Differential Power Analysis against the Miller Algorithm

Nadia El Mrabet, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre, Jean-Claude Bajard

Differential Power Analysis against the Miller Algorithm

July 24, 2008

Abstract

Pairings permit several protocol simplifications and original scheme creation, for example Identity Based Cryptography protocols. Initially, the use of pairings did not involve any secret entry, consequently, side channel attacks were not a threat for pairing based cryptography. On the contrary, in an Identity Based Cryptographic protocol, one of the two entries to the pairing is secret. Side Channel Attacks can be therefore applied to find this secret. We realize a Differential Power Analysis(DPA) against the Miller algorithm, the central step to compute the Weil, Tate and Ate pairing.

Keywords: Pairing, Miller Algorithm, Pairing Based Cryptography, SCA, DPA.

1 Introduction

Originally, the cryptographic use of pairings was in a destructive way. Pairings convert the discrete logarithm problem from an elliptic curve subgroup to the discrete logarithm problem in a finite field, this property was used for particular elliptic curves in the MOV [18] and Frey Ruck [9] attack. This pairing property allows constructing new protocols. The first constructive use of pairings was the tripartite key exchange of Joux A. [11]. It was followed by original protocols like Identity Based Cryptography (IBC), introduced by Boneh D. and Franklin M. in 2003 [3], or short signature schemes [10].

The use of pairings in IBC involves a secret entry during the pairing calculation. Several pairing implementations exist, for example [20] and [2]. Side Channel Attacks (SCA) against pairing based cryptography were first developed three years ago ([19], [21] and [12]). Page D. and Vercauteren F. suggested a Messerges style Differential Power Analysis (DPA) attack against the Duursma and Lee algorithm in affine coordinates [19]; but they did not developed it. In [21], Whelan C. and Scott M. concluded that if the secret is used as the first argument of a pairing calculation, then it is impossible to find it. We show that if P is secret during the calculation of the pairing between points P and Q (denoted $e(P,Q)$), then we can find it out.

We propose a general DPA attack against the Miller algorithm [16]. Most of the pairings calculations use the Miller algorithm as a computation central step. It is the case for the Weil [6, chap. 16], Tate [20] and Ate [17] pairings. We will show how the attack can be carried out when Jacobian coordinates are used. At our best knowledge, it is the first time that an algorithm in Jacobian coordinates is attacked and clearly depicted by means of DPA.

The paper is organised as follow, Section 2 recalls the Miller algorithm and the main pairing properties. Section 3 describes a general DPA attack, Section 4 presents the circuit that has been designed in order to perform the calculation and the attack, whereas Section 5 presents the results of the attack. Eventually, Section 6 concludes the paper and provides some perspectives for further works.

2 Pairings and the Miller algorithm

2.1 Short introduction to the pairing

Let G_1 and G_2 be two abelian groups of order l, and G_3 a multiplicative abelian group of the same order. A pairing is an application satisfying this definition:

DEFINITION 2.1:
A pairing is a function: $e : G_1 \times G_2 \rightarrow G_3$ *with the following properties :*

- *Bilinearity :*

$$\forall P, P' \in G_1, \forall Q \in G_2, e(P + P', Q) = e(P, Q).e(P', Q)$$

$$\forall P \in G_1, \forall Q, Q' \in G_2, e(P, Q + Q') = e(P, Q).e(P, Q')$$

- *Non-degeneracy :*

$$\forall P \in G_1 - \{0\}, \exists Q \in G_2, \textit{such that: } \; e(P, Q) \neq 1$$

$$\forall Q \in G_2 - \{0\}, \exists P \in G_1, \textit{such that: } \; e(P, Q) \neq 1$$

We will consider pairings defined over an elliptic curve E over a finite field $\mathbb{F}_q$, for q a prime number, or a power of a prime number different from 2 and 3. In characteristic 2 [12] and 3, the same scheme can be applied, but the equations are a little different. The equation of the elliptic curve E is $Y^2 = X^3 + aXZ^4 + bZ^6$ in Jacobian coordinates, with a and $b \in \mathbb{F}_q$. The integer a can be considered to be (-3) as Brier and Joye have shown in [4]. Let $l \in \mathbb{N}^*$, and $G_1 \subset E(\mathbb{F}_q)$, $G_2 \subset E(F_{q^k})$, $G_3 \subset \mathbb{F}^*_{q^k}$ be three groups of order l, and k be the smallest integer such that l divides $(q^k - 1)$, k is called the embedding degree.

