J. Cryptol. (2011) 24: 346–374
DOI: 10.1007/s00145-010-9087-5

Journal of
CRYPTOLOGY

Fault-Based Attack on Montgomery's Ladder Algorithm

Agustin Dominguez-Oviedo*
Department of Mechatronics, ITESM Campus Queretaro, Queretaro, Qro, Mexico

M. Anwar Hasan
Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada
ahasan@ece.uwaterloo.ca

Bijan Ansari*
Qualcomm Inc., San Diego, CA, USA

Received 1 September 2009
Online publication 20 October 2010

Abstract. In this paper we present invalid-curve attacks that apply to the Montgomery ladder elliptic curve scalar multiplication (ECSM) algorithm. An elliptic curve over the binary field is defined using two parameters, a and b. We show that with a different "value" for curve parameter a, there exists a cryptographically weaker group in nine of the ten NIST-recommended elliptic curves over $\mathbb{F}_{2^m}$. Thereafter, we present two attacks that are based on the observation that parameter a is not utilized for the Montgomery ladder algorithms proposed by López and Dahab (CHES 1999: Cryptographic Hardware and Embedded Systems, LNCS, vol. 1717, pp. 316–327, Springer, Berlin, 1999). We also present the probability of success of such attacks for general and NIST-recommended elliptic curves. In addition we give some countermeasures to resist these attacks.

Key words. Elliptic curve cryptography, Scalar multiplication, Montgomery ladder, Fault-based attacks.

1. Introduction

In 1996 a fault analysis attack was introduced by Boneh et al. [5]. This attack is based on fault injection in a device performing an RSA [33] or Rabin [32] digital signature as well as some identification protocols such as the Fiat-Shamir [12] and the Schnorr [37] schemes. Biehl et al. [3] proposed the first fault-based attack on elliptic curve cryptography (ECC) [21,26]. Their basic idea is to change the input points, elliptic curve parameters, or the base field in order to perform the operations in a weaker group where solving the elliptic curve discrete logarithm problem (ECDLP) is feasible. A basic assumption for this attack is that one of the two parameters of the governing elliptic curve

* This work was done when Agustín Domínguez-Oviedo and Bijan Ansari were at the University of Waterloo.

equation is not involved for point operations formulas. In this way, the computation could be performed in a cryptographically less secure elliptic curve. Later, Ciet and Joye [7] have shown how to recover the secret key by applying the same principle of invalid curves but using a less restrictive assumption of unknown but fixed faulty input point.

Recently, Fouque et al. [14] proposed a fault attack on the Montgomery ladder implementation elliptic curve scalar multiplication. They showed how to retrieve the secret scalar using elliptic curves defined over prime fields. They based their work on the fact that the y-coordinate is not used for the elliptic curve scalar multiplication (ECSM) and a computation after a fault may leave the original group and be in a twist of the original elliptic curve. A number of protections against active fault attacks have been reported in [3,4,7,15,31,42] and [10]. For a survey of methods of fault analysis attacks on ECC and countermeasures, the reader is referred to [2].

The invalid-curve attacks presented by Biehl et al. [3] and Ciet and Joye [7] apply to applications where the above-mentioned parameter is not used for the group formulas. However, for the Montgomery ladder algorithm [22] used in the ECSM, it is not the case since the parameter is utilized. In this paper we present fault-based attacks that apply to the Montgomery ladder algorithm on curves defined over the binary field. Our work takes advantage of the other parameter of the elliptic curve equation. After a brief review of the Montgomery algorithm, we first present some observations about the NIST-recommended curves over the binary field. Next, we present two invalid-curve-based attacks on the target algorithm. Finally, we present some possible countermeasures to the attacks presented in this paper.

2. Background

2.1. *Montgomery's Ladder Algorithm for ECSM*

Montgomery [27] presented a method to compute multiples of points for a special type of elliptic curve over prime fields. His technique has been generalized to other curves of cryptographic interests [6,22,28]. Also, an extension of this method has been described in the context of modular exponentiation [20]. The attacks presented here apply to the Montgomery ladder algorithm proposed by López and Dahab [22] for elliptic curves defined over the binary field.

The well-known simplified affine form of the Weierstrass equation of non-supersingular elliptic curves over the binary field is [19]

$$y^2 + xy = x^3 + ax^2 + b. \tag{1}$$

The binary double-and-add and the Montgomery ladder algorithms (and their variants) are among the most commonly used schemes for performing ECSM on curves defined by (1). Algorithm 1 below is a description of the Montgomery algorithm [27] in its most basic form.

Algorithm 1 Basic Montgomery's ladder ECSM.

Input: $P \in E(\mathbb{F}_q)$, $k = (k_{t-1} \cdots k_1\ k_0)_2$ with $k_{t-1} = 1$.
Output: $Q = kP$.

1. $Q_0 \leftarrow P$, $Q_1 \leftarrow 2P$.
2. For $i = t-2$ downto 0 do
   2.1 If $(k_i = 0)$ then
      2.1.1 $Q_1 \leftarrow Q_0 \uplus Q_1$, $Q_0 \leftarrow 2Q_0$;
   2.2 Else
      2.2.1 $Q_0 \leftarrow Q_0 \uplus Q_1$, $Q_1 \leftarrow 2Q_1$.
3. Return(Q_0).

In each iteration of the algorithm, the difference between Q_1 and Q_0 is equal to input point P. This fact leads to a formula for the x-coordinate of the sum of two points without their y-coordinates. Below, we present such a formula due to López and Dahab [22].

Let $P = (x, y)$ be the difference between P_1 and P_0, i.e., $P_1 - P_0 = P$. If P is known, then the x-coordinate of $P_0 \uplus P_1$ can be obtained as:

$$\mathrm{x}(P_0 \uplus P_1) = \begin{cases} x_0^2 + \frac{b}{x_0^2} & \text{if } P_0 = P_1, \\ x + \frac{x_0}{x_0 + x_1} + (\frac{x_0}{x_0 + x_1})^2 & \text{if } P_0 \neq P_1. \end{cases} \tag{2}$$

Additionally the y-coordinate of P_0, y_0, can be obtained from $P = (x, y)$, and the x-coordinates of P_0 and P_1 (i.e., x_0 and x_1, respectively) as follows:

$$y_0 = \frac{(x_0 + x)[(x_0 + x)(x_1 + x) + x^2 + y]}{x} + y. \tag{3}$$

As one can clearly see, (2) involves parameter b. As a result, the invalid-curve attacks presented by Biehl et al. [3] and Ciet and Joye [7] do not apply to the Montgomery algorithm.

2.2. *Elliptic Curve Discrete Logarithm Problem (ECDLP)*

The ECDLP is based on the difficulty of obtaining k given P and $Q(= kP)$ for some integer k and P, $Q \in E(\mathbb{F}_q)$ [19]. This principle has led to schemes equivalent to DLP-based cryptosystems, such as Diffie-Hellman key exchange [8], ElGamal public key encryption [11], ElGamal digital signatures [11], and DSA [13].

In practice for the ECDLP to be intractable, it is important to select appropriate domain parameters such as the finite field $\mathbb{F}_q$ where the curve E is defined, the curve E itself, and the base point P. When the order n of the base point P is a large prime, the fastest known algorithms to solve the ECDLP, namely the baby-step giant-step [40] and the Pollard's rho [30] algorithms, need $O(\sqrt{n})$ steps. Consequently, for security purposes it is necessary that the size of the underlying finite field be at least the double of the security level in bits. Security level of L bits is referred to as the best algorithm for breaking the system that takes approximately 2^L steps [19]. For example, for achieving

an 80-bit security level, the cryptosystem would require an elliptic curve defined over a finite field $\mathbb{F}_q$, where $q \approx 2^{160}$. With respect to the selection of the elliptic curve E, some types of curves are avoided for cryptographic applications since the ECDLP can be reduced. These curves include supersingular curves [24], anomalous curves [35,39], and curves over $\mathbb{F}_{2^m}$ for some non-prime values of m [16,18,23].

If the order of the base point P does not contain at least a large prime factor, then it is possible to use an extension for ECC of the Silver–Pohlig–Hellman algorithm [29] to solve the ECDLP as presented in Algorithm 2. This algorithm reduces the problem to subgroups of prime order. Let n be the order of the base point P with a prime factorization $n = \prod_{i=0}^{j-1} p_i^{e_i}$, where $p_i < p_{i+1}$. Suppose that $Q = lP$, where $P,\ Q \in E(\mathbb{F}_q)$ and $l \in [0, n-1]$. This algorithm obtains during the outer loop, the value of $l \bmod p_i^{e_i}$ for each $0 \leq i \leq j-1$. With these values $l \bmod n$ can be uniquely computed using the CRT [25]. It is important to note that at Step 1.3.2 one EC discrete logarithm needs to be computed. However, this operation is in a subgroup at the most of order p_{j-1}. It can be performed with the fastest known algorithms for ECDLP such as the Pollard's rho algorithm with an expected running time of $O(\sqrt{p_m})$, where p_m is the largest prime divisor of $\mathtt{ord}(P_i)$.

Algorithm 2 Silver–Pohlig–Hellman's algorithm for solving the ECDLP.

Input: $P \in E(\mathbb{F}_q)$, $Q \in \langle P \rangle$, $n = \mathtt{ord}(P) = \prod_{i=0}^{j-1} p_i^{e_i}$, where $p_i < p_{i+1}$.
Output: $l \bmod n$.

1. For $i = 0$ to $j-1$ do
   1.1 $Q' \leftarrow \mathcal{O}$, $l_i \leftarrow 0$.
   1.2 $P_i \leftarrow (n/p_i)P$.
   1.3 For $t = 0$ to $(e_i - 1)$ do
      1.3.1 $Q_{t,i} \leftarrow (n/p_i^{t+1})(Q \uplus Q')$.
      1.3.2 $W_{t,i} \leftarrow \log_{P_i} Q_{t,i}$. {ECDLP in a subgroup of order $\mathtt{ord}(P_i)$.}
      1.3.3 $Q' \leftarrow Q' - W_{t,i} p^t P$.
      1.3.4 $l_i \leftarrow l_i + p^t W_{t,i}$.
2. Use the CRT to solve the system of congruences $l \equiv l_i \pmod{p_i^{e_i}}$. This gives us $l \bmod n$.
3. Return(l).

Example 1. Let E be the curve $y^2 + xy = x^3 + 1$ over the field $\mathbb{F}_{2^{11}}$ given by the polynomial $f(z) = z^{11} + z^2 + 1$. Let us represent the elements of $\mathbb{F}_{2^{11}}$ in hexadecimal form. Consider the point $P = (\mathtt{0x10F},\mathtt{0x27A})$ whose order is $n = 92 = 2^2 \cdot 23$. Let $Q = (\mathtt{0x1FB},\mathtt{0x2C6})$. We can use Algorithm 2 to obtain $l = \log_P Q$ as follows.

- During the first loop for $i = 0$ we can obtain $l_0 = l \bmod 2^2$. We can find that $l_0 = W_{0,0} + 2W_{2,0} = 1 + 2 \cdot 0 = 1$.
- For the second loop for $i = 1$ we determine $l_1 = l \bmod 23$. It can be shown that $l_1 = W_{1,0} = 18$.
- Finally we have the following pair of congruences: $l \bmod 4 = 1$ and $l \bmod 23 = 18$. Solving using the CRT we have $l = 41$.

To resist the Silver–Pohlig–Hellman attack one can simply select an elliptic curve E such that its group order, $\#E(\mathbb{F}_{2^m})$, is prime or *almost prime*, i.e., $\#E(\mathbb{F}_{2^m}) = hn$, where n is a prime and h is small [19] (e.g., $h \in [1, 4]$).

