If something about the interdependence between the input and the measurements is known, this may help to learn s. To identify s, a DPA that is based on the previously defined model proceeds in two steps. First, from the knowledge about the algorithm the internal states $\left(z_{s',\xi_1}, \dots, z_{s',\xi_q}\right) = (h(s', \xi_1), \dots, h(s', \xi_q))$ are predicted for the queries $\xi_1, \dots, \xi_q$ and for possible candidates $s' \in S$. Then the correlation between $X = \left(\mathrm{HW}(z_{s',\xi_1}), \dots, \mathrm{HW}(z_{s',\xi_q})\right)$ and $Y = \left(o_{s,\xi_1}, \dots, o_{s,\xi_q}\right)$ is determined. Finally, the hypothesis s' with the highest correlation is selected as the guess for the secret.

For the leakage function, there is a linear relationship between the Hamming weight of the intermediate state $z_{s,\xi}$ and $o_{s,\xi}$. To detect such a linear dependency, for example Pearson's correlation coefficient can be applied to X and Y [MOP07]:

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}}. \tag{7}$$

Usually, the set of secrets S is of exponential size with respect to a security parameter n. Hence, it is computationally not feasible to calculate the correlation of the sampled values with each hypothesis s' in S. But the attacker can proceed iteratively. The set of S is decomposed into polynomially (in n) many appropriate subsets. For the decomposition to be meaningful, the correlation caused by two elements from the same subsets should be close while the correlation should be different for elements from distinct subsets. Then in every iteration, the correlation with a representative of each subset is calculated. The subset of the representative with the highest correlation is selected as input for the next iteration. This is continued in a tree based manner until only a subset with one element remains. We will present an approach that follow this strategy in Section 4. There, the subsets will be numbers in a certain interval.

3 Attacking the Tate Pairing with Secret P

In this section we consider $\mathcal{A}_s$ as a cryptographic algorithm with the (reduced) Tate pairing $\mathrm{e}(P, Q)$ as one of its components. We consider fields of characteristic 2 or of large prime characteristic p. The case where the second argument Q is the secret was handled in [WS06]. There, a DPA on the modular multiplication was used to recover the secret point Q. They conjectured that it may be more difficult to attack the pairing if P is the secret point. In the following, we will show an attack for the case where the first argument P is the secret while the argument Q is part of the input of $\mathcal{A}_s$. As justified in Section 2.3, we assume $P \in E(\mathbb{F}_q)$ and $Q \in E\left(\mathbb{F}_{q^k}\right)$, $Q \notin E(\mathbb{F}_q) \setminus \{\mathcal{O}\}$. Further assume that the Tate pairing is implemented on the basis of the Miller algorithm from Algorithm 1.

3.1 Attack Based on the Modular Multiplication

In [WS06], the modular multiplication of elements in $\mathbb{F}_q$ was exploited to leak information about the secret. There it was shown how s can be recovered from the repeated computation of $z_{s,\xi} = h(s, \xi) = h_1(s, \xi) \cdot h_2(s, \xi)$ with different values for ξ. The DPA presented in [WS06] recovers s iteratively in portions of w bits from the least significant bit (LSB) of s to the MSB of s. In this process, different hypotheses about a fraction of w bits of s are made. Under these hypotheses, a part of the value of $z_{s,\xi} = h_1(s, \xi) \cdot h_2(s, \xi)$ is predicted. Then the hypothesis that yields the highest correlation with the measurements according to (7) is identified. Finally, the corresponding fraction of s is fixed according to this hypothesis. As a requirement for the DPA, a fraction of $z_{s,\xi}$ needs to be predictable under an hypothesis for s and a given input ξ. Take $h_1(s, \xi) = s + \xi$ and $h_2(s, \xi) = s$ as an example. Given ξ and a hypothesis for the w LSBs of s we can predict the w LSBs of $z_{s,\xi} = (s + \xi)s$. This is because the w LSBs only depend on the w LSBs of s and ξ. For more details concerning the DPA see [WS06].

In the following we want to show how to recover the secret point P based on this DPA. To achieve this, we will use the fact that P is already defined over $\mathbb{F}_q$. Our attack is in two steps. First we give definitions for $h_1(P,Q)$ and $h_2(P,Q)$ that fulfill the requirements for the DPA. In our case the DPA will not give us the secret point P directly. Instead it will result in an s that is related to P. Thus, in a second step we show how P can be recovered from s.

