Degenerate Curve Attacks

Extending Invalid Curve Attacks to Edwards Curves and Other Models

Samuel Neves[1] and Mehdi Tibouchi[2]

[1] CISUC, Dept. of Informatics Engineering
University of Coimbra, Portugal
sneves@dei.uc.pt
[2] Okamoto Research Laboratory
NTT Secure Platform Laboratories, Japan
tibouchi.mehdi@lab.ntt.co.jp

Abstract Invalid curve attacks are a well-known class of attacks against implementations of elliptic curve cryptosystems, in which an adversary tricks the cryptographic device into carrying out scalar multiplication not on the expected secure curve, but on some other, weaker elliptic curve of his choosing. In their original form, however, these attacks only affect elliptic curve implementations using addition and doubling formulas that are independent of at least one of the curve parameters. This property is typically satisfied for elliptic curves in Weierstrass form but not for newer models that have gained increasing popularity in recent years, like Edwards and twisted Edwards curves. It has therefore been suggested (e.g. in the original paper on invalid curve attacks) that such alternate models could protect against those attacks.

In this paper, we dispel that belief and present the first attack of this nature against (twisted) Edwards curves, Jacobi quartics, Jacobi intersections and more. Our attack differs from invalid curve attacks proper in that the cryptographic device is tricked into carrying out a computation not on another elliptic curve, but on a group isomorphic to the multiplicative group of the underlying base field. This often makes it easy to recover the secret scalar with a single invalid computation.

We also show how our result can be used constructively, especially on curves over random base fields, as a fault attack countermeasure similar to Shamir's trick.

Keywords: Elliptic curve cryptography, Edwards curves, Implementation issues, Fault attacks, Countermeasures

1 Introduction

Elliptic curve cryptography (ECC) was introduced in the 1980s by Miller [44] and Koblitz [38], following the successful application of elliptic curves to integer factorization [39]. Compared to its finite field alternatives, ECC offers shorter keys, higher speeds, and additional structure that enables constructions such as bilinear pairings. ECC rests on the hardness of the elliptic curve discrete logarithm problem (ECDLP), which has remained intractable so far—for well-chosen curves.

Regardless of the theoretical security of elliptic curve cryptosystems, attacks targeting their *implementations* are numerous. One particularly powerful attack class is the *fault attack* [12, 13], which consists in injecting faults before or during a cryptographic operation, and inspecting the resulting output to recover key information. Fault attacks directed at elliptic curve scalar multiplication implementations were first published in [9] and further developed in many other works, including [11, 15, 20, 36].

A conceptually simpler attack pointed out by Antipa et al. [1] and extended in several further works [35, 37], the *invalid curve attack*, exploits implementations that fail to verify that input points to a scalar multiplication belong to the correct elliptic curve, and where point addition and doubling formulas are independent of at least one curve parameter. In such cases, the attacker can query its target with a specially-crafted point outside of the correct elliptic curve. Then, because the formulas used in the scalar multiplication do not depend on all curve parameters, the implementation really computes a normal scalar multiplication by the same scalar, but on a *different* curve depending on the invalid input point. Choosing invalid points in such a way that the corresponding curves are weak, the attacker can then quickly recover secret keys from observing the outputs (or the hashed outputs) of the scalar multiplications. Although the attack and recommended countermeasures are well-known to cryptographers, recent research has found that a number of widely-used cryptographic libraries in the wild are vulnerable [29].

The attack of Antipa et al. was originally introduced in the context of elliptic curves in Weierstrass form $y^2 = x^3 + ax + b$, where the usual formulas for point addition and doubling are independent of the curve parameter b. Nowadays, however, alternate elliptic curve models and addition laws are gaining prominence: models such as Montgomery [4, 45] and Edwards [7, 18] curves are being proposed for wide Internet usage[3], and several others are known to have desirable properties for cryptographic applications [10, 33, 34, 40, 53].

Invalid curve attacks generalize directly to those alternate models *provided that* the crucial property of independence of the arithmetic on at least one curve parameter is satisfied. But many of the newer models for elliptic curves, including Edwards curves, use all parameters in their most common addition formulas. It is thus reasonable to expect, then, that invalid curve attacks would not apply to those curves. In fact, the use of addition formulas depending on all curve parameters was specifically mentioned by Antipa et al. [1] as a possible countermeasure to thwart their attack.

Our contribution. In this paper, we re-examine the feasibility of invalid curve attacks against newer elliptic curve models like Edwards curves, and find that a new variant of the attack of Antipa et al. *will* indeed break the security of implementations that do not carry out proper point validation. The new attack

[3] See `https://tools.ietf.org/html/draft-irtf-cfrg-curves`

works by reducing the problem of finding the secret scalar to solving discrete logarithms not on a weaker elliptic curve, but in the multiplicative group of the base field, which is easy for typical curve sizes.

The idea behind the attack is roughly to let one of the parameters in the curve family vary, and consider the degenerate curves (those of genus 0) among them. On those special curves, the group law degenerates to the multiplicative group (or in rare cases, the additive group), and while in principle the group formulas should still involve the curve parameter that was made to vary, it often ends up being multiplied by the constant zero for all points on the degenerate curve. As a result, the same formulas as for scalar multiplication on the correct curve yield an exponentiation in the degenerate group.

When only a hash value of the result of the scalar multiplication is provided (as in hashed Diffie–Hellman), our new attack is somewhat less flexible than invalid curve attacks, since it is no longer possible to vary the weak curve as done by Antipa et al. However, using a baby-step-giant-step-like time-memory tradeoff, we show that we can still easily break curves over some of the largest fields commonly used for elliptic curve cryptography, such as $\mathbb{F}_{2^{521}-1}$.

This new attack underscores the importance of point validation even over newer elliptic curve models.

Finally, the properties we exploit in the attack can also be used constructively, to thwart fault attacks. We present a concrete countermeasure, similar to Shamir's trick [50], that detects faults injected during scalar multiplication particularly efficiently. This is done by lifting the computation on the elliptic curve over $\mathbb{F}_p$ to the composite order ring $\mathbb{Z}/pr\mathbb{Z}$ for some small constant r, and making sure that the component modulo r of the lifted curve is degenerate in the sense mentioned above. Then, verifying that the computation modulo r was correct becomes a simple field exponentiation, which is much faster than the usual scalar multiplication. This technique applies to Weierstrass curves as well as newer models.

Organization of the paper. In §2, we provide a rundown of some of the most common curve models and addition laws used in elliptic curve cryptography. In §3, we first recall the traditional invalid curve attack, and then present our extension of it to newer models of elliptic curves using the degenerate curve technique. In §4, we explain how the new attack can be applied when only a hash of the result of the scalar multiplication is available. And finally, in §5, we present our concrete fault attack countermeasure using degenerate curves.

2 Elliptic Curve Models

We begin by presenting the elliptic curve forms and respective group laws studied in this paper. This is not an exhaustive list; there are many other addition laws in the literature, and the interested reader can see an overview of many of them in [8]. Every base field $\mathbb{F}_p$ throughout this paper is assumed to have characteristic ≥ 5.