The most useful property in pairing based cryptography is bilinearity: $e([n]P, [m]Q) = e(P, Q)^{nm}$. A. Menezes gives a very good introduction to the pairing based cryptography [15]. Among all the pairings used in cryptography,

three of them are constructed in the same way. The Miller algorithm [16] is an important step for Weil, Tate and Ate pairings computation. These pairings are described in the appendix A.

2.2 Miller algorithm

The following description of the Miller algorithm is referenced in chap. 16 of [6]. The Miller algorithm is the central step for the Weil, Tate and Ate pairings computation. It is constructed like a double and add scheme using the construction of $[l]P$ and based on the notion of divisor, here we will only give the indispensable elements for the pairing computation.

The Miller algorithm constructs the rational function F_P associated to the point P, P is a generator of $G_1 \subset E(\mathbb{F}_q)$; and at the same time, it evaluates $F_P(Q)$ for a point $Q \in G_2 \subset E(F_{q^k})$. P is a l^{th} root of the function F_P (if we use the divisor notion: $Div(F_P) = lDiv(P) - lDiv(P_\infty)$).
The algorithm is such that the construction and the evaluation of the function F_P at point Q are done simultaneously.

Figure 1: Miller algorithm

Algorithm 1: Miller(P, Q, l)

Data: $l = (l_n \dots l_0)$(radix 2 representation), $P \in G_1$ and $Q \in G_2$;
Result: $F_P(Q) \in G_3$;
$1 : T \leftarrow P$, $f_1 \leftarrow 1$, $f'_1 \leftarrow 1$,$f_2 \leftarrow 1$, $f'_2 \leftarrow 1$
for $i = n - 1$ **to** 0 **do**
  $2 : T \leftarrow [2]T$
  $3 : f_1 \longleftarrow {f_1}^2 \times h_1(Q)$
  $4 : f'_1 \longleftarrow {f'_1}^2 \times h'_1(Q)$
  if $l_i = 1$ **then**
    $5 : T \leftarrow T + P$
    $6 : f_2 \longleftarrow f_2 \times h_2(Q)$
    $7 : f'_2 \longleftarrow f'_2 \times h'_2(Q)$
  end
end
return $\frac{f_1}{f'_1} \times \frac{f_2}{f'_2} \in \mathbb{F}^*_{q^k}$

In algorithm 1, the function h_1 is the equation $h_1(x, y) = 0$ of the tangent line to the curve E at point T (that corresponds to the doubling $[2]T$) and h'_1 corresponds to the vertical line at point $[2]T$. The second part is executed only if the digit i^{th} of l considered is equal to one, and h_2 corresponds to the straight line defined by T and P, h'_2 corresponds to the vertical line at point $(T + P)$. The Miller algorithm is used to compute the Weil, Tate and Ate pairings (see A).

2.3 Jacobian coordinates

We consider that the elliptic curve over $\mathbb{F}_q$ is given in Jacobian coordinates: $E : Y^2 = X^3 + aXZ^4 + bZ^6$, with a and $b \in \mathbb{F}_q$. In Jacobian coordinates, the point P is $P = (X_P, Y_P, Z_P)$. To write P in affine coordinates we use the relation $P = (\frac{X_P}{Z_P^2}, \frac{Y_P}{Z_P^3}, 1)$ for $Z_P \neq 0$. As described in [13] and [1], for optimisation reasons point Q is in affine coordinates. We need at least two coordinates to find the point P, the third can be found using the elliptic curve equation.

The equation of h_1 is:

$$h_1(x, y) = Z_3 Z^2 y - 2Y^2 - 3(X - Z^2)(X + Z^2)(Z^2 x - X)$$

where, $T = (X, Y, Z)$ and $[2]T = (X_3, Y_3, Z_3)$ with $Z_3 = 2YZ$.
Capital letters are elements of $\mathbb{F}_q$, while lower case letters are elements of $\mathbb{F}_{q^k}$.

3 DPA attack

The basic idea of DPA in the context of the cryptanalysis is to correlate the power consumed by the device with the processed data in order to retrieve the secret cipher key. In practice, the supply current measurements during a large number of encryptions are recorded then divided over two sets by means of a selection function and guesses on the secret key. The correct secret key is identified among all the guesses thanks to a statistical analysis.