3.1.1 Defining $h_1(P,Q)$ and $h_2(P,Q)$

As in [WS06] we will use the computation of the function $g_{U,V}(x,y)$ from (4) as the target of our attack. Recall the outline of Miller's algorithm from Section 2.3. With the notation introduced there, we write the value of the point R in iteration j as $R_j = (x_j, y_j)$. The function $g_{U,V}(x,y)$ is evaluated with $U = V = R_j$, $x = x_Q$, and $y = y_Q$. Note that R_j depends only on the secret P. With $\lambda_j = \lambda_{R_j,R_j}$ we denote the function λ from (5) that is calculated during the doubling of R_j. Inserting R_j, x_Q, and y_Q into (4) we get

$$g_{R_j,R_j}(x_Q,y_Q) = y_Q - y_j + \lambda_j(x_j - x_Q). \tag{8}$$

Notice that λ_j is multiplied with $(x_j - x_Q)$. Since we assumed $P \in E(\mathbb{F}_q)$ it follows that $x_j \in \mathbb{F}_q$ and we can write x_j as $x_j = \left({x_j}^{(1)}, 0, \ldots, 0\right)$. We further assume that no twist is used or that the embedding degree k is larger than 2. Then $Q \notin E(\mathbb{F}_q) \setminus \{\mathcal{O}\}$ implies $x_Q \notin \mathbb{F}_q$ [JN09]. Hence, x_Q is of the form $x_Q = \left({x_Q}^{(1)}, {x_Q}^{(2)}, \ldots, {x_Q}^{(k)}\right)$ and there exists an $i \geq 2$ such that ${x_Q}^{(i)} \neq 0$. Therefore, we get $x_j - x_Q = \left({x_j}^{(1)} - {x_Q}^{(1)}, -{x_Q}^{(2)}, \ldots, -{x_Q}^{(k)}\right)$. We set $h_1(s,\xi) = h_1(P) = \lambda_j$ and $h_2(s,\xi) = h_2(Q) = -{x_Q}^{(i)}$ for $-{x_Q}^{(i)} \neq 0$. Hence, h_1 only depends on P while h_2 only depends on Q. The latter is under our control.

We assumed $P \in E(\mathbb{F}_q)$ and $x_Q \notin \mathbb{F}_q$. This restriction allowed us to define an intermediate state $z_{s,\xi} = h_1(s,\xi)h_2(s,\xi)$, where $h_2(s,\xi)$ is independent of the secret and under our control. Therefore, we can apply the DPA on the modular multiplication to recover λ_j. The described restriction was not made in [WS06] and hence they were not able to mount their attack on schemes where P is secret.

3.1.2 Recovery of P from λ_j

According to (5), λ_j is a function of R_j. For given R_j this function is not invertible. But by solving (5) for y_{R_j} and combining the result with (1) we get a polynomial in x_{R_j} over $\mathbb{F}_q$ of degree 4. If λ_j was correctly recovered, then this polynomial has at least one root in $\mathbb{F}_q$. All roots can be determined in expected polynomial time in $\log q$, for example with the algorithm of Cantor-Zassenhaus. Since there are at most 4 roots in $\mathbb{F}_q$, we get at most 4 possible solutions for the x-coordinate of R_j. Inserting x_j into (5) gives one unique y-coordinate for each x_j. So we end up with at most 4 possible solutions for R_j. If j is known, these solutions directly translate to at most 4 solutions for P by applying (6). Now, P can be found by parametrizing $\mathcal{A}$ with all candidates. For example in an encryption scheme, the ciphertext of an arbitrary message could be decrypted with $\mathcal{A}_P$. Then the $\mathcal{A}_P$ that delivers the original message was parametrized by the correct key.

To summarize, we made use of a reasonable restriction of the arguments of the pairing in the sense of efficient implementations. This enabled us to extend the DPA presented in [WS06] also to the first argument.

3.2 Attack Based on the Modular Addition

In this section we handle the case where the SCA is based on the modular addition. It is a common assumption that the modular addition cannot be attacked with a DPA. However in Section 4, we will outline an approach for a DPA on the modular addition that might refute this assumption. Given a DPA on the modular addition and with Section 3.1, the strategy is quite obvious. For the DPA on the modular addition to be possible, we require fields $\mathbb{F}_q$ with $q = p$ prime and we assume that $\mathbb{F}_p$ is represented by $\mathbb{Z}_p$.