3. Parameter a and NIST-Recommended Curves

3.1. *Parameter a*

Theorem 1. *Let E and $\overline{E}$ be non-supersingular elliptic curves defined over $\mathbb{F}_{2^m}$. E and $\overline{E}$ given by the equations*

$$E : y^2 + xy = x^3 + ax^2 + b,$$
$$\overline{E} : y^2 + xy = x^3 + \bar{a}x^2 + \bar{b}$$

are isomorphic *over $\mathbb{F}_{2^m}$ if and only if $\mathrm{Tr}(a) = \mathrm{Tr}(\bar{a})$ and $b = \bar{b}$. If the last conditions are met, then there is an admissible change of variables $(x, y) \to (x, y + tx)$ that converts E into $\overline{E}$ for some $t \in \mathbb{F}_{2^m}^*$ that satisfies $\bar{a} = t^2 + t + a$.*

By Theorem 1 we can state that the number of isomorphism classes for elliptic curves defined by (1) is $2^{m+1} - 2$. The latter comes from the number of possible values for parameter b (i.e., $2^m - 1$) times the possible values of the trace function of parameter a (i.e., 2). With the last observation, for a fixed value of parameter b there are only two isomorphic classes of curves, one for each value of $\gamma \in \{0, 1\}$, where $\mathrm{Tr}(a) = \gamma$. Let us define two representative elliptic curves, E_0 and E_1, one for each of these isomorphic classes:

$$E_0 : y^2 + xy = x^3 + b \quad (a = 0), \tag{4}$$
$$E_1 : y^2 + xy = x^3 + x^2 + b \quad (a = 1). \tag{5}$$

Lemma 1. *Let E_0 and E_1 be two elliptic curves over $\mathbb{F}_{2^m}$ defined by (4) and (5), respectively.*

(i) *The only points that $E_0(\mathbb{F}_{2^m})$ and $E_1(\mathbb{F}_{2^m})$ share are $\mathcal{O}$ and $(0, \sqrt{b})$.*
(ii) *Let $(u, v) \in E_j(\mathbb{F}_{2^m})$, where $u \in \mathbb{F}_{2^m}^*$, $v \in \mathbb{F}_{2^m}$, and $j \in \{0, 1\}$. Then, there does not exist any point in $E_{\overline{j}}(\mathbb{F}_{2^m})$ of the form (u, w) for any $w \in \mathbb{F}_{2^m}$, where $\overline{j} = 1 - j$.*
(iii) *There exist two points of the form (u, v) and $(u, u + v)$ in either $E_0(\mathbb{F}_{2^m})$ or $E_1(\mathbb{F}_{2^m})$ for each $u \in \mathbb{F}_{2^m}^*$ and some $v \in \mathbb{F}_{2^m}$.*
(iv) *The orders of $E_0(\mathbb{F}_{2^m})$ and $E_1(\mathbb{F}_{2^m})$ satisfy the following:*

$$\#E_0(\mathbb{F}_{2^m}) + \#E_1(\mathbb{F}_{2^m}) = 2^{m+1} + 2. \tag{6}$$

Proof. First, if we solve the quadratic expressions resulting from (4) and (5) with $x = 0$, we obtain a unique solution $y = \sqrt{b}$. For $x \neq 0$, (1) has a solution for y if and only if

$$\mathrm{Tr}(x) + \mathrm{Tr}(a) + \mathrm{Tr}\left(\frac{b}{x^2}\right) = 0. \tag{7}$$

Since the only difference between (4) and (5) is the value of parameter a, we can conclude from (7) that if any value of $x \in \mathbb{F}_{2^m}^*$ does not have a solution with $a = j$, then it does with $a = \bar{j}$ for $j = 0$ or 1. Also this equation shows that it is not possible to have a solution for both E_0 and E_1 with the same $x \neq 0$.

Additionally, for a given value of $x \neq 0$ we have two distinct solutions that represent two elliptic curve points (i.e., a point and its negative). To this end, for $x \neq 0$, $\#E_0(\mathbb{F}_{2^m}) + \#E_1(\mathbb{F}_{2^m})$ consider exactly $2^{m+1} - 2$ points on both curves. In addition, the points $\mathcal{O}$ and $(0, \sqrt{b})$ are common and are counted twice in the sum of both orders, bringing the total up to $2^{m+1} + 2$ as shown in (6). □

Example 2. Let us consider $\mathbb{F}_{2^5}$ as represented by the irreducible polynomial $f(z) = z^5 + z^2 + 1$. Let us represent the elements of $\mathbb{F}_{2^5}$ in hexadecimal form. Let E_0 and E_1 be the curves $y^2 + xy = x^3 + 1$ and $y^2 + xy = x^3 + x^2 + 1$, respectively, defined over $\mathbb{F}_{2^5}$. $E_0(\mathbb{F}_{2^5})$ has an order of 44 with the following set of points:

```
{(0x00,0x01),(0x01,0x00),(0x01,0x01),(0x02,0x1F),(0x02,0x1D),(0x03,0x0C),(0x03,0x0F),
 (0x04,0x12),(0x04,0x16),(0x05,0x1A),(0x05,0x1F),(0x07,0x1F),(0x07,0x18),(0x09,0x1D),
 (0x09,0x14),(0x0B,0x16),(0x0B,0x1D),(0x0C,0x05),(0x0C,0x09),(0x0D,0x0B),(0x0D,0x06),
 (0x0F,0x19),(0x0F,0x16),(0x10,0x09),(0x10,0x19),(0x11,0x03),(0x11,0x12),(0x12,0x14),
 (0x12,0x06),(0x15,0x12),(0x15,0x07),(0x17,0x0B),(0x17,0x1C),(0x18,0x0F),(0x18,0x17),
 (0x1A,0x11),(0x1A,0x0B),(0x1B,0x0F),(0x1B,0x14),(0x1C,0x09),(0x1C,0x15),(0x1F,0x06),
 (0x1F,0x19),O}.
```

On the other hand, $E_1(\mathbb{F}_{2^5})$ has an order of 22 with the following set of points:

```
{(0x00,0x01),(0x06,0x10),(0x06,0x16),(0x08,0x17),(0x08,0x1F),(0x0A,0x18),(0x0A,0x12),
 (0x0E,0x07),(0x0E,0x09),(0x13,0x1C),(0x13,0x0F),(0x14,0x0D),(0x14,0x19),(0x16,0x02),
 (0x16,0x14),(0x19,0x04),(0x19,0x1D),(0x1D,0x1B),(0x1D,0x06),(0x1E,0x15),(0x1E,0x0B),
 O}.
```

3.2. *NIST-Recommended Curves*

Let $E(a, b)$ be a NIST-recommended elliptic curve defined over the binary field $\mathbb{F}_{2^m}$ with curve parameters a and b. In Table 1, each NIST-recommended randomly chosen elliptic curve over $\mathbb{F}_{2^m}$ is presented, where $m = 163$, 233, 283, 409, and 571. Then, for each of these curves its corresponding curve $\widehat{E}(\widehat{a}, b)$ is shown, where $\widehat{a} = 1 - \mathrm{Tr}(a)$. Similarly, Table 2 gives the NIST-recommended Koblitz curves. For each curve the "values" of m, $f(z)$, a, b, and $\#E(\mathbb{F}_{2^m})$ are listed, where $f(z)$ is the irreducible trinomial or pentanomial used as the reduction polynomial. For the random curves, parameter b is shown in hexadecimal form. For each case the group order $\#E(\mathbb{F}_{2^m})$ is given in decimal, followed by its prime factorization.

We notice that for each listed NIST-recommended curve E, the group $\widehat{E}(\mathbb{F}_{2^m})$ is cryptographically weaker; i.e., all the prime factors of $\#\widehat{E}(\mathbb{F}_{2^m})$ are smaller than the larger prime factor of $\#E(\mathbb{F}_{2^m})$, with only one exception for the case of $m = 283$ for Koblitz curves, where the orders of both $E(\mathbb{F}_{2^m})$ and $\widehat{E}(\mathbb{F}_{2^m})$ are *almost* prime. In Table 3, the size of each prime factor of the group orders of these elliptic curves is presented. Additionally, it can be shown by Rück's theorem [34] that $E(\mathbb{F}_{2^m})$ and $\widehat{E}(\mathbb{F}_{2^m})$, where $m \in \{163, 233, 283, 409, 571\}$, are cyclic groups for all the curves in Tables 1 and 2.

Curve specifications for $m = 163$: $f(z) = z^{163} + z^7 + z^6 + z^3 + 1$,
$b = 0x\ 00000002\ 0A601907\ B8C953CA\ 1481EB10\ 512F7874\ 4A3205FD$

Standard Curve B-163. $a = 1$
$\#E(\mathbb{F}_{2^{163}}) = 1169201309864722334562948488575278137851368640317 4$
$= (2)(584600654932361167281474244287639068925684320 1587)$

Weaker Curve. $\widehat{a} = 0$
$\#\widehat{E}(\mathbb{F}_{2^{163}}) = 1169201309864722334562947243770774693598123428444 4$
$= (2)^2\ (31)(907)(18908293)(192478327)(285644694766939633075451013 53)$

Curve specifications for $m = 233$: $f(z) = z^{233} + z^{74} + 1$,
$b = 0x\ 00000066\ 647EDE6C\ 332C7F8C\ 0923BB58\ 213B333B\ 20E9CE42\ 81FE115F\ 7D8F90AD$

Standard Curve B-233. $a = 1$
$\#E(\mathbb{F}_{2^{233}}) = 1380349269358112757486951172455405111167962547469002711075876726897 0926$
$= (2)(690174634679056378743475586227702555583981273734501355537938363448 5463)$

Weaker Curve. $\widehat{a} = 0$
$\#\widehat{E}(\mathbb{F}_{2^{233}}) = 1380349269358112757486951172455405069812481041399151910989132962622626 0260$
$= (2)^2\ (5)(283)(541)(584818873)(7831953276938460946 09)(9842010543696906015214412-$
$423419303)$

Curve specifications for $m = 283$: $f(z) = z^{283} + z^{12} + z^7 + z^5 + 1$,
$b = 0x\ 027B680A\ C8B8596D\ A5A4AF8A\ 19A0303F\ CA97FD76\ 45309FA2\ A581485A\ F6263E31\ 3B79A2F5$

Standard Curve B-283. $a = 1$
$\#E(\mathbb{F}_{2^{283}}) = 1554135113780583256735569525458815125313925184875380977821839305354008855557 4-$
757385742
$= (2)(7770675568902916283677847627294075626569625924376904889109196526770044277-$
$787378692871)$

Weaker Curve. $\widehat{a} = 0$
$\#\widehat{E}(\mathbb{F}_{2^{283}}) = 1554135113780583256735569525458815125313925757608042256181060550228238000770 8-$
578585076
$= (2)^2\ (7)(19)^2(5942982169)(48758898298463720443)(4552740729996075317094698 3)(11-$
$6544641275194419631177527)$

Table 1. NIST-recommended randomly chosen curves and their weaker counterparts over $\mathbb{F}_{2^m}$.

3.3. *Invalid-Curve Attacks on Montgomery's Ladder Algorithm*

Consider a cryptosystem that uses a *strong* elliptic curve $E(a, b)$ defined over $\mathbb{F}_{2^m}$ with curve parameters a and b (e.g., a NIST-recommended elliptic curve), where m is an odd number. Assume that $\widehat{E}(\widehat{a}, b)$ is a weaker curve defined over $\mathbb{F}_{2^m}$ with curve parameters $\widehat{a}$ and b, such that $\mathtt{Tr}(\widehat{a}) = 1 - \mathtt{Tr}(a)$. Consider that the attacker has the computational power for computing the EC discrete logarithm using the Silver–Pohlig–Hellman algorithm in the cryptographically weaker group $\widehat{E}(\mathbb{F}_{2^m})$. Also consider that $\widehat{E}(\mathbb{F}_{2^m})$ is a cyclic group, which implies that there are $\phi(\#\widehat{E}(\mathbb{F}_{2^m}))$ points of order $\#\widehat{E}(\mathbb{F}_{2^m})$. Additionally, for the attacks presented in this work we need to obtain $\#\widehat{E}(\mathbb{F}_{2^m})$. Using (6), this value can be obtained from $\#E(\mathbb{F}_{2^m})$, which is usually public or can be obtained with some point counting algorithms, e.g., [36,38]. Consider that the underlying ECSM algorithm is the Montgomery ladder (Algorithm 1). Since this algorithm does not utilize the curve parameter a, depending on the input point the computation can be carried out in either $E(\mathbb{F}_{2^m})$ or $\widehat{E}(\mathbb{F}_{2^m})$. Then, the idea behind the attacks presented below is to produce an incorrect result from the computation being performed in $\widehat{E}(\mathbb{F}_{2^m})$ due to a fault. Our contribution adopts the same single-bit flip fault model proposed in [5], which has been shown to be practical [41].

