An Updated Survey on Secure ECC Implementations: Attacks, Countermeasures and Cost

Junfeng Fan and Ingrid Verbauwhede

Katholieke Universiteit Leuven, ESAT/SCD-COSIC and IBBT
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
{jfan,iverbauwhede}@esat.kuleuven.be

Abstract. Unprotected implementations of cryptographic primitives are vulnerable to physical attacks. While the adversary only needs to succeed in one out of many attack methods, the designers have to consider all the known attacks, whenever applicable to their system, simultaneously. Thus, keeping an organized, complete and up-to-date table of physical attacks and countermeasures is of paramount importance to system designers. This paper summarises known physical attacks and countermeasures on Elliptic Curve Cryptosystems. For implementers of elliptic curve cryptography, this paper can be used as a road map for countermeasure selection in the early design stages.

Keywords: Elliptic curve cryptosystems, side-channel attacks, fault attacks.

1 Introduction

The advent of physical attacks on cryptographic device has created a big challenge for implementers. By monitoring the timing, power consumption, electromagnetic (EM) emission of the device or by inserting faults, adversaries can gain information about internal data or operations and extract the key without mathematically breaking the primitives. With new tampering methods and new attacks being continuously proposed and accumulated, designing a *secure* cryptosystem becomes increasingly difficult. While the adversary only needs to succeed in one out of many attack methods, the designers have to prevent all the applicable attacks simultaneously. Moreover, countermeasures of one attack may surprisingly benefit another attack. As a result, keeping abreast of the most recent developments in the field of implementation attacks and with the corresponding countermeasures is a never ending task.

In this paper we provide a systematic overview of implementation attacks and countermeasures of one specific cryptographic primitive: Elliptic Curve Cryptography (ECC) [32,39]. This survey is an updated version of a previous report [16], and has been influenced by Avanzi's report [2], by the books of Blake et al. [6] and by Avanzi et al. [3]. Due to the space limit, we only give a catalogue-like

D. Naccache (Ed.): Quisquater Festschrift, LNCS 6805, pp. 265–282, 2012.

summary of the known attacks and countermeasures. Implementers can use this paper as a road map. For the details of each attack or protection, we refer the readers to the original papers.

The rest of this paper is organised as follows. Section 2 gives a short introduction about the background of ECC. Section 3 and 4 gives details of known passive and active attacks on ECC, respectively. In Section 6, we discuss known countermeasures and their effectiveness. Section 6 gives several cautionary notes on the use of countermeasures. We conclude the paper in Section 7.

2 Background

We give a brief introduction to Elliptic Curve Cryptography in this section. A comprehensive introduction to ECC can be found in [6,3]. For a thorough summary of power analysis attacks, by far the most popular class of implementation attacks, we refer the reader to [35].

Throughout this paper we assume the notations below are defined as follows:

- $\mathbb{K}$: a finite field ($\mathbb{F}_p$ for prime field and $\mathbb{F}_{2^m}$ for binary field);
- $char(\mathbb{K})$: the characteristic of $\mathbb{K}$;
- $E(a_1, a_2, a_3, a_4, a_6)$: an elliptic curve with coefficients a_1, a_2, a_3, a_4, a_6;
- $P(x, y)$: a point with coordinates (x, y);
- $\mathcal{O}$: point at infinity;
- $E(\mathbb{K})$: a group formed by the points on an elliptic curve E defined over the finite field $\mathbb{K}$;
- $\#E$: the number of points on curve E, i.e. the order of E;
- *weak* curve: a curve whose order does not have big prime divisors;
- the order of point P: the smallest integer r such that $rP = \mathcal{O}$;
- affine coordinates: a point is represented with a two-tuple of numbers (x, y);
- projective coordinates: a point (x, y) is represented as (X, Y, Z), where $x = X/Z, y = Y/Z$;
- Jacobian projective coordinates: a point (x, y) is represented as (X, Y, Z), where $x = X/Z^2, y = Y/Z^3$.

2.1 Elliptic Curve Cryptosystems

An elliptic curve E over a field $\mathbb{K}$ can be defined by a *Weierstrass* equation:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

where $a_1, a_2, a_3, a_4, a_6 \in \mathbb{K}$ and $\Delta \neq 0$. Here Δ is the discriminant of E. A *Weierstrass* equation can be simplified by applying a change of coordinates. If $char(\mathbb{K})$ is not equal to 2 or 3, then E can be transformed to

$$y^2 = x^3 + ax + b \tag{2}$$

where $a, b \in \mathbb{K}$. If $char(\mathbb{K}) = 2$, then E can be transformed to

$$y^2 + xy = x^3 + ax^2 + b \tag{3}$$

if E is non-supersingular.

For cryptographic use, we are only interested in elliptic curves over a finite field. Elliptic curves defined over both prime fields and binary extension fields are used in reality. Given two points, $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$, the sum of P_1 and P_2 is again a point on the same curve under the addition rule. For example, for elliptic curve E over $\mathbb{F}_{2^m}$, one can compute $P_3(x_3, y_3) = P_1 + P_2$ as follows:

$$x_3 = \lambda^2 + a_1\lambda - a_2 - x_1 - x_2$$
$$y_3 = -y_1 - (x_3 - x_1)\lambda - a_1 x_3 - a_3$$

where

$$\lambda = \begin{cases} \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3} & (x_1, y_1) = (x_2, y_2), \\ \frac{y_1 - y_2}{x_1 - x_2} & \text{otherwise.} \end{cases}$$

Algorithm 1. Montgomery powering ladder [40]

Input: $P \in E(\mathbb{F})$ and integer $k = \sum_{i=0}^{l-1} k_i 2^i$.
Output: kP.