2.1 Weierstrass model

The canonical short Weierstrass form of an elliptic curve is given by the equation $y^2 = x^3 + ax + b$, with a point at infinity $\mathcal{O} = (0 : 1 : 0)$. Addition on Weierstrass curves is derived directly from the chord and tangent method [52, Chapter III.2]:

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned} \qquad \text{where } \lambda = \begin{cases} \frac{y_1-y_2}{x_1-x_2} & \text{if } (x_1, y_1) \neq (x_2, \pm y_2), \\ \frac{3x_1^2+a}{2y_1} & \text{if } (x_1, y_1) = (x_2, y_2). \end{cases} \tag{1}$$

2.2 Twisted Edwards model

Edwards curves were introduced in 2007 [7,18]. Here we look at their generalization, *twisted* Edwards curves [5], which cover more curves. A twisted Edwards curve is defined by the equation $ax^2 + y^2 = 1 + dx^2y^2$, with neutral affine point $\mathcal{O} = (0, 1)$. The general complete group law for twisted Edwards curves is

$$(x_3, y_3) = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2} \right). \tag{2}$$

An addition formula, no longer complete, which does not require the d parameter, was found by Hisil, Wong, Carter, and Dawson [25]:

$$(x_3, y_3) = \begin{cases} \left(\frac{x_1y_1+x_2y_2}{y_1y_2+ax_1x_2}, \frac{x_1y_1-x_2y_2}{x_1y_2-y_1x_2} \right) & \text{if } (x_1, y_1) \neq (x_2, y_2), (-x_1, -y_1) \\ \left(\frac{2x_1y_1}{y_1^2+ax_1^2}, \frac{y_1^2-ax_1^2}{2-y_1^2-ax_1^2} \right) & \text{if } (x_1, y_1) = (x_2, y_2) \end{cases}. \tag{3}$$

2.3 Huff's model

Huff curves are a recently rediscovered elliptic curve model [34] previously used in the study of a certain Diophantine equation [27]. They are defined by the equation $ax(y^2 - 1) = by(x^2 - 1)$, and have the affine neutral point $\mathcal{O} = (0, 0)$. Huff's addition formula, complete for points of odd order, is independent of the curve's parameters:

$$(x_3, y_3) = \left(\frac{(x_1 + x_2)(1 + y_1y_2)}{(1 + x_1x_2)(1 - y_1y_2)}, \frac{(y_1 + y_2)(1 + x_1x_2)}{(1 - x_1x_2)(1 + y_1y_2)} \right). \tag{4}$$

2.4 Hessian model

The Hessian form of an elliptic curve, introduced in [14] (also in [17,24,33,46,53]), is defined by the equation $x^3+y^3+1 = 3dxy$, with a point at infinity $\mathcal{O} = (1, -1, 0)$ as neutral element. The group law is given by

$$(x_3, y_3) = \begin{cases} \left(\frac{y_1^2x_2-y_2^2x_1}{x_2y_2-x_1y_1}, \frac{x_1^2y_2-x_2^2y_1}{x_2y_2-x_1y_1} \right) & \text{if } (x_1, y_1) \neq (x_2, y_2) \\ \left(\frac{y_1(1-x_1^3)}{x_1^3-y_1^3}, \frac{x_1(y_1^3-1)}{x_1^3-y_1^3} \right) & \text{if } (x_1, y_1) = (x_2, y_2). \end{cases} \tag{5}$$

2.5 Twisted Hessian model

The twisted Hessian form [6,8] is defined by equation $ax^3 + y^3 + 1 = dxy$, with neutral element $\mathcal{O} = (0, -1)$. Unlike the original Hessian form, twisted Hessian curves have an affine neutral point and complete addition formula

$$(x_3, y_3) = \left(\frac{x_1 - y_1^2 x_2 y_2}{a x_1 y_1 x_2^2 - y_2}, \frac{y_1 y_2^2 - a x_1^2 x_2}{a x_1 y_1 x_2^2 - y_2} \right). \tag{6}$$

2.6 Twisted Jacobi intersections

Jacobi intersections were suggested by Chudnovsky and Chudnovsky [14], and were among the first competitive candidates for fast single-coordinate arithmetic[4]. Here we present Hisil et al.'s generalization [26], defined by the intersection of $bs^2 + c^2 = 1$ and $as^2 + d^2 = 1$, with neutral affine point $\mathcal{O} = (0, 1, 1)$ and complete addition formula

$$(s_3, c_3, d_3) = \left(\frac{s_1 c_2 d_2 + c_1 d_1 s_2}{1 - ab s_1^2 s_2^2}, \frac{c_1 c_2 - b s_1 d_1 s_2 d_2}{1 - ab s_1^2 s_2^2}, \frac{d_1 d_2 - a s_1 c_1 s_2 c_2}{1 - ab s_1^2 s_2^2} \right). \tag{7}$$

2.7 Extended Jacobi quartics

Extended Jacobi quartics [14,26] are defined by the equation $y^2 = dx^4 + 2ax^2 + 1$, with $\mathcal{O} = (0, 1)$ and group law

$$(x_3, y_3) = \left(\frac{x_1 y_2 + y_1 x_2}{1 - d x_1^2 x_2^2}, \frac{(1 + d x_1^2 x_2^2)(y_1 y_2 + 2a x_1 x_2) + 2d x_1 x_2 (x_1^2 + x_2^2)}{(1 - d x_1^2 x_2^2)^2} \right). \tag{8}$$

3 Invalid Curve Attacks

3.1 Review of the Weierstrass curve case

We begin by describing the classic invalid curve attack against short Weierstrass curves $E_{a,b}\colon y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_p$. The key insight is that formulas defining the arithmetic on that curve, given by Eq. (1), do not depend on the parameter b of the curve equation. All the curves $E_{a,b'}$ for all b' actually share the same addition and doubling formulas.

Now consider a cryptographic device that performs scalar multiplications in $E_{a,b}(\mathbb{F}_p)$ by a constant secret scalar k, and that, furthermore, does not check that input points actually belong to that curve. An attacker trying to recover k can then query the device on an invalid point $\widetilde{P} = (\tilde{x}, \tilde{y}) \notin E_{a,b}(\mathbb{F}_p)$. That point belongs to a well-defined curve of the form $E_{a,b'}$, namely $E_{a,\tilde{b}}$ with $\tilde{b} = \tilde{y}^2 - \tilde{x}^3 - a\tilde{x}$.

[4] Miller [44] also suggested x-only arithmetic for Diffie–Hellman. However he suggested using division polynomials for scalar multiplication, which is far more computationally expensive.

As a result, on input $\widetilde{P}$, the device actually computes the scalar multiplication $k \cdot \widetilde{P}$ in the group $E_{a,\tilde{b}}(\mathbb{F}_p)$ and returns that value.

The discrete logarithm problem in the subgroup $\langle\widetilde{P}\rangle$ generated by $\widetilde{P}$ in $E_{a,\tilde{b}}(\mathbb{F}_p)$ will typically be much easier than in the original group $E_{a,b}(\mathbb{F}_p)$, and the attacker can even choose the invalid point and curve to make the problem particularly easy. This allows him to efficiently recover k modulo the order of $\langle\widetilde{P}\rangle$, and then all of k by repeating the process a few times with different invalid curves.