Curve specifications for $m = 409$: $f(z) = z^{409} + z^{87} + 1$,
$b =$ 0x 0021A5C2 C8EE9FEB 5C4B9A75 3B7B476B 7FD6422E F1F3DD67 4761FA99 D6AC27C8 A9A197B2 72822F6C D57A55AA 4F50AE31 7B13545F

Standard Curve B-409. $a = 1$
$\#E(\mathbb{F}_{2^{409}}) =$ 132211193758049719790383061606554207965680936592856243856929759660831554965474961041628744752435822193195973457673313505354 2
$=$ (2)(661055968790248598951915308032771039828404682964281219284648798304157774827374805208143723762179110965979867288366567526771)

Weaker Curve. $\widehat{a} = 0$
$\#\widehat{E}(\mathbb{F}_{2^{409}}) =$ 132211193758049719790383061606554207965680936592856243856929758448930761529049577288446939423450291745840511352336029816348 4
$=$ (2)^2(13)(43)(599)(1867)(4201)(10711)(37882813369962759934 7)(31017040828999712946665122892352599407801073958767427697543570603579682776895114772929)

Curve specifications for $m = 571$: $f(z) = z^{571} + z^{10} + z^5 + z^2 + 1$,
$b =$ 0x 02F40E7E 2221F295 DE297117 B7F3D62F 5C6A97FF CB8CEFF1 CD6BA8CE 4A9A18AD 84FFABBD 8EFA5933 2BE7AD67 56A66E29 4AFD185A 78FF12AA 520E4DE7 39BACA0C 7FFEFF7F 2955727A

Standard Curve B-571. $a = 1$
$\#E(\mathbb{F}_{2^{571}}) =$ 772907504603451668939070378186397468859785465941286999731447050290303828457912084907228799877883154616626776224385388897249374492563362614046905657660666482278638221057140 6
$=$ (2)(3864537523017258344695351890931987344298927329706434998657235251451519142289560424536143999389415773083133881121926944486246872462816813070234528288303332411393191105285703)

Weaker Curve. $\widehat{a} = 0$
$\#\widehat{E}(\mathbb{F}_{2^{571}}) =$ 772907504603451668939070378186397468859785465941286999731447050290303828457912084907248706754885876568358670188215481973696656971853832448250257811726165817200154104872229 2
$=$ (2)^2(7)(1153)(99262049966063)(641043691173743374578683)(3650231141108073953669603)(562516514411236993734142229508523209240999366989)(1832372106849886832903758716153488484939785889992701131641)

Table 1. (*Continued.*)

4. Basic Attack

Fault Model Let us assume that the adversary can inject a flip fault (single or multiple bit) into the x-coordinate that might occur at random locations of the input point $P = (P_x, P_y) \in E(\mathbb{F}_{2^m})$ of a device computing the ECSM utilizing Algorithm 1. Suppose that the resulting finite field pair after the fault injection is known and is $\widetilde{P} = (\widetilde{P}_x, P_y)$. Consider that the result $\widetilde{Q} = k\widetilde{P} = (\widetilde{Q}_x, \widetilde{Q}_y)$ is released.

4.1. *Attack Description*

For a given $\widetilde{P} = (\widetilde{P}_x, P_y)$ we can verify if there exists a point in $\widehat{E}(\mathbb{F}_{2^m})$ with the same x-coordinate, i.e., if $\exists\ \widehat{P} \in \widehat{E}(\mathbb{F}_{2^m})$ such that $\widehat{P} = (\widetilde{P}_x, \widehat{P}_y)$ for some $\widehat{P}_y \in \mathbb{F}_{2^m}$. In fact, by Lemma 1 we can expect that if we flip single or multiple bits of the x-coordinate such a point exists with a probability of about 1/2. Having $\widetilde{P}_x$, we can obtain $\widehat{P}_y$ as

$$\widehat{P}_y = \widetilde{P}_x \cdot \mathtt{Ht}\left(\widetilde{P}_x + b/\widetilde{P}_x^2 + \widehat{a}\right) \tag{8}$$

where $\mathtt{Ht}(\cdot)$ denotes the half-trace function of the argument [19]. When $\widehat{P} \in \widehat{E}(\mathbb{F}_{2^m})$, in a similar way we can obtain $\widehat{Q} = (\widetilde{Q}_x, \widehat{Q}_y) \in \widehat{E}(\mathbb{F}_{2^m})$ for some $\widehat{Q}_y \in \mathbb{F}_{2^m}$.

Curve specifications for $m = 163$: $f(z) = z^{163} + z^7 + z^6 + z^3 + 1, b = 1$

Standard Curve K-163. $a = 1$

$\#E(\mathbb{F}_{2^{163}}) = 11692013098647223345629483507196896966582371481 26$
$= (2)(58460065493236116728147417535984483483291185740 63)$

Weaker Curve. $\widehat{a} = 0$

$\#\widehat{E}(\mathbb{F}_{2^{163}}) = 11692013098647223345629473816263631617836683539492$
$= (2)^2 (653)(6521)(34101072914026637)(20129541232727197849723433)$

Curve specifications for $m = 233$: $f(z) = z^{233} + z^{74} + 1, b = 1$

Standard Curve K-233. $a = 0$

$\#E(\mathbb{F}_{2^{233}}) = 13803492693581127574869511724554051042283763955449008505312348098965372$
$= (2)^2 (34508731733952818937173779311385127605709409888622521263280870247413 43)$

Weaker Curve. $\widehat{a} = 1$

$\#\widehat{E}(\mathbb{F}_{2^{233}}) = 13803492693581127574869511724554050767520671933232537715337748796231814$
$= (2)(92269)(114861079)(130034039)(5062109767067236109)(98933113739063012876557-7490907)$

Curve specifications for $m = 283$: $f(z) = z^{283} + z^{12} + z^7 + z^5 + 1, b = 1$

Standard Curve K-283. $a = 0$

$\#E(\mathbb{F}_{2^{283}}) = 15541351137805832567355695254588151253139246935172245297183499990119263318817-690415492$
$= (2)^2 (38853377844514581418389238136470378132848117337930613242958749975298158 2-9704422603873)$

Other Curve. $\widehat{a} = 1$

$\#\widehat{E}(\mathbb{F}_{2^{283}}) = 15541351137805832567355695254588151253139262489661987042845498565703205244465-645555326$
$= (2)(77706755689029162836778476272940756265696312448309935214227492828516026 22-232822777663)$

Curve specifications for $m = 409$: $f(z) = z^{409} + z^{87} + 1, b = 1$

Standard Curve K-409. $a = 0$

$\#E(\mathbb{F}_{2^{409}}) = 13221119375804971979038306160655420796568093659285624385692975800915228451569-967642026930338311098320563854663624709254346 84$
$= (2)^2 (33052798439512429947595765401638551991420234148214060964232439502288071 1-289249191050673258457777458014096366590617731358671)$

Weaker Curve. $\widehat{a} = 1$

$\#\widehat{E}(\mathbb{F}_{2^{409}}) = 13221119375804971979038306160655420796568093659285624385692976010061003197882-486190980638079277513073339793817376225077823 42$
$= (2)(5616389)(90250595219)(53825825250806581242382638109975931)(24229267173791-84361670943884439581457811909443935034501042288725219735 1)$

Table 2. NIST-recommended Koblitz curves and, except for $m = 283$, their weaker counterparts over $\mathbb{F}_{2^m}$.

Having $\widehat{P}, \widehat{Q} \in \widehat{E}(\mathbb{F}_{2^m})$ one can obtain $l = k$ or $\#\widehat{E}(\mathbb{F}_{2^m}) - k \bmod n$ using Algorithm 2, where $n = \mathtt{ord}(\widehat{P})$. This would be possible because the computation is performed in the weaker group $\widehat{E}(\mathbb{F}_{2^m})$ and not in the original group $E(\mathbb{F}_{2^m})$. One can then exhaustively search for an integer k' that satisfies (i) $l = k' \bmod n$ or $\#\widehat{E}(\mathbb{F}_{2^m}) - k' \bmod n$ and (ii) $\widetilde{Q} = k'\widetilde{P}$. Thus, the idea of the basic attack is that the adversary with only one pair $(\widehat{P}, \widehat{Q})$ and some acceptable amount of exhaustive search will be able to retrieve the secret scalar k with a probability of success ρ. Let e be a parameter such that 2^e is the maximum acceptable amount of exhaustive search space. The complete attack procedure is presented as Algorithm 3.

Curve specifications for $m = 571$: $f(z) = z^{571} + z^{10} + z^5 + z^2 + 1$, $b = 1$

Standard Curve K-571. $a = 0$

$\#E(\mathbb{F}_{2^{571}}) = 7729075046034516689390703781863974688597854659412869997314470502903038284579120849072535914090826847338826851203301405845094699896266469247718729686468370014222934741106692$

$= (2)^2 (193226876150862917234767594546599367214946366485321749932861762572575957114478021226813397852270671183470671280082535146127367497406661731192968242161709250355573368527667 3)$

Weaker Curve. $\widehat{a} = 1$

$\#\widehat{E}(\mathbb{F}_{2^{571}}) = 7729075046034516689390703781863974688597854659412869997314470502903038284579120849072239152236863464511027612922707302864365614747905481375252905007399952980564988518187006$

$= (2)(835205577201087993065806 99)(59620168636271854235471070 1)(7760879540369714171579633139517983435067803444075923356781485100647555483423235449402799828439841075582403446581482649 7)$

Table 2. (*Continued.*)

Case	m	Curve		Size of each prime factor of $\#E(\mathbb{F}_{2^m})$ (in bits)
Randomly chosen curves	163	NIST B-163	E	2,163
		Weaker curve	$\widehat{E}$	2, 5, 10, 25, 28, 95
	233	NIST B-233	E	2, 233
		Weaker curve	$\widehat{E}$	2, 3, 9, 10, 30, 70, 113
	283	NIST B-283	E	2, 283
		Weaker curve	$\widehat{E}$	2, 3, 5, 33, 66, 86, 87
	409	NIST B-409	E	2, 409
		Weaker curve	$\widehat{E}$	2, 4, 6, 10, 11, 13, 14, 69, 284
	571	NIST B-571	E	2, 570
		Weaker curve	$\widehat{E}$	2, 3, 11, 47, 80, 82, 159, 191
Koblitz curves	163	NIST K-163	E	2, 163
		Weaker curve	$\widehat{E}$	2, 10, 13, 55, 85
	233	NIST K-233	E	2, 232
		Weaker curve	$\widehat{E}$	2, 17, 27, 27, 63, 100
	283	NIST K-283	E	2, 281
		Other curve	$\widehat{E}$	2, 284
	409	NIST K-409	E	2, 409
		Weaker curve	$\widehat{E}$	2, 23, 37, 116, 234
	571	NIST K-571	E	2, 569
		Weaker curve	$\widehat{E}$	2, 87, 89, 395

Table 3. Size of each prime factor of $\#E(\mathbb{F}_{2^m})$ and $\#\widehat{E}(\mathbb{F}_{2^m})$ (in bits) for the curves presented in Tables 1 and 2.

In Step 8 of Algorithm 3, $l = k$ or $\#\widehat{E}(\mathbb{F}_{2^m}) - k \mod n$ is obtained. The value of l has only partial information about k. The remaining part of the scalar might be obtained using an exhaustive search. The latter involves two main steps: (i) solve a system of congruences with a test candidate and the known part of the scalar (Step 11.2.1), and (ii) perform a scalar multiplication to verify if the solution of the system of congruences is the desired scalar (Step 11.2.2).

Let r be the exhaustive search space. This value depends on n and $\#\widehat{E}(\mathbb{F}_{2^m})$. In Step 11.2.1, for having a unique solution $\mod \#\widehat{E}(\mathbb{F}_{2^m})$ it is necessary that

$$\mathtt{lcm}(n, r) = \#\widehat{E}(\mathbb{F}_{2^m}). \tag{9}$$

Algorithm 3 Basic invalid-curve attack on Montgomery's ladder ECSM algorithm.

Input: E defined over $\mathbb{F}_{2^m}$, access to Algorithm 1, the base point $P = (P_x, P_y) \in E(\mathbb{F}_{2^m})$, the order $\#\widehat{E}(\mathbb{F}_{2^m})$, a parameter for acceptable amount of exhaustive search e.
Output: Scalar k with a probability of ρ.

Phase 1: Collect faulty output
1. Inject a fault in $P = (P_x, P_y)$ for obtaining $\widetilde{P} = (\widetilde{P}_x, P_y)$.
2. Compute $\widetilde{Q} = k\widetilde{P} = (\widetilde{Q}_x, \widetilde{Q}_y)$ Algorithm 1.
3. $T \leftarrow \widetilde{Q}_x + b/\widetilde{Q}_x^2 + \widehat{a}$.
4. If $(\mathtt{Tr}(T) = 0)$ then
   4.1 $\widehat{Q}_x \leftarrow \widetilde{Q}_x$, $\widehat{Q}_y \leftarrow \widetilde{Q}_x \cdot \mathtt{Ht}(T)$;
5. Else
   5.1 Go to Step 1.

