Article 25fa pilot End User Agreement

This publication is distributed under the terms of Article 25fa of the Dutch Copyright Act (Auteurswet) with explicit consent by the author. Dutch law entitles the maker of a short scientific work funded either wholly or partially by Dutch public funds to make that work publicly available for no consideration following a reasonable period of time after the work was first published, provided that clear reference is made to the source of the first publication of the work.

This publication is distributed under The Association of Universities in the Netherlands (VSNU) 'Article 25fa implementation' pilot project. In this pilot research outputs of researchers employed by Dutch Universities that comply with the legal requirements of Article 25fa of the Dutch Copyright Act are distributed online and free of cost or other barriers in institutional repositories. Research outputs are distributed six months after their first online publication in the original published version and with proper attribution to the source of the original publication.

You are permitted to download and use the publication for personal purposes. All rights remain with the author(s) and/or copyrights owner(s) of this work. Any use of the publication other than authorised under this licence or copyright law is prohibited.

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please contact the Library through email: copyright@ubn.ru.nl, or send a letter to:

University Library
Radboud University
Copyright Information Point
PO Box 9100
6500 HA Nijmegen

You will be contacted as soon as possible.

Chapter 3

Recent Developments in Side-Channel Analysis on Elliptic Curve Cryptography Implementations

Louiza Papachristodoulou, Lejla Batina and Nele Mentens

3.1 Introduction

The emerging need for secure communications in embedded systems is constantly threatened by sophisticated side-channel analysis (SCA) attacks. SCA attacks exploit various types of physical leakage of secret information from cryptographic devices. The physical leakage originates also from the power consumption [1], the electromagnetic radiation [2, 3], and the timing behavior [4] of the device. We focus on attacks exploiting power consumption leakage, namely power analysis attacks. These attacks are based on the principle that a switching event of a signal inside a device causes a current to be drawn from the power supply or to be drained to the ground, which is illustrated in Fig. 3.1 on the basis of a CMOS inverter. When the input switches from a logical 1 to a logical 0 or vice versa, the output makes the opposite transition, respectively charging or discharging the output capacitor. When the input remains constant, there is no switching current and no switching power consumption. This physical behavior is exploited by power analysis attacks to extract data that are processed internally in the device.

Within this area of power analysis of cryptographic implementations, there are various methods of analysis, such as Simple Power Analysis (SPA), Differential Power Analysis (DPA), and Collision Analysis (CA). SPA uses a single power trace or several traces, i.e., the instantaneous power consumption of a single run of an algorithm over a certain period of time. DPA uses statistical methods to extract

L. Papachristodoulou · L. Batina
Digital Security Group, Radboud University, P.O. Box 9010, 6500 Nijmegen,
GL, The Netherlands
e-mail: louiza@cryptologio.org

L. Batina
e-mail: lejla@cs.ru.nl

N. Mentens (✉)
KU Leuven, ESAT/COSIC & IMinds, Kasteelpark Arenberg 10, 3001 Leuven, Belgium
e-mail: nele.mentens@kuleuven.be

Fig. 3.1 Switching current at the output of a CMOS inverter

information from multiple traces [1]. CA exploits the leakage of two portions of traces when the same intermediate values are used [5].

Naive implementations of public-key cryptosystems are usually susceptible to SPA attacks because of, e.g., the use of conditional branches. In the RSA cryptosystem, these branches are present in the modular exponentiation algorithm when it is executed using an iteration of modular squarings and modular multiplications. The analogy of modular exponentiation in RSA is point multiplication in elliptic curve cryptosystems. Naive implementations use the double-and-add method consisting of consecutive point doublings and point additions, where a point addition is only executed when the corresponding key bit equals 1. This way, a single power trace reveals a logical 1 in the key through the presence of a point addition. One type of countermeasures balance the computation such that the power traces always look similar regardless of the processed key bits. Other countermeasures randomize the computation such that an attacker is not able to correlate the power traces with the processed data.

This chapter starts with an overview of elliptic curves used in cryptography in Sect. 3.2. Since the power analysis attacks we discuss, focus on the scalar multiplication algorithm, Sect. 3.3 presents different options for this algorithm. Section 3.4 elaborates on power analysis attacks on elliptic curve cryptosystems, while Sect. 3.5 gives an overview of countermeasures at the algorithmic level.

3.2 Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) was introduced around 1985 independently by Miller [6] and Koblitz [7]. It is broadly used for implementing asymmetric cryptographic protocols in embedded devices due to the small key length and memory

requirements compared to equivalent RSA implementations. For instance, a 256-bit field curve provides a security level of 128 bits, which is roughly equivalent to a 2048-bit RSA key (see [8] for more details).

3.2.1 *Coordinate Systems*

An elliptic curve $\mathcal{E}$ over the finite field K , denoted as $\mathcal{E}_K$, can be defined in terms of solutions (x, y) to one of the equations defined in Sect. 3.2.2. The pairs that verify these equations represent the *affine coordinates* of a point over the curve $\mathcal{E}$. From the addition rules on an elliptic curve, the necessary operations are addition, multiplication, and inversion over K . Inversion is the most expensive operation and can be avoided using other types of coordinate systems for the points $\mathcal{P} = (x, y)$. We hereby present the most commonly used coordinates systems that can be found in cryptographic implementations of elliptic curve protocols.

Projective coordinates

In the projective coordinate system, each point $\mathcal{P} = (x, y)$ is represented by three coordinates (X, Y, Z) , where $x = \frac{X}{Z}$, $y = \frac{Y}{Z}$.

Jacobian coordinates

In the Jacobian coordinates system, each point $\mathcal{P} = (x, y)$ is represented also by three coordinates (X, Y, Z) , with $x = \frac{X}{Z^2}$, $y = \frac{Y}{Z^3}$.

López-Dahab coordinates

In the López-Dahab system, the relation for the point (X, Y, Z) is $x = \frac{X}{Z}$, $y = \frac{Y}{Z^2}$.

3.2.2 *Forms of Elliptic Curves*

There are several forms of elliptic curves defined by their curve equation. Below we will treat some commonly used forms.

Weierstrass curves

An elliptic curve defined over a field K is defined by the Weierstrass equation

$$\mathcal{E}_K : y^2 + \alpha_1 xy + \alpha_3 y = x^3 + \alpha_2 x^2 + \alpha_4 x + \alpha_6. \quad (3.1)$$

Together with the point at infinity $\mathcal{O}$, the set $(\mathcal{E}_K \cup \mathcal{O}, +)$ forms an abelian group with neutral element $\mathcal{O}$.

When the characteristic of the field K is not 2 or 3, then the general Weierstrass form can be simplified to

$$\mathcal{E}_K : y^2 = x^3 + \alpha x + \beta. \quad (3.2)$$

In the following, it is assumed that $\text{char}(K) \neq 2, 3$. Adding the points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ gives a third point on the curve, namely $P + Q = (x_3, y_3)$ according to the formulas

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2, \\ y_3 = \lambda(x_1 - x_3) - y_1, \end{cases}$$

with $\lambda = \frac{y_1 - y_2}{x_1 - x_2}$ if $P \neq Q$ and $\lambda = \frac{3x_1^2 + \alpha}{2y_1}$ if $P = Q$.

For points represented in Jacobian coordinates, $P = (X_1, Y_1, Z_1)$ and $Q = (X_2, Y_2, Z_2)$, the addition of P and Q with $P \neq Q$ is $P + Q = (X_3, Y_3, Z_3)$ with

$$X_3 = F^2 - E^3 - 2BE^2, Y_3 = F(BE^2 - X_3) - DE^3, Z_3 = Z_1Z_2E, \quad (3.3)$$

where $A = X_1Z_2^2$, $B = X_2Z_2^2$, $C = Y_1Z_2^3$, $D = Y_2Z_1^3$, $E = A - B$, $F = C - D$. Jacobian addition needs $12\mathbf{M} + 4\mathbf{S}$, with $\mathbf{M}$ and $\mathbf{S}$ the number of multiplications and squarings over K , respectively. Point doubling can be performed very efficiently with only $3\mathbf{M} + 6\mathbf{S}$ using the formulas

$$X_3 = B^2 - 2A, Y_3 = B(A - X_3) - Y_1^4, Z_3 = Y_1Z_1, \quad (3.4)$$

where $A = X_1Y_1^2$, $B = \frac{1}{2}(3X_1^2 + \alpha Z_1^4)$.

Weierstrass curves are standardized and widely used in cryptography [9–12]. However, they have a main drawback regarding their side-channel resistance; namely their addition formulas are incomplete. As is obvious from the previous formulas, addition and doubling are handled differently and the point at infinity gives an exception case. In [13], Bosma and Lenstra presented complete formulas for Weierstrass curves, which had an exceptional case for the pair of points (P, Q) if and only if $P - Q$ is a point of order two. In [14], Renes, Costello, and Batina presented complete addition formulas for odd order elliptic curves $E/\mathbb{F}_q : y^2 = x^3 + \alpha x + b$ with $q \geq 5$, which require only 12 field multiplications $12\mathbf{M}$.¹ The complete addition formulas using Jacobian representation of a point are

$$\begin{cases} X_3 = (X_1Y_2 + X_2Y_1)(Y_1Y_2 - \alpha(X_1Z_2 + X_2Z_1) - 3bZ_1Z_2) \\ \quad - (Y_1Z_2 + Y_2Z_1)(\alpha X_1X_2 + 3b(X_1Z_2 + X_2Z_1) - \alpha^2Z_1Z_2), \\ Y_3 = (Y_1Y_2 + \alpha(X_1Z_2 + X_2Z_1) + 3bZ_1Z_2)(Y_1Y_2 - \alpha(X_1Z_2 + X_2Z_1) - 3bZ_1Z_2) \\ \quad + (3X_1X_2 + \alpha Z_1Z_2)(\alpha X_1X_2 + 3b(X_1Z_2 + X_2Z_1) - \alpha^2Z_1Z_2), \\ Z_3 = (Y_1Z_2 + Y_2Z_1)(Y_1Y_2 + \alpha(X_1Z_2 + X_2Z_1) + 3bZ_1Z_2) \\ \quad + (X_1Y_2 + X_2Y_1)(3X_1X_2 + \alpha Z_1Z_2) \end{cases}$$

Edwards curves

Edwards curves, introduced by Edwards in [15], were the first curves shown to have a complete addition law. Applications of Edwards and twisted Edwards curves in

¹For the overview on elliptic curves in this section, we omit counting the multiplications by a constant (M_α) and the additions A , which are used for extensive comparison results in some publications.

cryptography are extensively studied by Bernstein and Lange [16, 17]. An Edwards curve is defined over a field K with $\text{char}(K) \neq 2$ by the following equation:

$$\mathcal{E}_d : y^2 + x^2 = 1 + dx^2y^2, \quad (3.5)$$

where $d \in K \setminus \{0, 1\}$. The Edwards addition law for two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, in affine coordinates, is given by the following formulas:

$$P + Q = (x_3, y_3) = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2} \right). \quad (3.6)$$

This addition law is unified, i.e., the same formula can be used for both addition and doubling without exceptional cases. The neutral element is the point $(0, 1)$. If d is not a square, then the addition law is complete and there are no exceptional cases for the neutral element.

