Diversity and transparency for ECC

Jean-Pierre Flori* Jérôme Plût* Jean-René Reinhard* Martin Ekerå†

Generating and standardizing elliptic curves to use them in a cryptographic context is a hard task. There have been several attempts to define public elliptic curves for a general cryptographic use, such as NIST FIPS 186–2 curves [53], Brainpool curves [47], SECG curves [58], ANSSI FRP256v1 [41], Curve25519 [7], and OSCCA SM2 [54]. Recent years have seen some distrust cast on previously standardized curves and the emergence of the need to standardize new curves. Different parties have spoken their point of view on the (dis)trust they have on previously standardized curves whether it is because of the properties they satisfy or don't satisfy or the process used to generate them. Such analyses often come with a list of security and performance/implementation-related criteria a curve should satisfy, and a proposal on how to correctly generate such a curve in a way that can be trusted [10, 17, 18, 48, 2], together with a proposal of such a correctly generated curve [7, 17, 2].

We believe it is very important that the international standards do not *a priori* restrict practical uses of ECC to a single elliptic curve or to a very small family of related elliptic curves. Even though no attack might currently be known, the discovery of a weakness of this particular family is always a possibility. As for the choices of the curves themselves and given the current state-of-the-art, some trade-offs between speed and security have to be made. For example, most of the recently proposed curves [7, 17, 2, 54], and in particular the NIST standardized curves, rely on the use of special primes or particular forms of curves to achieve a very high speed, but don't attain optimal bit-security. Moreover, secure implementation of these particular curves may require some specific precautions against various attacks, such as side-channel attacks. Having at least a less speed-optimized but more general-looking curve, defined over a prime field whose characteristic looks random, in general Weierstraß form, and with a prime number of points seems primordial, especially if the former class of curves gets broken in the future. For ECC to be trusted and widely adopted, *diversity* is needed.

Secondly, even when a curve satisfies all common security criteria, whether it is completely generic, or with a few speed-optimized parameters, another criterion for inclusion in international standards is that one should know all the details about how the curve was generated, and be able to verify that the generation process actually ended up with the claimed curve and not anoher one in the same family satisfying the same conditions. Much has been said on this matter [10, 17, 48, 2], but as will become clear in the next sections, some arbitrary choices always have to be made: one has to fix some bounds, find suitable speed/security trade-offs, and so on. Therefore, *rigidity* as sometimes advertized [10] seems illusory to us. Nevertheless, *transparency* is achievable and needed for wide adoption of ECC.

In this note, we don't make an explicit proposal for an elliptic curve, but we deal with the following issues.

Security. We give a list of criteria that should be satisfied by a secure elliptic curve. Although a few of these criteria are incompatible, we detail what we think are the best choices for optimal security.

Transparency. We sketch a way to generate a curve in a fully transparent way so that it can be trusted and not suspected to belong to a (not publicly known to be) vulnerable class. In particular, since the computational cost of verifying the output of such a process may be quite high, we sketch out the format of a *certificate* that eases the computations. We think that this format might deserve being standardized.

1 Criteria for cryptographic elliptic curves

We give here a list of useful criteria for selecting elliptic curves for general cryptographic use. We sort

*Agence nationale de la sécurité des systèmes d'information
Secrétariat général de la défense et de la sécurité nationale
51, boulevard de La Tour-Maubourg
75700 Paris 07 SP
France
†National Communications Security Authority
Försvarsmakten
Lidingövägen 24
SE-107 85 Stockholm
Sweden

these conditions in several categories, with different importance being granted to each category. The first category (1.1) contains the minimal conditions under which the discrete logarithm problem may be hard in the point group. The second category (1.2) describes some properties that may improve security for some implementations, particularly in a context where side-channel attacks must be considered. The third one (1.3) is not related to any known attack; instead, it gives condition under which a curve may be considered as particular and therefore potentially particularly vulnerable to some yet-unknown attack. Finally, the last two categories group some properties which may be desirable about the curve, either for facilitating its implementation (1.4) or for specific protocols and algorithms (1.5), without weakening (too much) its security.

Incompatible conditions

We point out that several of these conditions are mutually incompatible. We list these here, in a roughly descending order of importance.

Choice of the cardinality of the base field and the curve coefficients. These may either be chosen pseudo-randomly [47, 41], for example as a precaution against some side-channel attacks (1.2.4) or against some possible future attacks (1.3.4); or very specific values may be chosen instead, in view of faster curve arithmetic (1.4.5) [53, 7, 54].