1: $R[0] \leftarrow P$, $R[1] \leftarrow 2P$.
2: **for** $i = l - 2$ downto 0 **do**
3: $\quad R[\neg k_i] \leftarrow R[0] + R[1]$, $R[k_i] \leftarrow 2R[k_i]$.
4: **end for**

Return $R[0]$.

2.2 Scalar Multiplication

The set of points (x, y) on E together with the point at infinity form an abelian group. Given a point $P \in E(\mathbb{K})$ and a scalar k, the computation kP is called point multiplication or scalar multiplication. Algorithm 1 shows the Montgomery powering ladder for scalar multiplication. The security of ECC is based on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP), namely, finding out k for two given points P and Q such that $Q = kP$.

3 Passive Attacks

In practice, execution of an Elliptic Curve Scalar Multiplication (ECSM) can leak information of k in many ways. The goal of the attacker is to retrieve the entire bit stream of k [1] using physical attacks. Physical attacks include mainly two types of attacks: Side Channel Analysis (SCA) and Fault Analysis (FA). In this section, we briefly recap the known SCA (also known as passive attacks) on an ECC implementation.

[1] Note that for some scenarios, the attackers only need to recover a few bits of k to break the scheme. For example, Nguyen and Shparlinski [43] have shown that a few bits of k from a couple of signatures are enough to break ECDSA [47].

Most SCA attacks are based on power consumption leakage. Most often, electromagnetic (EM) radiation is considered as an extension of the power consumption leakage and the attacks/countermeasures are applied without change [41]. For the sake of simplicity, we will only mention power traces as the side-channel to describe the known attacks. However, it is important to point out that EM radiation can serve as an better leakage source since radiation measurements can be made locally [18].

3.1 Simple Power Analysis

Simple power analysis (SPA) attacks make use of distinctive key-dependent patterns shown in the power traces [33]. As shown by Coron [14], when double-and-add algorithm is used for a point multiplication, the value of scalar bits can be revealed if the adversary can distinguish between point doubling and point addition from a power trace.

3.2 Template Attacks

A template attack [9] requires access to a fully controllable device, and proceeds in two phases. In the first phase, the profiling phase, the attacker constructs templates of the device. In the second phase, the templates are used for the attack. Medwed and Oswald [37] showed the feasibility of this type of attacks on an implementation of the ECDSA algorithm. In [23] a template attack on a masked Montgomery ladder implementation is presented.

3.3 Differential Power Analysis

Differential power analysis (DPA) attacks use statistical techniques to pry the secret information out of the measurements [33]. DPA sequentially feeds the device with N input points P_i, $i \in \{1, 2, .., N\}$. For each point multiplication, kP_i, a measurement over time of the side-channel is recorded and stored. The attacker then chooses an intermediate value, which depends on both the input point P_i and a small part of the scalar k, and transforms it to a hypothetical leakage value with the aid of a hypothetical leakage model. The attacker then makes a guess of the small part of the scalar. For the correct guess, there will be a correlation between the measurements and the hypothetical leakages. The whole scalar can be revealed incrementally using the same method.

3.4 Comparative Side-Channel Attacks

Comparative SCA [24] resides between a simple SCA and a differential SCA. Two portions of the same or different leakage trace are compared to discover the reuse of values. The first reported attack belonging to this category is the doubling attack [19]. The doubling attack is based on the assumption that even if the attacker does not know what operation is performed, he can detect when the same operations are performed twice. For example, for two point doublings, $2P$ and $2Q$, the attacker may not know what P and Q are, but he can tell if $P = Q$. Comparing two power traces, one for kP and one for $k(2P)$, it is possible to recover all the bits of k.

3.5 Refined Power Analysis

A refined power analysis (RPA) attack exploits the existence of special points: $(x, 0)$ and $(0, y)$. Feeding to a device a point P that leads to a special point $R(0, y)$ (or $R(x, 0)$) at step i under the assumption of processed bits of the scalar will generate exploitable side-channel leakage [21]. Especially, applying randomised projective coordinates, randomised EC isomorphisms or randomised field isomorphisms does not prevent this attack since zero stays after randomization.

3.6 Zero-Value Point Attack

A zero-value point attack (ZPA) [1] is an extension of RPA. Not only considering the points (i.e. $R[1]$ and $R[0]$) generated at step i, a ZPA also considers the value of auxiliary registers. For some special points P, some auxiliary registers will predictably have zero value at step i under the assumption of processed bits of the scalar. The attacker can then use the same procedure of RPA to incrementally reveal the whole scalar.

3.7 Carry-Based Attack

The carry-based attack [18] is designed to attack Coron's first countermeasure (also known as scalar randomisation). Instead of performing kP, Coron suggested to perform $(k + r\#E)P$ where r is a random number. The crucial observation here is that, when adding a random number a to a fixed number b, the probability of generating a carry bit $c = 1$ depends solely on the value of b (the carry-in has negligible impact [18]). If $(k + r\#E)$ is performed with a w-bit adder, where w is the digit size, the attacker can learn k digit by digit from the distribution of the carry bit.