Twisted Edwards curves, introduced in [18], are a generalization of Edwards curves and they have the form $E_{\alpha,d} : \alpha x^2 + y^2 = 1 + dx^2y^2$. The addition law for twisted Edwards curves is a generalization of Eq. (3.6):

$$P + Q = (x_3, y_3) = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - \alpha x_1x_2}{1 - dx_1x_2y_1y_2} \right). \quad (3.7)$$

The cost of addition and doubling on Edwards curves depends on the form of the curve and the coordinates chosen by the developer. An overview of all types of curves and coordinates is given in the Explicit Formulas Database [19]. The most efficient implementation of twisted Edwards curves is given by Bernstein et al. in [18]. It uses inverted twisted Edwards coordinates and needs $9\mathbf{M} + 1\mathbf{S}$ for addition and $3\mathbf{M} + 4\mathbf{S}$ for doubling.²

Montgomery curves

In [20], P. L. Montgomery defined the following form of elliptic curves over finite fields of odd characteristic:

$$\mathcal{E}_M : By^2 = x^3 + Ax^2 + x, \quad B(A^2 - 4) \neq 0. \quad (3.8)$$

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be points on $\mathcal{E}_M$. Then, the point $P_3 = (x_3, y_3) = P_1 + P_2$ can be calculated using the following formulas:

Addition formulas ($P_1 \neq P_2$)

$$\begin{cases} \lambda = (y_2 - y_1)/(x_2 - x_1) \\ x_3 = B\lambda^2 - A - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases}$$

²The developer can choose to use different formulas for addition and doubling in twisted Edwards curves for extra efficiency in the implementation or unified formulas for resistance against side-channel attacks.

Doubling formulas ($P_1 = P_2$)

$$\begin{cases} \lambda = (3x_1^2 + 2Ax_1 + 1)/(2By_1) \\ x_3 = B\lambda^2 - A - 2x_1 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases}$$

Montgomery arithmetic is very efficient with additional speed-up by computing only (X, Z) coordinates of intermediate points [21]. We set $(x, y) = (X/Z, Y/Z)$ and present the operations in projective coordinates, as described in [20]. We note here that the point $nP = (X_n, Y_n, Z_n)$ is the n -times multiple of the point $P = (X, Y, Z)$. The addition and doubling formulas for $(m+n)P = mP + nP$ are as follows: **Addition formulas** ($m \neq n$)

$$\begin{cases} X_{m+n} = Z_{m-n}[(X_m - Z_m)(X_n + Z_n) + (X_m + Z_m)(X_n - Z_n)]^2 \\ Z_{m+n} = X_{m-n}[(X_m - Z_m)(X_n + Z_n) - (X_m + Z_m)(X_n - Z_n)]^2 \end{cases}$$

Doubling formulas ($m = n$)

$$\begin{cases} 4X_n Z_n = (X_n + Z_n)^2 - (X_n - Z_n)^2 \\ X_{2n} = (X_n + Z_n)^2 (X_n - Z_n)^2 \\ Y_{2n} = (4X_n Z_n)((X_n - Z_n)^2 + ((A+2)/4)(4X_n Z_n)) \end{cases}$$

In [22], it is shown that the loop iteration in the Montgomery ladder using (X, Z) coordinates is performed in only $11\mathbf{M} + 4\mathbf{S}$. Moreover, as presented in [23], the number of additions and doublings in scalar multiplications on Montgomery form elliptic curves only depends on the bit length of the key and not on the bit patterns or the bit itself at a certain position of the scalar.

Hessian curves

A Hessian curve over a field K is defined by the cubic equation

$$\mathcal{E}_K : x^3 + y^3 + z^3 = dxyz, \quad (3.9)$$

where $d \in K$ and $d^3 \neq 27$. Hessian and twisted Hessian curves are interesting for cryptography due to their small cofactor 3 and their side-channel resistance [24–26]. Moreover, the Hessian addition formulas (also called Sylvester formulas) can be used for doubling, a fact that provides a form of unification.

In [27], Farashahi and Joye presented efficient unified formulas for *generalized* Hessian curves as follows:

$$\mathcal{E}_K : x^3 + y^3 + cz^3 = dxyz, \quad (3.10)$$

where $c, d \in K$, $c \neq 0$ and $d^3 \neq 27c$. The unified formulas are complete for certain parameter choices. More precisely, the group of K -rational points on a generalized Hessian curve has complete addition formulas, if and only if c is not a cube in K . The fastest known addition formulas on binary elliptic curves with $9\mathbf{M} + 3\mathbf{S}$ for

extended projective coordinates and $8\mathbf{M} + 3\mathbf{S}$ for mixed affine-projective addition are presented in [27]. The sum of two points P, Q represented in extended projective coordinates by $(X_i : Y_i : Z_i : A_i : B_i : C_i : D_i : E_i : F_i)$, where $A_i = X_i^2, B_i = Y_i^2, C_i = Z_i^2, D_i = X_i Y_i, E_i = X_i Z_i, F_i = Y_i Z_i$ and for $i = 1, 2$, is the point $P + Q = (X_3 : Y_3 : Z_3 : A_3 : B_3 : C_3 : D_3 : E_3 : F_3)$ with

$$\begin{cases} X_3 = cC_1F_2 + D_1A_2, \\ Y_3 = B_1D_2 + cE_1C_2, \\ Z_3 = A_1E_2 + F_1B_2, \end{cases}$$

with $A_3 = X_3^2, B_3 = Y_3^2, C_3 = Z_3^2, D_3 = X_3Y_3, E_3 = X_3Z_3, F_3 = Y_3Z_3$.

The complete addition formulas for twisted Hessian curves of cofactor 3 in [24] give the fastest results for prime-field curves with $8.77\mathbf{M}$ for certain curve parameters.

3.3 Scalar Multiplication Algorithms

ECC primitives are used for cryptographic protocols such as the Elliptic Curve Digital Signature Algorithm (ECDSA) for digital signatures, Elliptic Curve ElGamal as an encryption/decryption scheme, and Elliptic Curve Diffie-Hellman (ECDH) as a key exchange scheme. The main operation in all those protocols using ECC is scalar multiplication of a point $\mathcal{P}$ on a curve and with an integer k .

Computing the result of a scalar multiplication on an elliptic curve can be done in a similar way as exponentiation in RSA. A simple and efficient algorithm is binary scalar multiplication, where an n -bit scalar k is written in its binary form $(k_0, k_1, \dots, k_{n-1})_2$ with $k = \sum_i k_i 2^i, k_i \in \{0, 1\}$. The binary algorithm processes a loop, scanning the bits of the scalar (from the most significant bit to the least significant one, or the other way around) and performing a point doubling only if the current bit is 0 or a doubling and an addition if the bit is 1.

Scalar multiplication algorithms take as input a point $\mathcal{P}$ in affine or projective coordinates and the scalar k . The result is the point $[k]\mathcal{P}$ on the curve. During the execution of the algorithm, mixed coordinates can be used for an additional speed-up [28].

Scalar multiplication is a sensitive operation, since it manipulates the secret key k and returns the result according to the bits of the key. Naïve implementations of scalar multiplication with `if`-statements are subject to SCA, and more precisely timing attacks. Coron's randomization countermeasures of point or scalar, as presented in [29] and in Sect. 3.5, can thwart timing or DPA attacks, but not SPA attacks. SPA leakage is present when there is a difference in operation flow between only doubling or doubling-and-addition. Point additions and point doublings give different leakage patterns and since these operations are key dependent, the key can be retrieved quite easily. SPA-resistant algorithms are *regular* algorithms, which perform a constant operation flow regardless of the scalar value. A nice overview of fast and regular scalar multiplication algorithms is given in [30]. In this section, we present the most broadly used regular scalar multiplication algorithms.

3.3.1 Left-to-Right Double-and-Add-Always Algorithm

The double-and-add-always algorithm was initially proposed by Coron in [31] as a first attempt to avoid `if`-statements and therefore prevent the identification of different operations. The algorithm performs a point doubling followed by a point addition in a `for` loop, scanning the scalar bits from the most significant to the least significant one. Both operations are performed in every loop and according to the key bit, the final assignment to $\mathcal{R}_0$ will be either $\mathcal{R}_0$ or $\mathcal{R}_1$. There are no conditional

Algorithm 1: The left-to-right double-and-add-always algorithm

Input: $\mathcal{P}, k = (k_{x-1}, k_{x-2}, \dots, k_0)_2$

Output: $\mathcal{Q} = k \cdot \mathcal{P}$

$\mathcal{R}_0 \leftarrow \mathcal{P}$;

for $i \leftarrow x - 2$ **down to** 0 **do**

$\mathcal{R}_0 \leftarrow 2\mathcal{R}_0$;

$\mathcal{R}_1 \leftarrow \mathcal{R}_0 + \mathcal{P}$;

$\mathcal{R}_0 \leftarrow \mathcal{R}_{k_i}$;

end

return $\mathcal{R}_0$

statements in the algorithm, but there is one key-dependent assignment, which can leak secret information. Another important remark is that $\mathcal{R}_0$ is initialized by $\mathcal{P}$ instead of $\mathcal{O}$, in order to avoid exceptional cases given by the point at infinity.