Cofactor. While the existence of a very small torsion subgroup may lead to some attacks such as small-subgroup attack (1.2.1) or side-channel attacks (1.2.2), it is a necessary condition for the availability of some faster curve coordinates such as Edwards or Montgomery curves (1.4.4). Also note that allowing a square cofactor might induce a non-cyclic structure on the group of rational points (e.g., if c^2 divides the number of points and c divides $p-1$, then the full c-torsion, which is of rank two, might be rational).

Primality of the order of the quadratic twist. In general, we expect that the order of the quadratic twist will have at least one large prime divisor, for example larger than $p^{1/4}$ (1.3.3). If this order is itself a prime number, then this grants a supplementary layer of protection against some side-channel attacks (1.2.3). However, this property is relatively rare itself (by a factor proportional to $\log p$), which could raise concern that curves with a prime twist are themselves exceptional in some way.

Because of all these incompatibilities, each implementation might select an appropriate curve depending on the context, such as performance constraints or the likelihood of side-channel attacks. In particular, we think that international standards should include a family of curves where each of these contradictions is solved in a way that maximizes security: namely, curves with pseudo-random coefficients, defined over a pseudo-random base field, with a cofactor equal to one, and preferably with a secure quadratic twist.

Restriction to prime fields

We limit our discussion to elliptic curves defined over a prime field. In the case of extension fields, some attacks exist in particular cases [43, 51, 26, 37]. Some of these attacks may even be exploited to include a trapdoor in an elliptic curve [69]. We also note that, over finite fields with small characteristic, an index calculus technique related to that of [43] gives a quasi-polynomial solution to the multiplicative discrete logarithm problem [5]. For all these reasons, we consider elliptic curves over prime fields as probably much safer than elliptic curves over extension fields.

Notation

In all the remainder of this document, we shall use the following notation: $p \geq 5$ is a prime number, $k = \mathbb{F}_p$ is the finite field with p elements, $E : y^2 = x^3 + a\,x + b$ is an elliptic curve defined over k, $N = |E(\mathbb{F}_p)|$ is the order of the group of rational points of E, $t = p + 1 - N$ is the trace of the Frobenius automorphism of E, and q is the largest prime divisor of N and $c = N/q$ the cofactor.

1.1 Hardness of the discrete logarithm problem

We give here a list of conditions corresponding to known attacks on the discrete logarithm problems. We point out that some criteria commonly required for generating elliptic curves, such as the criteria on the discriminant and class number [47], are not known to lead to a direct attack. Therefore, we do not include them here, but in Section 1.3 instead.

1.1.1 Nonsingular curve

If the discriminant $4\,a^3 + 27\,b^2$ of the curve E is zero, then E is not an elliptic curve: it is a singular curve and its group of points is isomorphic to an additive or multiplicative group. Such (non-elliptic) curves must be excluded.

1.1.2 Large prime subgroup

Since discrete logarithms are computable in the group $E(k)$ with complexity $O(\sqrt{q})$, where q is the largest prime divisor of N, it is necessary that $\sqrt{q}$ attains the required security bound. In practice, it is advisable to select elliptic curves whose order N is the product of a large prime q and a very small *cofactor* $c = N/q$. A cofactor $c = 1$ yields an optimal security for a given bitsize whereas a very small cofactor might allow performance improvements.

For a given curve E, checking if this is the case requires computing the group order N, which is a moderately expensive task. For curves over large prime fields, the most efficient algorithms are variations of the SEA algorithm [61, 62, 20] with complexity $O(\log^4 q)$.

The probability that a random elliptic curve over $\mathbb{F}_p$ has a prime group order is bounded below [34] by $\frac{0.44}{\log p}$. This condition is the most restrictive during the generation of a curve in practice.

1.1.3 Absence of additive transfer

If $N = p$ then there exists an *additive transfer* reducing the discrete logarithm in $E(k)$ to that in the additive group of $\mathbb{F}_p$. Therefore, elliptic curves with trace 1 must be excluded.

1.1.4 Absence of multiplicative transfer

The *embedding degree* is the smallest integer e such that q divides $p^e - 1$ (that is the multiplicative order of p modulo q). The pairings attached to the elliptic curve E give a group homorphism from $E(k)$ to the multiplicative group $\mathbb{F}_{p^e}^\times$.