The transition on the output of a CMOS gate leads to a charge or a discharge of the parasite capacitance C on inputs of downstream gates. In particular, when the output switches from 0 to 1, the supply line sources a current and C is charged (Figure 2 arrow (a)). Conversely, when the output switches from 1 to 0, the current coming from the capacitance is discharged to the ground (Figure 2 arrow (b)). Transitions from 0 to 0 and from 1 to 1 do not participate in the overall power consumption of the circuit.

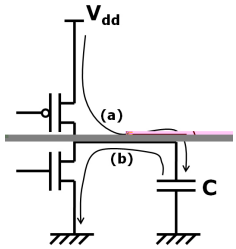


Figure 2: Power consumption in a CMOS circuit

Power dissipated by the circuit can be monitored by using a small resistor in series between Vdd and the power supply. The current measured on this resistor

is the sum of all the currents due to gates that switch from 0 to 1. We now introduce some theoretical issues that allow the reader to understand the principle underlying the DPA attack. We consider the output of a gate whose state depends on both the plain text under ciphering (primary inputs) and the secret key. It is called target node.
We consider now a sequence of input patterns $P_0, P_1, \ldots, P_n$ that generate the transitions $T_1(P_0 \rightarrow P_1), T_2(P_1 \rightarrow P_2), \ldots, T_n(P_{n-1} \rightarrow P_n)$ on the circuit primary inputs. A logic simulation of the circuit while monitoring the target node allows classifying these input transitions in two sets, according to a guess on the key:

- PA, composed by the transitions that make the target node to commute from 0 to 1 and therefore that make the target gate to consume;
- PB, composed by the transitions that do not lead the target gate to participate to the power consumed by the circuit (i.e., transitions from 0 to 0, from 1 to 1, and from 1 to 0 on the target node).

Figure 3 represents the power consumption of the device when stimulated by numerous input vectors.We assume here that the guess on the secret key is correct. In other word, the simulation is performed with the key actually used in the circuit from which power consumptions are collected. Each rectangle represents the total power consumed by the circuit when a new vector is applied to the inputs. In this figure, and just for clarity of explanation, the power consumption is represented by a rectangle corresponding to the average of the consumption over the transition time. In the following, this issue will be re-defined in a more precise way.

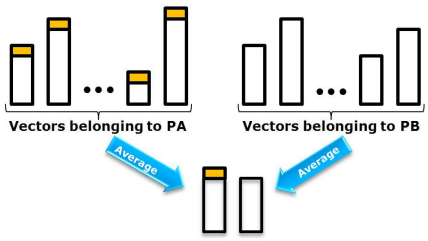


Figure 3: Classification of sets PA and PB

The set of transitions on the circuit inputs is split in the two sets: in the left part there are the PA transitions and the related consumptions while in the right part there are the PB transitions and their corresponding consumptions. A part of the power consumption related to the transitions belonging to PA is due to the power consumed by the target gate (shaded rectangles). Obviously, the commutation from 0 to 1 of non-target nodes also contribute to the power consumption of the circuit but input transitions that leads to such commutations

are assumed to be evenly distributed to sets PA and PB. If a large number of transitions are considered, mean consumptions related to sets PA and PB are almost equal, except for the contribution of the target node.

In other words, since we classified the two sets in such a way that the set PA always leads to a component of power consumption that is not present in the set PB, the difference between the two mean powers computed from set PA and set PB must show a noticeable difference.

During a DPA attack, the target node is chosen in such a way that it depends on a small part of the key only, so that all the key suppositions can be considered. For example, when DPA is conducted against Data Encryption Standard (for which this type of attack has first introduced and it has shown its efficiency [14]), the target node is chosen as the output of an SBox that depends on 6 input bits and on 6 (of 54) secret key bits. Thus, only 64 key guesses are needed, instead of 2^{54}.

For each key guess, the two sets PA and PB are created according to the results of the logic simulation and the key guess under evaluation. The power mean values are calculated for each set using the power traces measured on the circuit under attack for each transition. Finally, the differences of the mean values of the two sets are calculated. When the key guess is correct (and only in this case), PA actually includes the input transitions that lead to a transition 0 to 1 on the target node while PB does not include any of these transitions. The difference between the mean power obtained from PA and PB can be observed in this case. On the contrary, when the curves are classed in PA or PB independently from the actual value of the secret key, the two average curves do not present any noticeable difference.