3.3.2 Right-to-Left Double-and-Add-Always Algorithm

The binary right-to-left double-and-add-always algorithm of [32] is shown below as Algorithm 2. The steps of the algorithm are similar to Algorithm 1 with the following differences:

- The bits of the scalar are scanned from the least significant to the most significant one.
- Two temporary registers are used instead of three and they are both effectively used, without any dummy operations.

Similar to Algorithm 3.3.1, there are no conditional statements in this algorithm, but there is a key-dependent assignment, which can be vulnerable to attacks. However, there are several attacks that can be mounted on the left-to-right, but not on the right-to-left algorithm (for instance the Doubling attack, described in Sect. 3.4, is only applicable on the left-to-right algorithm).

Algorithm 2: Binary right-to-left double-and-add-always algorithm**Input:** $\mathcal{P}, k = (k_{x-1}, k_{x-2}, \dots, k_0)_2$ **Output:** $\mathcal{Q} = k \cdot \mathcal{P}$ $\mathcal{R}_0 \leftarrow \mathcal{O};$ $\mathcal{R}_1 \leftarrow \mathcal{P};$ **for** $i \leftarrow 0$ **up to** $x-1$ **do** $b \leftarrow 1 - k_i;$ $\mathcal{R}_b \leftarrow 2\mathcal{R}_b;$ $\mathcal{R}_b \leftarrow \mathcal{R}_b + \mathcal{R}_{k_i};$ **end****return** $\mathcal{R}_0$

3.3.3 Montgomery Ladder

The Montgomery Ladder, initially presented by Montgomery in [20] as a way to speed up scalar multiplication on elliptic curves, and later used as the primary secure and efficient choice for resource-constrained devices, is one of the most challenging algorithms for simple side-channel analysis due to its natural regularity of operations. A comprehensive security analysis of the Montgomery ladder, given by Joye and Yen in [33], showed that the regularity of the algorithm makes it intrinsically protected against a large variety of implementation attacks (SPA, some fault attacks, etc.). The Montgomery ladder is described in Algorithm 3. For a specific choice of projective coordinates, as described in Sect. 3.2.2 and in [21], one can do computations with only X and Z coordinates, which makes this option more memory efficient than other algorithms.

Algorithm 3: The Montgomery Ladder**Input:** $\mathcal{P}, k = (k_{x-1}, k_{x-2}, \dots, k_0)_2$ **Output:** $\mathcal{Q} = k \cdot \mathcal{P}$ $\mathcal{R}_0 \leftarrow \mathcal{P};$ $\mathcal{R}_1 \leftarrow 2\mathcal{P};$ **for** $i \leftarrow x-2$ **down to** 0 **do** $b \leftarrow 1 - k_i;$ $\mathcal{R}_b \leftarrow \mathcal{R}_0 + \mathcal{R}_1;$ $\mathcal{R}_{k_i} \leftarrow 2 \cdot \mathcal{R}_{k_i};$ **end****return** $\mathcal{R}_0$

3.3.4 Side-Channel Atomicity

Side-channel atomicity is an SPA countermeasure proposed by Chevallier-Mames et al. [34], in which individual operations are implemented in such a way that they have an identical side-channel profile (e.g., for any branch and any key-bit related subroutine). In short, it is suggested in [34] that the point doubling and addition operations are implemented such that the same code is executed for both operations. This renders the operations indistinguishable by simply inspecting a suitable side-channel. One could, therefore, implement a point multiplication as described in Algorithm 4.

Algorithm 4: Side-Channel Atomic double-and-add algorithm

Input: $\mathcal{P}, k = (k_{x-1}, k_{x-2}, \dots, k_0)_2$

Output: $\mathcal{Q} = k \cdot \mathcal{P}$

$R_0 \leftarrow \mathcal{O}; R_1 \leftarrow \mathcal{P}; i \leftarrow x - 1;$

$n \leftarrow 0;$

while $i \geq 0$ **do**

$\mathcal{R}_0 \leftarrow \mathcal{R}_0 + \mathcal{R}_n;$

$n \leftarrow n \oplus k_i;$

$i \leftarrow i - \neg n;$

end

return $\mathcal{R}_0$

There are certain choices of coordinates and curves for which this approach can be deployed by using unified or complete addition formulas for the group operations. The unified and complete formulas of Weierstrass, Edwards and Hessian curves are described in Sect. 3.2.2.

3.4 Side-Channel Attacks on ECC

Attacking implementations of elliptic curve cryptography (ECC) with natural protection against side-channel attacks, e.g., implementations using Edwards curves, is quite challenging. This form of elliptic curves, proposed by Edwards in 2007 [15] and promoted for cryptographic applications by Bernstein and Lange [18], showed some advantages compared to elliptic curves in Weierstrass form. For instance, the fast and complete formulas for addition and doubling put these types of curves forward as more appealing for memory-constrained devices and at the same time resistant to classical simple power analysis (SPA) techniques. Recently, due to the work of Renes et al., complete formulas have been published for curves in Weierstrass form [14].

Although considered a very serious threat against ECC implementations, differential power analysis (DPA), as proposed in [1], cannot be applied directly to

ECC-based algorithms and protocols. Soon after the first DPA paper by Kocher et al., Coron showed how to attack the scalar multiplication operation by DPA techniques [31]. However, the idea does not apply to other ECC protocols where the secret is either not a scalar involved in the scalar multiplication algorithm or a scalar that is used only once, like, e.g., in ECDSA or in ephemeral Diffie-Hellman. The latter is incompatible with the requirement of DPA to collect a large number of power traces of computations on the same secret data.

When attacking ECDSA, two secrets could be of interest for the attacker. She could go for an ephemeral key or a secret scalar that becomes a part of the signature. The idea of attacking the ephemeral key is to get reveal a few key bits (from just one measurement) and then proceed with some sort of theoretical cryptanalysis to recover the remaining bits. This kind of special attacks is often used in combination with lattice techniques similar to [35, 36], in order to derive the whole private key from a few bits of multiple ephemeral keys.

The richness of the mathematical structures behind public-key systems and other algorithm-dependent features, that are special for both RSA and ECC, created opportunities for many unique side-channel attacks exploiting those features. The first work to propose new techniques was the paper of Fouque and Vallette [37]. They introduce a new attack against scalar multiplication (or modular exponentiation) that looks for identical patterns within power traces due to the same intermediate results occurring within the computation. In this way, the so-called “doubling attack” only requires two queries to the device. This work has started a new line of research on new attack techniques that reside somewhere between SPA and DPA, of which the most notable are collision [5, 37–41] and template attacks [36, 42, 43].

Collision-based attacks exploit the fact that when processing the same data, the same computations will result in the same (or very similar) patterns in the power consumption traces. However, although the idea is easily verifiable, the efficiency of most of the so far introduced collision-based attacks is shown only on simulated traces; no practical experiments on real ECC implementations have confirmed those results. To the best of our knowledge, only two practical collision-based attacks on exponentiation algorithms were published, each of which rely on very specific assumptions and deal with very special cases. Hanley et al. exploit collisions between input and output operations within the same trace [44]. On the other hand, Wenger et al. performed a hardware-specific attack on consecutive rounds of a Montgomery ladder implementation [45]. However, both attacks are very restrictive in terms of applicability to various ECC implementations as they imply some special implementation options, such as, e.g., the use of López–Dahab coordinates, where field multiplications use the same key-dependent coordinate as input to two consecutive rounds. A class of attacks similar to collision-based attacks is sometimes also called horizontal attacks and they were first defined for modular exponentiation by Clavier et al. [46]. Their attack is inspired by Walter’s work [38] and it requires a unique power trace, in this way rendering classical randomization countermeasures. In another work [47], the authors have introduced a general framework enabling to model both horizontal and classical attacks (the latter are called vertical attacks in this work) in a simple way.

Their follow-up paper [41] introduced horizontal attacks for ECC but the results were obtained from simulations only and no real measurements were used.

We observe that the trend of attacks is shifting more toward this type of collision and horizontal attacks as known randomization-based countermeasures are effective in protecting ECC against SPA and DPA attacks. Therefore, in the remainder of this chapter we focus on collision-based/inspired attacks and we give a detailed example of one such attack i.e. the Online Template Attack (OTA).

3.4.1 Collision-Correlation Attacks

As mentioned above, collision attacks exploit leakages by comparing two portions of the same or different traces exploiting the same power being consumed when values are reused. The Big Mac attack [38] is the first theoretical attack on public-key cryptosystems, in which only a single trace is required to observe key dependencies and collisions during an RSA exponentiation. Witteman et al. performed a similar attack on the RSA modular exponentiation even in the presence of blinded messages [48]. Clavier et al. introduced horizontal correlation analysis, as a type of attack where a single power trace is enough to recover the private key [46]. They also extended the Big Mac attack using different distinguishers, i.e., types of statistical tests.