Therefore, elliptic curves with an embedding degree small enough that discrete logarithms are computable in $\mathbb{F}_{p^e}^\times$ must be excluded.

Over the base field $\mathbb{F}_p$, supersingular curves have an embedding degree one and must be excluded. They are exactly the curves with trace zero and can therefore easily be detected.

1.1.5 Index calculus

Index calculus techniques developed to compute discrete logarithms in multiplicative subgroups of finite fields have been extended to elliptic curves in a variety of ways. However, in the current state-of-the-art, when the curve is defined over a prime field, it is more expensive to correctly lift the curve and the points defining the DLP than to directly solve it [66, 63, 44, 64, 65]. Therefore, we do not believe that there is any additional check to perform in the case of a prime base field.

1.2 Implementation-dependent security

While the existence of some attacks, such as side-channel attacks or attacks against badly designed protocols, mainly depends upon the implementation, in some circumstances the choice of the curve itself might have an impact on the efficiency of these attacks or on the ease of implementation of appropriate counter-measures. We give here a list of criteria which might improve the security of some implementations.

1.2.1 Absence of small subgroups

If the point group contains a small subgroup, then it may be possible to trick some implementations into revealing information about the secret key [46] or compromising the output of a key exchange. If the curve does not have a small subgroup (for example if it has a prime number of points) then such attacks are inoperant. Otherwise, protecting against them requires a few more point operations.

1.2.2 Absence of special points

The *special points* of an elliptic curve are the points (x, y) such that one of the two coordinates is zero. In the presence of such special points, there exist side-channel attacks [38] exposing private information.

Several protections against these attacks exist [50]. One of them is simply ensuring that the curve does not contain any special point.

Special points of the form $(x, 0)$ exist if the curve has an even order. Special points of the form $(0, y)$ exist if the coefficient b is a square in $\mathbb{F}_p$.

1.2.3 Twist security

The *quadratic twist* of the elliptic curve E is the curve E' with equation $dy^2 = x^3 + a\,x + b$, where d is a non-square element of k. For a given abscissa x_0, exactly one of the curves E, E' contains a point (x_0, y).

An attacker may manipulate a badly written implementation into using the quadratic twist E' in place of the original curve E, either through side-channel attacks [31], or through attacks on a badly designed protocol.

Such attacks may be easily mitigated by checking that the manipulated points are on the original curve E and not on its twist E'. A supplementary layer of protection against these attacks can be obtained if the twist E' satisfy security conditions similar to those of the curve E itself. Nevertheless, it should be noted that the original curve and its twist will never share the same exact behavior against side-channel attacks, e.g. for exactly one of the two curves the coefficient b is a square, which is suboptimal against side channel attacks as the curve contains a special point of the form $(0, y)$. Moreover twist security won't protect against potential side channel attacks where one would detect if the computation took place on the curve or its twist and gain information on some bits of the secret.

Using a variant of the technique of Galbraith and McKee [34], we find that the probability that a random elliptic curve on $\mathbb{F}_p$ is both secure and twist-secure seems bounded below by $\frac{0.5}{\log^2 p}$ and above by $\frac{5}{\log^2 p}$. This estimate means that including twist-security in the conditions is particularly expensive, since all necessary checks will be performed on a quadratic (in $\log p$) number of elliptic curves. In particular, in view of the certificates mentioned in Section 2, both the size and the cost of validation of the certificate increase by a linear factor. Although this property of twist-security is quite rare, the curves satisfying it are not special in the sense of Section 1.3.

1.2.4 Non-special base field

Some common parameter choices, such as the NIST [53], Curve25519 [7], and SM2 [54] elliptic curves, use as their base field a prime field $\mathbb{F}_p$ where p is a prime number of a "special form", such as pseudo-Mersenne or generalized Mersenne numbers or values of cyclotomic polynomials [39]. While the use of such prime numbers speeds up the modular arithmetic, they are also more vulnerable to some side-channel attacks [25, 59, 6, 70, 60, 30]. The use of a non-special, pseudo-random base field prevents this class of attacks.

1.2.5 Unified group law

Some curve families admit a *unified* or *complete* addition law: these formulas have no exceptional cases such as $P+P$, $P+(-P)$ or $P+0$ for Weierstraß curves. When using such formulas, a point multiplication is computed in constant time relatively to the scalar, which adds a layer of protection against some side-channel attacks. However, this does not offer an absolute protection [30]. Moreover, these families all have a non-trivial cofactor, which could be considered as a threat in light of 1.2.1 or 1.2.2 above.