The whole attack can thus be summarized as follows:

1. Find a curve $E_{a,\tilde{b}}(\mathbb{F}_p)$ and a point $\widetilde{P}$ on it such that discrete logarithms in $\langle\widetilde{P}\rangle$ are easy;
2. Query the cryptographic device on $\widetilde{P}$ to get $k \cdot \widetilde{P}$;
3. Solve the discrete logarithm in the easy group, revealing $k \bmod \operatorname{ord}(\widetilde{P})$;
4. Repeat until k is recovered in its entirety.

Finding a curve and point such that discrete logarithms are easy can be done in several different ways. The original approach, inspired by [41], was to use invalid curves containing subgroups of very small orders and an input point in those subgroups; such curves are easy to find, but quite a few queries are needed to recover all of k.

Another approach is to use a curve of smooth order [43]: this is somewhat harder to construct, but may allow a full recovery of k in a single query. Alternatively, using a singular curve [35] yields a discrete logarithm problem in a form of the multiplicative group over $\mathbb{F}_p$ (or the additive group when $a = 0$), which is typically easy to solve and again makes the single-query recovery of k possible [28, §3.7].

The attack also extends to the situation when the cryptographic device only returns a hash of the resulting point of the scalar multiplication (the hashed Diffie–Hellman setting): in that case, the small subgroup approach is typically the most efficient. That is the approach taken by Jager, Schwenk and Somorovsky in their paper attacking ECDH key exchange in actually deployed TLS libraries [29].

3.2 Parameter-independent formulas

The invalid curve attack translates easily to the case of alternate curve models for which the addition and doubling formulas are independent of at least one of the curve parameters: when querying the cryptographic device on a point $\widetilde{P}$ outside of the valid curve E, the computations still amount to a scalar multiplication on a different curve $\widetilde{E}$ in the same family, obtained by adjusting the independent parameter appropriately.

This is the case for (twisted) Hessian and Huff curves. Additionally, efficient d-less formulas exist for Edwards curves (cf. Eq. (3)), Jacobian quartics and Jacobian intersections [26].

On the other hand, in the case of addition laws depending on all curve parameters, the result of sending an arbitrary invalid input point to the device can no longer be interpreted as a scalar multiplication on a well-defined invalid curve: the attack of Antipa et al. does not generalize directly to that setting.

3.3 Our new approach: the degenerate curve attack against Edwards curves

As is easily observed in Eq. (2), the typical Edwards addition formulas depend on all curve parameters and are therefore not vulnerable to the original invalid curve attack as described above. However, there is one interesting property of this addition law that helps us transfer elliptic curve discrete logarithms to the curve's underlying field, rendering them solvable by sieve methods [16, 21].

Theorem 1. *Let $E_{a,d}$ be a twisted Edwards curve over $\mathbb{F}_p$. The subset $\widetilde{G} \subset \mathbb{F}_p^2$ of the affine plane consisting of points of the form $(0, y)$, $y \neq 0$, endowed with the addition law defined by the same formula as $E_{a,d}$, given by Eq. (2), forms a group isomorphic to $\mathbb{F}_p^*$ under the isomorphism $y \mapsto (0, y)$.*

Proof. The map $\varphi\colon \mathbb{F}_p^* \to \widetilde{G}$, $y \mapsto (0, y)$ is by definition a bijection. It suffices to check that it is a homomorphism to conclude. But this is indeed the case since adding the points $(0, y_1)$ and $(0, y_2)$ yields, according to Eq. (2):

$$\varphi(y_1) + \varphi(y_2) = \left(\frac{0 \cdot y_2 + y_1 \cdot 0}{1 + d \cdot 0 \cdot 0 \cdot y_1 y_2}, \frac{y_1 y_2 - a \cdot 0 \cdot 0}{1 - d \cdot 0 \cdot 0 \cdot y_1 y_2} \right) = (0, y_1 y_2) = \varphi(y_1 y_2)$$

as required. □

As a result, given a cryptographic device performing scalar multiplications in the group $E_{a,d}(\mathbb{F}_p)$ without input point validation, as in the original attack of §3.1, an attacker can send as input an invalid point $\widetilde{P}$ of the form $(0, \tilde{y})$, and receive as result the scalar multiplication of $\widetilde{P}$ by the secret k in the group $\widetilde{G}$, namely $(0, \tilde{y}^k)$. Therefore, recovering k is reduced to solving the discrete logarithm problem in the multiplicative group $\mathbb{F}_p^*$, which as we have mentioned above is much easier than in $E_{a,d}(\mathbb{F}_p)$ owing to well-known subexponential attacks.

For elliptic curve sizes used in practice (up to 500 or so bits), the finite field discrete log is easy! By choosing y as a generator of $\mathbb{F}_p^*$ (which is always a cyclic group), the attacker can thus recover all of k in a single query. This yields our generalization of invalid curve attacks to the case of Edwards curves: we call this attack a *degenerate curve attack* for reasons that will become apparent shortly.

Remark 1. An obvious but important observation is that, while we have described our attack in affine coordinates, it also works in the (likely) case when the device performs its computation in projective coordinates, using the projective versions of the same group operations. It is straightforward to check, for example, that $(0 : Y_1 : 1) + (0 : Y_2 : 1) = (0 : Y_1 Y_2 : 1)$ (and generalizations with other values of the Z-coordinates go through similarly).

One can wonder why, despite the dependence of the group law Eq. (2) on all curve parameters, we can still find an invalid curve in the affine plane where the same formulas induce a group structure. A rough explanation is as follows. First, the y-axis $Y\colon x = 0$ in the plane is actually a limit (in the usual sense of one-parameter families) of the twisted Edwards curves $E_{a,d}$ for fixed d: it is the fiber above $a = \infty$. This is easily seen by rewriting the equation of $E_{a,d}$ in terms of $a' = 1/a$, as $x^2 + a'y^2 = a'(1 + dx^2y^2)$, and setting $a' = 0$. Since Y is of genus 0, the Edwards group law should degenerate on Y (minus a finite number of points) as the additive or the multiplicative group. The expression of the group law need not a priori be the same as on the original curve $E_{a,d}$ itself, but it does turn out to be the case, because the only term depending on the parameter a cancels out along $Y\colon x = 0$.

Now the line Y is not itself singular (although it should perhaps really be seen as the non-reduced double line $x^2 = 0$), but it is where the family degenerates, hence the name of our attack.

3.4 Degenerate curve attacks against other models

The idea of the previous attack generalizes easily to other models of elliptic curves, including all of those mentioned in §2. We now describe those generalizations in affine coordinates below; they of course also work in projective coordinates.