3.8 Address-Bit DPA

The address-bit attack (ADPA) [38] explores the link between the register address and the key. The first ADPA applied to ECC is by Itoh et al. [25]. For example, an implementation of Alg. 1 performs point addition and doubling regardless to the value of the key bit, but the address of the doubled point depends solely on k_i. As a result, k_i can be recovered if the attacker can distinguish between data read from $R[0]$ and from $R[1]$.

4 Fault Attacks

Besides passive side-channel analysis, adversaries can actively disturb the cryptographic devices to derive the secret. Faults on the victim device can be induced with a laser beamer, glitches in clock, a drop of power supply and so on. Readers who are interested in these methods are referred to [34].

In this section, we give a short description of the known fault analysis on ECC. Based on the scalar recovery method, we divide fault attacks on ECC into three categories, namely, safe-error based analysis, weak-curve based analysis and differential fault analysis.

4.1 Safe-Error Analysis

The concept of safe-error was introduced by Yen and Joye in [49, 30]. Two types of safe-error are reported: C safe-error and M safe-error.

C safe-error. The C safe-error attack exploits dummy operations which are usually introduced to achieve SPA resistance. Taking the add-and-double-always algorithms [14, Alg. 1] as an example, the dummy addition in step 3 makes safe-error possible. The adversary can induce temporary faults during the execution of the dummy point addition. If the scalar bit $k_i = 1$, then the final results will be faulty. Otherwise, the final results are not affected. The adversary can thus recover k_i by checking the correctness of the results.

M safe-error. The M safe-error attack exploits the fact that faults in some memory blocks will be cleared. The attack was first proposed by Yen and Joye [49] to attack RSA. However, it also applies to ECSM. Assuming that $R[k_i]$ in Alg. 1 is loaded from memory to registers and overwritten by $2R[k_i]$, then faults in $R[1]$ will be cleared only if $k_i = 1$. By simply checking whether the result is affected or not, the adversary can reveal k_i.

4.2 Weak Curve Based Analysis

In 2000, Biehl et al. [5] described the first weak curve fault attack on an ECC implementation. The key observation is that a_6 in the diffinition of E (Eq.1) is not used in the addition formulae. As a result, the addition formulae for curve E generates correct results for any curve E' that differs from E only in a_6:

$$E' : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a'_6. \tag{4}$$

Thus, the adversary can cheat an ECC processor with a point $P' \in E'(\mathbb{F})$ where E' is a cryptographically *weak* curve. The adversary can then solve ECDLP on E' and find out k.

The method of *moving* a scalar multiplication from a strong curve E to a weak curve E' often requires fault induction. With the help of faults, the adversary makes use of invalid points [5], invalid curves [12] and twist curves [17] to hit a weak curve. These methods are described below.

Invalid point attacks. Invalid point attack lets the scalar multiplication start with a point P' on the weak curve E'. If kP is performed without checking the validity of P, then no faults need to be induced. If the ECC processor does check the validity of P, the adversary will try to change the point P right after the point validation. In order to do so, the attacker should be able to induce a fault at a specific timing.

Invalid curve attacks. Ciet and Joye [12] refined the attack in [5] by loosening the requirements on fault injection. They show that any *unknown* faults, including permanent faults in non-volatile memory or transient faults caused on the bus, in *any* curve parameters, including field representation and curve parameters a_1, a_2, a_3, a_4, may cause the scalar multiplication being performed on a weak curve.

Twist curve based FA. In 2008, Fouque et al. [17] noticed that many cryptographically strong curves have weak twist curves. A scalar multiplication kP not using the y-coordinate gives correct results for point on both the specified curve E and it's quadratic twist, and the result of kP on weak twists can leak k. On an elliptic curve defined over a prime field $\mathbb{F}_p$, a random $x \in \mathbb{F}_p$ corresponds to a point on either E or its twist with probability one half. As a result, a random fault on the x-coordinate of P has a probability of one half to hit a point on the (weak) twist curve.

4.3 Differential FA

The Differential Fault Attack (DFA) uses the difference between the correct results and the faulty results to deduce certain bits of the scalar.

Biehl-Meyer-Müller DFA. Biehl et al. [5] reported the first DFA on an ECSM. We use an right-to-left multiplication algorithm (Alg. 2) to describe this attack. Let Q_i and R_i denote the value of Q and R at the end of the i^{th} iteration, respectively. Let $k(i) = k \text{ div } 2^i$. Let Q'_i be the value of Q if faults have been induced. The attack reveals k from the Most Significant Bits (MSB) to the Least Significant Bits (LSB).

1. Run ECSM once and collect the correct result (Q_{l-1}).
2. Run the ECSM again and induce an one-bit flip on Q_i, where $l - m \leq i < l$ and m is *small*.
3. Note that Q_{l-1}=Q_i+$(k(i)2^i)P$ and Q'_{l-1}=Q'_i+$(k(i)2^i)P$. The adversary then tries all possible $k(i) \in \{0, 1, .., 2^m - 1\}$ to generate Q_i and Q'_i. The correct value of $k(i)$ will result in a $\{Q_i, Q'_i\}$ that have only one-bit difference.

The attack works for the left-to-right multiplication algorithm as well. It also applies if k is encoded with any other deterministic codes such as Non-Adjacent-Form (NAF) and w-NAF. It is also claimed that a fault induced at random moments during an ECSM is sufficient [5].