As before we need an intermediate state of the form $z_{s,\xi} = h(s, \xi) = h_1(s, \xi) + h_1(s, \xi) \mod p$. But this time the DPA proceeds from MSB to LSB in fractions of w bits.

The modular addition of x_j and $-x_Q$ is part of (8). We set $h_1(s, \xi) = s = x_j$ and $h_2(s, \xi) = -{x_Q}^{(1)}$. If we once again assume that $x_j \in \mathbb{F}_p$, this choice results in $h_1(s, \xi) + h_2(s, \xi) \mod p = x_j + \left(-{x_Q}^{(1)}\right) \mod p$. Using the hypothesis about the MSBs of x_j and by controlling Q we are also able to predict parts of $z_{s,\xi} = x_j - {x_Q}^{(i)} \mod p$. Hence, we can apply a DPA to learn $s = x_j$.

Actually, it is not required that $x_j \in \mathbb{F}_p$ when we attack the modular addition. It is only required that each component ${x_j}^{(i)}$ of x_j is added to the corresponding component ${x_Q}^{(i)}$ of x_Q. Then the DPA can be applied independently to each component for a complete recovery of x_j.

Knowing the x-coordinate of x_j the y-coordinate can be recovered by solving the Weierstrass equation in (1). Because the equation has degree two in y, this results in two roots y_j and y'_j and leaves us with two possible points R_j and R'_j. Based on (6) we are able to determine P and P'. As before, the correct solution can be identified by parametrization of the algorithm with both candidates.

3.3 Countermeasures

In this section we describe a heuristic countermeasure against the attack on the Tate pairing. In [PV06] and [WS06], point blinding of P or Q by a random point T has been proposed as protection against DPA as well as against fault attacks. By bilinearity of the pairing we get

$$\begin{array}{rcl} \mathrm{e}(P+T, Q) & = & \mathrm{e}(P, Q)\,\mathrm{e}(T, Q) \\ \mathrm{e}(P, Q+T) & = & \mathrm{e}(P, Q)\,\mathrm{e}(P, T). \end{array} \tag{9}$$

Division by $\mathrm{e}(T, Q)$ or $\mathrm{e}(P, T)$ in a second step will cancel the effect of the blinding. Both approaches will circumvent our DPA of the modular multiplication as well as the DPA of the modular addition. The reason is that the randomization will inhibit the prediction of the internal state $z_{s,\xi}$.

The implementation of (9) is quite expensive because an additional pairing as well as an inversion in $\mathbb{F}_{q^k}$ is required. In [PV06], it has been proposed to choose T once at device initialization. Then β is determined as $\beta = \mathrm{e}(P,T)^{-1}$. At every invocation of the pairing b is chosen uniformly at random from $\mathbb{Z}_2$. The elements T and β are updated according to $T \leftarrow (-1)^b 2T$ and $\beta \leftarrow \beta^{(-1)^b 2}$. The result of the pairing is $\mathrm{e}(P,Q) = \mathrm{e}(P,Q+T)\beta$.

We propose a slight modification. To do so, notice that the domain of the second argument of the pairing is the equivalence class $E(\mathbb{F}_{q^k})/lE(\mathbb{F}_{q^k})$ (see (2)). If we now choose a random point T initially from $E(\mathbb{F}_{q^k})$ with order r relatively prime to l, then $T+Q \sim Q$. Hence $\mathrm{e}(P,Q+T) = \mathrm{e}(P,Q)$. Although only a moderate improvement in efficiency, this saves the storage of element β.

4 DPA of Modular Addition

Often it is implicitly assumed that it is difficult to perform a DPA on the modular addition. In this section we will present our approach to refute this statement. In Section 4.1 we describe and theoretically justify why the proposed DPA should work in practice. Because we did not yet have the possibility to test its real-life applicability we will give some simulation results in Section 4.2.

4.1 Description and Analysis of the DPA

Let $s, \xi \in \mathbb{Z}_p$ and consider the operation $z_{s,\xi} = s + \xi \mod p$. With the model from Section 2.4 our DPA will try to determine s based on q measurements of the form $o_{s,\xi} = \alpha\,\mathrm{HW}(z_{s,\xi}) + n$ for different choices of ξ. We follow the strategy of Section 2.4 of a correlation analysis according to (7).