1.3 Normality of the curve

The criteria we present here do not correspond to known attacks on elliptic curves. Rather, they are properties that random curves should satisfy with overwhelming probability. When generating curves, checking for these conditions should reject a negligible proportion of curves. If on the other hand a curve does not satisfy one of these conditions then, even though we do not know any precise attack, this curve is slightly more likely to be vulnerable.

More concretely this means that during the process of generating a secure curve, in general, curves will be discarded because they don't satisfy one of the criteria of Section 1.1. When a curve passes the checks for these criteria, the computations needed for the criteria of the current section will be performed and will also pass with overwhelming probability. Therefore, these computations will only be performed once in general, on the final curve.

Expected smoothness of random numbers. Many common number-theoretic computations depend on the factorization of a parameter (for example, the order of some group), and become easier when this parameter is smooth (i.e. when it has only small prime divisors). We recall here [19] that the probability that a number n be B-smooth is approximately u^{-u}, where $u \simeq \log n / \log B$.

We must determine a threshold at which the factorization of these numbers is considered "too smooth". The first choice for a probability threshold would be of the order of $1/\sqrt{p}$, corresponding to the complexity of the discrete logarithm in the group $E(k)$. For example, for a fixed exponent α, a number $x \approx p^\alpha$ has a probability $\approx 1/\sqrt{p}$ of being $(\frac{\log p}{2 \log \log p})^{2\alpha}$-smooth. However, numeric computations suggest that this choice may be too lenient at cryptographic sizes: for random numbers of 256 bits, the threshold probability of 2^{-128} corresponds to numbers which are 727-smooth. In a computational view, this bound is extremely low: any algorithm involving polynomials or matrices of this size would be easy to implement. We instead use the smoothness bound $B = p^{1/4}$, corresponding to a probability of 1/256. This means that we expect that only 1/256 of pseudo-random elliptic curves will be rejected as "too exceptional", while computations in these groups are likely to involve linear or polynomial algebra with size about $p^{1/4}$, which will likely remain out of reach for reasonable values of p.

1.3.1 Discriminant of the endomorphism ring

The *endomorphism field* of the curve E is the field K generated by its Frobenius endomorphism φ. Since φ is a root of the equation $\varphi^2 - t\varphi + p = 0$, K is an imaginary quadratic number field. The discriminant of φ is the value $D_\varphi = t^2 - 4p < 0$. It is the discriminant of the order $\mathcal{O}_\varphi = \mathbb{Z}[\varphi] \subset K$ and is greater than $O(\sqrt{p})$ with probability $1 - O(1/\sqrt{p})$.

The discriminant of K is the (fundamental) discriminant D_K of its maximal order $\mathcal{O}_K$. It is closely related to the square-free part of D_φ: $D_\varphi = D_K f_\varphi^2$ for $f_\varphi \in \mathbb{Z}$ called the conductor of the order $\mathcal{O}_\varphi$, and D_K or $D_K/4$ is a square-free integer. The *endomorphism ring* of the curve E is some order $\mathcal{O}_E$ in K: $\mathcal{O}_\varphi \subset \mathcal{O}_E \subset \mathcal{O}_K$. Let us denote its discriminant by D_E. The best method we know to compute $\mathcal{O}_E$ for an ordinary elliptic curve is of subexponential complexity [13].

The expected value for D_E is $D_K \approx D_E \approx D_\varphi$; we know [21] that the square-free part of a random integer n is less than $\sqrt{n}$ with probability approximately $1.66/\sqrt{n}$. For cryptographic sizes, this means that, with overwhelming probability, we should expect $D_E \geq D_K > \sqrt{D_\varphi}$.

The best method we know to compute D_K requires the factorization of D_φ and is therefore of subexponential complexity. For cryptographic values of D_φ, this is a possible but quite expensive task. However, since almost all curves satisfy the condition $D_K > \sqrt{D_\varphi}$, this condition is extremely unlikely to lead to the rejection of a curve and the computation will therefore in practice be performed only once for the final curve. Moreover, once the factorization of D_φ is known, it is very easy to check that the produced factorization is correct.

It should be noted that this criteria automatically eliminates the two smallest discriminants $D_K = -4$ and $D_K = -3$, corresponding to the special curves with j-invariant 1728 or 0.