Phase 2: Obtain k partially using the Silver–Pohlig–Hellman algorithm
6. $\widehat{P}_x \leftarrow \widetilde{P}_x$, $\widehat{P}_y \leftarrow \widetilde{P}_x \cdot \mathtt{Ht}(\widetilde{P}_x + b/\widetilde{P}_x^2 + \widehat{a})$.
7. Obtain $n = \mathtt{ord}(\widehat{P})$.
8. Utilize Algorithm 2 with $(\widehat{P}, \widehat{Q}, n)$ to obtain $l \mod n$.

Phase 3: Exhaustive search and verification
9. Find the smallest value of r for $\mathtt{lcm}(n, r) = \#\widehat{E}(\mathbb{F}_{2^m})$ (see (11)).
10. If $(r = 1)$ then
    10.1 Compute $R = l\widetilde{P}$ using Algorithm 1.
    10.2 If $(R = \widetilde{Q})$ then return(l); else return($\#\widehat{E}(\mathbb{F}_{2^m}) - l$).
11. Else if $(r \leq 2^e)$ then
    11.1 $k' \leftarrow 0$.
    11.2 While $(k' < r)$ do
        11.2.1 Solve the system of congruences $k'' \equiv k' \pmod r$ and $k'' \equiv l \pmod n$.
        11.2.2 Compute $R = k''\widetilde{P}$ using Algorithm 1.
        11.2.3 If $(R = \widetilde{Q})$ then return(k'');
        11.2.4 Else if $(R = -\widetilde{Q})$ then return($\#\widehat{E}(\mathbb{F}_{2^m}) - k''$);
        11.2.5 Else $k' \leftarrow k' + 1$.
12. Else return("failure").

For efficiency, r should be selected as the minimum value that satisfies (9). Let $\#\widehat{E}(\mathbb{F}_{2^m}) = 2^{e_0} p_1^{e_1} p_2^{e_2} \cdots p_{u-1}^{e_{u-1}}$ be the prime factorization of $\#\widehat{E}(\mathbb{F}_{2^m})$, where $e_j \geq 1$ for $j \in [0, u-1]$. Let $n = 2^{f_0} p_1^{f_1} p_2^{f_2} \cdots p_{u-1}^{f_{u-1}}$ be the prime factorization of $n = \mathtt{ord}(\widehat{P})$, where $0 \leq f_j \leq e_j$ for $j \in [0, u-1]$. Similarly, let $r = 2^{g_0} p_1^{g_1} p_2^{g_2} \cdots p_{u-1}^{g_{u-1}}$ be the prime factorization of r. Using notation similar to that utilized by Menezes et al. [25] with regard to `lcm`, we can express (9) as

$$2^{\max(f_0,g_0)} p_1^{\max(f_1,g_1)} p_2^{\max(f_2,g_2)} \cdots p_{u-1}^{\max(f_{u-1},g_{u-1})} = 2^{e_0} p_1^{e_1} p_2^{e_2} \cdots p_{u-1}^{e_{u-1}}. \tag{10}$$

The exponents of the minimum value of r that satisfies (10) are

$$g_j = \begin{cases} 0, & \text{if } e_j = f_j, \\ e_j, & \text{otherwise,} \end{cases} \tag{11}$$

for $j \in [0, u-1]$.

Note that if $r > 2^e$, Algorithm 3 returns in Step 12 "failure". This means that from a specific pair $(\widetilde{P}, \widetilde{Q})$ the exhaustive search space required to obtain uniquely the value of k (i.e., r) is more than the maximum admissible exhaustive search space (i.e., 2^e). For example for a weaker group $\widehat{E}(\mathbb{F}_{2^m})$ from the NIST-recommended curves, as we show below, the probability of failure is quite low even for small values of e. Moreover, in the case of no success with a particular pair $(\widetilde{P}, \widetilde{Q})$, the attacker can repeat the attack procedure until an inevitable success is achieved.

The probability of success of Algorithm 3 (i.e., ρ) depends on the maximum acceptable amount of exhaustive search 2^e and the order of point $\widehat{P}$. Assume that point $\widehat{P}$ is taken randomly from group $\widehat{E}(\mathbb{F}_{2^m})$. In a cyclic group, it is well known that the number of elements of order d is $\phi(d)$. Here $\#\widehat{E}(\mathbb{F}_{2^m})$ is not prime, and consequently not all the points in $\widehat{E}(\mathbb{F}_{2^m})$ have an order $\#\widehat{E}(\mathbb{F}_{2^m})$. Moreover, if $\#\widehat{E}(\mathbb{F}_{2^m})$ has several prime factors (i.e., it is expected since $\widehat{E}(\mathbb{F}_{2^m})$ is assumed to be a weaker group), the order of the points could have any combination of those prime factors or their respective prime powers. For example, the number of points with the full order $\#\widehat{E}(\mathbb{F}_{2^m})$ is $\phi(\#\widehat{E}(\mathbb{F}_{2^m}))$. In contrast, there is only one point of order two which corresponds to $(0, \sqrt{b})$.

4.2. *Obtaining the Probability of Success ρ*

Let $\#\widehat{E}(\mathbb{F}_{2^m}) = 2^{n_0} p_1^{n_1} p_2^{n_2} \cdots p_{u-1}^{n_{u-1}}$ be the prime factorization of $\#\widehat{E}(\mathbb{F}_{2^m})$, where $n_j \geq 1$ for $j \in [0, u-1]$ and $p_j < p_{j+1}$ for $j \in [1, u-2]$. Assume that point $\widehat{P}$ is taken randomly from the group $\widehat{E}(\mathbb{F}_{2^m})$. Here we will obtain the probability of success ρ, first for specific values and then for an arbitrary value of e.

- *Case 1:* $e = 0$. If $e = 0$, then the attack will succeed when $\mathtt{ord}(\widehat{P}) = \#\widehat{E}(\mathbb{F}_{2^m})$. The number of points in $\widehat{E}(\mathbb{F}_{2^m})$ of order $\#\widehat{E}(\mathbb{F}_{2^m})$ is

$$\phi\big(\#\widehat{E}(\mathbb{F}_{2^m})\big) = 2^{n_0-1} \prod_{j=1}^{u-1} p_i^{n_i} \left(1 - \frac{1}{p_i}\right),$$

and for this case the probability ρ is

$$\rho_{e=0} = \frac{\phi(\#\widehat{E}(\mathbb{F}_{2^m}))}{\#\widehat{E}(\mathbb{F}_{2^m})} = \frac{1}{2}\prod_{j=1}^{u-1}\left(1-\frac{1}{p_j}\right). \tag{12}$$

Clearly, this value is bounded to $1/2$. If $p_1 \gg 1$, then $\rho_{e=0}$ would be close to $1/2$ (e.g., all the Koblitz curves in Example 4.2).

- *Case 2:* $e = 1$. For $e = 1$, this probability can be obtained as follows:

$$\rho_{e=1} = \begin{cases} \prod_{j=1}^{u-1}(1-\frac{1}{p_j}), & \text{if } n_0 = 1, \\ \frac{1}{2}\prod_{j=1}^{u-1}(1-\frac{1}{p_j}), & \text{otherwise.} \end{cases} \tag{13}$$

- *Case 3:* $e = 2$. For $e = 2$, we can have two cases. First, if $p_1 \neq 3$, then $\rho_{e=2}$ is

$$\rho_{e=2} = \begin{cases} \prod_{j=1}^{u-1}(1-\frac{1}{p_j}), & \text{if } n_0 = 1 \text{ or } 2, \\ \frac{1}{2}\prod_{j=1}^{u-1}(1-\frac{1}{p_j}), & \text{otherwise.} \end{cases} \tag{14}$$

Secondly, if $p_1 = 3$, then it is necessary to take into account points of order $\#\widehat{E}(\mathbb{F}_{2^m})/h$, with $h \in [1, 3]$. In this case $\rho_{e=2}$ is

$$\rho_{e=2} = \begin{cases} \frac{5}{6}\prod_{j=2}^{u-1}(1-\frac{1}{p_j}), & \text{if } n_0 = 1 \text{ or } 2, \text{ and } n_1 = 1, \\ \frac{2}{3}\prod_{j=2}^{u-1}(1-\frac{1}{p_j}), & \text{if } n_0 = 1 \text{ or } 2, \text{ and } n_1 \geq 2, \\ \frac{1}{6}\prod_{j=2}^{u-1}(1-\frac{1}{p_j}), & \text{if } n_0 \geq 3, \text{ and } n_1 = 1, \\ \frac{1}{3}\prod_{j=2}^{u-1}(1-\frac{1}{p_j}), & \text{otherwise.} \end{cases} \tag{15}$$

- *Case 4: Arbitrary e with some conditions*. Let

$$\#\widehat{E}(\mathbb{F}_{2^m}) = 2^{n_0} p_1^{n_1} p_2^{n_2} \cdots p_{t-1}^{n_{t-1}} p_t^{n_t} p_{t+1}^{n_{t+1}} \cdots p_{u-1}^{n_{u-1}}.$$

Assume that $\#\widehat{E}(\mathbb{F}_{2^m})$ splits completely in e bits such that

$$\log_2\left(2^{n_0} p_1^{n_1} \cdots p_{t-1}^{n_{t-1}}\right) \leq e \quad \text{and} \quad \log_2(p_t) > e.$$

If these conditions are satisfied, then the number of points whose order divides $p_t^{n_t} p_{t+1}^{n_{t+1}} \cdots p_{u-1}^{n_{u-1}}$ is

$$s = \sum_{i=0}^{g-1} \phi\left(2^{j_0(i)} p_1^{j_1(i)} p_2^{j_2(i)} \cdots p_{t-1}^{j_{t-1}(i)} p_t^{n_t} p_{t+1}^{n_{t+1}} \cdots p_{u-1}^{n_{u-1}}\right), \tag{16}$$

where

$$
\begin{aligned}
g &= (n_0+1)(n_1+1)\cdots(n_{t-1}+1),\\
j_0(i) &= i \bmod (n_0+1),\\
j_1(i) &= \left\lfloor \frac{i}{n_0+1} \right\rfloor \bmod (n_1+1),\\
j_2(i) &= \left\lfloor \frac{i}{(n_0+1)(n_1+1)} \right\rfloor \bmod (n_2+1),\\
&\vdots\\
j_{t-1}(i) &= \left\lfloor \frac{i}{(n_0+1)(n_1+1)\cdots(n_{t-2}+1)} \right\rfloor \bmod (n_{t-1}+1).
\end{aligned}
$$

It can be shown that

$$
\sum_{i=0}^{g-1} \phi\left(2^{j_0(i)} p_1^{j_1(i)} p_2^{j_2(i)} \cdots p_{t-1}^{j_{t-1}(i)}\right) = 2^{n_0} p_1^{n_1} p_2^{n_2} \cdots p_{t-1}^{n_{t-1}}.
$$

Since the function ϕ is *multiplicative*,[1] we can reduce (16) and obtain

$$
s = 2^{n_0} p_1^{n_1} \cdots p_{t-1}^{n_{t-1}} p_t^{n_t-1} (p_t-1) p_{t+1}^{n_{t+1}-1} (p_{t+1}-1) \cdots p_{u-1}^{n_{u-1}-1} (p_{u-1}-1).
$$

In this case ρ is as follows:

$$
\rho = \frac{s}{\#\widehat{E}(\mathbb{F}_{2^m})} = \frac{(p_t-1)(p_{t+1}-1)\cdots(p_{u-1}-1)}{p_t p_{t+1} \cdots p_{u-1}}. \tag{17}
$$

- *Case 5: Arbitrary e*. When we cannot split $\#\widehat{E}(\mathbb{F}_{2^m})$ in the form as in the previous case, we can proceed as follows. First, search for the smallest prime factor such that $\log_2(p_i) > e$. Let t be the index of this prime factor. Let $d = p_t^{n_t} p_{t+1}^{n_t} \cdots p_{u-1}^{n_{u-1}}$. From all the possible combinations of the prime factors $p_0 p_1 \cdots p_{t-1}$ and their respective powers, we need to consider only those whose product with d has a value of r that satisfies (9) and $r \le e$. The complete procedure for this case is stated in Algorithm 4. This algorithm also includes the computation of ρ for Cases 1–4.

Algorithm 4 Probability of success ρ for Algorithm 3.

Input: The order $\#\widehat{E}(\mathbb{F}_{2^m}) = 2^{n_0} p_1^{n_1} p_2^{n_2} \cdots p_{t-1}^{n_{t-1}} p_t^{n_t} p_{t+1}^{n_t} \cdots p_{u-1}^{n_{u-1}}$, a parameter for acceptable amount of exhaustive search e, where $0 \le e < \log_2(p_{u-1})$.
Output: Probability of success ρ.