Sign change FA. In 2006, Blömer et al. [7] proposed the sign change fault (SCF) attack. It attacks implementations where scalar is encoded in Non-Adjacent Form. When using curves defined over the prime field, the sign change of a point

Algorithm 2. Right-To-Left (upwards) binary method for point multiplication

Input: $P \in E(\mathbb{F})$ and integer $k = \sum_{i=0}^{l-1} k_i 2^i$.
Output: kP.

1: $R \leftarrow P$, $Q \leftarrow \mathcal{O}$.
2: **for** $i = 0$ to $l - 1$ **do**
3: If $k_i = 1$ then $Q \leftarrow Q + R$.
4: $R \leftarrow 2R$.
5: **end for**

Return Q.

Table 1. Physical Attacks on Elliptic Curve Cryptography Implementations

Attack	Single Execution	Multiple Executions	Chosen Base Point	Using Output Point	Incremental key Recovery
SPA	√				
DPA		√			√
Template attack †	√				
Doubling attack		√	√		
RPA		√	√		√
ZPA		√	√		√
Carry-based attack		√			
ADPA		√			√
Safe-error attack		√			√
Weak-curve attack	√*	√*		√	√
Differential FA		√		√	√

† Attack is reported to recover only a small number of bits of the scalar.

* It may need more than one trial to hit a weak curve.

implies only a sign change of its y-coordinate. The SCF attack does not force the elliptic curve operations to leave the original group $E(\mathbb{F}_p)$, thus P is always a valid point.

4.4 Summary of Attacks

Physical attacks have different application conditions and complexities. For example, SPA and Template SPA require a single trace, while DPA and ADPA require multiple traces. Besides, some attacks make use of the final results while others don't. These conditions reveal the applicability of each attack and suggest possible protections. Table 1 summarises the attacks and their application conditions.

5 Countermeasures

Many protection methods have been proposed to counteract the reported attacks. However, countermeasures are normally proposed to prevent an implementation from a specific attack. It has been pointed out that a countermeasure against one attack may benefit another one. In this section, we discuss the cross relationship between known attacks and countermeasures. We first give a summary of known countermeasures. The computational overhead of each countermeasure is estimated using a curve that achieves 128-bit security. The Montgomery power ladder without y-coordinates is used as the benchmark.

Table 3 summarises the most important attacks and their countermeasures. The different attacks, grouped into passive attacks, active attacks and combined

attacks are listed column-wise, while each row represents one specific countermeasure. Let A_j and C_i denote the attack in the j^{th} row and countermeasure in the i^{th} column, respectively. The grid (i, j), the cross of the i^{th} column and the j^{th} row, shows the relation between A_j and C_i.

- $\surd$: C_i is an effective countermeasure against A_j.
- $\times$: C_i is attacked by A_j.
- **H**: C_i helps A_j.
- **?**: C_i might be an effective countermeasure against A_j, but the relation between C_i and A_j is unclear or unpublished.
- Blank : C_i and A_j are irrelevant (C_i is very likely not effective against A_j).

It is important to make a difference between $\times$ and *blank*. Here $\times$ means C_i is attacked by A_j, where *blank* means that the use of C_i does not affect the effort or results of A_j at all. For example, scalar randomisation using a 20-bit random number can be attacked by a doubling attack, so we put a $\times$ at their cross. The Montgomery powering ladder is designed to thwart SPA, and it does not make a DPA attack harder or easier, so we leave the cell a *blank*.

Below we discuss each countermeasure and its relation to the listed attacks.

5.1 SPA Countermeasures

Indistinguishable Point Operation Formulae (IPOF) [8]. IPOF try to eliminate the difference between point addition and point doubling. The usage of unified formulae for point doubling and addition [8] is a special case of IPOF. However, even when unified formulae are in use, the implementation of the underlying arithmetic, especially the operations with conditional instructions, may still reveal the type of the point operation (addition or doubling) [48, 46]. When using add-and-double method, the Hamming weight of the secret scalar can be easily leaked.

Double-and-add-always [14]. The *double-and-add-always* algorithm, introduced by Coron, ensures that the sequence of operations during a scalar multiplication is independent of the scalar by inserting of a dummy point additions. Due to the use of dummy operations, it makes C safe-error fault attack possible.

Atomic block [10]. Instead of making the group operations indistinguishable, one can rewrite them as sequences of side-channel atomic blocks that are indistinguishable for simple SPAs.

If dummy atomic blocks are added, then this countermeasure may enable C safe-error attacks. Depending on the implementation, it may also enable M safe-error attack.

Montgomery Powering Ladder. The Montgomery ladder [40, 30] for ECC, shown as Alg. 1, shows protection against SPA since the scalar multiplication is performed with a fixed pattern inherently unrelated to each bit of the scalar.

Table 2. Countermeasures and overhead

Cost estimation: negligible (< 10%), low (10%-50%) and high (> 50%)

Countermeasures	**Target Attacks**	**Computation Overhead**
Indistinguishable Point Operation	SPA	Low
Double-and-add-always	SPA	Low
Atomic block	SPA	Negligible
Montgomery Powering Ladder $^{+y}$	SPA	Low
Montgomery Powering Ladder $^{-y}$	SPA	-
Scalar randomisation	DPA	Low
Random key splitting	DPA	High
Base point blinding	DPA	Negligible
Random projective coordinates	DPA	Negligible
Random EC isomorphism	DPA	Low
Random field isomorphism	DPA	Low
Random register address	ADPA	Low
Point Validation	Invalid Point	Negligible
Curve Integrity Check	Invalid Curve	Negligible
Coherence Check	DFA	Low †
Combined curve check	Sign change	Low
Co-factor multiplication	Small group (RPA)	Negligible

$^{+y}$ Using y-coordinate; $^{-y}$ Not using y-coordinate;
† Depends on the number of coherence checks performed in each ECSM.