1.3.2 Class number and class group

The class number $h(\mathcal{O}_E)$ of the order $\mathcal{O}_E$ is the minimal degree of a number field over which E admits a faithful lift. It is also the degree of the Hilbert class polynomial used in the theory of complex multiplication. Therefore, a large class number may prevent the use of any attacks based on complex multiplication. Under the generalized Riemann hypothesis, the best method we know to compute the class number is subexponential [40, 11], and requires at least a few days of computation for cryptographic sizes. Since this method also computes the group stucture of the ideal class group of $\mathcal{O}_E$, it can be used to produce a small and easily verifiable certificate for the class number. Note that the value $h(\mathcal{O}_E)$ is easily computed [24, 7.24] as a multiple of the more classical class number $h(K)$ of the maximal order $\mathcal{O}_K$ in K which is therefore enough to work with. Another justification is that it is possible to transfer the discrete logarithm problem onto a curve with maximal endomorphism ring through isogenies.

The class number has a negligible probability of being $(\log p)^{O(1)}$-smooth. While the best known algorithms for computing the class number are subexponential [40, 11], for any bound B, it is possible to prove that $h(K)$ is not B-smooth in time $O(B \log p)$, which is polynomial if B is polynomial.

On the other hand, the class number of K is minorated [49], under the generalized Riemann hypothesis, by $h(K) \geq \frac{\pi}{3e} \frac{\sqrt{|D_K|}}{\log|D_K|}$.

1.3.3 Cardinality of the quadratic twist

The order of the quadratic twist of E is $2(p+1) - N$. This number lies in the same interval $[p+1-2\sqrt{p}, p+1+2\sqrt{p}]$ as the curve order itself. Therefore it has a negligible probability of being $(\log p)^{O(1)}$-smooth and probability 1/256 of being $p^{1/4}$-smooth.

We recall from paragraph 1.2.3 above that the smoothness of this number has a direct influence on the effectiveness of some side-channel attacks.

It should be noted that the curve and its quadratic twist share the same endomorphism ring. Therefore the discriminant and class number criteria are automatically satisfied by the twist if they are satisfied by the original curve. This is not true as far as the embedding degree is concerned. The original curve and its twist have distinct embedding degrees.

1.3.4 Non-special base field

In the case of the *multiplicative* discrete logarithm problem, the Special Number Field Sieve allows faster computations of discrete logarithms modulo p when p is a *special* prime number, i.e. when it is a value of a polynomial of low degree with small coefficients evaluated at a small value.

It is hard to check whether a given (prime) number is special. However, most prime numbers used in standard elliptic curves (for example in the FIPS 186–2 curves [53], in Curve25519 [7], and in the SM2 curve [54]) are explictly given as special primes as these allow faster arithmetic over the base field. Even though we don't know any attack against curves with such special parameters, it is legitimate to consider them as exceptional.

A related question would be to detect curves whose number of points N is a special number. As noted above, this cannot be easily checked. Nonetheless, the only way we are aware of which could lead to the construction of such a curve would be through the use of complex multiplication. Therefore, the discriminant of the number field associated to the generated curve would be unusually small and this would be detected by other checks from this section.

1.3.5 Embedding degree

For any bound m, the probability that the embedding degree (1.1.4) of E is at most m is [4] $O(m^2 \cdot p^{-1} \cdot (\log p)^5 \cdot (\log\log p)^2)$. This implies that, with probability $1 - 1/\sqrt{p}$, the embedding degree of E is at least $p^{1/4-o(1)}$.

1.3.6 Multiplicative group of the base field

The multiplicative structure of the base field $\mathbb{F}_p^\times$ is directly related to the factorization of $p-1$. In particular, if $p-1$ is smooth (i.e. all its prime divisors are small), then the multiplicative discrete logarithm problem is easy; $p-1$ has a negligible probability of being $(\log p)^{O(1)}$-smooth and probability $1/256$ of being $p^{1/4}$-smooth.

1.4 Convenience of implementation

We list here a few criteria that may make the implementation of an elliptic curve more convenient without weakening the security. We note that some of these conditions (namely 1.4.1, 1.4.2, 1.4.3 and 1.4.4) are satisfied by an asymptotically non-zero proportion of all elliptic curves: in this case, we believe it very unlikely that such a wide class of curve would be inherently weak against a future attack. However, the last two conditions (1.4.5 and 1.4.6) correspond to choices that contradict some conditions from sections 1.2 and 1.3 above.