It is important to note that the actual attack is performed by measuring and analyzing the instantaneous power consumptions over the whole transition period, and not using the time-averaged value as shown in Figure 3. A real attacker (without any knowledge about the hardware implementation of the circuit) cannot know the exact instant when the target node commutes. So the attacker needs to analyze the evolution of the power consumption in function of the time. Figure 4 shows the appearance of the result of an attack over a period of 1ns and 8 key guesses. Each of the 8 curves represents the difference between the mean powers issued from the transitions classified in sets PA and PB, in function of time. The curve that shows the higher peak (bold line in Figure 4) corresponds to the correct key guess (if the attack has been successfully conducted).

4 Description of the circuit

In order to validate the DPA attack on the Miller algorithm, we implemented a circuit able to calculate the h_1 function. We focused on an 8-bit architecture that performs all the calculations in modulo-251. Even if it is smaller than an actual circuit for pairing calculation, it allows anyway validating the attack

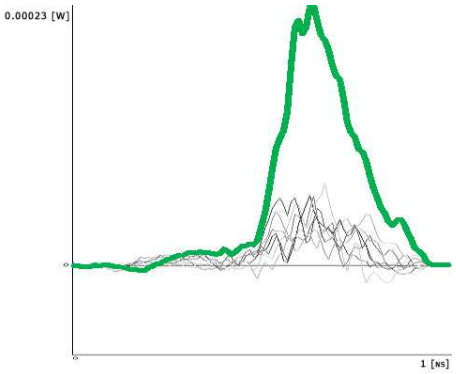
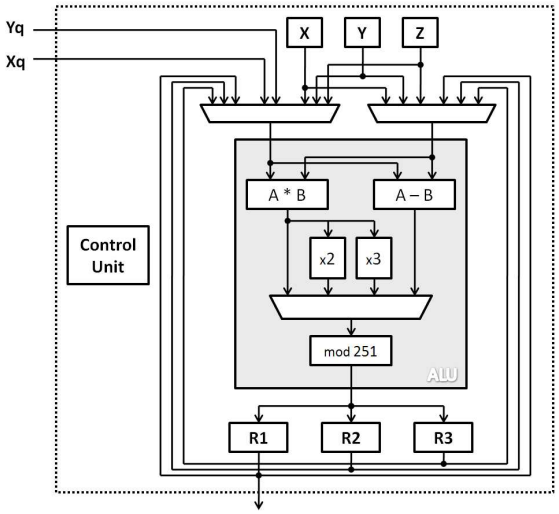


Figure 4: Appearance of the result of the DPA attack

principle. Indeed, DPA is performed focusing on a target node that depends on a small part of the key only (usually no more than 8 bits so that exhaustive analysis can be done). Therefore, also for a bigger architecture the DPA will concentrate on a sub-part of the circuit that operates on 8-bit only.



Algorithme to compute $h_1(x_Q, y_Q)$	
1	$R1 = R3 = Z^2$
2	$R1 = R1 \times x_Q$
3	$R1 = R1 - X$
4	$R2 = X - R3$
5	$R3 = X + R3$
6	$R2 = 3 \times R2 \times R3$
7	$R3 = R2 \times R1$
8	$R1 = Z^2$
9	$R1 = R1 \times y_Q$
10	$R1 = R1 \times Y$
11	$R1 = 2 \times R1 \times Z$
12	$R2 = 2 \times Y^2$
13	$R1 = R1 - R2$
14	$R1 = R1 - R3$
15	$R1 = h_1(x_Q, y_Q)$

Figure 5: Circuit and algorithm for the calculation of $h_1(x_Q, y_Q)$

The circuit is sketched in the left of Figure 5. The datapath includes 3 registers for the 3 components of the secret key (X, Y, Z), 3 working registers $(R1, R2, R3)$ and an ALU that implements a multiplication and a subtraction between two input variables X and Y, and the multiplications by 2 and by 3 of the result of the multiplication. The control unit (CU) drives all the signals to perform the calculation of the function h_1. In particular, the CU processes

the input data (x_Q, y_Q) and computes the final result in 15 clock cycles. The algorithm implemented by the CU is described in the right of Figure 5.

The circuit has been described in VHDL then synthesised using a 130nm technology provided by STm [22]. The overall area is equal to 79059 equivalent gates and the circuit can operate at 120 MHz.

5 DPA attack against Miller algorithm

5.1 Context of the attack

The aim of identity based encryption is that anyone is able to receive and more importantly to read an encrypted message with almost no help, even if this person is not a cryptographer. The user's public key is equal to his identity, and a trusted authority (T_A) sends him his private key. This trusted authority creates all the private keys related to an Identity Based (IB) protocol. The general scheme of identity based encryption is described in [5].