The *doubling attack*, proposed by Fouque and Vallette [37] and described previously, is a special type of collision attack relevant to ECC. The main assumption of this attack is that an adversary can distinguish collisions of power trace segments (within a single or more power traces) when the device under attack performs twice the same computation, even if the adversary is not able to tell which exact computation is done. Collision of two computations will not reveal the value of the operand. Yen et al. extended this attack to the *Refined Doubling Attack (RDA)* [39], where the adversary is assumed to be able to detect the collision between two modular squarings, i.e., detecting if the squared value is the same or not. Collisions of computations cannot be distinguished; the only knowledge obtained is that $k_i = k_{i-1}$ if a collision is detected. Based on the derived relationship between every two adjacent private key bits (either $k_i = k_{i-1}$ or $k_i \neq k_{i-1}$) and a given bit (e.g., k_0 or k_{m-1}), all other private key bits can be derived uniquely. RDA is a powerful attack technique that works against some scalar multiplication algorithms, which are resistant against the doubling attack (e.g., the Montgomery power ladder).

3.4.2 Horizontal Attacks and Variants

An interesting class of side-channel attacks is the *Horizontal Analysis* attack, where a single trace is used to recover the secret scalar. The main characteristic of the traces that makes horizontal attacks possible lies in the fact that the operation sequences of doubling-adding and doubling-doubling can be distinguished. The attacker applies

the classical correlation analysis using different parts of time samples in the same side-channel trace to recover the secret scalar bit-by-bit. This technique can be useful to attack protected implementations, where the secret value or unknown input is blinded. The first horizontal attacks were applied to RSA implementations; extension of those to ECC implementations is straight-forward, since scalar multiplication and exponentiation algorithms have the same operation steps.

The so-called *Big Mac* attack from Walter [38] is the first attack of this kind, where squarings (**S**) are distinguished from multiplications (**M**) and the secret exponent of an RSA exponentiation can be recovered from a single execution curve. The distinction is possible by averaging and comparing the cycles performed in the multiplier of the device during long-integer multiplication, since more cycles are needed for **SM** than for **SS**. This attack can be directly applied to ECC implementations.

The term *horizontal* was first introduced by Clavier et al. in [46], where the authors performed a horizontal correlation analysis to compute the correlation factor on several segments extracted from a single execution curve of a known message RSA encryption. More specifically, their proposed method starts by finding a sequence of elementary calculations $(C_i)_j$ (with $i, j \in \mathbb{Z}$ indicating the sequence of the calculation and the order of the execution respectively) that processes the same mathematical operation (e.g., field multiplication) and depends on the same part of the secret scalar. The outputs O_{i_j} of the calculations $C_i(X_i)$ that depend on the same input value X_i will give high correlation results and in this way, they can be distinguished from outputs of computations with different input values. Horizontal correlation analysis was performed on RSA using the Pearson correlation coefficient in [46] and triangular trace analysis of the exponent in [49].

The most recent attack, proposed by Bauer et al. in [41], is a type of *horizontal collision correlation* attack on ECC, which combines atomicity and randomization techniques. Based on the basic assumption of collision attacks that an adversary is able to distinguish when two field multiplications have at least one common operand, their attack consists of the following steps:

- Identify two elementary calculations C_1, C_2 that are processed N times with inputs from the same distribution. The correlation between the random output values O_1, O_2 must depend on the same secret sub-part s .
- For each of the N processings of C_i get an observation l_j^i , with $j \in \{1, \dots, N\}$.
- Compute the Pearson correlation coefficient on the two samples of observations $\rho = \rho((l_j^1)_j, (l_j^2)_j)$.
- Deduce information on the secret scalar from ρ using an appropriate distinguisher that shows which observation is more similar to the real secret value.

The horizontal collision correlation attack is shown by simulated traces to be applicable to atomic implementations and to implementations based on unified addition formulas over Edwards curves.

Two recent publications on blinded asymmetric algorithms propose the combination of horizontal and vertical techniques, in an attempt to provide more practical attacks against blinded implementations and avoid the complex signal processing phase. Bauer et al. [50] at Indocrypt 2013, presented an attack on RSA blinded

exponentiation based on this approach. They took advantage of the side-channel leakage of the entire long-integer modular multiplication without splitting the trace into parts of single precision multiplications. However, their attack requires a small public exponent (no greater than $2^{16} + 1$) and an exponent blinding factor smaller than 32 bits. Their observation that the scalar blinding does not mask a large part of the secret value, led Feix et al. [51] a year later to exploit this vulnerability vertically on a ECC implementation. The most significant part of the blinded scalar can be recovered with a horizontal attack. The least significant part of the scalar is retrieved using vertical analysis (several execution traces) and the information leaked in the previous steps of the attack.

3.4.3 Template Attacks

The most powerful SCA attack from an information theoretic point of view is considered to be a template attack (TA). Template attacks, as introduced in the original paper by Chari et al. in [52], are a combination of statistical modeling and power analysis attacks consisting of two phases, as follows:

- The first phase is the *profiling* or *template-building* phase, where the adversary builds templates to characterize the device by executing a sequence of instructions on fixed data. Focusing on an “interesting pattern” or finding the points of interest is very common in this phase.
- The second phase is the *template-matching phase*, in which the adversary matches or correlates the templates to actual traces of the device. By applying some signal processing and classification algorithms to the templates, it is possible to find the best matching for the traces.

In this type of attacks, the adversary is assumed to have in his possession a device which behaves similar to the device under attack (target device), in order to build template traces. In his device he can simulate the same algorithms and implementations that run in the target device. For the template-matching phase several distinguishers and classification algorithms are proposed; in the next section the most common classifiers are presented.

The practical application of TAs is shown on several cryptographic implementations such as RC4 in [53] and elliptic curves in [54]. Medwed and Oswald demonstrated in [42] a practical template attack on ECDSA. However, their attack required an offline DPA attack on the EC scalar multiplication operation during the template-building phase, in order to select the points of interest. They also need 33 template traces per key bit. Furthermore, attacks against ECDSA and other elliptic curve signature algorithms only need to recover a few bits of the ephemeral scalar for multiple scalar multiplications with different ephemeral scalars and can then employ lattice techniques to recover the long-term secret key [35, 36, 43]. It is not possible to obtain several traces in the context of ephemeral Diffie-Hellman: an attacker only gets a single trace and needs to recover sufficiently many bits of this ephemeral scalar

from side-channel information to be able to compute the remaining bits through, for example, Kangaroo techniques [55, 56]. With these techniques and by using pre-computation tables, it is possible to exploit partial information on the subkeys and recover the last l unknown bits of the key in $\mathcal{O}(\sqrt[3]{l})$ group operations [57].

3.4.4 Common Distinguishers

In this section, the most common distinguishers used in SCA for correlation analysis and template-matching are presented. Machine learning techniques for classification and clustering are broadly used in SCA, in order to distinguish between traces with high noise ratios.

According to [58], unsupervised clustering is generally useful in side-channel analysis when profiling information is not available and an exhaustive partitioning is computationally infeasible. The authors presented an attack on an FPGA-based elliptic curve scalar multiplication using the k -means method. In [59], Perin et al. used unsupervised learning to attack randomized exponentiations.

Lerman et al. showed in [60] that machine learning techniques give better classification results when there is limited ability of the adversary to perform profiling of the device and in a high dimensionality context, where many parameters affect the leakage of the device. Indeed, combining three side-channel leakages and a clustering-based approach for non-profiled attacks, gives higher success rates than traditional template attacks, as shown by Specht et al. in [61].

The success results of online template attacks (presented in the next section) are significantly improved in [62] by using the k -nearest neighbor approach, naïve Bayes classification and the support vector machine method for template classification. In order to explain these techniques, we first give the definition of the Euclidean distance and the Pearson correlation coefficient.

Euclidean Distance The Euclidean distance between two points is defined as the square root of the sum of the squares of the differences between the corresponding point values:

$$d_{EUC} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

In the SCA setting, a realization of a random variable X corresponds to x . A sample of n observations or traces of X is denoted by $(x_i)_{1 \leq i \leq n}$, where the index i denotes the different observations or the different time when an observation occurs in the same trace.

Pearson correlation The Pearson correlation coefficient measures the linear independence between two observations X and Y :

$$\rho(X, Y) = \frac{n \sum_i x_i y_i - \sum_i x_i \sum_i y_i}{\sqrt{n \sum_i x_i^2 - (\sum_i x_i)^2} \sqrt{n \sum_i y_i^2 - (\sum_i y_i)^2}}$$

In SCA, the Pearson correlation coefficient is used to describe the difference in the Hamming weight of the observations.

Naïve Bayes Classification The naïve Bayes classification method is based on probability concepts, and more precisely on the Bayes theorem for conditional probabilities of independent events [63]. According to the conditional probability model, let $\mathbf{x} = (x_1, \dots, x_n)$ be the vector of problem instances (independent variables) to be classified, each one having a feature n . Each instance is assigned a probability $p(c_k | x_1, \dots, x_n)$, for k possible classes. The set of classes $c_1, c_2, \dots, c_k$ is mutually exclusive and exhaustive.

Using Bayes' theorem, the posterior conditional probability is $p(c_k | \mathbf{x}) = \frac{p(c_k) p(\mathbf{x} | c_k)}{p(\mathbf{x})}$. Assuming that each event and posterior probability is independent on each previous event, the conditional distribution over the class variable c is $p(c_k | x_1, \dots, x_n) = \frac{1}{Z} p(c_k) \prod_{i=1}^n p(x_i | c_k)$ where the evidence $Z = p(\mathbf{x})$ is a scaling factor dependent only on $x_1, \dots, x_n$, that is, a constant if the values of the feature variables are known.

The naïve Bayes classifier is a function that combines the naïve Bayes probability model with a decision rule. One common rule is to pick the hypothesis that is most probable; that is the maximum value of the a posteriori probability. For each class c_i , the class index that gives the maximum value for an event is chosen as classifier.

K-Nearest Neighbor The k -Nearest Neighbor Classification (kNN) is a classification method based on the closest instances of the training set to the unlabeled data. Basically, according to [63], it consists of the following two steps:

1. Choose the number of k closest instances (from the training set) to the sample.
2. The majority label (class) for the chosen closest instances will be class for the unlabeled data.