---

1. If $(e = 0)$ then return$(\rho_{e=0})$ using (12);
2. Else if $(e = 1)$ then return$(\rho_{e=1})$ using (13);

[1] If $\gcd(m, n) = 1$, then $\phi(mn) = \phi(m)\phi(n)$.

3. Else if $(e = 2)$ then return$(\rho_{e=2})$ using (14) or (15);
4. Else if $\#\widehat{E}(\mathbb{F}_{2^m})$ splits completely in e bits such that $\log_2(2^{n_0} p_1^{n_1} \cdots p_{t-1}^{n_{t-1}}) \leq e$ and $\log_2(p_t) > e$
   4.1 Return(ρ) using (17);
5. Else
   5.1 Search for the smallest prime factor such that $\log_2(p_i) > e$. Set t with this index.
   5.2 $d \leftarrow p_t^{n_t} p_{t+1}^{n_{t+1}} \cdots p_{u-1}^{n_{u-1}}$.
   5.3 $\rho \leftarrow 0$.
   5.4 For $j_{t-1} = 0$ to n_{t-1} do
   For $j_{t-2} = 0$ to n_{t-2} do
   $\vdots$
   For $j_0 = 0$ to n_0 do
   $h \leftarrow 2^{j_0} p_1^{j_1} \cdots p_{t-2}^{j_{t-2}} p_{t-1}^{j_{t-1}}$.
   Find the smallest value of r for $\mathtt{lcm}(d \cdot h, r) = \#\widehat{E}(\mathbb{F}_{2^m})$.
   If $(r \leq 2^e)$ then
   $\rho \leftarrow \rho + \phi(h)$.
   5.5 $\rho \leftarrow \rho(p_t - 1)(p_{t+1} - 1) \cdots (p_{u-1} - 1)/(2^{n_0} p_1^{n_1} p_2^{n_2} \cdots p_{t-1}^{n_{t-1}} p_t p_{t+1} \cdots p_{u-1})$.
   5.6 Return(ρ).

4.3. *Probability of Success ρ for $\widehat{E}(\mathbb{F}_{2^m})$ from NIST-Recommended Curves*

Table 4 presents the probability of success of Algorithm 3 for $\widehat{E}(\mathbb{F}_{2^m})$ from the NIST-recommended curves. This shows the probability of obtaining the scalar k using a single faulty point $\widetilde{P} \in \widehat{E}(\mathbb{F}_{2^m})$ and specific values of parameter e. We notice that with the minimum amount of exhaustive search (i.e., $e = 0$) the values are close to $1/2$, especially for the Koblitz curve cases where the relation between the two smallest prime factors of $\#\widehat{E}(\mathbb{F}_{2^m})$ is greater (e.g., $p_1/2 \approx 10.8 \times 10^6$ for the example of the Koblitz curve over $\mathbb{F}_{2^{409}}$). Also for the Koblitz curve examples, it can be noticed that with $e = 2$ their probabilities are close to unity, as shown in the fifth column of Table 4. In contrast, for the randomly chosen curves, similar values close to unity are obtained with $e = 10$, as illustrated in the rightmost column of this table.

Table 5 shows the minimum value of parameter e for obtaining a probability ρ larger than some specific values. From this table it can be noticed that for practical situations e could be quite small for an exhaustive search (e.g., say 14) and still have a reasonably high probability of success ρ (e.g., $\rho > \frac{999}{1000}$).

Cost of Algorithm 3 Most of the computational cost of Algorithm 3 is involved in phases 2 and 3, i.e., obtaining k partially using the Silver–Pohlig–Hellman algorithm (Algorithm 2) and the exhaustive search with verification process, respectively. The cost of both phases depends on the order of $\widehat{P}$, i.e., n, and the order $\#\widehat{E}(\mathbb{F}_{2^m})$. Let us consider the cost of each phase.

[3] The case of $m = 283$ for Koblitz curves is omitted for this and any subsequent table since there does not exist a cryptographically weaker group $\widehat{E}(\mathbb{F}_{2^m})$.

Case	m	ρ				
		$e=0$	$e=1$	$e=2$	$e=5$	$e=10$
Randomly	163	0.48333745	0.48333745	0.96667491	0.98278616	0.99943089
chosen	233	0.39784981	0.39784981	0.79569963	0.99462453	0.99677211
curves	283	0.40601504	0.40601504	0.81203008	0.94736842	0.96992481
	409	0.44966230	0.44966230	0.89932460	0.93679646	0.99732494
	571	0.42819973	0.42819973	0.85639945	0.99913270	0.99913270
Koblitz	163	0.49915775	0.49915775	0.99831549	0.99831549	0.99908107
curves	233	0.49999457	0.99998915	0.99998915	0.99998915	0.99998915
	409	0.49999991	0.99999982	0.99999982	0.99999982	0.99999982
	571	0.49999999	0.99999999	0.99999999	0.99999999	0.99999999

Table 4. Probability of success ρ of obtaining k with Algorithm 3 for $\widehat{E}(\mathbb{F}_{2^m})$ from NIST-recommended curves[3] for a given parameter e.

Case	m	Parameter e (in bits)		
		$\rho > 1 - \frac{1}{100}$	$\rho > 1 - \frac{1}{1000}$	$\rho > 1 - \frac{1}{1\times 10^6}$
Randomly	163	7	10	17
chosen	233	5	12	20
curves	283	11	14	14
	409	8	12	23
	571	5	5	15
Koblitz	163	2	10	15
curves	233	1	1	18
	409	1	1	1
	571	1	1	1

Table 5. Minimum value of parameter e for obtaining a probability ρ larger than some given values for $\widehat{E}(\mathbb{F}_{2^m})$ from NIST-recommended curves.

- *Silver–Pohlig–Hellman's algorithm (phase 2 of Algorithm 3).* Step 1.3.2 of the Silver–Pohlig–Hellman algorithm (Algorithm 2), which is the only step in this algorithm with significant cost, needs to compute one EC discrete logarithm. This operation can be performed with a fast algorithm for ECDLP such as Pollard's rho algorithm [30] with an expected number of point operations of about $3\sqrt{p_{t-1}}$, where p_{t-1} is the largest prime divisor of n. This running time can be further reduced using a parallelized version of the Pollard's rho algorithm [43] to about $(\sqrt{\pi p_{t-1}/2})/M$ point operations, where M is the number of processors used for solving the ECDLP instance. Additionally, as shown by Gallant et al. [17], if a Koblitz curve over $\mathbb{F}_{2^m}$ is utilized, then the parallelized version of the Pollard's rho algorithm can take about $(\sqrt{\pi p_{t-1}/m})/(2M)$ point operations.
- *Exhaustive search and verification (phase 3 of Algorithm 3).* With $n = \mathtt{ord}(\widehat{P})$ and $\#\widehat{E}(\mathbb{F}_{2^m})$, the exhaustive search space r is obtained using (9) (see Step 9 of Algorithm 3). Thus, assuming $t \approx m$, the phase 3 of Algorithm 3 will require r scalar multiplications in the worst case which represents at most $(3mr)/2$ point operations if a binary method is utilized.

Example 3. Let us consider the cost of phases 2 and 3 of Algorithm 3 for $\widehat{E}(\mathbb{F}_{2^m})$ from the NIST-recommended curve K-163. For a single processor, the cost of phase 2 is of about $3\sqrt{p_4} \approx 2^{43.6}$ point operations, where p_4 is the largest prime factor of $\#\widehat{E}(\mathbb{F}_{2^m})$ (see Table 2). Now, assume that we have $M = 10{,}000$ computers for solving the instance of the ECDLP. In this case the expected number of point operations for each processor is approximately $(\sqrt{\pi p_4/163})/20000 \approx 2^{24.9}$. For the phase 3 cost, from Tables 3 and 5 we can notice that with a probability greater than $\frac{999}{1000}$ the exhaustive search space will be less than 2^{10}, which implies a number of point operations $<3(163)(2^{10})/2 \approx 2^{17.9}$.

5. Attack with Unknown Faulty Base Finite Field Pair $\widetilde{P}$

Fault Model Let us assume that the adversary can inject a single-bit flip fault into the x-coordinate of the input point $P_i = (P_{i,x}, P_{i,y}) \in E(\mathbb{F}_{2^m})$ of a device computing the ECSM utilizing Algorithm 1 for some i. Suppose that the resulting finite field pair after the fault injection $\widetilde{P}_i = (\widetilde{P}_{i,x}, P_{i,y})$ is unknown. Also, consider that the fault location is at a random position of the x-coordinate. Consider that the result $\widetilde{Q}_i = k\widetilde{P}_i = (\widetilde{Q}_{i,x}, \widetilde{Q}_{i,y})$ is computed.

5.1. *Attack Description*

Under this scenario the attacker might retrieve the secret scalar as follows. First, it is necessary to collect some faulty outputs of the form $\widetilde{Q}_i = k\widetilde{P}_i = (\widetilde{Q}_{i,x}, \widetilde{Q}_{i,y})$ for which there exists a point $\widehat{Q}_i \in \widehat{E}(\mathbb{F}_{2^m})$ such that $\widehat{Q}_i = (\widetilde{Q}_{i,x}, \widehat{Q}_{i,y})$ for some $\widehat{Q}_{i,y} \in \mathbb{F}_{2^m}$. In fact, with two different points $\widehat{Q}_i \in \widehat{E}(\mathbb{F}_{2^m})$, where $i \in \{0, 1\}$, and some acceptable amount of exhaustive search it is possible to obtain k with a high probability.

Let $\widehat{P}_i$ be a point in $\widehat{E}(\mathbb{F}_{2^m})$ with the same x-coordinate as $\widetilde{P}_i = (\widetilde{P}_{i,x}, P_{i,y})$, i.e., $\widehat{P}_i = (\widetilde{P}_{i,x}, \widehat{P}_{i,y}) \in \widehat{E}(\mathbb{F}_{2^m})$ for some $\widehat{P}_{i,y} \in \mathbb{F}_{2^m}$. Since $\widetilde{P}_i$ (and consequently $\widehat{P}_i$) is unknown, we need to guess it among those finite field pairs that differ from each P_i in only one bit of their x-coordinate. Let c_i be the number of possible candidates for $\widehat{P}_i$, where $i \in \{0, 1\}$. Let $R_{i,j}$ be a candidate for $\widehat{P}_i$, where $i \in \{0, 1\}$ and $j \in [0, c_i - 1]$. Initially, by Lemma 1 we can expect that c_i is about $m/2$. However, this amount could be further reduced depending on the order of $\widehat{Q}_i$. This is possible because we know that $\mathtt{ord}(\widehat{Q}_i) \leq \mathtt{ord}(\widehat{P}_i)$, and more precisely $\mathtt{ord}(\widehat{Q}_i) | \mathtt{ord}(\widehat{P}_i)$. Let η_i be the reduction factor due to the latter condition such that $c_i \approx \eta_i \frac{m}{2}$.

After collecting the faulty outputs we can construct two tables A_i of c_i entries with the output of the Silver–Pohlig–Hellman algorithm for each $(R_{i,j}, \widehat{Q}_i, n_{i,j})$, where $i \in \{0, 1\}$, $j \in [0, c_i - 1]$, and $n_{i,j} = \mathtt{ord}(R_{i,j})$. These tables are illustrated in Fig. 1. Thus, having $l_{i,j} \bmod n_{i,j}$ in each entry of Tables A_0 and A_1, we could distinguish those that are likely to be equivalent to either k or $\#\widehat{E}(\mathbb{F}_{2^m}) - k$. The idea is to search entry pairs v and w that satisfy either

$$l_{0,v} \equiv l_{1,w} \big(\bmod \gcd(n_{0,v}, n_{1,w})\big) \quad \text{or} \tag{18}$$

$$l_{0,v} \equiv \#\widehat{E}(\mathbb{F}_{2^m}) - l_{1,w} \big(\bmod \gcd(n_{0,v}, n_{1,w})\big). \tag{19}$$

In practical situations where $m \geq 163$ it is more likely to have a unique candidate pair that satisfies either (18) or (19), because it is expected that $n_{i,j} \gg c_i$ for $i \in \{0, 1\}$ and

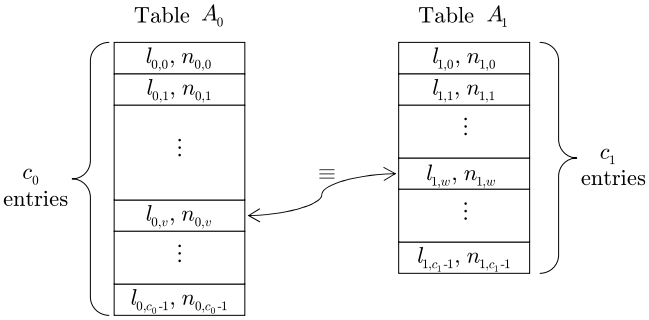

Fig. 1. Tables A_0 and A_1 with the output of the Silver–Pohlig–Hellman algorithm for each $(R_{i,j}, \widehat{Q}_i, n_{i,j})$, where $i \in \{0, 1\}$, $j \in [0, c_i - 1]$, and $n_{i,j} = \mathtt{ord}(R_{i,j})$.