It avoids the usage of dummy instructions and also resists the *normal* doubling attack. However, it is attacked by the relative doubling attack proposed by Yen et al. [50]. This attack can reveal the relation between two adjacent secret scalar bits, thereby seriously decreases the number of key candidates.

With Montgomery powering ladder, y-coordinate is not necessary during the scalar multiplication, which prevents sign-change attacks. However, for curves that have weak twist curves, using Montgomery powering ladder without y-coordinate is vulnerable to twist curve attacks.

Joye and Yen pointed out that Montgomery powering ladder may be vulnerable to M safe-error attacks (See [30] for details). They also proposed a modified method that allows to detect faults in both R[0] or R[1].

5.2 DPA Countermeasures

Scalar randomisation [14]. This method blinds the private scalar by adding a multiple of $\#E$. For any random number r and $k' = k + r\#E$, we have $k'P = kP$ since $(r\#E)P = \mathcal{O}$. Coron suggested choosing r to be around 20-bit.

The scalar randomisation method was analysed in [44] and judged weak if implemented as presented. Also, due to the fact that $\#E$ for standard curves has a long run of zeros, the blinded scalar, k', still has a lot of bits unchanged.

It makes the safe-error and sign-change attacks more difficult. On the other hand, it is shown in [18] that the randomisation process leaks the scalar under the carry-based attack. Moreover, as mentioned in [19] the 20-bit random value for blinding the scalar k is not enough to resist the doubling attack.

Base point blinding [14]. This method blinds the point P, such that kP becomes $k(P + R)$. The known value $S = kR$ is subtracted at the end of the computation. The mask S and R are stored secretly in the cryptographic device and updated at each iteration.

It can resist DPA/DEMA as explained in [14]. In [19], the authors conclude that this countermeasure is still vulnerable to the doubling attack since the point which blinds P is also doubled at each execution. This countermeasure makes RPA/ZPA more difficult since it breaks the assumption that the attacker can freely choose the base point (the base point is blinded).

This countermeasure might make the weak-curve based attacks more difficult since the attacker does not know the masking point R. In an attack based on an invalid point, the adversary needs to find out the faulty points P' and $Q' = kP'$. With the point blinding, it seams to be more difficult to reveal either P' or Q'. However, in the case of an invalid curve attack, base point blinding does not make a difference.

While neither blinding the base point or the scalar is effective to prevent the doubling attack, the combined use of them seems to be effective [19].

Random projective coordinates [14]. This method randomizes the homogeneous projective coordinates (X, Y, Z) with a random $\lambda \neq 0$ to $(\lambda X, \lambda Y, \lambda Z)$. The random variable λ can be updated in every execution or after each doubling or addition. This countermeasure is effective against differential SCA. It fails to resist the RPA as zero is not effectively randomized.

Random key splitting [11]. The scalar can be split in at least two different ways: $k = k_1 + k_2$ or $k = \lfloor k/r \rfloor r + (k \bmod r)$ for a random r.

This countermeasure can resist DPA/DEMA attacks since it has a random scalar for each execution. In [19], the authors have already analysed the effectiveness of Coron's first countermeasure against the doubling attack. If we assume that the scalar k is randomly split into two full length scalars, the search space is extended to 2^{81} for a 163-bit k (the birthday paradox applies here). This is enough to resist the doubling attack. It can also help to thwart RPA/ZPA if it is used together with base point randomisation [21, 1, 22]. However, this countermeasure is vulnerable to a carry-based attack if the key is split as follows: choose a random number $r < \#E$, and $k_1 = r$, $k_2 = k - r$.

Random EC isomorphism [29]. This method first applies a random isomorphism of the form $\psi : (x, y) \mapsto (r^2x, r^3y)$ and then proceeds by computing $Q = k \cdot \psi(P)$ and outputting $\psi^{-1}(Q)$.

Table 3. Attacks versus Countermeasures

C1: Indistinguishable Point Operation C2: Double-and-add-always C3: Atomic block
C4: Montgomery Powering Ladder $^{+y}$ C5: Montgomery Powering Ladder $^{-y}$ C6: Scalar randomization
C7: Random key splitting C8: Base point blinding C9: Random projective coordinates
C10: Random EC isomorphism C11: Random field isomorphism C12: Random register address
C13: Point Validation C14: Curve Integrity Check C15: Coherence Check
C16: Combined curve check C17: Co-factor multiplication

Attacks		**Countermeasures**																
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
Passive Attacks	SPA	√	√	√	√	√												
	DPA						× [44]	√	× [44]	√	√	√						
	Template SPA						× [37]	**?**	× [37]	√	√	√						
	Doubing		× [19]		× [50]	× [50]	× [19]	**?**	× [19]	**?**	**?**	**?**						
	RPA						√	√	√	× [21]	× [21]	× [21]						√*
	ZPA						√	√	√	× [1]	× [1]	× [1]						
	Carry-based						× [18]	×‡										
	ADPA												√					
Active Attacks	C Safe-error		**H** [49]		√	√												
	M Safe-error				√	√												
	Invalid Point								**?**					√				
	Invalid Curve														√			
	Twist Curve				√	**H** [17]								√				
	BMM DFA								**?**					√		√†		
	Sign change				**H** [7]	√			**?**								√	

† The countermeasures is effective only when the Montgomery powering ladder is used.
* The countermeasures is effective only when the attacker makes use of points of small order.
‡ C7 can be attacked if it splits the k as follows: $k_1 \leftarrow r$, $k_2 \leftarrow k - r$, where r is randomly selected.