The distance metric plays an important role, in order to determine the closest instance. In kNN , the Euclidean distance or the Manhattan distance can be used. The value k indicates the number of the already-classified closest instances that are chosen in order to classify the next unlabeled data. The default value is 1, but with larger value for k it is possible to obtain higher success rate with less template traces. Figure 3.2 shows an example with $k = 2$ and $k = 4$ close instances; if the new sample is closer to A, it will be classified in the “A-class.”

Fig. 3.2 kNN method for $k = 2$ and $k = 4$

Fig. 3.3 SVM: distance to the hyperplane for two sets of training data

SVM A Support Vector Machine (SVM) is a supervised learning model that produces a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples. Figure 3.3 shows such a hyperplane. An optimal hyperplane, as defined in [64], is the one that gives the largest minimum distance to the training points, because in this way noisy data will still be classified correctly. Therefore, the optimal separating hyperplane maximizes the margin of the training data. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. The classifier in an SVM can be nonlinear for data sets that are not easily separable, but in the following analysis a linear classifier gives very good results.

3.4.5 A Special Case: Online Template Attacks

In this section, we present in more detail an adaptive template attack technique, which is called an *Online Template Attack* (OTA) and is initially proposed by Batina et al. in [65]. This technique resides between horizontal and template attacks. The attacker is able to recover a complete scalar after obtaining only one power trace of a scalar multiplication from the device under attack. This attack is characterized as *online*, because templates are created *after* the acquisition of the target trace. While the same terminology is used, OTA is not a typical template attack; i.e., no preprocessing template-building phase is necessary. OTA functions by acquiring one target trace from the device under attack and comparing patterns of certain operations from this trace with templates obtained from the attacker's device that runs the same implementation. Pattern matching is performed at suitable points in the algorithm, where key-bit related assignments take place by using an automated module based on the Pearson correlation coefficient.

The attacker needs only very limited control over the device used to generate the online template traces. The main assumption is that the attacker can choose the input point to a scalar multiplication, an assumption that trivially holds even without any modification to the template device in the context of ephemeral Diffie-Hellman. It

also holds in the context of ECDSA, if the attacker can modify the implementation on the template device or can modify internal values of the computation. This is no different than for previous template attacks against ECDSA.

OTA is defined as a side-channel attack with the following conditions:

1. The attacker obtains only one power trace of the cryptographic algorithm involving the targeted secret data. This trace is called the *target trace*. The device, from which the target trace is obtained, is the *target device*. The fact that only one target trace is necessary for the attack, makes it possible to attack scalar multiplication algorithms with ephemeral scalar and with randomized scalar.
2. The attacker is generating template traces *after* having obtained the target trace. These traces are called (*online*) *template traces*.
3. The attacker obtains the template traces on the target device or a similar device³ *with very limited control over it*, i.e., access to the device to run several executions with chosen public inputs. The attacker does not rely on the assumption that the secret data are the same for all template traces.
4. At least one assignment in the exponentiation algorithm is made depending on the value of particular scalar bit(s), but there are no branches with key-dependent computations. Since the doubling operation is attacked, this key-dependent assignment should be during doubling. As a counterexample, it is noted that the binary right-to-left add-always algorithm for Lucas recurrences [32] is resistant to the proposed attack, because the result of the doubling is stored in a non-key-dependent variable.

The attack methodology is as follows:

- Acquire a full target trace from the device under attack, during the execution of a scalar multiplication.
- Locate the doubling and addition operations performed in each round.
- Find multiples of $m\mathcal{P}$, where $m \in \mathbb{Z}$, $m \leq k$ and k is the scalar. These points are used to create the template traces.

The methodology offers a generic attack framework, which does not require any previous knowledge of the leakage model nor a specific type of curve. It is applicable to various forms of curves (Weierstrass, Edwards and Montgomery curves), scalar multiplication algorithms and implementations. Contrary to the doubling attack [37], OTA can be launched against right-to-left algorithms and the Montgomery ladder.

The basic idea, as depicted in Fig. 3.4 consists of comparing the traces for the inputs $\mathcal{P}$ (target trace) and $2\mathcal{P}$ (online template trace) while executing scalar multiplication and then finding similar patterns between them, based on the hypothesis on a bit for a given operation. The target trace is obtained only once. For every bit of the scalar, an online template trace with input $k\mathcal{P}$, $k \in \mathbb{Z}$ is obtained, where k is chosen as a function of a hypothesis on this bit.

In the original paper, pattern matching is performed using the Pearson correlation coefficient, $\rho(X, Y)$, which measures the linear relationship between two vari-

³Similar device means the same type of microcontroller running the same algorithm.

Fig. 3.4 Schematic representation of OTA [66]

ables X and Y . For power traces, the correlation coefficient shows the relationship between two points of the trace, which indicates the Hamming-weight leakage of key-dependent assignments during the execution of a cryptographic algorithm. Extension to other distinguishers from machine learning is performed in [62].

The template matching corresponds to a list of correlation coefficients that show the relationship between all samples from the template trace to the same consecutive amount of samples in the target trace. If the hypothesis on the given key bit is correct, then the pattern match between the template traces at the targeted operation will be high (in experiments it reached 99 %).

In this way, the first i bits of the key can be recovered. Knowledge of the first i bits provides the attacker with complete knowledge of the internal state of the algorithm just before the $(i + 1)$ th bit is processed. Since at least one operation in the loop depends on this bit, a hypothesis can be made about the $(i + 1)$ th bit, a template trace based on this hypothesis is computed, and this trace is correlated with the target trace at the relevant predetermined point of the algorithm.

OTA on scalar multiplication algorithms The core idea and feasibility of the attack is demonstrated through an example based on the double-and-add-always algorithm described in Algorithm 1. Table 3.1 shows two executions of the algorithm for two different scalars $k = 100$ and $k = 110$. The first execution of the loop always starts by doubling the input point $\mathcal{P}$, for all values of k . It is assumed that $k_{x-1} = 1$. Depending on the second-most significant key bit k_{x-2} , the output of the first iteration of the algorithm will be either $2\mathcal{P}$ or $3\mathcal{P}$. For any point $\mathcal{P}$ it is, therefore, possible to get a power trace for the operation $2\mathcal{P}$, i.e., the attacker lets the algorithm execute the first two double-and-add iterations. In the proposed setup, the authors could zoom into the level of one doubling, which will be the template trace. Then, the attacker can

Table 3.1 Two executions of the double-and-add-always algorithm [65]

$k = 100$	$k = 110$
$R_0 = P$	$R_0 = P$
$R_0 = 2P, R_1 = 3P$, return $2P$	$R_0 = 2P, R_1 = 3P$, return $3P$
$R_0 = 4P, R_1 = 5P$, return $4P$	$R_0 = 6P, R_1 = 7P$, return $6P$

perform the same procedure with $2\mathcal{P}$ as the input point to obtain the online template trace that he wants to compare with the target trace. If it is assumed, that the second-most significant bit of k is 0, then he compares the $2\mathcal{P}$ template with the output of the doubling in the first iteration. Otherwise, he compares it with the online template trace for $3\mathcal{P}$.

Assuming that the first $(i - 1)$ bits of k are known, he can derive the i th bit by computing the two possible states of $\mathcal{R}_0$ after this bit has been treated and recover the key iteratively. Note that only the assignment in the i th iteration depends on the key bit k_i , but none of the computations do, so it is necessary to compare the trace of the doubling operation in the $(i + 1)$ th iteration with the original target trace. To decide whether the i th bit of k is zero or one, the trace that the doubling operation in the $(i + 1)$ th iteration would give for $k_{i+1} = 0$ is compared with the target trace. For completeness, he can compare the target trace with a trace obtained for $k_{i+1} = 1$ and verify that it has a lower pattern match percentage; in this case, the performed attack needs two template traces per key bit. However, if during the acquisition phase the noise level is low and the signal is of good quality, an efficient attack can be performed with only the target trace and a single trace for the hypothetical value of $\mathcal{R}_{k_{i+1}}$.

Attacking the right-to-left double-and-add-always algorithm of [32] can be done in a similar way, since it is a type of key-dependent assignment OTA. The attacker targets the doubling operation and notes that the input point will be doubled either in the first (if $k_0 = 0$) or in the second iteration of the loop (if $k_0 = 1$). If k is fixed he can easily decide between the two by inputting different points, since if $k_0 = 1$ he will see the common operation $2\mathcal{O}$. If k is not fixed, he simply measures the first two iterations and again uses the operation $2\mathcal{O}$ if the template generator should use the first or second iteration. Once he is able to obtain clear traces, the attack itself follows the general description of an OTA.

Montgomery Ladder The main observation that makes OTA attacks applicable to the Montgomery ladder is that at least one of the computations, namely the doubling in the main loop, directly depends on the key bit k_i . For example, if it is assumed that the first three bits of the key are 100, then the output of the first iteration will be $R_0 = 2\mathcal{P}$. If it is assumed that the first bits are 110, then the output of the first iteration will be $R_0 = 3\mathcal{P}$. Therefore, if the attacker compares the pattern of the output of the first iteration of Algorithm 3 with scalar $k = 100$, he will observe a higher correlation with the pattern of $R_0 = 2\mathcal{P}$ than with the pattern of $R_0 = 3\mathcal{P}$. This is demonstrated in the working example of Table 3.2.

Table 3.2 Two executions of the Montgomery ladder [65]

$k = 100$	$k = 110$
$R_0 = P, R_1 = 2P$	$R_0 = P, R_1 = 2P$
$b = 1 \quad R_1 = 3P, R_0 = 2P$	$b = 0 \quad R_0 = 3P, R_1 = 4P$
$b = 1 \quad R_1 = 5P, R_0 = 4P$	$b = 1 \quad R_1 = 7P, R_0 = 6P$

Side-channel atomicity

Simple atomic algorithms do not offer any protection against online template attacks, because the regularity of point operations does not prevent mounting this sort of attack. The point $2\mathcal{P}$, as an output of the third iteration of Algorithm 4, will produce a power trace with a pattern that is very similar to the trace that would have the point $2\mathcal{P}$ as an input. Therefore, the attack will be the similar to the one described for the binary left-to-right double-and-add-always algorithm; the only difference is that instead of the output of the second iteration of the algorithm, the attacker has to focus on the pattern of the third iteration. In general, when an attacker forms a hypothesis about a certain number of bits of k , the hypothesis will include the point in time where $\mathcal{R}_0$ will contain the predicted value. This means that he would have to acquire a larger target trace to allow all hypotheses to be tested.