Extended Jacobi quartics. Let $E_{a,b}\colon y^2 = dx^4 + 2ax^2 + 1$ be an extended Jacobi quartic curve over $\mathbb{F}_p$, and consider the set $\widetilde{G}$ of points in $\mathbb{F}_p^2$ of the form $(0, y)$, $y \neq 0$. Endow this set with the same addition law as $E_{a,d}$, defined by Eq. (8). It then forms a group isomorphic to $\mathbb{F}_p^*$ under the isomorphism $\varphi\colon y \mapsto (0, y)$. Indeed, this map is a bijection and we have:

$$\begin{aligned}\varphi(y_1) + \varphi(y_2) &= \left(\frac{0 \cdot y_2 + y_1 \cdot 0}{1 - d \cdot 0 \cdot 0}, \frac{(1 + d \cdot 0 \cdot 0)(y_1y_2 + 2a \cdot 0 \cdot 0) + 2d \cdot 0 \cdot 0 \cdot 0}{(1 - d \cdot 0 \cdot 0)^2}\right)\\ &= (0, y_1y_2) = \varphi(y_1y_2),\end{aligned}$$

so φ is an isomorphism as required.

Therefore, we can carry out our attack as before, by sending to a device performing scalar multiplications on $E_{a,d}$ the invalid input point $(0, y)$ for some generator y of $\mathbb{F}_p^*$.

In this case, the y-axis appears as the degenerate limit of the family $E_{a,d}$ for fixed a and varying d, taken for $d = \infty$.

Twisted Jacobi intersections. Let $E_{a,b}\colon as^2 + c^2 = bs^2 + d^2 = 1$ be a twisted Jacobi intersection over $\mathbb{F}_p$, and consider the sets $\widetilde{G}_1$ and $\widetilde{G}_2$ of points in $\mathbb{F}_p^3$ of the form $(0, c, 0)$, $c \neq 0$, and $(0, 0, d)$, $d \neq 0$, respectively. Endow both of these sets with the same addition law as $E_{a,b}$, defined by Eq. (7). Then they form groups isomorphic to $\mathbb{F}_p^*$ under the isomorphisms $\varphi_1\colon c \mapsto (0, c, 0)$ and $\varphi_2\colon d \mapsto (0, 0, d)$

respectively. Indeed, those maps are both bijections and we have:

$$\begin{aligned}\varphi_1(c_1)+\varphi_1(c_2) &= \left(\frac{0\cdot c_2\cdot 0+c_1\cdot 0\cdot 0}{1-ab\cdot 0\cdot 0}, \frac{c_1c_2-b\cdot 0\cdot 0\cdot 0\cdot 0}{1-ab\cdot 0\cdot 0}, \frac{0\cdot 0-b\cdot 0\cdot c_1\cdot 0\cdot c_2}{1-ab\cdot 0\cdot 0}\right)\\ &= (0,c_1c_2,0)=\varphi_1(c_1c_2)\end{aligned}$$

and similarly for φ_2 by symmetry.

This provides two families of invalid points using which we can carry out our attack exactly as before.

Twisted Hessian curves. The case of twisted Hessian curves is somewhat less interesting, since this model has a group law independent of the curve parameter d, and hence the original invalid curve attack applies to it. Nevertheless, we can mention for completeness that our approach generalizes rather directly to those curves as well.

Indeed, if $E_{a,d}\colon ax^3+y^3+1=dxy$ is a twisted Hessian curve, the map $\varphi\colon y\mapsto(0,-y)$ defines an isomorphism between $\mathbb{F}_p^*$ and the set of elements of the form $(0,y)$, $y\neq 0$ in $\mathbb{F}_p^2$ endowed with the same addition law as $E_{a,d}$, defined by Eq. (6). Indeed:

$$\begin{aligned}\varphi(y_1)+\varphi(y_2) &= \left(\frac{0+y_1^2\cdot 0\cdot y_2}{-a\cdot 0\cdot y_1\cdot 0+y_2}, \frac{-y_1y_2^2-a\cdot 0\cdot 0}{-a\cdot 0\cdot y_1\cdot 0+y_2}\right)\\ &= (0,-y_1y_2)=\varphi(y_1y_2).\end{aligned}$$

Huff curves. As with Hessian curves, Huff curves have a parameter-independent group law and hence are not the most relevant setting for us, but we can again extend our attack to them.

For the Huff curve $E_{a,b}\colon ax(y^2-1)=by(x^2-1)$ with the group law of Eq. (4), we can consider the set $\widetilde{G}$ of points in $\mathbb{F}_p^2$ of the form $(0,y)$. The sum of two such points under the addition law given by the same formula is given by:

$$(0,y_1)+(0,y_2)=\left(\frac{0\cdot(1+y_1y_2)}{1\cdot(1-y_1y_2)}, \frac{(y_1+y_2)\cdot 1}{1\cdot(1+y_1y_2)}\right)=\left(0,\frac{y_1+y_2}{1+y_1y_2}\right).$$

Thus, if we consider the map $\varphi\colon\mathbb{F}_p^*\to\widetilde{G}$ defined outside -1 by $\varphi(t)=(0,(1-t)/(1+t))$, it is easy to check that $\varphi(t_1)+\varphi(t_2)=\varphi(t_1t_2)$, and therefore we again have a group isomorphic to $\mathbb{F}_p^*$ to carry out our attack.

Remark 2. It may be worth noting that for some curve models, we are also able to find degenerate curves on which the addition law induces a group structure isomorphic to the twisted form of the multiplicative group (i.e. the subgroup of order $p+1$ of elements of norm 1 in $\mathbb{F}_{p^2}^*$). Huff curves offer a simple concrete example: consider the set of points of the form $(x,x)\in\mathbb{F}_p^2$ with the Huff addition law of Eq. (4). The sum of two such points is given by $(x_1,x_1)+(x_2,x_2)=(x_3,x_3)$ where

$$x_3=\frac{x_1+x_2}{1-x_1x_2}.$$

When -1 is a quadratic nonresidue in $\mathbb{F}_p$, this is well-known to be the so-called "compressed form" of the twisted multiplicative group [49].

4 The Hashed Case

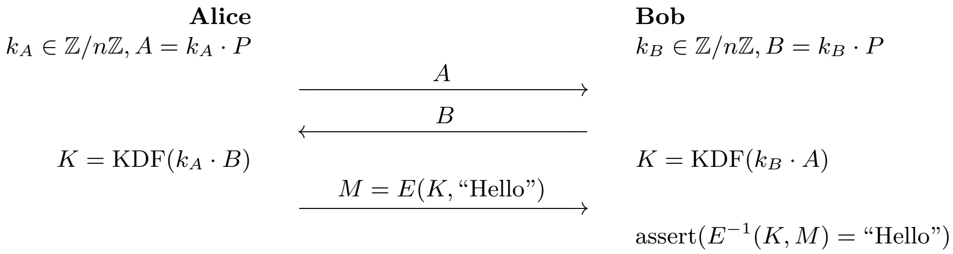
Alice
$k_A \in \mathbb{Z}/n\mathbb{Z}, A = k_A \cdot P$
Bob
$k_B \in \mathbb{Z}/n\mathbb{Z}, B = k_B \cdot P$
A
B
$K = \mathrm{KDF}(k_A \cdot B)$
$K = \mathrm{KDF}(k_B \cdot A)$
$M = E(K, \text{“Hello”})$
$\mathrm{assert}(E^{-1}(K, M) = \text{“Hello”})$

Figure 1. Basic unauthenticated elliptic curve Diffie–Hellman protocol, under which invalid curve attacks may be mounted. The protocol works over a curve $E_{a,b}(\mathbb{F}_p)$, with a generator point P of prime order n. $\mathrm{KDF}(\cdot)$ is an arbitrary key-derivation function taking points of $E_{a,b}(\mathbb{F}_p)$ as input; $E(K, M)$ is taken to be some authenticated encryption primitive, e.g., AES–GCM.