The important point during an IB protocol is that the decryption involves a pairing computation between the private key of the user and a public key. We call the public key the part of the message used during the pairing calculation involving the secret key.

A potential attacker can know the algorithm used, the number of iterations and the exponent. The secret is only one of the argument of the pairing. The secret key influences neither the time execution nor the number of iterations of the algorithm, which is different from RSA or AES protocols.

From here on, the secret will be denoted P and the public parameter (or the point used by the attacker) Q. We are going to describe a DPA attack against the Miller algorithm. We restrict this study to the case where the secret is used as the first argument of the pairing. If the secret is used as the second argument, the same attack can be applied. We assume that the algorithm is implemented on an electronic device like a smart card and used in a protocol involving IBC. The attacker can send as many known entry Q for the decryption operation of IBC as he wants, and he can collect the power consumption curves.

5.2 Convention

5.2.1 Multiplication in $\mathbb{F}_q$

We describe the attacks as if we have $k = 1$, and then Q coordinates are element of $\mathbb{F}_q$. This way, the multiplication $Z_P^2 x_Q$ is a multiplication in $\mathbb{F}_q$. The DPA attack works also when $k > 1$. Even if the multiplication $Z_P^2 x_Q$ becomes a multiplication between an element of $\mathbb{F}_q$ and an element of $\mathbb{F}_{q^k}$, we can consider a multiplication between two $\mathbb{F}_q$ elements.

Indeed, $x_Q \in \mathbb{F}_{q^k}$ is written : $x_Q = \sum_{i=0}^{k-1} x_{Q_i} \xi^i$, with $(1, \xi, \xi^2, \ldots, \xi^{k-1})$ a basis of $\mathbb{F}_{q^k}$, and; there exists a polynomial R such that $deg(R) = k$ with ξ root of R, $(R(\xi) = 0)$. Then $Z_P^2 x_Q = \sum_{i=0}^{k-1} \left(Z_P^2 \times x_{Q_i}\right) \xi^i$, is composed of k

products in $\mathbb{F}_q$. So we can focus on one of these k products in $\mathbb{F}_q$ to apply the DPA attack as described.

By the same way, to compute the difference $(Z_P^2 x_Q - X)$, we compute a difference between elements of $\mathbb{F}_q$ as in the affine case.
Indeed, if $Z_P^2 x_Q = \sum_{i=0}^{k-1} (Z_P^2 x_Q)_i \xi^i$ then

$$Z_P^2 x_Q - X = \left((Z_P^2 x_Q)_0 - X\right) + \sum_{i=1}^{k-1} (Z_P^2 x_Q)_i \xi^i.$$

5.2.2 First iteration

We describe the attack for the first iteration. It is the simplest case, because we know that for this iteration $T = P$. We can provide the attack for the j^{th} iteration. For this iteration we find $T = [j]P$, where $[j]P$ represents the scalar multiplication of point P by the integer j.

We know l, the order of the point Q,(as P and Q have the same order). By counting the number of clock cycles we can find the number d of iterations we have made before the DPA attack. Then reading the binary decomposition of l gives us directly j. We consider that at the beginning $j = 1$, if $l_{n-1} = 0$ then $j \leftarrow 2j$, else $j \leftarrow 2j + 1$, and we go on, until we arrive to the $(n - 1 - d)^{th}$ bit of l.

If the attack is done during the j^{th} iteration of the Miller algorithm, we find the coordinates of $[j]P$. In order to find P, we just have to compute j' the inverse of j modulo l, and then $P = [j'][j]P$.

Furthermore, we present the attack against the basic Miller algorithm. The attack can be straightforwardly generalised to the optimised Miller algorithm given in [13].

5.3 Description of the attack and results

In order to retrieve the secret key $P = (X_P, Y_P, Z_P)$, the circuit has to be used to perform some calculations while the power consumption of the physical device is monitored. In particular, the measure of the consumed power must be done during a time slot when the circuit calculates a result that depends on both the secret key and some controllable input data. We decided to observe the power consumption when the circuit performs the multiplication between Z_P^2 (a part of the secret key) and x_Q (the input data). This operation is done during the second control step (see Figure 5). To retrieve the second part of the key (X_P) we focused on the subtraction between the previously performed multiplication and the key (third control step in Figure 5). The last part of the key (Y_P) can be mathematically inferred from X_P and Z_P^2. Indeed, the elliptic curve equation $E : Y^2 = X^3 + aXZ^4 + bZ^6$ is a quadratic equation in Y_P. The square root of $\sqrt{X_P^3 + aX_P Z_P^4 + bZ_P^6}$ gives us two possibilities for the value of Y_P, testing them by an execution of the Miller

algorithm will give the correct coordinates for P.