Random field isomorphism [29]. This method makes use of isomorphisms between fields. To compute $Q = kP$, it first randomly chooses a field F' isomorphic to F through isomorphism ϕ, then computes $Q = \phi^{-1}(k(\phi(P)))$.

Random EC isomorphism and random field isomorphism have similar strength and weakness as random projective coordinates.

Random register address [26, 27]. This method randomises the register addresses to break the link between key bits and register address. In Alg. 1, the address of the destination register for point doubling is k_i. If k is not randomised, then the attacker can recover k_i with address-bit DPA. May et al. proposed Random Register Renaming (RRR) as a countermeasure on a special processor [36]. Itoh et al. [26] proposed a way to randomise register address for double-and-add-always, Montgomery powering ladder and window method. Izumi et al. [27] showed that the MPL version is still vulnerable and proposed an improved version.

5.3 FA Countermeasures

Point Validation [5, 12]. Point Validation (PV) verifies if a point lies on the specified curve or not. PV should be performed before and after scalar multiplication. If the base point or result does not belong to the original curve, no output should be given. It is an effective countermeasure against invalid point attacks and BMM differential fault attacks. If the y-coordinate is used, it is also effective against a twist-curve attack.

Curve Integrity Check [12]. The curve integrity check is to detect faults on curve parameters. Before starting an ECSM the curve parameters are read from the memory and verified using an error detecting code (i.e. cyclic redundancy check) before an ECSM execution. It is an effective method to prevent invalid curve attacks.

Coherence Check [20]. A coherence check verifies the intermediate or final results with respect to a valid pattern. If an ECSM uses the Montgomery powering ladder, we can use the fact that the difference between $R[0]$ and $R[1]$ is always P. This can be used to detect faults during an ECSM [15].

Combined curve check [7]. This method uses a reference curve to detect faults. This countermeasure makes use of two curves: a reference curve $E_t := E(\mathrm{F}_t)$ and a combined curve E_{pt} that is defined over the ring Z_{pt}. In order to compute kP on curve E, it first generate a combined point P_{pt} from P and a point $P_t \in E_t(\mathrm{F}_t)$ (with prime order). Two scalar multiplications are then performed: $Q_{pt} = kP_{pt}$ on E_{pt} and $Q_t = kP_t$ on E_t. If no error occurred, Q_t and $Q_{pt} \pmod t$ will be equal. Otherwise, the one of the results is faulty and the results should be aborted. It is an effective countermeasure against sign-change fault attack.

Co-factor multiplication [45]. To prevent small subgroup attacks, most protocols can be reformulated using cofactor multiplication. For instance, the Diffie-Hellman protocol can be adapted as follows: a user first computes $Q = h \cdot P$ and then $R = k \cdot Q$ if $Q \neq O$.

This method is an effective countermeasures against Goubin's RPA if the exploited special points are of small order. However, it does not provide protection against ZPA (since it does not necessarily use points of small order) and the combined attack.

6 Some Cautionary Notes

In this section, we discuss several issues on the selection and implementation of countermeasures.

6.1 On the Magic of Randomness

As shown in Table 3, adding randomness into data, operation and address serves as a primary method to prevent differential power (and some fault analysis). One underlying assumption of randomisation is that only a few bits of the scalar are leaked from each (randomised) execution, and these pieces of information can not be aggregated. In other words, since DPA (or DFA) recover the scalar incrementally, multiple (randomised) executions do not leak more bits of k than one execution. However, the history has shown that randomness may not work as good as expected. A good example is the use of a Hidden Markov Model (HMM) to analyze Oswald-Aigner randomised exponentiation [31] and random scalar splitting [42]. Another example is the horizontal analysis [13] that uses only a single trace. It is not clear whether there is an efficient and general aggregation algorithm to break randomised executions. However, randomness as a protection to DPA (and DFA) should definitely be used with caution.

6.2 Countermeasure Selection

While unified countermeasures to tackle both the passive and active attacks are attractive, they are very likely weaker than what is expected. Baek and Vasyltsov extended Shamir's trick, which was proposed for RSA-CRT, to secure ECC from DPA and FA [4]. However, Joye showed in [28] that a non-negligible portion of faults was undetected using the unified countermeasure and settings in [4].

For the selection of countermeasures, we believe three principles should be followed: Complete, Specific and Additive.

Complete: An adversary needs to succeed in only one out of many possible attack methods to win, but the implementation has to be protected from all applicable attacks.

Specific: For an ECC processor designed for a specific application, normally not all the attacks are applicable. For example, RPA and ZPA is not applicable if an ECC processor is designed solely for ECDSA since the base point is fixed.

Additive: The combination of two perfect countermeasures may introduce new vulnerabilities. Therefore, selected countermeasures should be evaluated to make sure they are additive.