The previous section considered attacks on a cryptographic device that performs elliptic curve scalar multiplications without validation of input points, and returns the actual result of the scalar multiplication. This is a somewhat idealized attack model, however.

One real-world protocol where a similar situation arises is (static) Diffie–Hellman key exchange over elliptic curves, one variant of which is presented in Fig. 1. In an invalid curve attack on that protocol, Bob would send Alice his invalid point B, and Alice would use it to compute the product $k_A \cdot B$ where k_A is her static secret key. The resulting point $k_A \cdot B$ is not directly sent back to Bob, however, but used to derive a key $K = \mathrm{KDF}(k_A \cdot B)$ used in subsequent communication. In effect, what Bob receives is the image of $k_A \cdot B$ under a fixed, public one-way function, usually with low collision probability (in Fig. 1, it would be the authentication message M).

We model that situation by considering an oracle which, on input of a point P (still unvalidated), computes the scalar multiplication $k \cdot P$ by a fixed secret k, and returns the image $H(k \cdot P)$ of the result under a public hash function H. In that more restrictive setting, degenerate curve attacks are not as devastating as previously described, but we will see that it is often still possible to recover k quite quickly in practice, depending on the smoothness of the order $p-1$ of $\mathbb{F}_p^*$ (or of $p+1$ in the case of degenerate groups isomorphic to the twisted multiplicative group; we will describe the attack in the $\mathbb{F}_p^*$ case to fix ideas).

The idea is simply to apply the Pohlig–Hellman algorithm [47]. Using the naive variant of the algorithm, the attacker can, for each prime divisor ℓ of $p-1$, choose a point $\widetilde{P}$ of order ℓ in the degenerate group, obtain $H(k \cdot \widetilde{P})$ from the oracle, and perform an exhaustive search in the subgroup $\langle \widetilde{P} \rangle$ to find the point $\widetilde{Q}$ such that $H(k \cdot \widetilde{P}) = H(\widetilde{Q})$, revealing $k \bmod \ell$. Prime powers are dealt with similarly, and in the end the attacker recovers all of k with only a few oracle queries, in time quasilinear in the largest prime factor $P_1(p-1)$ of $p-1$. Furthermore, if a higher query complexity is acceptable, we can use Shanks' baby-step giant-step time-memory tradeoff [51] to recover k in time quasilinear in $\sqrt{P_1(p-1)}$, also using a number of queries and a space complexity quasilinear in $\sqrt{P_1(p-1)}$.

In general, even $\sqrt{P_1(p-1)}$ need not be much smaller than the complexity of the discrete logarithm problem in the original curve. However, newer models like Edwards curves are often used over special base fields $\mathbb{F}_p$ with particularly efficient arithmetic. Table 1 lists those efficient primes for usual curve sizes together with the bit size of $P_1(p-1)$, and we can see that for many of them, the degenerate curve attack is quite efficient: for example, for curves over the Mersenne prime field $\mathbb{F}_{2^{521}-1}$ (used to construct the highest security elliptic curves, including E-521 [2]), the complexity of an $\mathbb{F}_p^*$ degenerate curve attack would be around $O(2^{44})$, which is very practical. And it would be $O(2^{57.5})$, also quite fast, over $\mathbb{F}_{2^{448}-2^{224}-1}$, the field of definition of Ed448-Goldilocks [22].

5 A Fault Attack Countermeasure

Soon after the announcement of the Bellcore attack on RSA, Shamir proposed a countermeasure [50] that relies on the Chinese remainder theorem to detect faults during modular exponentiation. The basic idea of Shamir is to replace computations modulo a prime p by computations in the ring modulo the composite pr, where r is a small randomly-selected integer, and then compare the result modulo r against an independent equivalent computation modulo r.

While Shamir's trick[5] works well on RSA, due to its simple structure, it is trickier to apply this countermeasure to the elliptic curve case. Nevertheless, countermeasures based on Shamir's trick have been devised. The first one was invented by Blömer, Otto, and Seifert [11] (BOS), and consisted of two elliptic curve scalar multiplications—one over $\mathbb{Z}/pr\mathbb{Z}$, the other over $\mathbb{Z}/r\mathbb{Z}$. Baek and Vasyltov [3] suggested the use of the curve $Y^2Z + pYZ^3 = X^3 + aXZ^4 + BZ^6 \in \mathbb{Z}/pr\mathbb{Z}$, where $B = y^2 + py - x^3 - ax$, which clearly is equivalent to the original when reduced modulo p. This method is limited to projective coordinates, since not every intermediate result may have an inverse in the extended ring. Their method also has some potential weaknesses owing to its reliance on random integers r instead of adequately selected primes [31]. It has been recently pointed out that the original BOS countermeasure is not correct when coupled with group laws containing exceptions [48], and thus group laws used in BOS-like countermeasures must be *test-free*.

More recently, Joye [30,32] proposed a variant of the BOS countermeasure, where one works instead over $\mathbb{Z}/pr^2\mathbb{Z}$ (resp. $\mathbb{Z}/r^2\mathbb{Z}$). To accelerate the second scalar multiplication, Joye takes advantage of the isomorphism between the set of points of $E(\mathbb{Z}/r^2\mathbb{Z})$ that reduce to the neutral point modulo r, and the additive

[5] Not to be confused with Shamir's double-exponentiation trick, pointed out by ElGamal [19, p. 471] and originally discovered by Straus [54].

Table 1. For primes p suitable for fast elliptic curve cryptography [23], size in bits of the largest prime factor of $p-1$ and $p+1$, and complexity of our BSGS-style hashed Diffie–Hellman attack in $\mathbb{F}_p^*$ ($(p-1)$ attack) and in the twisted multiplicative group ($(p+1)$ attack).