$j \in [0, c_i - 1]$. Nevertheless, even if there is not a unique candidate pair it is possible to verify which one is equivalent to k or $\#\widehat{E}(\mathbb{F}_{2^m}) - k$ after performing an exhaustive search similar to the attack presented in the previous subsection. The complete attack procedure is presented in Algorithm 5. Let e be a parameter such that 2^e is the maximum acceptable amount of exhaustive search per candidate pair found in Step 5 of Algorithm 5. Also, let us define σ as the probability of success for retrieving the scalar k using Algorithm 5.

Algorithm 5 Invalid-curve attack with unknown faulty base point $\widetilde{P}$.

Input: E defined over $\mathbb{F}_{2^m}$, access to Algorithm 1, base point $P_i = (P_{i,x}, P_{i,y}) \in E(\mathbb{F}_{2^m})$ with $i \in \{0, 1\}$, the order $\#\widehat{E}(\mathbb{F}_{2^m})$, a parameter for acceptable amount of exhaustive search e.
Output: Scalar k with a probability of σ

Phase 1: Collect faulty outputs
1. $i \leftarrow 0$.
2. While $(i < 2)$ do
   2.1 Inject a fault in $P_i = (P_{i,x}, P_{i,y})$ for obtaining $\widetilde{P}_i = (\widetilde{P}_{i,x}, P_{i,y})$.
   2.2 Compute $\widetilde{Q}_i = k\widetilde{P}_i = (\widetilde{Q}_{i,x}, \widetilde{Q}_{i,y})$ using Algorithm 1.
   2.3 $T_1 \leftarrow \widetilde{Q}_{i,x} + b/\widetilde{Q}_{i,x}^2 + \widehat{a}$.
   2.4 If $(\mathtt{Tr}(T_1) = 0)$ then
      2.4.1 $\widehat{Q}_{i,x} \leftarrow \widetilde{Q}_{i,x}$, $\widehat{Q}_{i,y} \leftarrow \widetilde{Q}_{i,x} \cdot \mathtt{Ht}(T_1)$, $i \leftarrow i + 1$.

Phase 2: Construct tables
3. For $i = 0$ to 1 do
4. $T_2 \leftarrow 1$.
   4.1 For $j = 0$ to $m - 1$ do
      4.1.1 $R_x \leftarrow P_{i,x} + T_2$.
      4.1.2 $T_3 \leftarrow R_x + b/{R_x}^2 + \widehat{a}$.
      4.1.3 If $(\mathtt{Tr}(T_3) = 0)$ then
         (a) $R_y \leftarrow R_x \cdot \mathtt{Ht}(T_3)$.
         (b) Obtain $n = \mathtt{ord}(R)$.

(c) If $(\mathtt{ord}(\widehat{Q}_i)|n)$ then
(i) Utilize Algorithm 2 with $(R, \widehat{Q}_i, n)$ to obtain $l \bmod n$.
(ii) Store (l, n) in Table A_i.

4.1.4 $T_2 = T_2 \ll 1$.

Phase 3: Searching for candidate pairs

5. For some entries v and w in Tables A_0 and A_1, respectively, search for candidate pairs that satisfy $l_v \equiv l_w \pmod{\mathtt{gcd}(n_v, n_w)}$ or $l_v \equiv \#\widehat{E}(\mathbb{F}_{2^m}) - l_w \pmod{\mathtt{gcd}(n_v, n_w)}$.
6. For the candidate pairs where $l_v \equiv \#\widehat{E}(\mathbb{F}_{2^m}) - l_w \pmod{\mathtt{gcd}(n_v, n_w)}$ set $l_w \leftarrow \#\widehat{E}(\mathbb{F}_{2^m}) - l_w \pmod{n_w}$ in Table A_1.

Phase 4: Exhaustive search and verification

7. For each candidate pair do
   7.1 Solve the system of congruences $l \equiv l_v \pmod{n_v}$ and $l \equiv l_w \pmod{n_w}$.
   7.2 $n \leftarrow \mathtt{lcm}(n_v, n_w)$.
   7.3 Find the smallest value of r for $\mathtt{lcm}(n, r) = \#\widehat{E}(\mathbb{F}_{2^m})$.
   7.4 If $(r = 1)$ then
      7.4.1 Compute $R = l\widetilde{P}$ using Algorithm 1.
      7.4.2 If $(R = \widetilde{Q})$ then return(l);
      7.4.3 Else if $(R = -\widetilde{Q})$ then return$(\#\widehat{E}(\mathbb{F}_{2^m}) - l)$.
   7.5 Else if $(r \leq 2^e)$ then
      7.5.1 $k' \leftarrow 0$.
      7.5.2 While $(k' < r)$ do
         (a) Solve the system of congruences $k'' \equiv k' \pmod{r}$ and $k'' \equiv l \pmod{n}$.
         (b) Compute $R = k''\widetilde{P}$ using Algorithm 1.
         (c) If $(R = \widetilde{Q})$ then return(k'');
         (d) Else if $(R = -\widetilde{Q})$ then return$(\#\widehat{E}(\mathbb{F}_{2^m}) - k'')$;
         (e) Else $k' \leftarrow k' + 1$.
8. Return("failure").

Number of Entries of Tables A_0 and A_1 Let $\#\widehat{E}(\mathbb{F}_{2^m}) = 2^{e_0} p_1^{e_1} p_2^{e_2} \cdots p_{u-1}^{e_{u-1}}$ be the prime factorization of $\#\widehat{E}(\mathbb{F}_{2^m})$. As stated before, the number of entries of Table A_i, c_i, depends on the reduction factor η_i. The latter in turn depends on the order of $\widehat{Q}_i$ and the order of the candidate points for $\widehat{P}_i$, $R_{i,j}$, where $i \in \{0, 1\}$ and $j \in [0, c_i - 1]$. Assuming that the points $R_{i,j}$ are taken randomly from the group $\widehat{E}(\mathbb{F}_{2^m})$, it can be shown that η_i depending on $\mathtt{ord}(\widehat{Q}_i)$ has the following bounds:

$$\eta_{\max} \leq \eta_i \leq 1,$$

where $\eta_{\max} = \frac{1}{2}\prod_{j=1}^{u-1}(1 - \frac{1}{p_j})$. The lower bound of the above expression corresponds to the case when $\mathtt{ord}(\widehat{Q}_i) = \#\widehat{E}(\mathbb{F}_{2^m})$. In this case the reduction factor is maximum (i.e., $\eta_{\max}$), and consequently the number of entries of Table A_i is minimum (i.e., $c_{\min} \approx \frac{\eta_{\max} m}{2}$). On the other hand, theoretically the upper bound of η_i holds only when $\mathtt{ord}(\widehat{Q}_i)$ is the point of order two $(0, \sqrt{b})$. However, for the cases where $p_1 \gg 2$ (e.g.,

Case	m	$\eta_{\max}$	$\overline{\eta}$	$\frac{\eta_{\max} m}{2} \approx c_{\min}$	$\frac{m}{2} \approx c_{\max}$	$\frac{\overline{\eta} m}{2} \approx \overline{c}$
Randomly	163	0.483	0.665	39.4	81.5	54.2
chosen	233	0.398	0.574	46.3	116.5	66.9
curves	283	0.406	0.573	57.5	141.5	81.1
	409	0.450	0.623	92.0	204.5	127.3
	571	0.428	0.603	122.2	285.5	172.1
Koblitz	163	0.499	0.686	40.7	81.5	55.9
curves	233	0.499	0.749	58.2	116.5	87.4
	409	0.499	0.749	102.2	204.5	153.4
	571	0.499	0.749	142.7	285.5	214.1

Table 6. Minimum, maximum, and average number of entries of Tables A_i for $\widehat{E}(\mathbb{F}_{2^m})$ from the NIST-recommended curves.

$\widehat{E}(\mathbb{F}_{2^m})$ for the Koblitz curves of Table 2), if $\mathtt{ord}(\widehat{Q}_i) = \#\widehat{E}(\mathbb{F}_{2^m})/2^{e_0}$, then the reduction factor is close to unity. For these cases the number of entries of Table A_i is maximum (i.e., $c_{\max} \approx \frac{m}{2}$). In Table 6 the values of $\eta_{\max}$, $c_{\min}$, and $c_{\max}$ are given for each $\widehat{E}(\mathbb{F}_{2^m})$ from the NIST-recommended curves. Also, this table shows the average cases for η_i and c_i (i.e., $\overline{\eta}$ and $\overline{c}$, respectively).

Algorithm 5 needs to compute in total $c_0 + c_1$ EC discrete logarithms using the Silver–Pohlig–Hellman algorithm. This number is fixed since the search for candidate pairs and the exhaustive search phases are performed after the tables' construction. If we merge these three phases, a speedup on average can be achieved. Let us describe two approaches that one could take to combine these phases.

1. We can first completely construct Table A_0. Then, each time an entry of Table A_1 is obtained, we can verify whether this entry satisfies the congruence in (18) or (19) with any entry of A_0. For each candidate pair found (if any) we proceed with the exhaustive search and verification process. If the verification fails, then we continue to obtain the next entry of Table A_1 and repeat the process until the scalar is obtained. Even when using this approach the number of EC discrete logarithms in the worst case is the same as that using Algorithm 5 (i.e., $c_0 + c_1$); on average it is roughly $c_0 + \frac{1}{2}c_1$.
2. Another approach is to construct Tables A_0 and A_1 in an alternate way. Each time an entry in A_i is obtained, we can search Table $A_{\bar{i}}$ for candidate pairs that satisfy either Congruence (18) or (19) for $i \in \{0, 1\}$. For each candidate pair found (if any) we proceed with the exhaustive search and verification process. This process is repeated until a candidate pair passes the verification process, i.e., the scalar is found. Let Tables A_0 and A_1 be of the same size, i.e., $c_0 = c_1$. For this case the average number of EC discrete logarithms is $\approx \frac{4}{3}c_0$. In Appendix A we show how the latter value is obtained. This appendix also includes the case where $c_0 \neq c_1$.

5.2. *Obtaining the Probability of Success* σ

The probability of success σ of Algorithm 5 depends on parameter e and the order of both $\widehat{P}_0$ and $\widehat{P}_1$. Consider that the latter two points are taken randomly from the group $\widehat{E}(\mathbb{F}_{2^m})$. For each trio $(\widehat{P}_i, \widehat{Q}_i, n_i)$, the Silver–Pohlig–Hellman algorithm provides $l_i \bmod n_i$, where $i \in \{0, 1\}$ and $n_i = \mathtt{ord}(\widehat{P}_i)$. Utilizing these values, a system of

congruences is solved and a solution $\mathtt{mod}\, n$ is obtained, where $n = \mathtt{lcm}(n_0, n_1)$ (see Step 7.2). This "combination" of modulus n_i might reduce the exhaustive search space in comparison with the individual case of n_0 or n_1. This observation permits us to obtain a relation between the probabilities of success ρ and σ for Algorithms 3 and 5, respectively. In this case ρ is the probability that from an individual pair (l_i, n_i), $i = 0$ or 1, we could obtain the scalar using exhaustive search for a given value of e. Then we can express σ as follows:

$$\sigma = 2\rho - \rho^2 + \lambda. \tag{20}$$

The first two terms represent the probability that for a given e we could obtain the scalar from at least one of the two pairs. The third term, λ, is the probability that the "combination" does succeed in obtaining the scalar with exhaustive search when neither pair individually does so for a given value of e. Equation (20) gives an explicit lower bound for σ, i.e., $\sigma \geq 2\rho - \rho^2$. In fact, for the cases of $\widehat{E}(\mathbb{F}_{2^m})$ from the NIST-recommended curves we notice that $\sigma \approx 2\rho - \rho^2$ for $e \geq 2$.

For obtaining a more precise value of σ one can check, from all the possible order values of two points (i.e., $\widehat{P}_0$ and $\widehat{P}_1$), which ones provide sufficient scalar information for obtaining the rest using exhaustive search for a given parameter e. Additionally we need to consider the probability of occurrence of every point order combination. The complete procedure is provided in Algorithm 6.