Practical results The feasibility and efficiency of OTA is shown in [65] with practical attacks on the double-and-add-always scalar multiplication running on the ATmega163 microcontroller [67] in a smart card. The scalar multiplication algorithm is based on the curve arithmetic of the Ed25519 implementation presented in [68], which is available online at <http://cryptojedi.org/crypto/#avrnacl>. The elliptic curve used in Ed25519 is the twisted Edwards curve $E: -x^2 + y^2 = 1 + dx^2y^2$ with $d = -(121665/121666)$ and base point

$$\mathcal{P} = (15112221349535400772501151409588531511454012693041857206046113283949847762202, \\ 46316835694926478169428394003475163141307993866256225615783033603165251855960).$$

For more details on Ed25519 and this specific curve, see [69, 70]. The whole underlying field and curve arithmetic is the same as in [68]. This means in particular that points are internally represented in extended coordinates as proposed in [71]. In this coordinate system, a point $\mathcal{P} = (x, y)$ is represented as $(X : Y : Z : T)$ with $x = X/Z$, $y = Y/Z$, and $x \cdot y = T/Z$.

Experimental results of OTA with extended projective coordinates of 256 bits, extended projective coordinates with reduced 255-bit input and input points with affine compressed coordinates are presented in [65]. The attack targets the output of the doubling operation and then performs pattern matching based on the Pearson correlation coefficient.

The correct key bit guess ($k_2 = 0$) gives 97 % correlation of the target trace with the template trace for $2\mathcal{P}$. On the other hand, the correlation of the target trace with the template trace for $3\mathcal{P}$ is at most 83 %. These high correlation results hold when one key bit is attacked. For every key bit, the pattern matching will give peaks as in Fig. 3.5. Attacking five bits with one acquisition gives lower numbers for pattern matching for both the correct and the wrong scalar guess, mainly due to the noise that is higher for longer acquisitions. However, the difference between correct and wrong assumptions is still remarkable; correct bit assumptions have 84–88 % matching patterns, while the correlation for the wrong assumptions drops to 50–72 %. To determine the value of one bit, it is thus necessary to compute only one template trace, and decide on the value of the targeted bit depending on whether the

Fig. 3.5 Pattern Matching $2P$ (blue) to target and $3P$ to target (brown) [65]

correlation is above or below a certain threshold (in this case, the threshold can be set to 80 %).

Error-Detection and Correction The idea of Online Template Attacks was extended by Dugardin et al. in [66] with an adaptive template attack on scalar multiplication. The authors propose a generic method to distinguish matching templates and using two templates per key bit, they manage to detect and correct errors for wrong bit assumptions. This fact increases the success rate of the attack significantly compared to the original OTA, reaching 99.8 % when 100 average template traces are used. They also take advantage of the horizontal and vertical leakage, which occurs in the broadly used software implementation of mbedTLS during the modular multiplication of large numbers (256-bit elements).

Classification Algorithms for Template Attacks The fact that the template-building phase in OTA is not necessary, significantly simplifies the process of retrieving the key, leaving the overhead of the attack in the template-matching phase. The template-matching technique used for both OTA papers [65, 66] is based on the Pearson correlation coefficient. In [62], more efficient techniques from the field of Machine Learning are used as distinguishers, and the proposed attack reaches a success rate of 100 % with only 20 template traces per key bit. This work is the first step toward a framework for “automating” the template-matching phase. The attack can be classified as a form of OTA having the same attack model and assumptions. The proposed classification techniques from the field of Machine Learning (k -Nearest Neighbor, Naïve Bayes, SVM) provide an efficient and simplified way to match templates during a Template Attack with very high success rates. A practical application of this attack is demonstrated on the scalar multiplication algorithm for the Brainpool curve BP256r1 implemented in mbedTLS (formerly PolarSSL, version 1.3.7).

3.5 Countermeasures

To prevent first-order DPA attacks [1, 4], it is not sufficient to make the operations time-constant and the power traces indistinguishable. The most common countermeasure applied in ECC implementations is randomization of the secret values. In this way, developers make it more difficult to extract useful information from secret values. This section first covers different types of randomization. Further, we focus specifically on countermeasures against OTA attacks.

3.5.1 Randomization Countermeasures

Scalar randomization Instead of a point multiplication with the scalar k , the blinded scalar k' is used, which is computed as follows:

$$k' = k + \#E \cdot r.$$

Here, $\#E$ is the number of points on the curve and r is a random number [31]. Because kP and $k'P$ always result in the same point on the elliptic curve, this method is effective against first-order DPA attacks when the random number is changed for every execution of the point multiplication.

Projective coordinate randomization In addition to scalar randomization, another countermeasure against DPA attacks on elliptic curve point multiplication is projective coordinate randomization. This countermeasure exploits the fact that the Z -coordinate can be chosen randomly when using projective coordinates [31]. This comes down to choosing a different Z -coordinate for each point multiplication during the conversion of the input point P to projective coordinates.

Base point splitting Using this technique, the scalar multiplication is not performed on the point P , but on the point $P + R$, where R is a random point on the curve. After the point multiplication $k(P + R)$, the value kR is subtracted from the result.

Elliptic curve isomorphism randomization The idea to protect scalar multiplication by transforming a curve through various random morphisms, was initially proposed by Joye and Tymen in [72]. Assume that ϕ is a random isomorphism from $\mathcal{E}_K \rightarrow \mathcal{E}'_K$, which maps $\mathcal{P} \in \mathcal{E}_K \rightarrow \mathcal{P}' \in \mathcal{E}'_K$. Multiplying $\mathcal{P}'$ with k will give $\mathcal{Q}' = [k]\mathcal{P}' \in \mathcal{E}'_K$. With the inverse map ϕ^{-1} we can get back to $\mathcal{Q} = [k]\mathcal{P}$. An attacker needs to know the internal representation of the point in order to perform a successful attack, so if $\mathcal{P}'$ is on a curve that the adversary does not know, he cannot create input points in the correct representation.

3.5.2 OTA Countermeasures

Given that an attacker needs to predict the intermediate state of an algorithm at a given point in time, we can assume that the countermeasures that are used to prevent DPA will also have an effect on the OTA. There are methods for changing the representation of a point, which can prevent OTA and make the result unpredictable to the attacker. Most notably those countermeasures are randomizing the projective representation of points and randomizing the coordinates through a random field isomorphism as described in [73]. However, inserting a point in affine coordinates and changing to (deterministic) projective coordinates during the execution of the scalar multiplication (compressing and decompressing of a point), does not affect the OTA type of attack, as it is shown with practical experiments in [65].

References

1. Kocher P, Jaffe J, Jun B. Differential power analysis. In: Wiener M, editor. *Advances in cryptology CRYPTO '99*, vol. 1666. LNCS, Springer; 1999. p. 388–97.
2. Gandolfi K, Moutel C, Olivier F. Electromagnetic analysis: concrete results. In: *Proceedings of third international work-shop, Cryptographic hardware and embedded systems—CHES 2001*, Paris, France, May 14–16, 2001. Generators; 2001. p. 251–61. doi:[10.1007/3-540-44709-1_21](https://doi.org/10.1007/3-540-44709-1_21). http://dx.doi.org/10.1007/3-540-44709-1_21.
3. Quisquater J-J, Samyde D. Electro magnetic analysis (EMA): measures and counter-measures for smart cards. In: *Proceedings of the international conference on research in smart cards: smart card programming and security. E-SMART '01*. London, UK, UK: Springer; 2001. p. 200–10. ISBN:3-540-42610-8. <http://dl.acm.org/citation.cfm?id=646803.705980>.
4. Kocher PC. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Kobitz N, editor. *Advances in cryptology CRYPTO '96*, vol. 1109. LNCS, Springer; 1996. p. 104–13.
5. Schramm K, Wollinger T, Paar C. A new class of collision attacks and its application to DES. In: Johansson T, editor. *Fast software encryption*, vol. 2887. LNCS, Springer; 2003. p. 206–22.
6. Miller VS. Use of elliptic curves in cryptography. In: Williams HC, editor. *Proceedings of advances in cryptology—CRYPTO '85*, Santa Barbara, California, USA, August 18–22, 1985, vol. 218. Lecture notes in computer science. Springer; 1985. p. 417–26. ISBN:3-540-16463-4. doi:[10.1007/3-540-39799-X_31](https://doi.org/10.1007/3-540-39799-X_31). http://dx.doi.org/10.1007/3-540-39799-X_31.
7. Kobitz N. Elliptic curve cryptosystems. *Math Comput*. 1987;48:203–9.
8. Blake NSI, Seroussi G. *Advances in elliptic curve cryptography*, vol. 317. Cambridge University Press; 1999.
9. ANSI-X9.62. Public key cryptography for the financial services industry: the elliptic curve digital signature algorithm (ECDSA), 1998.
10. ANSI-X9.63. Public key cryptography for the financial services industry: key agreement and key transport using elliptic curve cryptography, 1998.
11. BSI. RFC 5639—Elliptic curve cryptography (ECC) brainpool standard curves and curve generation. Technical report Bundesamt für Sicherheit in der Informationstechnik (BSI), 2010.
12. NIST. FIPS Publication 186-4—Digital signature standard (DSS). Tech. rep. National Institute of Standards and Technology (NIST), 2013. <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>.
13. Bosma W, Lenstra H. Complete systems of two addition laws for elliptic curves. *J Number Theory* 1995;53(2):229–40. ISSN:0022-314X. <http://dx.doi.org/10.1006/jnth.1995.1088>. <http://www.sciencedirect.com/science/article/pii/S0022314X85710888>.
14. Renes J, Costello C, Batina L. Complete addition formulas for prime order elliptic curves. In: Fischlin M, Coron J-S, editors. *Proceedings of progress in cryptology EUROCRYPT 2016* (35th international conference on cryptology in Europe, Vienna, Austria, May 8–12, 2016), vol. 9665. LNCS, Springer. p. 403–28.
15. Edwards HM. A normal form for elliptic curves. In: Koç ÇK, Paar C, editors. *Bulletin of the American mathematical society*, vol. 44. 2007. p. 393–422. <http://www.ams.org/journals/bull/2007-44-03/S0273-0979-07-01153-6/home.html>.
16. Bernstein DJ, Lange T. A complete set of addition laws for incomplete Edwards curves. In: *IACR cryptology ePrint archive* 2009. p. 580. <http://eprint.iacr.org/2009/580>.
17. Bernstein DJ, Lange T. Faster addition and doubling on elliptic curves. In: Kurosawa K, editor. *Advances in cryptology ASIACRYPT 2007*, vol. 4833. LNCS, Springer; 2007. p. 29–56. <http://cr.yp.to/papers.html#newelliptic>.
18. Bernstein DJ, Birkner P, Joye M, Lange T, Peters C. Twisted edwards curves. In: Vaudenay S, editor. *Progress in cryptology AFRICACRYPT 2008*, vol. 5023. LNCS, Springer; 2008. p. 389–405. <http://cr.yp.to/papers.html/#twisted>.
19. Bernstein DJ, Lange T. Explicit formulas database. <http://www.hyperelliptic.org/EFD/>.
20. Montgomery PL. Speeding the pollard and elliptic curve methods of factorization. *Math Comput*. 1987;48(177):243–64.