p	$\log_2 P_1(p-1)$	$(p-1)$ attack	$\log_2 P_1(p+1)$	$(p+1)$ attack
$2^{191}-19$	90	$O(2^{45})$	93	$O(2^{46.5})$
$2^{196}-15$	64	$O(2^{32})$	165	$O(2^{82.5})$
$2^{216}-2^{108}-1$	107	$O(2^{53.5})$	19	$O(2^{9.5})$
$2^{221}-3$	73	$O(2^{36.5})$	42	$O(2^{21})$
$2^{224}-2^{96}+1$	46	$O(2^{23})$	157	$O(2^{78.5})$
$2^{226}-5$	127	$O(2^{63.5})$	49	$O(2^{24.5})$
$2^{230}-27$	101	$O(2^{50.5})$	136	$O(2^{68})$
$2^{251}-9$	235	$O(2^{117.5})$	70	$O(2^{35})$
$2^{255}-19$	236	$O(2^{118})$	95	$O(2^{47.5})$
$2^{266}-3$	37	$O(2^{17.5})$	125	$O(2^{62.5})$
$2^{285}-9$	237	$O(2^{118.5})$	60	$O(2^{30})$
$2^{291}-19$	259	$O(2^{129.5})$	114	$O(2^{57})$
$2^{322}-2^{161}-1$	133	$O(2^{66.5})$	64	$O(2^{32})$
$2^{336}-3$	166	$O(2^{83})$	214	$O(2^{107})$
$2^{338}-15$	166	$O(2^{83})$	204	$O(2^{102})$
$2^{369}-25$	192	$O(2^{96})$	252	$O(2^{126})$
$2^{383}-31$	88	$O(2^{44})$	97	$O(2^{48.5})$
$2^{389}-21$	247	$O(2^{123.5})$	311	$O(2^{155.5})$
$2^{401}-31$	48	$O(2^{24})$	209	$O(2^{104.5})$
$2^{416}-2^{208}-1$	60	$O(2^{30})$	96	$O(2^{48})$
$2^{448}-2^{224}-1$	115	$O(2^{57.5})$	49	$O(2^{24.5})$
$2^{450}-2^{225}-1$	88	$O(2^{44})$	54	$O(2^{27})$
$2^{452}-3$	88	$O(2^{44})$	266	$O(2^{133})$
$2^{468}-17$	209	$O(2^{104.5})$	164	$O(2^{82})$
$2^{480}-2^{240}-1$	163	$O(2^{81.5})$	36	$O(2^{18})$
$2^{489}-21$	263	$O(2^{131.5})$	260	$O(2^{130})$
$2^{495}-31$	158	$O(2^{79})$	319	$O(2^{159.5})$
$2^{521}-1$	88	$O(2^{44})$	1	$O(2^{0.5})$

Algorithm 1: Fault attack countermeasure for elliptic curves with degenerate points allowing "shortcut" scalar multiplications.

```
Input:
A curve $E(\mathbb{F}_p)$;
A point $P = (x, y) \in E(\mathbb{F}_p)$;
A scalar exponent $k \in \mathbb{Z}$;
A security parameter $b$;
An efficiently-computable "shortcut" map $f(k, P) : E(\mathbb{F}_r) \to E(\mathbb{F}_r)$ implementing scalar multiplication by $k$.
Output: $k \cdot P$
begin
    $r \leftarrow$ random $b$-bit prime
    $E_r \leftarrow \mathrm{DegenerateCurve}(r)$   // Pick degenerate curve, model-dependent
    $P_r \leftarrow (x_r, y_r) \in E_r(\mathbb{F}_r)$   // Pick appropriate degenerate point on $E_r$
    $E' \leftarrow E \times E_r \ / \ \mathbb{Z}/pr\mathbb{Z}$
    $P' \leftarrow \Big(\mathrm{CRT}_{p,r}(x(P), x_r), \mathrm{CRT}_{p,r}(y(P), y_r)\Big) \in E'(\mathbb{Z}/pr\mathbb{Z})$
    $Q' \leftarrow k \cdot P'$
    if $Q' \bmod r \neq f(k, P' \bmod r)$ then   // Check for fault
        return "error"
    else
        return $\Big(x(Q') \bmod p, y(Q') \bmod p\Big)$
    end
end
```

group $\mathbb{F}_r^+$. For example, the set of affine points $(\alpha r, 1) \in E(\mathbb{Z}/r^2\mathbb{Z})$, coupled with the Edwards group law, yields the useful identity $k \cdot (\alpha r, 1) = (k \cdot \alpha r, 1) \pmod{r^2}$, which can be used to detect a fault very efficiently. Our proposed countermeasure is conceptually similar, but takes advantage of the multiplicative and additive identities of degenerate curves described in §3 instead. The countermeasure is described, in its most general form, in Algorithm 1.

One can view our proposed countermeasure as the BOS [11] countermeasure coupled with a "shortcut" $f(k, P)$ to compute the second scalar multiplication—$k \cdot P$ in $E(\mathbb{F}_r)$—much faster than by using the standard formulas. This shortcut takes different forms depending on which curve shape we are working over. Generically, we begin by picking a curve E_r over $\mathbb{F}_r$ for which there is at least one point for which scalar multiplication is easy to compute. Then, the extended curve E' is the direct product $E'(\mathbb{Z}/pr\mathbb{Z}) = E(\mathbb{F}_p) \times E_r(\mathbb{F}_r)$, and the countermeasure consists of checking whether $k \cdot P' \in E'$, reduced modulo r, equals the same multiplication performed independently in E_r. The correctness of this method follows from the correctness of BOS [11]; our concrete contribution is the shortcuts taken to reduce the computation overhead of the scalar multiplication in E_r. The following considers two popular shapes—Weierstrass and Edwards curves—but others are similarly easy to derive.

5.1 Weierstrass curves

In Weierstrass curves, we may take advantage of the unique singular curve $y^2 = x^3$. This curve is notable for degenerating into the *additive* group $\mathbb{F}_r^+$ via the map $(x, y) \mapsto x/y$ and $\infty \mapsto 0$, with inverse $t \mapsto (t^{-2}, t^{-3})$ and $0 \mapsto \infty$ [28, §3.7]. This immediately suggests a very efficient shortcut map for E_r:

$$f(k, P) = \left((kt)^{-2}, (kt)^{-3}\right),$$

where $t = x/y$ or $t = 0$ if $P = \infty$.

The resulting correctness test only requires a few multiplications modulo r, which is more efficient than both BOS [11] and Baek–Vasyltsov [3], and is comparable with Joye's approach [30]. Note that the inversions are avoidable by using projective coordinates.

5.2 Edwards curves

Unlike Weierstrass curves, Edwards curves do not have any additive degeneration. However, we can use the results of §3.3 to devise a similar countermeasure using a multiplicative degeneration. The shortcut map for E_r is

$$f(k, P) = \left(0, y^k\right) \ ,$$

where $P = (0, y)$ for any $y \notin \{0, 1\}$ that generates the group $\mathbb{F}_r^*$. In this case the computational overhead is larger than in the Weierstrass case—a modular exponentiation modulo r—but is still far cheaper than a scalar multiplication.

5.3 Comparison with previous countermeasures

The above methods offer some advantages relatively to previous Shamir-inspired fault attack countermeasures:

Only one full-fledged scalar multiplication is required. This is in contrast with Blömer–Otto–Seifert [11, §8] which requires 2 scalar multiplications—one modulo pr, another modulo r. In the case of Weierstrass curves, our countermeasure is faster than any other targeting the same curve shape.

Works both in affine and projective coordinates This is in contrast with Baek–Vasyltsov [3], which due to working on Weierstrass curves, breaks down when faced with the corner cases in the addition and doubling formulas of those curves.

Although our method may not suit every use case, it is another useful tool for hardened implementations of elliptic curves. It is particularly suitable for implementations of curves over random primes, which hardware implementers tend to favor [42], since multiplication modulo pr is straightforward to implement, and the overhead remains small. On the other hand, highly structured primes, usually very close to a power of 2, would likely suffer a higher performance impact, since modular reduction would no longer be a linear-time operation.