6.3 Implementation Issues

An obvious yet widely ignored fact is that the implementing process (coding in software or hardware) may also introduce vulnerabilities. For instance, an implementation of Montgomery powering ladder will inevitably use registers or memory entries for intermediate results. These temporary memory entries are not visible on the algorithm level, and safe-errors may be introduced in those memory locations. In order to avoid vulnerabilities introduced during the implementation process, an systematic analysis at the each representation level (from C to netlist) should be performed.

7 Conclusion

In this paper we give a systematic overview of the existing implementation attacks and countermeasures on ECC. While we have no intentions to provide new countermeasures, we do give a complete overview of a wide range of attacks and the common classes of countermeasures. We strongly believe that keeping track of the ever evolving field of implementation attacks is of crucial importance to a cryptosystem designer. This paper provides a digest of existing attacks and countermeasures, and Table 3 can be used for countermeasures selection during the early design stages.

Acknowledgement. This work was supported in part by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy), by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II and ICT-2009-238811 (UNIQUE), by the Research Council K.U.Leuven: GOA TENSE and by IBBT.

References

1. Akishita, T., Takagi, T.: Zero-Value Point Attacks Elliptic Curve Cryptosystem. In: Boyd, C., Mao, W. (eds.) ISC 2003. LNCS, vol. 2851, pp. 218–233. Springer, Heidelberg (2003)
2. Avanzi, R.: Side Channel Attacks on Implementations of Curve-Based Cryptographic Primitives. Cryptology ePrint Archive, Report 2005 /017, `http://eprint.iacr.org/`
3. Avanzi, R.M., Cohen, H., Doche, C., Frey, G., Lange, T., Nguyen, K., Vercauteren, F.: Handbook of Elliptic and Hyperelliptic Curve Cryptography. CRC Press, Boca Raton (2005)
4. Baek, Y.-J., Vasyltsov, I.: How to Prevent DPA and Fault Attack in a Unified Way for ECC Scalar Multiplication – Ring Extension Method. In: Dawson, E., Wong, D.S. (eds.) ISPEC 2007. LNCS, vol. 4464, pp. 225–237. Springer, Heidelberg (2007)

5. Biehl, I., Meyer, B., Müller, V.: Differential Fault Attacks on Elliptic Curve Cryptosystems. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 131–146. Springer, Heidelberg (2000)
6. Blake, I., Seroussi, G., Smart, N., Cassels, J.W.S.: Advances in Elliptic Curve Cryptography. London Mathematical Society Lecture Note Series. Cambridge University Press, New York (2005)
7. Blömer, J., Otto, M., Seifert, J.-P.: Sign Change Fault Attacks on Elliptic Curve Cryptosystems. In: Breveglieri, L., Koren, I., Naccache, D., Seifert, J.-P. (eds.) FDTC 2006. LNCS, vol. 4236, pp. 36–52. Springer, Heidelberg (2006)
8. Brier, E., Joye, M.: Weierstraß Elliptic Curves and Side-Channel Attacks. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 335–345. Springer, Heidelberg (2002)
9. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 13–28. Springer, Heidelberg (2003)
10. Chevallier-Mames, B., Ciet, M., Joye, M.: Low-Cost Solutions for Preventing Simple Side-Channel Analysis: Side-Channel Atomicity. IEEE Trans. Computers 53(6), 760–768 (2004)
11. Ciet, M., Joye, M.: (Virtually) Free Randomization Techniques for Elliptic Curve Cryptography. In: Qing, S., Gollmann, D., Zhou, J. (eds.) ICICS 2003. LNCS, vol. 2836, pp. 348–359. Springer, Heidelberg (2003)
12. Ciet, M., Joye, M.: Elliptic Curve Cryptosystems in the Presence of Permanent and Transient Faults. Des. Codes Cryptography 36(1), 33–43 (2005)
13. Clavier, C., Feix, B., Gagnerot, G., Roussellet, M., Verneuil, V.: Horizontal Correlation Analysis on Exponentiation. In: Soriano, M., Qing, S., López, J. (eds.) ICICS 2010. LNCS, vol. 6476, pp. 46–61. Springer, Heidelberg (2010)
14. Coron, J.: Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. In: Koç, Ç.K., Paar, C. (eds.) CHES 1999. LNCS, vol. 1717, pp. 292–302. Springer, Heidelberg (1999)
15. Dominguez-Oviedo, A.: On Fault-based Attacks and Countermeasures for Elliptic Curve Cryptosystems. PhD thesis, University of Waterloo, Canada (2008)
16. Fan, J., Guo, X., De Mulder, E., Schaumont, P., Preneel, B., Verbauwhede, I.: State-of-the-art of Secure ECC Implementations: A Survey on Known Side-channel Attacks and Countermeasures. In: HOST, pp. 76–87. IEEE Computer Society, Los Alamitos (2010)
17. Fouque, P., Lercier, R., Réal, D., Valette, F.: Fault Attack on Elliptic Curve Montgomery Ladder Implementation. In: Fifth International Workshop on Fault Diagnosis and Tolerance in Cryptography - FDTC, pp. 92–98 (2008)
18. Fouque, P., Réal, D., Valette, F., Drissi, M.: The Carry Leakage on the Randomized Exponent Countermeasure. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 198–213. Springer, Heidelberg (2008)
19. Fouque, P.-A., Valette, F.: The Doubling Attack – Why Upwards Is Better than Downwards. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 269–280. Springer, Heidelberg (2003)
20. Giraud, C.: An RSA Implementation Resistant to Fault Attacks and to Simple Power Analysis. IEEE Trans. Computers 55(9), 1116–1120 (2006)
21. Goubin, L.: A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 199–210. Springer, Heidelberg (2002)