The main observation is that the modular reduction helps us to enforce a *linear* decrease in the correlation $\rho_{s,s'}$ of a hypotheses s' with increasing distance to s. This will allow us to distinguish a hypothesis that is close to s from a hypothesis that is more distant. Therefore we can iteratively reduce the search space for s from $\mathbb{Z}_p$ to $\{s\}$. To understand this effect, take ξ such that $s + \xi < p$ but $s' + \xi \geq p$ for hypothesis s'. Different from the first sum, a modular reduction is performed for the second sum: $s' + \xi \mod p = s' + \xi - p$. Hence, even if $s + \xi$ and $s' + \xi$ have a small Hamming distance, the subtraction of p might cause a large Hamming distance of $s + \xi \mod p$ and $s' + \xi \mod p$. To describe this more formal we define:

$$\begin{array}{rcl} C_{s'} & = & \{\xi \in \mathbb{Z}_p | (s+\xi < p) \wedge (s'+\xi < p)\} \\ & \cup & \{\xi \in \mathbb{Z}_p | (s+\xi \geq p) \wedge (s'+\xi \geq p)\}\,. \end{array}$$

Our analysis is based on the following assumption:

Assumption 1. *We assume that p is such that the elements in $\mathbb{Z}_p \backslash C_{s'}$ do not contribute to the correlation of s and s'.*

Let $\delta_{s,s'}$ be the probability that $\xi \in C_{s'}$. Further let $\tilde{\rho}_{s,s'}$ be the correlation caused by elements from $C_{s'}$. Under Assumption 1 we can express the correlation of s' with s as $\rho_{s,s'} = \delta_{s,s'}\tilde{\rho}_{s,s'}$. We will show that we can select the inputs to the leakage oracle and the hypotheses such that this correlation significantly decreases with increasing distance to s.

We will now explain how we control the correlation $\rho_{s,s'} = \delta_{s,s'}\tilde{\rho}_{s,s'}$. The first factor $\delta_{s,s'}$ is the important part that causes the linear dependency of $\rho_{s,s'}$ on $|s' - s|$. We control it by restricting the domain of ξ. For the second factor we choose the hypotheses such that $\tilde{\rho}_{s,s'}$ is more ore less constant.

To produce a strong linear dependency of $\delta_{s,s'}$ on $|s' - s|$ we have to restrict the interval from which we choose ξ. Assume in iteration i of the DPA we know that s is within the interval $[a, b-1]$. Then we select the input to the leakage oracle in iteration i from the interval $\Xi_i = [p-b, p-a-1]$ uniformly at random. Hence, the probability that $\xi \in C_{s'}$ is $\delta_{s'} = 1 - |s - s'|/(b-a)$.

Next we will show an appropriate selection of the hypotheses that will keep the part $\tilde{\rho}_{s,s'}$ constant. Assume the search space $[a, b-1]$ in iteration i is of size n bits. Further assume that we limit the number of hypotheses to 2^w. Then we can cover the whole search space by selecting the j-th hypothesis as $s'_j = a + (j + 1/2)2^{n-w}$ for $0 \le j < 2^w$. We can interpret s'_j as the hypothesis that s is within the interval $[a + j2^{n-w}, a + (j+1)2^{n-w} - 1]$ that is centered around s'_j.

We will now estimate the correlation $\tilde{\rho}_{s,s'_j}$ between s and s'_j caused by elements $\xi \in \Xi_i \cap C_{s'_j}$. The correlation will be split into the part caused by the $n - w$ LSBs and into the part caused by the w MSBs.

The special form of $s'_j = a + (j + 1/2)2^{n-w}$ ensures that the $n - w$ LSBs of s'_j are determined by a. With $\xi \in \Xi_i \cap C_{s'_j}$ we get $(s'_j + \xi \mod p) - (s + \xi \mod p) = s'_j - s$. Hence, the correlation in the $n - w$ LSBs is independent of j. We will define this constant part of the correlation as $c(n-w)/n$ with $0 \le c \le 1$.

Next we will look at the correlation caused by the w MSBs. First assume s'_j is the closest hypothesis to s. It follows that $\left|s'_j - s\right| < 2^{n-w-1}$. Therefore a difference in the w MSBs of $(s'_j + \xi \mod p)$ and $(s_j + \xi \mod p)$ can only be caused by a carry at bit $n - w$. Over the random choices of ξ, this carry will only effect a minor decrease in the correlation of the w MSBs and we ignore this. Hence, the correlation of the closest hypothesis is assumed to be 1 in the w MSBs.