References

1. Antipa, A., Brown, D.R.L., Menezes, A., Struik, R., Vanstone, S.A.: Validation of elliptic curve public keys. In: Desmedt, Y. (ed.) PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography. Lecture Notes in Computer Science, vol. 2567, pp. 211–223. Springer, Heidelberg, Germany, Miami, USA (Jan 6–8, 2003)
2. Aranha, D.F., Barreto, P.S.L.M., Pereira, G.C.C.F., Ricardini, J.E.: A note on high-security general-purpose elliptic curves. Cryptology ePrint Archive, Report 2013/647 (2013), http://eprint.iacr.org/
3. Baek, Y., 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.) Information Security Practice and Experience, Third International Conference, ISPEC 2007, Hong Kong, China, May 7–9, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4464, pp. 225–237. Springer (2007), http://dx.doi.org/10.1007/978-3-540-72163-5_18
4. Bernstein, D.J.: Curve25519: New Diffie-Hellman speed records. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006: 9th International Conference on Theory and Practice of Public Key Cryptography. Lecture Notes in Computer Science, vol. 3958, pp. 207–228. Springer, Heidelberg, Germany, New York, NY, USA (Apr 24–26, 2006)
5. Bernstein, D.J., Birkner, P., Joye, M., Lange, T., Peters, C.: Twisted Edwards curves. In: Vaudenay, S. (ed.) AFRICACRYPT 08: 1st International Conference on Cryptology in Africa. Lecture Notes in Computer Science, vol. 5023, pp. 389–405. Springer, Heidelberg, Germany, Casablanca, Morocco (Jun 11–14, 2008)
6. Bernstein, D.J., Chuengsatiansup, C., Kohel, D., Lange, T.: Twisted Hessian curves. In: Lauter, K.E., Rodríguez-Henríquez, F. (eds.) Progress in Cryptology - LATINCRYPT 2015: 4th International Conference on Cryptology and Information Security in Latin America. Lecture Notes in Computer Science, vol. 9230, pp. 269–294. Springer, Heidelberg, Germany, Guadalajara, Mexico (Aug 23–26, 2015)
7. Bernstein, D.J., Lange, T.: Faster addition and doubling on elliptic curves. In: Kurosawa, K. (ed.) Advances in Cryptology – ASIACRYPT 2007. Lecture Notes in Computer Science, vol. 4833, pp. 29–50. Springer, Heidelberg, Germany, Kuching, Malaysia (Dec 2–6, 2007)
8. Bernstein, D.J., Lange, T.: Explicit-formulas database (2015), https://hyperelliptic.org/EFD/, accessed May 1st, 2015.
9. Biehl, I., Meyer, B., Müller, V.: Differential fault attacks on elliptic curve cryptosystems. In: Bellare, M. (ed.) Advances in Cryptology – CRYPTO 2000. Lecture Notes in Computer Science, vol. 1880, pp. 131–146. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2000)
10. Billet, O., Joye, M.: The Jacobi model of an elliptic curve and side-channel analysis. In: Proceedings of the 15th International Conference on Applied Algebra, Algebraic Algorithms and Error-correcting Codes. pp. 34–42. AAECC'03, Springer-Verlag, Berlin, Heidelberg (2003), https://eprint.iacr.org/2002/125
11. Blömer, J., Otto, M., Seifert, J.: Sign change fault attacks on elliptic curve cryptosystems. In: Breveglieri, L., Koren, I., Naccache, D., Seifert, J. (eds.) Fault Diagnosis and Tolerance in Cryptography, Third International Workshop, FDTC 2006, Yokohama, Japan, October 10, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4236, pp. 36–52. Springer (2006), http://dx.doi.org/10.1007/11889700_4

12. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of checking cryptographic protocols for faults (extended abstract). In: Fumy, W. (ed.) Advances in Cryptology – EUROCRYPT'97. Lecture Notes in Computer Science, vol. 1233, pp. 37–51. Springer, Heidelberg, Germany, Konstanz, Germany (May 11–15, 1997)
13. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of eliminating errors in cryptographic computations. Journal of Cryptology 14(2), 101–119 (2001)
14. Chudnovsky, D.V., Chudnovsky, G.V.: Sequences of numbers generated by addition in formal groups and new primality and factorization tests. Advances in Applied Mathematics 7(4), 385–434 (Dec 1986), http://dx.doi.org/10.1016/0196-8858(86)90023-0
15. Ciet, M., Joye, M.: Elliptic curve cryptosystems in the presence of permanent and transient faults. Des. Codes Cryptography 36(1), 33–43 (2005), http://dx.doi.org/10.1007/s10623-003-1160-8
16. Coppersmith, D., Odlyzko, A.M., Schroeppel, R.: Discrete logarithms in $GF(p)$. Algorithmica 1(1), 1–15 (Jan 1986), http://dx.doi.org/10.1007/BF01840433
17. Desboves, A.: Résolution, en nombres entries et sous la forme la plus générale, de l'équation cubique, homogène, à trois inconnues. Nouvelles annales de mathématiques, journal des candidats aux écoles polytechnique et normale 5(3), 545–579 (1886), http://www.numdam.org/item?id=NAM_1886_3_5__545_0
18. Edwards, H.M.: A normal form for elliptic curves. Bulletin of the American Mathematical Society 44(3), 393–422 (2007), http://dx.doi.org/10.1090/S0273-0979-07-01153-6
19. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory 31, 469–472 (1985)
20. Fouque, P., Lercier, R., Réal, D., Valette, F.: Fault attack on elliptic curve Montgomery ladder implementation. In: Breveglieri, L., Gueron, S., Koren, I., Naccache, D., Seifert, J. (eds.) Fifth International Workshop on Fault Diagnosis and Tolerance in Cryptography, 2008, FDTC 2008, Washington, DC, USA, 10 August 2008. pp. 92–98. IEEE Computer Society (2008), http://dx.doi.org/10.1109/FDTC.2008.15
21. Gordon, D.M.: Discrete logarithms in $GF(p)$ using the number field sieve. SIAM J. Discret. Math. 6(1), 124–138 (Feb 1993), http://dx.doi.org/10.1137/0406010
22. Hamburg, M.: Ed448-Goldilocks. Workshop on Elliptic Curve Cryptography Standards (2015)
23. Harris, B., et al.: The Pareto frontiers of sleeveless primes. The Curves mailing list (Oct 2014), https://moderncrypto.org/mail-archive/curves/2014/000324.html
24. Hesse, O.: Über die Elimination der Variabeln aus drei algebraischen Gleichungen vom zweiten Grade mit zwei Variabeln. Journal für die reine und angewandte Mathematik 28, 68–96 (1844), http://resolver.sub.uni-goettingen.de/purl?GDZPPN002144069
25. Hisil, H., Wong, K.K.H., Carter, G., Dawson, E.: Twisted Edwards curves revisited. In: Pieprzyk, J. (ed.) Advances in Cryptology – ASIACRYPT 2008. Lecture Notes in Computer Science, vol. 5350, pp. 326–343. Springer, Heidelberg, Germany, Melbourne, Australia (Dec 7–11, 2008)
26. Hisil, H., Wong, K.K., Carter, G., Dawson, E.: An exploration of affine group laws for elliptic curves. J. Mathematical Cryptology 5(1), 1–50 (2011), http://dx.doi.org/10.1515/jmc.2011.005
27. Huff, G.B.: Diophantine problems in geometry and elliptic ternary forms. Duke Mathematical Journal 15(2), 443–453 (1948)
28. Husemöller, D.: Elliptic Curves, Graduate Texts in Mathematics, vol. 111. Springer-Verlag, New York, 2 edn. (2004)