To avoid the cost of the circuit manufacturing and the drawbacks of the real experiments like experiment setup or noise management, we used an integrated simulation environment for the Differential Power Analysis [8] that returns power consumption traces from transistor level simulations. This DPA suite also executes the statistical analysis as described in Section 3. Historically the transistor level simulation of a big circuit was a very time consuming process. Nowadays, thanks to new algorithms and higher computation power of computers, it is possible to simulate circuits with hundreds of thousands transistors in a reasonable time, typically in the order of minutes. We used the Synopsys NanoSim [23] simulator that is an advanced circuit simulator for analog, high-performance digital and mixed-signal circuits. As proven in [8], the error obtained by electrical simulation is worthless with respect to the purpose of the validation.

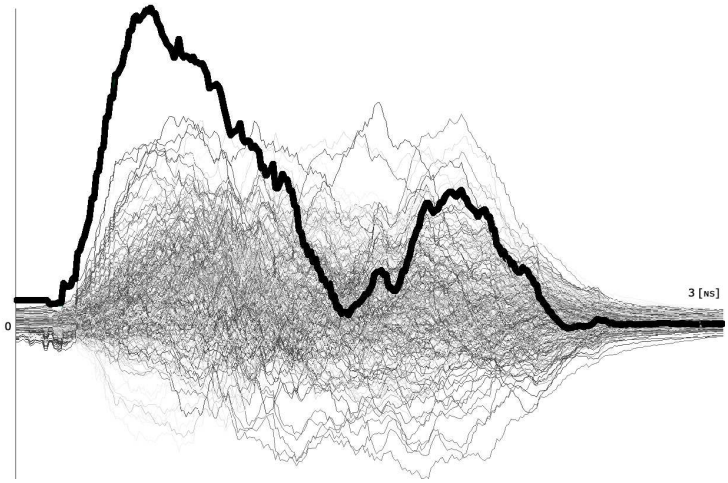


Figure 6: DPA attack against multiplication

The attack of the first part of the key (Z component) has been carried out using 256 input vectors (256 transitions). As target node we considered the output of the multiplication, before the Modulo-251 operation. Since Z is coded over 8 bits, $256 = 2^8$ key guesses must be evaluated. Figure 6 shows the result of the attack. The curve corresponding to the correct key is clearly identifiable among the 256 curves issued from the 256 key guesses since it has the most noticeable pick among all the curves. This curve, and more generally the curve corresponding to the highest integral value, corresponds to the biggest difference between power measures issued from sets PA and PB and thus means that transitions have been correctly assigned to the set PA (the target node effectively switches from 0 to 1). On the time axis, the instant 0 corresponds to beginning of the second control step.

Concerning the second part of the key (control step 3), the experiment disclosed that a higher number of input vectors must be used in order to retrieve the correct key. It depends on the fact that the circuit that implements the difference is much smaller than the multiplier, so its power consumption is drowned in the global power consumption of the device and thus less relevant. However, increasing the number of power traces allows generally to show up even very small power consumptions. We used all the combinations of input transitions for the variable x_Q, thus 65280 input vectors. Figure 7 shows the result of attack. Also in this case the curve corresponding to the correct key is identifiable but the confidence with which the decision is taken is less significant. In particular, by just showing the curve that presents the highest peak, the correct key guess cannot be found. The secret can be found if the method used to search the right curve is based on a more sophisticated analysis that calculates the integral of the curve. More details on search methods for DPA can be found in [8].