For a hypothesis s'_k that is not closest to s we get $|s'_k - s| \ge \left|s'_k - s'_j\right| - \left|s'_j - s\right| \ge 2^{n-w-1}$. Then the correlation in the w MSBs will be smaller because the sum of $s + \xi \mod p$ and $s'_k + \xi \mod p$ differ by more than 2^{n-w-1}. We assume the worst case for a wrong hypothesis and estimate the correlation in the w MSB also as 1.

Combining our estimations for the $n - w$ LSBs and the w MSBs we obtain

$$\tilde{\rho}_{s,s'} = 1\frac{w}{n} + c\frac{(n-w)}{n} \text{ with } 0 \le c \le 1.$$

For the overall correlation this results in

$$\rho_{s,s'_j} = \delta_{s'}\tilde{\rho}_{s,s'_j} = \left(1 - \frac{|s - s'|}{b-a}\right)\frac{w + c(n-w)}{n} \text{ with } 0 \le c \le 1. \tag{10}$$

This shows the linear decrease of the correlation ρ_{s,s'_j} with increasing distance $|s - s'|$. The second part $\tilde{\rho}_{s,s'_j}$ is independent of $|s - s'|$ and approaches 1 for $w \to n$. It defines the slope of the decrease of the correlation. In the worst case, for $c = 0$ we get $\tilde{\rho}_{s,s'} = w/n$. This shows that increasing w will increase the robustness of the DPA. However, the number of hypotheses is exponential in w. Hence, we can not increase w arbitrarily.

Equation (10) implies our strategy how to reduce the search space in an iteration from $[a, b-1]$ to $[a', b'-1]$: Among all 2^w hypotheses, we select the hypothesis s' with the highest correlation $\rho_{s,s'}$. Then $[a', b'-1]$ with $(b'-a') < (b-a)$ will be centered around s'. Hence, another parameter that influences the robustness is the ratio by which we reduce the search space from iteration to iteration. Let $[a', b'-1]$ be the outcome of iteration i. The DPA will fail if $s \notin [a', b'-1]$. On the one hand, increasing the ratio $(b'-a')/(a-b)$ will also increase the robustness. On the other hand, this will increase the number of iterations and thus the number of required queries.

4.2 Simulation Results

So far, our analysis did not consider noise that will influence the number of required queries to the leakage oracle. For our assumption of independent additive Gaussian noise with variance σ^2 the correlation $\tilde{\rho}_{s,s'}$ is scaled by a constant factor of $1/\sqrt{1 + \sigma^2/(n/4)}$ [MOP07]. Thus, the noise will not invalidate the strategy. Rather the number of required queries has to be increased to maintain the confidentiality of the correct hypothesis.

We plan to analyze the feasibility of the proposed DPA also in practice. But so far, we can only provide simulation results. As one example of practical relevance, we take an elliptic curve defined over $\mathbb{F}_p$ with $n = \operatorname{ld} p \approx 128$ bit. For the number of hypotheses we choose 2^w with $w = 16$. We perform our simulation with $\sigma^2 = 400$ and $q = 10000$ queries per iteration. We select the search space reduction ratio as $(b'-a')/(b-a) = 2^{-8}$. Hence, at iteration i the search space $[a, b-1]$ is of size $a - b = 2^{n-i8}$. Figure 1 shows the correlation $\rho_{s,s'}$ in iteration $i = 2$ ($w/n = 2/15$) without and with noise. We can see that our estimation predicts the simulation results quite well, also in the case of noise.

These results let us believe that the DPA is actually possible. We hope that we will be able to reproduce them also in practice.

5 Fault Attack on Pairing Based Cryptography

Until now we considered only passive attacks, where the adversary uses the information observed from some side channel. In this section we will look at fault attacks that are a very powerful type of active side channel attacks. By corrupting the data the algorithm works on or by interfering with the algorithm execution the adversary produces corrupted outputs and uses these to recover the secret key. In the context of PBC only few successful fault attacks are known [PV06, WS07].