22. Ha, J., Park, J., Moon, S., Yen, S.: Provably Secure Countermeasure Resistant to Several Types of Power Attack for ECC. In: Kim, S., Yung, M., Lee, H.-W. (eds.) WISA 2007. LNCS, vol. 4867, pp. 333–344. Springer, Heidelberg (2008)
23. Herbst, C., Medwed, M.: Using Templates to Attack Masked Montgomery Ladder Implementations of Modular Exponentiation. In: Chung, K.-I., Sohn, K., Yung, M. (eds.) WISA 2008. LNCS, vol. 5379, pp. 1–13. Springer, Heidelberg (2009)
24. 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. (eds.) CHES 2008. LNCS, vol. 5154, pp. 15–29. Springer, Heidelberg (2008)
25. Itoh, K., Izu, T., Takenaka, M.: Address-Bit Differential Power Analysis of Cryptographic Schemes OK-ECDH and OK-ECDSA. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 129–143. Springer, Heidelberg (2003)
26. Itoh, K., Izu, T., Takenaka, M.: A Practical Countermeasure against Address-Bit Differential Power Analysis. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 382–396. Springer, Heidelberg (2003)
27. Izumi, M., Ikegami, J., Sakiyama, K., Ohta, K.: Improved countermeasure against Address-bit DPA for ECC scalar multiplication. In: DATE, pp. 981–984. IEEE, Los Alamitos (2010)
28. Joye, M.: On the security of a unified countermeasure. In: FDTC 2008: Proceedings of the 5th Workshop on Fault Diagnosis and Tolerance in Cryptography, pp. 87–91. IEEE Computer Society, Los Alamitos (2008)
29. Joye, M., Tymen, C.: Protections against Differential Analysis for Elliptic Curve Cryptography. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 377–390. Springer, Heidelberg (2001)
30. Joye, M., Yen, S.-M.: The Montgomery Powering Ladder. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 291–302. Springer, Heidelberg (2003)
31. Karlof, C., Wagner, D.: Hidden Markov Model Cryptanalysis. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 17–34. Springer, Heidelberg (2003)
32. Koblitz, N.: Elliptic Curve Cryptosystem. Math. Comp. 48, 203–209 (1987)
33. Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
34. Kömmerling, O., Kuhn, M.G.: Design principles for tamper-resistant smartcard processors. In: USENIX Workshop on Smartcard Technology – SmartCard 1999, pp. 9–20 (1999)
35. Mangard, S., Oswald, E., Popp, T.: Power analysis Attacks: Revealing the Secrets of Smart Cards. Springer, Heidelberg (2007)
36. May, D., Muller, H.L., Smart, N.P.: Random Register Renaming to Foil DPA. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 28–38. Springer, Heidelberg (2001)
37. Medwed, M., Oswald, E.: Template Attacks on ECDSA. In: Chung, K.-I., Sohn, K., Yung, M. (eds.) WISA 2008. LNCS, vol. 5379, pp. 14–27. Springer, Heidelberg (2009)
38. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Power analysis attacks of modular exponentiation in smartcards. In: Koç, Ç.K., Paar, C. (eds.) CHES 1999. LNCS, vol. 1717, pp. 144–157. Springer, Heidelberg (1999)
39. Miller, V.S.: Use of Elliptic Curves in Cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986)

40. Montgomery, P.L.: Speeding the Pollard and elliptic curve methods of factorization. Mathematics of Computation 48(177), 243–264 (1987)
41. De Mulder, E., Örs, S., Preneel, B., Verbauwhede, I.: Differential power and electromagnetic attacks on a FPGA implementation of elliptic curve cryptosystems. Computers & Electrical Engineering 33(5-6), 367–382 (2007)
42. Muller, F., Valette, F.: High-Order Attacks Against the Exponent Splitting Protection. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 315–329. Springer, Heidelberg (2006)
43. Nguyen, P.Q., Shparlinski, I.: The insecurity of the elliptic curve digital signature algorithm with partially known nonces. Des. Codes Cryptography 30(2), 201–217 (2003)
44. Okeya, K., Sakurai, K.: Power Analysis Breaks Elliptic Curve Cryptosystems even Secure against the Timing Attack. In: Roy, B., Okamoto, E. (eds.) INDOCRYPT 2000. LNCS, vol. 1977, pp. 178–190. Springer, Heidelberg (2000)
45. Smart, N.P.: An Analysis of Goubin's Refined Power Analysis Attack. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 281–290. Springer, Heidelberg (2003)
46. Stebila, D., Thériault, N.: Unified Point Addition Formulæ and Side-Channel Attacks. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 354–368. Springer, Heidelberg (2006)
47. Vanstone, S.: Responses to NIST's proposal. Communications of the ACM 35, 50–52 (1992)
48. Walter, C.D.: Simple Power Analysis of Unified Code for ECC Double and Add. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 191–204. Springer, Heidelberg (2004)
49. Yen, S.M., Joye, M.: Checking Before Output Not Be Enough Against Fault-Based Cryptanalysis. IEEE Trans. Computers 49(9), 967–970 (2000)
50. Yen, S.-M., Ko, L.-C., Moon, S.-J., Ha, J.C.: Relative Doubling Attack Against Montgomery Ladder. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 117–128. Springer, Heidelberg (2006)