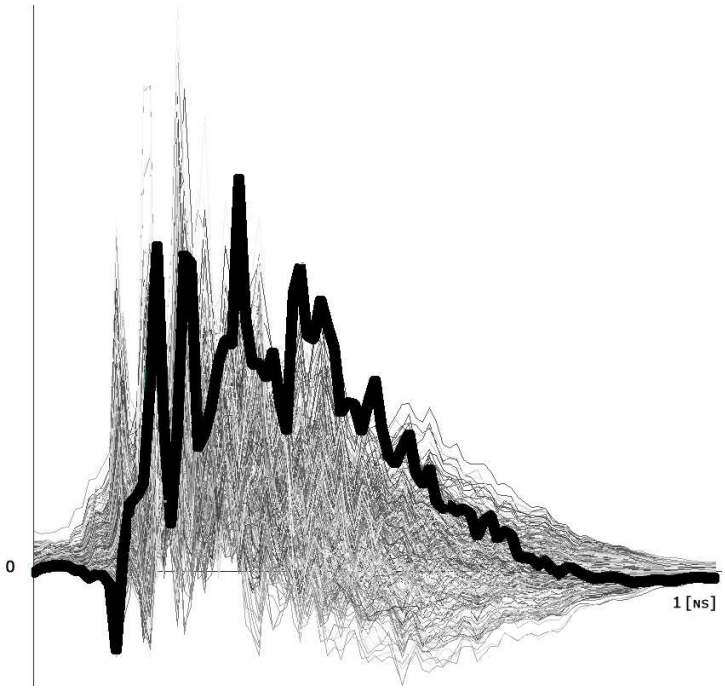


Figure 7: DPA attack against difference

6 Conclusion and further work

We have given the definition of pairings and highlight that a secret is used in Identity Based Cryptography. In the past, pairing based cryptography did not involve any secret, which is not the case for Identity Based Cryptography. We presented and realised the first DPA attack against the Miller algorithm in Jacobian coordinates.

Experimental results performed using electrical level simulation proved that without any countermeasure the secret key can be discovered. As future work, we will study some countermeasure techniques based on the three Coron countermeasures [7]. The third Coron Countermeasure is the more efficient against the attack we described. This countermeasure uses the Jacobian homogeneity property:

$$\left(\forall\ \lambda \in \mathbb{F}_q^*, (X, Y, Z) = (\lambda^2 X, \lambda^3 Y, \lambda Z)\right).$$

A spin off to the first Coron countermeasure was proposed by Page D. and Vercauteren F. [19], it uses the pairing bilinearity property. Indeed, $e_*([s]P, [r]Q) = e_*(P, Q)^{sr}$ ($e_*(P, Q)$ represents a pairing, no matter which), thus for a couple (r, s) such that $rs \equiv 1\ mod(l)$, this countermeasure involves only two scalar multiplications on the curve before the pairing computation. This countermeasure is efficient against the DPA attack, but it can be passed by by increasing the number of curve traces. More theoretical countermeasure to DPA attacks can be found in [6, chap. 27].

References

[1] Bajard J.C. and El Mrabet N.: *Pairing in cryptography: an arithmetic point de view*, Advanced Signal Processing Algorithms, Architectures, and Implementations XVI, part of SPIE, August 2007.

[2] Bertoni G. M., Chen L., Fragneto P., Harrison K.A. and Pelosi G.: *Computing Tate pairing on smartcards*, Proceedings of Ches'05, Workshop on Cryptographic Hardware and Embedded Systems 2005 (CHES 2005) Edinburgh, Scotland.

[3] Boneh D., Franklin M.: *Identity-based encryption from the Weil pairing.* SIAM Journal of Computing, 32, 586–615, 2003

[4] Brier E., Joye M.: *Point multiplication on elliptic curves through isogenies*, AAECC 2003, LNCS., vol. 2643, 2003, 43–50.

[5] Blake F., Seroussi G., Smart N. (editors): *Advances in Elliptic Curve Cryptography*, Series: London Mathematical Society Lecture Note Series (No. 317),Cambridge University Press,2005

[6] Cohen, H., Frey, G. (editors): *Handbook of elliptic and hyperelliptic curve cryptography.* Discrete Math. Appl., Chapman & Hall/CRC (2006)

[7] Coron J.S.: *Countermesures to DPA for Elliptic Curve Cryptography*

[8] Di Natale G., Flottes M.-L. and Rouzeyre B.: *An Integrated Validation Environment for Differential Power Analysis* IEEE International Symposium on Electronic Design, Test & Applications (DELTA 2008), Hong Kong, January 2008, pp. 527-532

[9] Frey G., Müller M., Rück H.G.: *The Tate Pairing and the Discrete Logarithm Applied to Elliptic Curve Cryptosystems*, IEEE Transactions Inf. Theory, 45, 1717-1719;1999

[10] Galbraith S.: Pairings, Chapter IX, *Advances in Elliptic Curve Cryptography*, F. Blake and G. Seroussi and N. Smart editors, Series: London Mathematical Society Lecture Note Series (No. 317),Cambridge University Press,2005

[11] Joux A.: *one round protocol for tripartite Diffie-Hellman*, Algorithmic Number Theory: Fourth International Symposium, Lecture Notes in Computer Science, 1838 (2000), 385-393. Full version: Journal of Cryptology, 17 (2004), 263-276

[12] Tae Hyun K., Tsuyoshi T., Dong-Guk H., Ho Won K. and Jongin L.: *Side Channel Attacks and Countermesures on Pairing Based Cryptosystems over Binary Fields*, The 5th International Conference on Cryptology and Network Security (CANS 2006), LNCS 4301, Springer-Verlag 2006, 168-181.