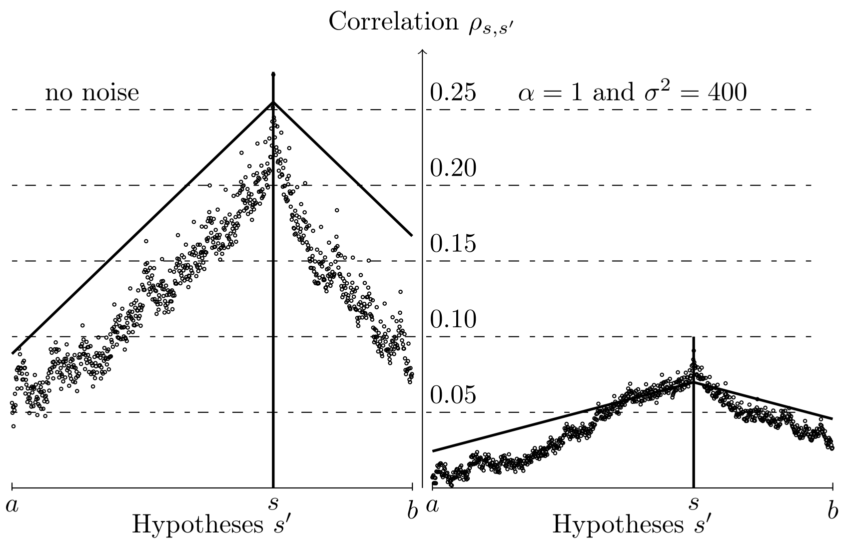

Figure 1: Correlation for hypotheses s' with $w = 16$ bit, $n = 120$ bit, and $q = 10000$ queries. The vertical lines mark the secret s. The solid lines represent our estimation according to (10) with $c = 1.12 \cdot 10^{-3}$. In the presence of noise, the correlation is down-scaled by a factor of $\sqrt{1 + \sigma^2/(n/4)} \approx 3.65$.

In [WS07] the authors considered fault attacks against several pairing algorithms. In particular, fault attacks on their algorithm for the eta pairing were presented. This algorithm (see Algorithm 2 for details) computes the eta pairing iteratively as

$$\eta\,(P, Q) = \prod_{i=0}^{m-1} \left(\frac{g_{m-1-i}}{v_{m-1-i}} \right)^{2^{m-1-i}} . \tag{11}$$

In the i-th iteration the intermediate result is squared and then multiplied by a factor g_j/v_j (for $j = m - 1 - i$) with

$$\begin{aligned} g_j &= (y_Q + y_T + \lambda_j(x_Q + x_T + 1), \lambda_j + x_Q + 1, \lambda_j + x_Q, 0) \\ v_j &= (x_Q + x_T + 1, 1, 1, 0)\,. \end{aligned} \tag{12}$$

We are especially interested in two most promising attacks from [WS07]. In the first attack the fault is induced into the value v_j of (11). In the second attack the value g_j is corrupted. Both attacks were used for the iteration with $j = 0$ to recover the second argument of the pairing. However, if the secret is used as the first argument, the authors of [WS07] did not present how to recover it for the second attack.

We extend the fault attacks of [WS07] in two directions. On the one hand, we show that the restriction to the iteration with $j = 0$ is not necessary. On the other hand, we show that for the second attack the secret can be recovered independently of which argument is the secret. We do not need to modify the assumptions about the faults involved. We still assume the most realistic and general fault type — a random fault in certain value, stored in a memory cell.

The general idea behind the fault attacks of [WS07] is based on the fact that a fault in g_j or v_j for some j only effects the factor g_j/v_j. A corrupted result $\eta(P,Q)'$ divided by the correct pairing value yields then a relatively simple equation over $\mathbb{F}_{2^{4m}}$:

$$\frac{\eta(P,Q)'}{\eta(P,Q)} = \left(\frac{g_j'}{g_j}\right)^{2^j}, \tag{13}$$

since all other factors are canceled (similarly for v_j).

We consider at first the loop iteration with $j = 0$ and a fault in the constant factor $g_0^{(1)}$ of g_0. Thus we get the following equation from (13):

$$N := \frac{\eta(P,Q)'}{\eta(P,Q)} = \frac{g_0'}{g_0} \tag{14}$$

with $g_0' = \left({g_0'}^{(1)}, {g_0}^{(2)}, {g_0}^{(3)}, {g_0}^{(4)}\right)$. From (12) we have $g_j^{(4)} = 0$ and $g_j^{(3)} = g_j^{(2)} + 1$. Thus we achieve the following system of equations over $\mathbb{F}_{2^m}$ using knowledge of how multiplication in $\mathbb{F}_{2^{4m}}$ is performed:

$$\begin{cases} {g_0'}^{(1)} &= N^{(1)}g_0^{(1)} + \left(N^{(3)} + N^{(4)}\right)g_0^{(2)} + N^{(3)} \\ {g_0}^{(2)} &= N^{(2)}g_0^{(1)} + \left(N^{(1)} + N^{(3)}\right)g_0^{(2)} + N^{(3)} + N^{(4)} \\ {g_0}^{(2)} + 1 &= N^{(3)}g_0^{(1)} + \left(N^{(1)} + N^{(2)} + N^{(4)}\right)g_0^{(2)} + N^{(1)} + N^{(4)} \\ 0 &= N^{(4)}g_0^{(1)} + \left(N^{(2)} + N^{(3)}\right)g_0^{(2)} + N^{(2)} \end{cases},$$

where every equation is associated with one of the four components of g_0'. The secret point yields a possible solution of this system of equations.