29. Jager, T., Schwenk, J., Somorovsky, J.: Practical invalid curve attacks on TLS-ECDH. In: Pernul, G., Ryan, P.Y.A., Weippl, E.R. (eds.) Computer Security - ESORICS 2015 - 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9326, pp. 407–425. Springer, Heidelberg, Germany, Vienna, Austria (2015)
30. Joye, M.: Fault-resistant calculcations on elliptic curves (Jun 2013), http://www.google.com/patents/US8457303, US Patent 8,457,303
31. Joye, M.: On the security of a unified countermeasure. In: Breveglieri, L., Gueron, S., Koren, I., Naccache, D., Seifert, J. (eds.) Fifth International Workshop on Fault Diagnosis and Tolerance in Cryptography, 2008, FDTC 2008, Washington, DC, USA, 10 August 2008. pp. 87–91. IEEE Computer Society (2008), http://dx.doi.org/10.1109/FDTC.2008.8
32. Joye, M.: Elliptic curve cryptosystems in the presence of faults. In: Fischer, W., Schmidt, J. (eds.) 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013. p. 73. IEEE Computer Society (2013), http://conferenze.dei.polimi.it/FDTC13/shared/FDTC-2013-keynote-2.pdf
33. Joye, M., Quisquater, J.J.: Hessian elliptic curves and side-channel attacks. In: Koç, Çetin Kaya., Naccache, D., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2001. Lecture Notes in Computer Science, vol. 2162, pp. 402–410. Springer, Heidelberg, Germany, Paris, France (May 14–16, 2001)
34. Joye, M., Tibouchi, M., Vergnaud, D.: Huff's model for elliptic curves. In: Hanrot, G., Morain, F., Thomé, E. (eds.) Algorithmic Number Theory, 9th International Symposium, ANTS-IX, Nancy, France, July 19-23, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6197, pp. 234–250. Springer (2010), http://dx.doi.org/10.1007/978-3-642-14518-6_20
35. Karabina, K., Ustaoğlu, B.: Invalid-curve attacks on (hyper)elliptic curve cryptosystems. Advances of Mathematics of Communications 4(3), 307–321 (2010), http://cryptolounge.net/pdf/KarUst10.pdf
36. Kim, T., Tibouchi, M.: Bit-flip faults on elliptic curve base fields, revisited. In: Boureanu, I., Owesarski, P., Vaudenay, S. (eds.) ACNS 14: 12th International Conference on Applied Cryptography and Network Security. Lecture Notes in Computer Science, vol. 8479, pp. 163–180. Springer, Heidelberg, Germany, Lausanne, Switzerland (Jun 10–13, 2014)
37. Kim, T., Tibouchi, M.: Invalid curve attacks in a GLS setting. In: Tanaka, K., Suga, Y. (eds.) IWSEC 15: 10th International Workshop on Security, Advances in Information and Computer Security. Lecture Notes in Computer Science, vol. 9241, pp. 41–55. Springer, Heidelberg, Germany, Nara, Japan (Aug 26–28, 2015)
38. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of Computation 48, 203–209 (1987), http://dx.doi.org/10.1090/S0025-5718-1987-0866109-5
39. Lenstra, Jr., H.W.: Factoring integers with elliptic curves. Annals of Mathematics 126(3), 649–673 (Nov 1987), http://www.jstor.org/stable/1971363
40. Liardet, P.Y., Smart, N.P.: Preventing SPA/DPA in ECC systems using the Jacobi form. In: Koç, Çetin Kaya., Naccache, D., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2001. Lecture Notes in Computer Science, vol. 2162, pp. 391–401. Springer, Heidelberg, Germany, Paris, France (May 14–16, 2001)
41. Lim, C.H., Lee, P.J.: A key recovery attack on discrete log-based schemes using a prime order subgroup. In: Kaliski Jr., B.S. (ed.) Advances in Cryptology – CRYPTO'97. Lecture Notes in Computer Science, vol. 1294, pp. 249–263. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 1997)

42. Lochter, M., Merkle, J., Schmidt, J.M., Schütze, T.: Requirements for standard elliptic curves. Cryptology ePrint Archive, Report 2014/832 (2014), http://eprint.iacr.org/2014/832
43. Menezes, A.: Another look at HMQV. Journal of Mathematical Cryptology 1, 47–64 (Jan 2007), http://dx.doi.org/10.1515/JMC.2007.004
44. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) Advances in Cryptology – CRYPTO'85. Lecture Notes in Computer Science, vol. 218, pp. 417–426. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 1986)
45. Montgomery, P.L.: Speeding the Pollard and elliptic curve methods of factorization. Mathematics of Computation 48(177), 243–264 (1987), http://www.ams.org/journals/mcom/1987-48-177/S0025-5718-1987-0866113-7/
46. Mumford, D.: On the equations defining Abelian varieties. I. Inventiones mathematicae 1(4), 287–354 (1966), http://dash.harvard.edu/handle/1/3597241
47. Pohlig, S.C., Hellman, M.E.: An improved algorithm for computing logarithms over GF(p) and its crytographic significance. IEEE Trans. Inf. Theory 24, 106–110 (1978)
48. Rauzy, P., Moreau, M., Guilley, S., Najm, Z.: Using modular extension to provably protect ECC against fault attacks. Cryptology ePrint Archive, Report 2015/882 (2015), http://eprint.iacr.org/2015/882
49. Rubin, K., Silverberg, A.: Compression in finite fields and torus-based cryptography. SIAM J. Comput. 37(5), 1401–1428 (2008)
50. Shamir, A.: How to check modular exponentiation (May 1997), presented at the rump session of EUROCRYPT'97.
51. Shanks, D.: Class number, a theory of factorization, and genera. In: Lewis, D.J. (ed.) 1969 Number Theory Institute. Proceedings of Symposia in Pure Mathematics, vol. 20, pp. 415–440. American Mathematical Society, Providence, Rhode Island (1971)
52. Silverman, J.H.: The Arithmetic of Elliptic Curves, Graduate Texts in Mathematics, vol. 106. Springer-Verlag, 2 edn. (2009), http://www.math.brown.edu/~jhs/AECHome.html
53. Smart, N.P.: The Hessian form of an elliptic curve. In: Koç, Çetin Kaya., Naccache, D., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2001. Lecture Notes in Computer Science, vol. 2162, pp. 118–125. Springer, Heidelberg, Germany, Paris, France (May 14–16, 2001)
54. Straus, E.G.: Addition chains of vectors (problem 5125). The American Mathematical Monthly 71(7), 806–808 (1964), http://www.jstor.org/stable/2310929