[13] Koblitz N., Menezes A. J.: *Pairing-based cryptography at high security levels*, Proceedings of the Tenth IMA International Conference on Cryptography and Coding, Springer-Verlag, LNCS 3796, 2005, 13-36;

[14] Kocher P., Jaffe J. and Jun B.: Differential power analysis, Advances in Cryptology Ű Crypto 1999, Lecture Notes in Comput. Sci., vol. 1666, Springer-Verlag, Berlin, 1999, 388–397.

[15] Menezes A.: *An introduction to pairing-based cryptography* Notes from lectures given in Santander, Spain, 2005
http://www.cacr.math.uwaterloo.ca/∼ajmeneze/publications/pairings.pdf

[16] Miller V.: *The Weil pairing and its efficient calculation*, J. Cryptology, 17 (2004), 235-261.

[17] Matsuda S., Kanayama N., Hess F. and Okamoto E.: *Optimised versions of the Ate and Twisted Ate Pairings* http://eprint.iacr.org/2007/013 .

[18] MenezesA., Okamoto T. and Vanstone S.A.: *Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field*, IEEE Trans. Inf . Theory 39, numéro 5, pages 1639-1646, 1993.

[19] Page D. and Vercauteren F.: *Fault and Side Channel Attacks on Pairing Based Cryptography*, IEEE Transactions on Computers, vol. 55, no. 9, pp. 1075-1080, Sept., 2006.

[20] Scott M.: *Computing the Tate Pairing*, Lecture Notes in Computer Science 3376, Springer-Verlag, 2005. ed.Cryptography track - RSA-2005, - , San Francisco, USA, 293 - 304.

[21] Whelan C. and Scott M.: *Side Channel Analysis of Practical Pairing Implementation: Which Path is More Secure ?*, Lecture Notes in Computer Science, Volume 4341 Springer-Verlag ed.VietCrypt 2006, 25-SEP-06 - 28-SEP-06, Hanoi, Vietnam, 99 - 114.

[22] *http://www.st.com*

[23] *http://www.synopsys.com*

A Pairing algorithm

A.1 Definition of Weil pairing

The Weil pairing is defined for $P \in G_1$, and $Q \in G_2$. We denote it e_W, such that:

$$\begin{aligned} e_W : G_1 \times G_2 &\mapsto G_3 \\ (P,Q) &\mapsto e_W(P,Q) = \frac{F_P(Q)}{F_Q(P)} \end{aligned}$$

To calculate this pairing, The Miller algorithm is applied twice. The Miller Lite is the calculation of $F_P(Q)$, and the calculation of $F_Q(P)$ is the Full Miller part.

A.2 Definition of Tate pairing

By the same way, we define the Tate pairing, denoted e_T:

$$\begin{aligned} e_T : G_1 \times G_2 &\mapsto G_3 \\ (P,Q) &\mapsto e_T(P,Q) = F_P(Q)^{\frac{q^k-1}{l}} \end{aligned}$$

Here, $F_P(Q)$ is evaluated with the Miller Lite algorithm. Then, this value is raised to the power $\frac{q^k-1}{l}$ by a classical exponentiation algorithm which can use a sliding window approach.

A.3 Definition of the Ate pairing

Let t be the trace of the Frobenius endomorphism on E see [6, chap. 5] for more details. Let $T = t - 1$, and $N = gcd(T^k - 1, q^k - 1)$. $P \in G_1$ and $Q \in G_2$, then the Ate pairing between P and Q is e_A :

$$\begin{aligned} G_2 \times G_1 &\mapsto G_3 \\ (Q,P) &\mapsto e_A(Q,P) = F_{T,Q}(P)^{\frac{c_T(q^k-1)}{N}} \end{aligned}$$

with $c_T = \sum_{i=0}^{k-1} T^{k-1-i} q^i \equiv\ kq^{k-1}\ mod(l)$