Since we do not know ${g_0'}^{(1)}$, we ignore the first equation and get three equations with two unknowns $g_0^{(1)}$ and $g_0^{(2)}$. When analyzing (14) in the form $N = g_0' \cdot g_0^{-1}$, one can prove that the random value ${g_0'}^{(1)}$ effects all components of N. Furthermore over the random choice of ${g_0'}^{(1)}$ the components of N are not equal to zero and the given system of equations has exactly one solution with high probability (for more details see [Lis11]). Thus we can use for example the third and the fourth equation in order to eliminate $g_0^{(1)}$:

$$g_0^{(2)} = \frac{N^{(2)}N^{(3)} + N^{(4)} + N^{(1)}N^{(4)} + N^{(4)}N^{(4)}}{N^{(3)}N^{(3)} + N^{(2)}N^{(4)} + N^{(2)}N^{(3)} + N^{(4)} + N^{(1)}N^{(4)} + N^{(4)}N^{(4)}}.$$

Then we compute $g_0^{(1)}$ from the fourth equation.

$$g_0^{(1)} = \left(\left(N^{(2)} + N^{(3)}\right)g_0^{(2)} + N^{(2)}\right)/N^{(4)}.$$

Recovery of P and Q from $g_0^{(1)}$ and $g_0^{(2)}$ From (12) we have $g_0^{(2)} = x_T^2 + x_Q$ and $g_0^{(1)} = y_Q + y_T + \left(x_T^2 + 1\right)(x_Q + x_T + 1)$. Point T can be easily expressed in coordinates of P [Ste07]. For the iteration with $j = 0$ we have $x_T = x_P^{2^{m-2}}$ and $y_T = y_P^{2^{m-2}} + \tau(m-1)$ with $\tau(i) = 0$ if $i \equiv 0, 1 \pmod 4$ and $\tau(i) = 1$ else. Hence, we can derive:

- For the secret point P:

$$x_P = \left(g_0^{(2)} + x_Q\right)^2 \text{ and}$$

$$y_P = \left(g_0^{(1)} + y_Q + \left(x_P^{2^{m-1}} + 1\right)\left(x_Q + x_P^{2^{m-2}} + 1\right) + \tau(m-1)\right)^4.$$

- For the secret point Q:

$$x_Q = g_0^{(2)} + x_P^{2^{m-1}} \text{ and}$$

$$y_Q = g_0^{(1)} + y_P^{(m-2)} + \tau(m-1) + \left(x_P^{2^{m-1}} + 1\right)\left(x_Q + x_P^{2^{m-2}} + 1\right).$$

Thus we can completely recover the secret point in both cases when corrupting the constant component of g_0 in the iteration with $j = 0$ (see Section 4.3.3 in [Lis11] for more details and a numerical example).

Attacking an Arbitrary Iteration Next we show, that the restriction of this fault attack to the iteration with $j = 0$ is not necessary. When we go back to (13) we can derive the following equation for every j:

$$N = \left(\frac{\eta(P,Q)'}{\eta(P,Q)}\right)^{2^{4m-j}} = \frac{g_j'}{g_j}.$$

This equation is similar to (14), with the sole exception of equations for x_T and y_T, which depend on j. The equation for the fault in v_j is similar. For more details see Section 4.3.4 in [Lis11].

6 Open Problems and Conclusion

For the most efficient implementations of the Tate pairing the first argument is defined over the base field and the second argument is defined over the extension field [BKLS02]. Extending the results of [WS06] we showed that it is in principle possible to attack the pairing no matter whether the secret is the first or the second argument of the pairing.

Several countermeasures like point blinding have been proposed to protect the pairing against SCAs [WS06, PV06, WS07]. They are all heuristic in the sense that they prevent a special attack and that their effectiveness is not rigorously proven. Hence, an important field of research is to find sound models that allow provable secure countermeasures that are efficient enough for the implementation on constraint devices like smart cards.