Algorithm 6 Probability of success σ for Algorithm 5.

Input: The order $\#\widehat{E}(\mathbb{F}_{2^m}) = 2^{n_0} p_1^{n_1} \cdots p_{u-1}^{n_{u-1}}$, a parameter for acceptable amount of exhaustive search e, where $e \geq 0$.
Output: Probability of success σ.

1. $\sigma = 0$
2. For $J_{u-1} = 0$ to n_{u-1} do
   For $J_{u-2} = 0$ to n_{u-2} do
   $\vdots$
   For $J_0 = 0$ to n_0 do
   $D \leftarrow 2^{J_0} p_1^{J_1} \cdots p_{u-1}^{J_{u-1}}$
   $N \leftarrow \phi(D)$
   For $j_{u-1} = 0$ to n_{u-1} do
   For $j_{u-2} = 0$ to n_{u-2} do
   $\vdots$
   For $j_0 = 0$ to n_0 do
   $d \leftarrow 2^{j_0} p_1^{j_1} \cdots p_{u-1}^{j_{u-1}}$
   $n \leftarrow \mathtt{lcm}(D, d)$
   Find the smallest value of r for $\mathtt{lcm}(n, r) = \#\widehat{E}(\mathbb{F}_{2^m})$.
   If $(r \leq 2^e)$ then
   $\sigma \leftarrow \sigma + N \cdot \phi(d)$.
3. $\sigma = \sigma / (\#\widehat{E}(\mathbb{F}_{2^m}))^2$
4. Return(σ).

5.3. *Probability of Success σ for $\widehat{E}(\mathbb{F}_{2^m})$ from NIST-Recommended Curves*

Table 7 presents the probability of success of Algorithm 5 for $\widehat{E}(\mathbb{F}_{2^m})$ from the NIST-recommended curves. This shows the probability of obtaining the scalar k for specific values of parameter e. These values were obtained using Algorithm 6. We notice that the probability of success is better in comparison with the basic attack. In fact, for $e \geq 2$ the relation between the probability of success of both attacks is $\sigma \approx 2\rho - \rho^2$. In Table 8, we list the minimum value of parameter e for obtaining a probability σ larger than some specific values. This table shows that even with small values of e (e.g., say 14) the probability of success is quite high (e.g., $\sigma > \frac{999,999}{1,000,000}$).

Cost of Algorithm 5 The most significant computational cost of Algorithm 5 is involved in phases 2 and 4, i.e., construction of tables and the exhaustive search with verification process, respectively. Let us consider the cost of each phase.

- *Construction of tables (phase 2 of Algorithm 5).* Compared with the basic attack presented in the previous subsection (Algorithm 3), Algorithm 5 needs to perform $c_0 + c_1$ instances of the Silver–Pohlig–Hellman algorithm (Algorithm 2) instead of

Case	m	σ				
		$e=0$	$e=1$	$e=2$	$e=5$	$e=10$
Randomly	163	0.74921865	0.74921865	0.99895820	0.99973864	0.99999970
chosen	233	0.71998855	0.71998855	0.95998473	0.99998410	0.99999555
curves	283	0.73265871	0.73265871	0.97687829	0.99722992	0.99926508
	409	0.74515657	0.74515657	0.99354209	0.99797754	0.99999814
	571	0.73469332	0.73469332	0.97959110	0.99999925	0.99999925
Koblitz	163	0.74999822	0.74999822	0.99999763	0.99999763	0.99999939
curves	233	0.74999999	0.99999999	0.99999999	0.99999999	0.99999999
	409	0.74999999	0.99999999	0.99999999	0.99999999	0.99999999
	571	0.74999999	0.99999999	0.99999999	0.99999999	0.99999999

Table 7. Probability of success σ of obtaining k with Algorithm 5 for $\widehat{E}(\mathbb{F}_{2^m})$ from the NIST-recommended curves for a given parameter e.

Case	m	Parameter e (in bits)		
		$\sigma > 1 - \frac{1}{100}$	$\sigma > 1 - \frac{1}{1000}$	$\sigma > 1 - \frac{1}{1\times 10^6}$
Randomly	163	2	5	10
chosen	233	5	5	12
curves	283	3	9	14
	409	2	6	12
	571	3	5	5
Koblitz	163	2	2	10
curves	233	1	1	1
	409	1	1	1
	571	1	1	1

Table 8. Minimum value of parameter e for obtaining a probability σ larger than some given values for $\widehat{E}(\mathbb{F}_{2^m})$ from the NIST-recommended curves.

one, where c_i is the size of Table A_i for $i \in \{0, 1\}$. Similar to the cost of phase 2 of Algorithm 3, the cost to construct the tables with a single processor is about $3(c_0 + c_1)\sqrt{p_{t-1}}$ point operations, where p_{t-1} is the largest prime divisor of $\#\widehat{E}(\mathbb{F}_{2^m})$. If M processors are used, then about $(c_0 + c_1)\sqrt{\pi p_{t-1}/2}/M$ point operations are required. If a Koblitz curve over $\mathbb{F}_{2^m}$ is utilized, then this cost can be reduced to about $(c_0 + c_1)(\sqrt{\pi p_{t-1}/m})/(2M)$ point operations. These costs clearly depend directly on values of c_i, which depends on the order of $\widehat{Q}_i$ and the order of the candidate points for $\widehat{P}_i$. As discussed earlier, the bounds for c_i are approximately $\frac{\eta_{\max} m}{2} \leq c_i \leq \frac{m}{2}$, where $\eta_{\max}$ is the maximum reduction factor which depends on $\#\widehat{E}(\mathbb{F}_{2^m})$.

- *Exhaustive search and verification (phase 4 of Algorithm 5).* In phase 3 of Algorithm 5 using Tables A_0 and A_1, a search for candidate pairs that satisfy either (18) or (19) is performed. As discussed earlier, for today's applications where $m \geq 163$ it is expected to have a unique candidate pair. In this way, in phase 4 an exhaustive search is performed in order to obtain the full value of the scalar. Here, the exhaustive search space r is obtained in Steps 7.2 and 7.3. Thus, assuming $t \approx m$, the phase 4 of Algorithm 5 will require r scalar multiplications in the worst case which represents at most $(3mr)/2$ point operations if a binary method is utilized.

Example 4. Let us consider the cost of phases 2 and 4 of Algorithm 5 for $\widehat{E}(\mathbb{F}_{2^m})$ from the NIST-recommended curve K-163. Let us use the minimum and maximum values of c_i from Table 6 to give an interval for each cost. For a single processor, the cost of phase 2 is approximately in the interval $[6c_{\min}\sqrt{p_4}, 6c_{\max}\sqrt{p_4}] \approx [2^{49.9}, 2^{50.9}]$ point operations, where p_4 is the largest prime factor of $\#\widehat{E}(\mathbb{F}_{2^m})$ (see Table 2). Now, assume that we have $M = 10{,}000$ computers for solving the instances of the ECDLP. In this case the expected number of point operations for each processor is approximately in the interval $[\frac{c_{\min}(\sqrt{\pi p_4/163})}{10000}, \frac{c_{\max}(\sqrt{\pi p_4/163})}{10000}] \approx [2^{31.2}, 2^{32.2}]$. For the phase 4 cost, from Tables 3 and 5 we can notice that with a probability greater than $\frac{999}{1000}$ the exhaustive search space will be $r \leq 4$. Here the cost of phase 4 is negligible.

6. Countermeasures

The attacks presented in the previous section only need one or two faulty outputs to break the given instance of ECSM with a high probability of success. Hence, this may constitute a threat to cryptosystems using the Montgomery ladder ECSM for elliptic curves over the binary field. Therefore, some countermeasures are needed. In the following, we will describe possible protections against the attacks presented in this paper.

Group Formulas Change A possible countermeasure is to use alternative group formulas that include both elliptic curve parameters a and b. However, such formulas are likely to require more computations and hence cause a degradation in terms of performance. Additionally, if this approach is the only protection used, no errors due to faults are detected, and this might constitute a risk for other attacks such as the DFA attack presented by Biehl et al. [3].

Curve Selection The attacks presented in this paper assume that $\widehat{E}(\mathbb{F}_{2^m})$ is a cryptographically weaker group where the ECDLP could be solved in a reasonable period of time for a given $E(\mathbb{F}_{2^m})$. However, this assumption is not true if both $\#E(\mathbb{F}_{2^m})$ and $\#\widehat{E}(\mathbb{F}_{2^m})$ are almost prime. From the NIST-recommended curves, the only curve that satisfies this condition is referred to as K-283. Although, this curve selection criterion is an effective countermeasure against the fault-based attacks presented in this paper, it might be too restrictive from a practical point of view. Moreover, the following two countermeasures represent a possible solution without limiting the use of particular group $E(\mathbb{F}_{2^m})$ even when the order of $\widehat{E}(\mathbb{F}_{2^m})$ is not an almost prime number.

Point Verification (PV) It is important to verify that the input point is in $E(\mathbb{F}_{2^m})$. In the case where this checking could be bypassed, it is more important to verify whether or not the output is on the original elliptic curve. This countermeasure not only prevents the attacks presented in this paper, but also others such as those described by Biehl et al. [3], Ciet and Joye [7], and Antipa et al. [1]. It is important to note that this verification needs to be implemented in a secure environment. Otherwise, the attacker might bypass this protection and carry out an invalid-curve attack such as one of those described earlier in this paper.

Coherency Check (CC) In addition to PV, which could be applied to any ECSM algorithm, the Montgomery ladder ECSM algorithm also permits us to detect errors in scalar multiplication using a coherency check (CC). We can use the fact that the temporary pair (Q_0, Q_1) is of the form $(l \cdot P, (l+1)P)$ for some integer l at any value of i during the loop of Montgomery's algorithm. Since the difference between Q_1 and Q_0 should be P at any iteration, one can check this during and after the ECSM operation. Note that if the attacker is able to modify the input point P in the way described in Algorithms 3 and 5, the operation $Q_1 - Q_0$ needs to be implemented using group formulas that include both curve parameters, a and b, or at least parameter a to avoid performing this checking operation in $\widehat{E}(\mathbb{F}_{2^m})$. This approach for error detection is presented in more detail in [9].

7. Conclusion

In this paper we have presented two invalid-curve attacks that apply to the Montgomery ladder ECSM algorithms proposed by López and Dahab [22]. These attacks exploit the fact that parameter a is not used in the group formulas for these particular algorithms. In this way, if $\widehat{E}(\mathbb{F}_{2^m})$ is a weaker group with the same parameters as the original group $E(\mathbb{F}_{2^m})$ except for parameter a and we are able to inject a fault in the input point as described in Algorithms 3 and 5, then we would retrieve the scalar k with a high probability of success. For the purpose of the NIST-recommended curves, we have shown that there exists a weaker group for nine of the ten cases that include the randomly chosen and Koblitz curves. The only exception is the curve K-283, for which $\#E(\mathbb{F}_{2^m})$ and $\#\widehat{E}(\mathbb{F}_{2^m})$ are almost prime. Also, we have obtained the theoretical probability of success for each of the presented attacks. Additionally, we have determined numerical values of the probabilities of success for $\widehat{E}(\mathbb{F}_{2^m})$ from the NIST-recommended curves. And finally, we have presented some countermeasures to prevent the attacks described in this paper.

Appendix A. Average Number of EC Discrete Logarithms for Algorithm 5

In this appendix we include the computations of the average number of EC discrete logarithms for Algorithm 5 using the second improved approach described on page 365. As assumed in Sect. 5, the fault location is at a random position of the x-coordinate of the base point P. This assumption implies that the value of $k \bmod n_i$ is at a random position in Tables A_i, where $n_i = \mathtt{ord}(\widehat{P}_i)$ and $i \in \{0, 1\}$. Let us define the random variable w as the number of entries needed for having $k \bmod n_i$ in both tables. The order of the possible values of w is shown in Fig. A.1 for the case $c_0 < c_1$.