1.4.1 Fast Jacobian coordinates

Choosing a curve of the form $y^2 = x^3 - 3x + b$ (that is with $a = -3$) enables to save 2 out of 10 multiplications [55] required to double a point in Jacobian coordinates. A random elliptic curve over $\mathbb{F}_p$ is isomorphic to a curve with $a = -3$ with probability $1/2$ if $p \equiv 3 \pmod 4$ and $1/4$ if $p \equiv 1 \pmod 4$.

1.4.2 Number of points at most p

If the number of points N is greater than p, then it might be impossible to represent numbers up to $N-1$ in the same memory size as coordinates of points of E. Exactly one half of all curves satisfy this.

1.4.3 Easy computation of square roots

The *point compression* method allows representing one point (x, y) of E by only its abscissa x and one bit discriminating between the two possible values $\pm y$. However, recovering y requires computing a square root in k. This is easier when $p \equiv 3 \pmod 4$ since in this case, $c^{(p+1)/2}$ is a square root of c if c is a square. (Similar formulas exist as soon as $p \not\equiv 1 \pmod 8$.)

1.4.4 Equations other than Weierstraß

There exist several other representations of elliptic curves than Weierstraß coordinates, such as Edwards curves [27], twisted Edwards curves [8], Jacobi curves [12], and Montgomery curves [52]. Only a finite proportion of curves are isomorphic to a curve in these families. For example, about 35% of all elliptic curves are isomorphic to an Edwards curve, and about 40% are isomorphic to a Montgomery curve [57] or, equivalently, to a twisted Edwards curve.

Each of these families requires at least the presence of a point of order two on the curve. This is a special point in the sense of 1.2.2 above and could introduce a weakness in some implementations. The corresponding automorphisms also speed up the Pollard rho method for discrete logarithms by a small factor [35]. Moreover, in the case of a non-prime base field, there exist some attacks [29] against several of these curve families which slightly weaken the discrete logarithm problem.

1.4.5 Fast base field arithmetic

The choice of a base field of a special form, such as the field of integers modulo a pseudo-Mersenne [7] or generalized Mersenne prime [53], allows the implementation of a faster, dedicated arithmetic.

However, taking full advantage of these optimizations needs restricting the implementation to a particular, very small family of elliptic curves. Therefore, we think that, for optimal security and in view of paragraphs 1.2.4 and 1.3.4 above, the most secure implementations should be able to work with a general base field.

1.4.6 Special coefficients

Most of the formulas for elliptic curve arithmetic involve the use of the curve coefficients. Choosing special values for these coefficients, such as integers with a small absolute value, allows a faster implementation.

However, as in the previous paragraph, we think that restricting implementation to benefit from these optimizations might be a security threat.

1.5 Families of curves with particular properties

We give here a short list of families of curves with particular properties which might be useful in some specific contexts. These families are small enough to contradict the “normality” conditions of Section 1.3.

1.5.1 Curves with a fast endomorphism

Some families of curves have an easily computable endomorphism. This allows a faster implementation of point multiplication [36, 33], with a theoretical gain of up to 50%.

However, the construction of all these families relies on the fact that the discriminant of the endomorphism field is small [67, 68], and is therefore in contradiction with paragraph 1.3.1.

Moreover, in the presence of an endomorphism of order m, the Pollard rho method for computing discrete logarithms becomes faster by a factor of up to $\sqrt{m}$ [26, 35].

1.5.2 Pairing-friendly curves

Some families of elliptic curves allow a fast pairing computation. This construction has various applications in cryptography, such as one-round three-way key exchange [42], short signatures [16], and identity-based cryptography [14, 56, 15].

The main requirement for the existence of a fast pairing is that the embedding degree is small. This is in direct contradiction with the requirement from paragraph 1.1.4 above.

Moreover, most of the constructions for pairing-friendly curves [32, 28] either use very sparse families of curves, or use complex multiplication to construct adequate curves, which requires a small class group, in contradiction with paragraph 1.3.2.

2 Transparent generation of curves

The selection of an elliptic curve for cryptographic purposes involves checking a long list of properties, including several for which arbitrary bounds have to be set: such as the threshold at which we consider the class number to be exceptionally small, or the order of the twisted curve to be exceptionnally smooth. This implies that the generated curve will always depend on arbitrary choices, including the choice of the sampling function for elliptic curves. That is why we think *rigidity* as often advertized [10] is illusory, and we prefer the notion of *transparency*.