References

[BF03] Dan Boneh and Matthew Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[BKLS02] Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott. Efficient Algorithms for Pairing-Based Cryptosystems. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368. Springer, 2002.

[BLS03] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. On the Selection of Pairing-Friendly Groups. In Mitsuru Matsui and Robert J. Zuccherato, editors, *Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 17–25. Springer, 2003.

[BLS04] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, 2004.

[BS04] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *ACM Conference on Computer and Communications Security*, pages 168–177, 2004.

[BSS05] Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart, editors. *Advances in Elliptic Curve Cryptography*. Number 317 in London Mathematical Society Lecture Note Series. Cambridge University Press, 2005.

[FST10] David Freeman, Michael Scott, and Edlyn Teske. A Taxonomy of Pairing-Friendly Elliptic Curves. *Journal of Cryptology*, 23:224–280, 2010. 10.1007/s00145-009-9048-z.

[GPSW06] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98. ACM, 2006.

[JN09] Marc Joye and Gregory Neven, editors. *Identity-Based Cryptography*, volume 2 of *Cryptology and Information Security*. IOS Press, 2009.

[KTH+06] Tae Kim, Tsuyoshi Takagi, Dong-Guk Han, Ho Kim, and Jongin Lim. Side Channel Attacks and Countermeasures on Pairing Based Cryptosystems over Binary Fields. In David Pointcheval, Yi Mu, and Kefei Chen, editors, *Cryptology and Network Security*, volume 4301 of *Lecture Notes in Computer Science*, pages 168–181. Springer Berlin / Heidelberg, 2006. 10.1007/11935070_11.

[Lis11] Gennadij Liske. Fault attacks in pairing-based cryptography. Master's thesis, University of Paderborn, 2011.

[MFN09] Nadia El Mrabet, Marie Lise Flottes, and Giorgio Di Natale. A practical Differential Power Analysis attack against the Miller algorithm. In *Research in Microelectronics and Electronics, 2009. PRIME 2009. Ph.D.*, pages 308–311, July 2009.

[Mil04] Victor S. Miller. The Weil Pairing, and Its Efficient Calculation. *Journal of Cryptology*, 17(4):235–261, 2004.

[MOP07] S. Mangard, E. Oswald, and T. Popp. *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer-Verlag New York Inc, 2007.

[MOV93] Alfred J. Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Trans. Inform. Theory*, 39(5):1639–1646, 1993.

[Mra09] Nadia El Mrabet. What about vulnerability to a fault attack of the miller's algorithm during an identoty based protocol? In Jong Hyuk Park, Hsiao-Hwa Chen, Mohammed Atiquzzaman, Changhoon Lee, Tai hoon Kim, and Sang-Soo Yeo, editors, *Advances in Information Security and Assurance Third International Conference and Workshops*, volume 5576 of *Lecture Notes in Computer Science*, pages 122–134. Springer, June 2009.

[Pau07] Paulo S. L. M. Barreto and Steven D. Galbraith and Colm Ó hÉigeartaigh and Michael Scott. Efficient pairing computation on supersingular abelian varieties. *Designes, Codes and Cryptography*, 42(3):239–271, 2007.

[PV04] D. Page and F. Vercauteren. Fault and Side-Channel Attacks on Pairing Based Cryptography. Cryptology ePrint Archive, Report 2004/283, 2004. `http://eprint.iacr.org/`.

[PV06] Dan Page and Frederik Vercauteren. A fault attack on pairing-based cryptography. *IEEE Transactions on Computers*, 55(9):1075–1080, 2006.

[SCA06] Michael Scott, Neil Costigan, and Wesam Abdulwahab. Implementing Cryptographic Pairings on Smartcards. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 134–147. Springer, 2006.

[Sil09] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer Science+Business Media, 2 edition, 2009.

[Ste07] Steven D. Galbraith and Colm Ó hÉigeartaigh and Caroline Sheedy. Simplified pairing computation and security implications. *Journal of Mathematical Cryptology*, 1(3):267–281, August 2007.

[WS06] C. Whelan and M. Scott. Side Channel Analysis of Practical Pairing Implementations: Which Path is More Secure? *Progress in Cryptology-VIETCRYPT 2006*, pages 99–114, 2006.

[WS07] Claire Whelan and Michael Scott. The importance of the final exponentiation in pairings when considering fault attacks. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *Pairing*, volume 4575 of *Lecture Notes in Computer Science*, pages 225–246. Springer, 2007.