Case: $c_0 = c_1$ In this case the accumulative probability distribution $F(w)$ for some given values is as follows:

$$
\begin{aligned}
&F(1) = 0, && F(2) = 1/c_0^2,\\
&F(3) = 2/c_0^2, && F(4) = 4/c_0^2,\\
&F(5) = 6/c_0^2, && F(6) = 9/c_0^2,\\
&F(7) = 12/c_0^2, && F(8) = 16/c_0^2,\\
&\vdots && \vdots\\
&F(2c_0 - 1) = (c_0 - 1)/c_0, && F(2c_0) = 1.
\end{aligned}
$$

We can write $F(w)$ as

$$
F(w) = \begin{cases} (w^2 - 1)/(4c_0^2), & w \text{ odd, and } 1 \le w \le 2c_0 - 1,\\ w^2/(4c_0^2), & w \text{ even, and } 2 \le w \le 2c_0. \end{cases}
$$

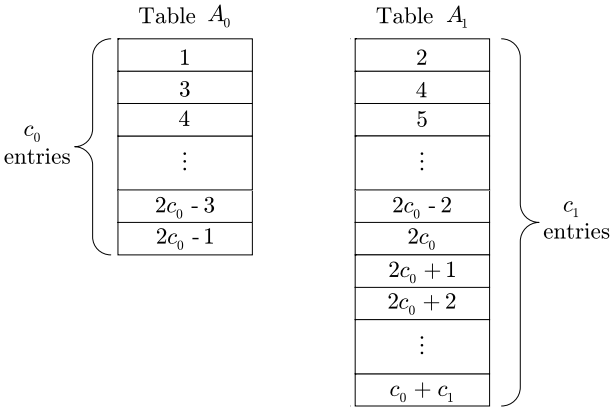

Fig. A.1. Values of the random variable w according to entries of Tables A_0 and A_1 considering $c_0 < c_1$.

Then the probability distribution $f(w) = F(w) - F(w-1)$ is

$$f(w) = \begin{cases} (w-1)/(2c_0^2), & w \text{ odd, and } 1 \le w \le 2c_0 - 1, \\ w/(2c_0^2), & w \text{ even, and } 2 \le w \le 2c_0. \end{cases}$$

The mean μ can be expressed by

$$\mu = \sum_{w=1}^{2c_0} w f(w).$$

After performing a change of variables (i.e., $y = \frac{w-1}{2}$ and $y = \frac{w}{2}$ for the odd and even number cases, respectively), μ can be rewritten as

$$\mu = \sum_{y=0}^{c-1} \frac{y(2y+1)}{c_0^2} + \sum_{y=1}^{c_0} \frac{2y^2}{c_0^2} = \frac{8c_0^2 + 3c_0 + 1}{6c_0} \approx \frac{4}{3}c_0 \quad (\text{for } c_0 \gg 1). \tag{A.1}$$

Case: $c_0 < c_1$ Similar to the previous case, we can write $F(w)$ for some given values as follows:

$$\begin{aligned} &F(1) = 0, && F(2) = 1/(c_0c_1), \\ &F(3) = 2/(c_0c_1), && F(4) = 4/(c_0c_1), \\ &F(5) = 6/(c_0c_1), && F(6) = 9/(c_0c_1), \\ &\vdots && \vdots \\ &F(2c_0 - 1) = (c_0 - 1)/c_1, && F(2c_0) = c_0/c_1, \\ &F(2c_0 + 1) = (c_0 + 1)/c_1, && F(2c_0 + 2) = (c_0 + 2)/c_1, \\ &\vdots && \vdots \\ &F(c_0 + c_1 - 1) = (c_1 - 1)/c_1, && F(c_0 + c_1) = 1. \end{aligned}$$

We express $F(w)$ as

$$F(x) = \begin{cases} (x^2 - 1)/(4c_0c_1), & x \text{ odd, and } 1 \le x \le 2c_0 - 1, \\ x^2/(4c_0c_1), & x \text{ even, and } 2 \le x \le 2c_0, \\ (x - c_0)/c_1, & 2c_0 + 1 \le x \le c_0 + c_1. \end{cases}$$

For this case the probability distribution $f(x)$ is

$$f(x) = \begin{cases} (x-1)/(2c_0c_1), & x \text{ odd, and } 1 \le x \le 2c_0 - 1, \\ x/(2c_0c_1), & x \text{ even, and } 2 \le x \le 2c_0, \\ 1/c_1, & 2c_0 + 1 \le x \le c_0 + c_1. \end{cases}$$

We can obtain the mean μ as follows:

$$\mu = \sum_{x=1}^{c_0+c_1} x f(x).$$

After performing a change of variables (i.e., $y = \frac{x-1}{2}$ and $y = \frac{x}{2}$ for the odd and even number cases, respectively, where $1 \leq x \leq 2c_0$), μ can be expressed as

$$\begin{aligned} \mu &= \sum_{y=0}^{c_0-1} \frac{y(2y+1)}{c_0 c_1} + \sum_{y=1}^{c_0} \frac{2y^2}{c_0 c_1} + \sum_{x=2c_0+1}^{c_0+c_1} \frac{x}{c_1}, \\ \mu &= \frac{3c_1^2 - c_0^2 + 6c_0c_1 + 3c_1 + 1}{6c_1}. \end{aligned} \tag{A.2}$$

Case: $c_0 > c_1$ This case is very similar to the previous case. In fact, from (A.2) we can perform the changes of variables $c_0 \leftarrow c_1$ and $c_1 \leftarrow c_0$ to obtain the mean for this case:

$$\mu = \frac{3c_0^2 - c_1^2 + 6c_0c_1 + 3c_0 + 1}{6c_0}. \tag{A.3}$$

References

[1] A. Antipa, D.R.L. Brown, A. Menezes, R. Struik, S.A. Vanstone, Validation of elliptic curve public keys, in *PKC 2003: Public Key Cryptography*. LNCS, vol. 2567 (Springer, Berlin, 2003), pp. 211–223

[2] A. Becker, Methods of fault analysis attacks on elliptic curve cryptosystems. Diploma thesis, Technische Universität Darmstadt (2006)

[3] I. Biehl, B. Meyer, V. Müller, Differential fault attacks on elliptic curve cryptosystems, in *CRYPTO 2000: Advances in Cryptology*. LNCS, vol. 1880 (Springer, Berlin, 2000), pp. 131–146

[4] J. Blömer, M. Otto, J.-P. Seifert, Sign change attacks on elliptic curve cryptosystems, in *FDTC 2005: Fault Diagnosis and Tolerance in Cryptography*. LNCS, vol. 4236 (Springer, Berlin, 2006), pp. 36–42

[5] D. Boneh, R.A. DeMillo, R.J. Lipton, On the importance of eliminating errors in cryptographic computations. *J. Cryptol.* **14**(2), 101–119 (2001)

[6] E. Brier, M. Joye, Weierstraß elliptic curves and side-channel attacks, in *Public Key Cryptography*. LNCS, vol. 2274 (Springer, Berlin, 2002), pp. 335–345

[7] M. Ciet, M. Joye, Elliptic curve cryptosystems in the presence of permanent and transient faults. *Des. Codes Cryptogr.* **36**(1), 33–43 (2005)

[8] W. Diffie, M.E. Hellman, New directions in cryptography. *IEEE Trans. Inf. Theory* **22**(6), 644–654 (1976)

[9] A. Domínguez-Oviedo, M.A. Hasan, Algorithm-level error detection for ECSM. CACR Technical Reports CACR 2009-05, University of Waterloo, Tech. Rep. (2009)

[10] A. Domínguez-Oviedo, M.A. Hasan, Error detection and fault tolerance in ECSM using input randomization. *IEEE Trans. Dependable Secure Comput.* **6**, 175–187 (2009)

[11] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory* **31**(4), 469–472 (1985)

[12] U. Feige, A. Fiat, A. Shamir, Zero-knowledge proofs of identity. *J. Cryptol.* **1**(2), 77–94 (1988)

[13] FIPS 186 Digital Signature Standard (DSS). Federal Information Processing Standards Publication 186. National Institute for Standards and Technology (1994)

[14] P.-A. Fouque, R. Lercier, D. Réal, F. Valette, Fault attack on elliptic curve Montgomery ladder implementation, in *Proceedings of the Workshop on Fault Diagnosis and Tolerance in Cryptography* (2008), pp. 92–98
[15] J. Francq, J.-B. Rigaud, P. Manet, A. Tria, A. Tisserand, Error detection for borrow-save adders dedicated to ECC unit, in *FDTC '08: Proceedings of the 2008 5th Workshop on Fault Diagnosis and Tolerance in Cryptography* (IEEE Computer Society, Washington, 2008), pp. 77–86
[16] G. Frey, Applications of arithmetical geometry to cryptographic constructions, in *Proceedings of the Fifth International Conference on Finite Fields and Applications* (Springer, Berlin, 2001), pp. 128–161
[17] R. Gallant, R. Lambert, S. Vanstone, Improving the parallelized Pollard lambda search on anomalous binary curves. *Math. Comput.* **69**(232), 1699–1705 (2000)
[18] P. Gaudry, F. Hess, N.P. Smart, Constructive and destructive facets of Weil descent on elliptic curves. *J. Cryptol.* **15**(1), 19–46 (2002)
[19] D. Hankerson, A. Menezes, S.A. Vanstone, *Guide to Elliptic Curve Cryptography* (Springer, Berlin, 2003)
[20] M. Joye, S.-M. Yen, The Montgomery powering ladder, in *CHES 2002*. LNCS (Springer, Berlin, 2002), pp. 291–302
[21] N. Koblitz, Elliptic curve cryptosystems. *Math. Comput.* **48**, 203–209 (1987)
[22] J. López, R. Dahab, Fast multiplication on elliptic curves over $GF(2^m)$ without precomputation, in *CHES 1999: Cryptographic Hardware and Embedded Systems*. LNCS, vol. 1717 (Springer, Berlin, 1999), pp. 316–327
[23] M. Maurer, A. Menezes, E. Teske, Analysis of the GHS Weil descent attack on the ECDLP over characteristic two finite fields of composite degree. *LMS J. Comput. Math.* **5**, 127–174 (2002)
[24] A. Menezes, T. Okamoto, S.A. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Trans. Inf. Theory* **39**(5), 1639–1646 (1993)
[25] A. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography* (CRC Press, Boca Raton, 2001)
[26] V.S. Miller, Use of elliptic curves in cryptography, in *CRYPTO 1985: Advances in Cryptology*. LNCS, vol. 218 (Springer, Berlin, 1986), pp. 417–426
[27] P.L. Montgomery, Speeding the Pollard and elliptic curve methods of factorization. *Math. Comput.* **48**, 243–264 (1987)
[28] K. Okeya, K. Sakurai, Efficient elliptic curve cryptosystems from a scalar multiplication algorithm with recovery of the y-coordinate on a Montgomery-form elliptic curve, in *CHES 2001: Cryptographic Hardware and Embedded Systems*. LNCS, vol. 2162 (Springer, Berlin, 2001), pp. 126–141
[29] S. Pohlig, M. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE Trans. Inf. Theory* **24**, 106–110 (1978)
[30] J.M. Pollard, Monte Carlo methods for index computation (mod p). *Math. Comput.* **32**, 918–924 (1978)
[31] S. Pontarelli, G. Cardarilli, M. Re, A. Salsano, Error detection in addition chain based ECC point multiplication, in *IEEE International On-Line Testing Symposium* (2009), pp. 192–194
[32] M.O. Rabin, Digitalized signatures and public-key functions as intractable as factorization. Cambridge, MA, USA, Tech. Rep. (1979)
[33] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **21**(2), 120–126 (1978)
[34] H.-G. Rück, A note on elliptic curves over finite fields. *Math. Comput.* **49**(179), 301–304 (1987)
[35] T. Satoh, K. Araki, Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Comment. Math. Univ. St. Pauli* **47**, 81–92 (1998)
[36] T. Satoh, B. Skjernaa, Y. Taguchi, Fast computation of canonical lifts of elliptic curves and its application to point counting. *Finite Fields Appl.* **9**, 89–101 (2003)
[37] C.-P. Schnorr, Efficient signature generation by smart cards. *J. Cryptol.* **4**(3), 161–174 (1991)
[38] R. Schoof, Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comput.* **44**(170), 483–494 (1985)
[39] I.A. Semaev, Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Math. Comput.* **67**, 353–356 (1998)
[40] D. Shanks, Class number, a theory of factorization, and genera, in *Proceedings of the Symposium in Pure Mathematics*, vol. 20 (American Mathematical Society, Providence, 1971), pp. 415–440
[41] S. Skiribogatov, R. Anderson, Optical fault induction attacks, in *CHES 2002: Cryptographic Hardware and Embedded Systems*. LNCS, vol. 2523 (Springer, Berlin, 2002), pp. 2–12

[42] R. Stern, N. Joshi, K. Wu, R. Karri, Register transfer level concurrent error detection in elliptic curve crypto implementations, in *FDTC '07: Proceedings of the Workshop on Fault Diagnosis and Tolerance in Cryptography* (IEEE Computer Society, Washington, 2007), pp. 112–119

[43] P.C. van Oorschot, M.J. Wiener, Parallel collision search with cryptanalytic applications. *J. Cryptol.* **12**(1), 1–28 (1999)