21. Stam M. On montgomery-like representations for elliptic curves over GF(2k). In: Desmedt YG, editor. Proceedings of public key cryptography PKC 2003: 6th international workshop on practice and theory in public key cryptography Miami, FL, USA, January 6–8, 2003. Berlin, Heidelberg: Springer; 2002. p. 240–54. ISBN: 978-3-540-36288-3. doi:[10.1007/3-540-36288-6_18](https://doi.org/10.1007/3-540-36288-6_18). http://dx.doi.org/10.1007/3-540-36288-6_18.
22. Izu T, Möller B, Takagi T. Improved elliptic curve multiplication methods resistant against side channel attacks. In: Progress in cryptology—INDOCRYPT 2002, third international conference on cryptology in India, Hyderabad, India, December 16–18, 2002. p. 296–313. doi:[10.1007/3-540-36231-2_24](https://doi.org/10.1007/3-540-36231-2_24). http://dx.doi.org/10.1007/3-540-36231-2_24.
23. Okeya K, Kurumatani H, Sakurai K. Elliptic curves with the montgomery-form and their cryptographic applications. In: Proceedings public key cryptography, third international workshop on practice and theory in public key cryptography, PKC 2000, Melbourne, Victoria, Australia, January 18–20, 2000. p. 238–57. doi:[10.1007/978-3-540-46588-1_17](https://doi.org/10.1007/978-3-540-46588-1_17). http://dx.doi.org/10.1007/978-3-540-46588-1_17.
24. Bernstein DJ, Chuengsatiansup C, Kohel D, Lange T. Twisted hessian curves. In: Proceedings of progress in cryptology—LATINCRYPT 2015—4th international conference on cryptology and information security in Latin America, Guadalajara, Mexico, August 23–26, 2015. p. 269–94. doi:[10.1007/978-3-319-22174-8_15](https://doi.org/10.1007/978-3-319-22174-8_15). http://dx.doi.org/10.1007/978-3-319-22174-8_15.
25. Hisil H, Wong KK-H, Carter G, Dawson E. Faster group operations on elliptic curves. In: Brankovic L, Susilo W, editors. Seventh Australasian information security conference (AISC 2009), vol. 98. CRPIT. Wellington, New Zealand: ACS; 2009. p. 7–19.
26. Joye M, Quisquater J. Hessian elliptic curves and side-channel attacks. In: Proceedings of Cryptographic hardware and embedded systems—CHES 2001, third international workshop, Paris, France, May 14–16, 2001. Generators, 2001. p. 402–10. doi:[10.1007/3-540-44709-1_33](https://doi.org/10.1007/3-540-44709-1_33). http://dx.doi.org/10.1007/3-540-44709-1_33.
27. Farashahi RR, Joye M. Efficient arithmetic on hessian curves. In: Proceedings of Public key cryptography—PKC 2010, 13th international conference on practice and theory in public key cryptography, Paris, France, May 26–28, 2010. p. 243–60. doi:[10.1007/978-3-642-13013-7_15](https://doi.org/10.1007/978-3-642-13013-7_15). http://dx.doi.org/10.1007/978-3-642-13013-7_15.
28. Cohen H, Miyaji A, Ono T. Efficient elliptic curve exponentiation using mixed coordinates. In: Ohta K, Pei D, editors. Proceedings of advances in cryptology—ASIACRYPT '98, international conference on the theory and applications of cryptology and information security, Beijing, China, October 18–22, 1998. Lecture notes in computer science, vol. 1514. Springer; 1998. p. 51–65. ISBN: 3-540-65109-8. doi:[10.1007/3-540-49649-1_6](https://doi.org/10.1007/3-540-49649-1_6). http://dx.doi.org/10.1007/3-540-49649-1_6.
29. Coron J. Resistance against differential power analysis for elliptic curve cryptosystems. In: Koç ÇK, Paar C, editors. Proceedings of cryptographic hardware and embedded systems, first international workshop, CHES '99, Worcester, MA, USA, August 12–13, 1999. Lecture notes in computer science, vol. 1717. Springer; 1999. p. 292–302. ISBN: 3-540-66646-X. doi:[10.1007/3-540-48059-5_25](https://doi.org/10.1007/3-540-48059-5_25). http://dx.doi.org/10.1007/3-540-48059-5_25.
30. Rivain M. Fast and regular algorithms for scalar multiplication over elliptic curves. In: IACR Cryptology ePrint Archive, 2011. p. 338. <http://eprint.iacr.org/2011/338>.
31. Coron J-S. Resistance against differential power analysis for elliptic curve cryptosystems. In: Koç ÇK, Paar C, editors. Cryptographic hardware and embedded systems CHES'99, vol. 1717. LNCS, Springer; 1999. p. 292–302. <http://saluc.engr.uconn.edu/refs/sidechannel/coron99resistance.pdf>.
32. Joye M. Highly regular right-to-left algorithms for scalar multiplication. In: Paillier P, Verbauwhede I, editors. Cryptographic hardware and embedded systems CHES 2007, vol. 4727. LNCS, Springer; 2007. p. 135–47.
33. Joye M, Yen S. The montgomery powering ladder. In: Kaliski BS, Koç ÇK, Paar C, editors. Cryptographic hardware and embedded systems CHES 2002, vol. 2523. LNCS, Springer; 2002. p. 291–302.
34. Chevallier-Mames B, Ciet M, Joye M. Low-cost solutions for preventing simple sidechannel analysis: side-channel atomicity. IEEE Trans Comput. 2004;53(6):760–8.

35. Benger N, van de Pol J, Smart NP, Yarom Y. Ooh aah... just a little bit: a small amount of side channel can go a long way. In: Proceedings of cryptographic hardware and embedded systems—CHES 2014—16th international workshop, Busan, South Korea, September 23–26, 2014. p. 75–92. doi:[10.1007/978-3-662-44709-3_5](https://doi.org/10.1007/978-3-662-44709-3_5). http://dx.doi.org/10.1007/978-3-662-44709-3_5.
36. Römer T, Seifert J. Information leakage attacks against smart card implementations of the elliptic curve digital signature algorithm. In: Attali I, Jensen T, editors. Smart card programming and security, vol. 2140. LNCS, Springer; 2001. p. 211–19.
37. Fouque P-A, Valette F. The doubling attack why upwards is better than downwards. In: Walter CD, Koç ÇK, Paar C, editors. Cryptographic hardware and embedded systems CHES 2003, vol. 2779. LNCS, Springer; 2003. p. 269–80.
38. Walter CD. Sliding windows succumbs to big mac attack. In: Koç ÇK, Naccache D, Paar C, editors. Cryptographic hardware and embedded systems CHES 2001, vol. 2162. LNCS, Springer; 2001. p. 286–99.
39. Yen S, Ko L, Moon S, Ha J. Relative doubling attack against montgomery ladder. In: Won DH, Kim S, editors. Information security and cryptology ICISC 2005, vol. 3935. LNCS, Springer; 2005. p. 117–28.
40. Homma N, Miyamoto A, Aoki T, Satoh A, Shamir A. Collision-based power analysis of modular exponentiation using chosen-message pairs. In: Oswald E, Rohatgi P, editors. Cryptographic hardware and embedded systems—CHES 2008, vol. 5154. LNCS, Springer; 2008. p. 15–29.
41. Bauer A, Jaulmes É, Prouff E, Wild J. Horizontal collision correlation attack on elliptic curves. In: Lange T, Lauter K, Lisonek P, editors. Selected areas in cryptography, vol. 8282. LNCS, Springer; 2014. p. 553–70.
42. Medwed M, Oswald E. Template attacks on ECDSA. In: Chung K-I, Sohn K, Yung M, editors. Information security applications, vol. 5379. LNCS, Springer; 2009. p. 14–27.
43. Mulder ED, Hutter M, Marson ME, Pearson P. Using bleichenbacher’s solution to the hidden number problem to attack nonce leaks in 384-Bit ECDSA. In: Bertoni G, Coron J-S, editors. Cryptographic hardware and embedded systems CHES 2013, vol. 8086. LNCS, Springer; 2013. p. 435–52. https://online.tugraz.at/tug_online/voe_main2.getvolltext?pCurrPk=71281.
44. Hanley N, Kim H, Tunstall M. Exploiting collisions in addition chain-based exponentiation algorithms using a single trace. Cryptology ePrint Archive, Report 2012/485.2012.
45. Wenger E, Korak T, Kirschbaum M. Analyzing side-channel leakage of RFID suitable lightweight ECC hardware. In: Hutter M, Schmidt J-M, editors. Radio frequency identification, vol. 8262. LNCS, Springer; 2013. p. 128–44.
46. Clavier C, Feix B, Gagnerot G, Roussellet M, Verneuil V. Horizontal correlation analysis on exponentiation. In: Soriano M, Qing S, Lopez J, editors. Information and communications security, vol. 6476. LNCS, Springer; 2010. p. 46–61.
47. Bauer A, Jaulmes É, Prouff E, Wild J. Horizontal and vertical side-channel attacks against secure RSA implementations. In: Proceedings topics in cryptography—CT-RSA 2013—the cryptographers’ track at the RSA conference 2013, San Francisco, CA, USA, February 25–March 1, 2013. p. 1–17. doi:[10.1007/978-3-642-36095-4_1](https://doi.org/10.1007/978-3-642-36095-4_1). http://dx.doi.org/10.1007/978-3-642-36095-4_1.
48. Witteman M, van Woudenberg J, Menarini F. Defeating RSA multiply-always and message blinding countermeasures. In: Kiayias A, editor. Topics in cryptography CT-RSA 2011, vol. 6558. LNCS, Springer; 2011. p. 77–88.
49. Clavier C, Feix B, Gagnerot G, Giraud C, Roussellet M, Verneuil V. ROSETTA for single trace analysis. In: Galbraith S, Nandi M, editors. Progress in cryptology INDOCRYPT 2012, vol. 7668. LNCS, Springer; 2012. p. 140–55.
50. Bauer A, Jaulmes É. Correlation analysis against protected SFM implementations of RSA. In: Proceedings progress in cryptography—INDOCRYPT 2013—14th international conference on cryptology in India, Mumbai, India, December 7–10, 2013. p. 98–115. doi:[10.1007/978-3-319-03515-4_7](https://doi.org/10.1007/978-3-319-03515-4_7). http://dx.doi.org/10.1007/978-3-319-03515-4_7.
51. Feix B, Roussellet M, Venelli A. Side-channel analysis on blinded regular scalar multiplications. Cryptology ePrint Archive, Report 2014/191. <http://eprint.iacr.org/2014>.

52. Chari S, Rao JR, Rohatgi P. Template attacks. In: Cryptographic hardware and embedded systems—CHES 2002, 4th international workshop, Redwood Shores, CA, USA, August 13–15, 2002, Revised Papers. 2002. p. 13–28. doi:[10.1007/3-540-36400-5_3](https://doi.org/10.1007/3-540-36400-5_3). http://dx.doi.org/10.1007/3-540-36400-5_3.
53. Rechberger C, Oswald ME. Practical template attacks. In: Lim CH, Yung M, editors. Information security applications, vol. 3325. Lecture notes in computer science. Springer; 2004. p. 440–56.
54. Mulder ED, Buysschaert P, Örs SB, Delmotte P, Preneel B, Vandenbosch G, Verbauwheide I. Electromagnetic analysis attack on an FPGA implementation of an elliptic curve cryptosystem. In: IEEE international conference on computer as a tool. Belgrade, Serbia & Montenegro; 2005. p. 1879–82. doi:[10.1109/EURCON.2005.1630348](https://doi.org/10.1109/EURCON.2005.1630348). <http://www.sps.ele.tue.nl/members/m.j.bastiaans/spc/demulder.pdf>.
55. Avanzi RM. Generic algorithms for computing discrete logarithms. In: Handbook of elliptic and hyperelliptic curve cryptography, 2005. p. 476–94. doi:[10.1201/9781420034981.pt5](https://doi.org/10.1201/9781420034981.pt5). <http://dx.doi.org/10.1201/9781420034981.pt5>.
56. Pollard JM. Kangaroos, monopoly and discrete logarithms. J. Crypt. 2000;13(4):437–47. doi:[10.1007/s001450010010](https://doi.org/10.1007/s001450010010). <http://dx.doi.org/10.1007/s001450010010>.
57. Lange T, van Vredendaal C, Wakker M. Kangaroos in side-channel attacks. In: Smart card research and advanced applications—13th international conference, CARDIS 2014, Paris, France, November 5–7, 2014. Revised selected papers. 2014. p. 104–21. doi:[10.1007/978-3-319-16763-3_7](https://doi.org/10.1007/978-3-319-16763-3_7). http://dx.doi.org/10.1007/978-3-319-16763-3_7.
58. Heyszl J, Ibing A, Mangard S, Santis FD, Sigl G. Clustering algorithms for non-profiled single-execution attacks on exponentiations. In: Smart card research and advanced applications—12th international conference, CARDIS 2013, Berlin, Germany, November 27–29, 2013. Revised Selected papers. 2013. p. 79–93. doi:[10.1007/978-3-319-08302-5_6](https://doi.org/10.1007/978-3-319-08302-5_6). http://dx.doi.org/10.1007/978-3-319-08302-5_6.
59. Perin G, Imbert L, Torres L, Maurine P. Attacking randomized exponentiations using unsupervised learning. In: Constructive side-channel analysis and secure design—5th international workshop, COSADE 2014, Paris, France, April 13–15, 2014. Revised selected papers, 2014. p. 144–60. doi:[10.1007/978-3-319-10175-0_11](https://doi.org/10.1007/978-3-319-10175-0_11). http://dx.doi.org/10.1007/978-3-319-10175-0_11.
60. Lerman L, Poussier R, Bontempi G, Markowitch O, Standaert F-X. Template attacks vs. machine learning revisited (and the curse of dimensionality in side-channel analysis). In: Mangard S, Poschmann AY, editors. Constructive side-channel analysis and secure design. Lecture notes in computer science (LNCS). Springer; 2015. p. 20–33.
61. Specht R, Heyszl J, Kleinstauber M, Sigl G. Improving non-profiled attacks on exponentiations based on clustering and extracting leakage from multi-channel high-resolution EM measurements. In: Constructive side-channel analysis and secure design—6th international workshop, COSADE 2015, Berlin, Germany, April 13–14, 2015. Revised selected papers, 2015. p. 3–19. doi:[10.1007/978-3-319-21476-4_1](https://doi.org/10.1007/978-3-319-21476-4_1). http://dx.doi.org/10.1007/978-3-319-21476-4_1.
62. Özgen E, Papachristodoulou L, Batina L. Classification algorithms for template matching. In: IEEE international symposium on hardware oriented security and trust, HOST 2016, McLean, VA, USA; 2016 (to appear).
63. Bramer M. Chapter 3, introduction to classification: naïve bayes and nearest neighbour. In: Principles of data mining, undergraduate topics in computer science. London: Springer; 2013. p. 21–37. ISBN: 978-1-4471-4883-8. doi:[10.1007/978-1-4471-4884-5_3](https://doi.org/10.1007/978-1-4471-4884-5_3). http://dx.doi.org/10.1007/978-1-4471-4884-5_3.
64. Alpaydin E. Chapter 13, kernel machines. In: Introduction to machine learning.
65. Batina L, Chmielewski L, Papachristodoulou L, Schwabe P, Tunstall M. Online template attacks. In: Proceedings progress in cryptology—INDOCRYPT 2014—15th international conference on cryptology in India, New Delhi, India, December 14–17, 2014. p. 21–36.
66. Dugardin M, Papachristodoulou L, Najm Z, Batina L, Danger J, Guilley S. Dismantling real-world ECC with horizontal and vertical template attacks. In: Constructive side-channel analysis and secure design—7th international workshop, COSADE 2016, Graz, Austria, April 14–15, 2016 (to appear).

67. Corporation A. ATMEL AVR32UC technical reference manual. ARM Doc Rev.32002F, 2010. <http://www.atmel.com/images/doc32002.pdf>.
68. Hutter M, Schwabe P. NaCl on 8-bit AVR microcontrollers. In: Youssef A, Nitaj A, editors. Progress in cryptology AFRICACRYPT 2013, vol. 7918. LNCS, Springer; 2013. p. 156–72.
69. Bernstein DJ, Duif N, Lange T, Schwabe P, Yang BY. High-speed high-security signatures. In: Preneel B, Takagi T, editors. Cryptographic hardware and embedded systems CHES 2011, vol. 6917. LNCS. see also full version [14]. Springer; 2011. p. 124–42.
70. Bernstein DJ, Duif N, Lange T, Schwabe P, Yang B-Y. High-speed high-security signatures. J Crypt Eng. 2012;2(2):77–89. <http://cryptojedi.org/papers/#ed25519>, see also short version [13].
71. Hisil H, Wong KK-H, Carter G, Dawson E. Revisited edwards curves. In: Pieprzyk J, editor. Advances in cryptology ASIACRYPT, vol. 5350. LNCS, Springer; 2008. p. 326–43.
72. Joye M, Tymen C. Protections against differential analysis for elliptic curve cryptography. In: Proceedings of Cryptographic hardware and embedded systems—CHES 2001, third international workshop, Paris, France, May 14–16, 2001. Generators, 2001. p. 377–90. doi:10.1007/3-540-44709-1_31. http://dx.doi.org/10.1007/3-540-44709-1_31.
73. Joye M. Smart-card implementation of elliptic curve cryptography and DPA-type attacks. In: Quisquater J-J, Paradinas P, Deswarte Y, Kalam A AE, editors. Smart card research and advanced applications VI, vol. 135. IFIP international federation for information processing. Kluwer Academic Publishers, Springer; 2004. p. 115–25.