We point out that the standard parameters for ECC include not only the curve, but also its definition field and a point generating a prime-order group in the curve. While we know of no weakness related to the choice of this point, precaution still commands that this choice should also be justified.

Checking that the generation process, with a known algorithm and a public seed, indeed produced the claimed curve is a computationnally expensive task. We give here the outlines of a certificate format for this. This allows any program receiving elliptic curve parameters for cryptographic purposes to check, at a moderate computational cost, that the curve is indeed suitable, and moreover, that it is the first suitable curve found by the sampling function.

2.1 Generating an elliptic curve

This procedure is in two steps. First, a *generation program* checks elliptic curves, as provided by a *sampling function*, until a suitable elliptic curve is found. This programs outputs the elliptic curve parameters together with a *certificate* proving that the curve is actually suitable for cryptographic purposes. Moreover, the certificate should also prove that none of the curves previously tried by the generating program was suitable. Then, a *validation program* can use the certificate data to validate the generating process. The certificate enables this second program to have a significantly shorter runtime than the first one.

The list of conditions to be checked and the way to sample curve parameters would be indicated in the certificate header. We do not fully specify how every condition presented in the previous sections should be written down in the certificate when it leads to the rejection of a curve. For most of the ones presented in Section 1, checking whether a curve is suitable is very fast. However, three conditions in particular are more expensive: namely, the condition that the curve order is prime or only includes the expected small cofactor (this condition is expected to be the one condition leading to rejection of most of the unsuitable curves, so that this check will be performed a large number of times), the computation of the endomorphism ring and class number, and the computation of the embedding degree (both of which will typically be performed only once, for the final curve).

2.2 Certifying the curve order

The most restrictive condition in practice is that the curve order must be prime, or a prime number times a very small cofactor. This means that we expect that a linear number (in $\log p$) of curves will be rejected because of a composite group order, whereas the first curve found with a prime (or only including the expected small cofactor) group order will be retained. Note that in a cryptographic context, when a small cofactor c is allowed, it is usually because a special form of curve is used and therefore N will always be divisible by c. We do not take into account the case where c does not automatically divide the order of the tested curves (or is just a bound on the allowed cofactor) though the following treatment can be easily be extended to deal with this situation.

2.2.1 Rejected curve order

In general, proving that the curve order is not prime, or has an unexpected cofactor, is easy: namely, if $n < 2(\sqrt{p}-1)^2$ is a composite number coprime to c and $P = 0$ is a point such that $n \cdot P = 0$, then the curve order N has a composite factor coprime to c. Namely, let $d = \gcd(n, N)$; d is coprime to c. Since $P = 0$, we have $d = 1$. If $d = N$ then N divides n and is coprime to c. Since $n/2$ does not meet the lower Hasse bound $(\sqrt{p}-1)^2$, we see that $N = n$, and therefore N is composite and coprime to c (which had to be 1). If on the other hand $d = N$, then d is a strict divisor of N coprime to c.

The certificate is then the list $(N/c, a, c \cdot G)$, where N is the computed curve order, c is the cofactor, a is a witness of composition of N/c, and G is a random point (such that $c \cdot G = 0$).

Obstructions to producing such a certificate can occur when a cofactor $c = 1$ is expected: the actual cofactor d might be a product of small primes dividing c but still be different from c. Typically, d will be a multiple of c and the group of points will be cyclic. It is then easy to produce a point P of small order e a multiple of c and dividing d, but also to check that P has order exactly e; most other cases can be resolved in a similar way. The most problematic one is when $N = d^2eq$, q is prime, $de = c$, $d > 1$, $e \geq 1$, and the full d-torsion is rational, because no rational point of small order greater than c exists. Nevertheless, a certificate similar to the general case can still be be issued: (cq, a, G) where G has order cq. The Hasse bound indeed ensures that $N/(cq) = 1$ and the curve is not suitable.

In practice, it is possible to do better than above. Indeed, over a prime base field, the most efficient methods to compute the curve order are variants of the ℓ-adic SEA algorithm [61, 62, 20]. This algorithm computes the order N of the curve by computing $N \pmod{\ell}$ (or $N \pmod{\ell^k}$) for several auxiliary primes $\ell < O(\log p)$ (and small exponents k). The fact that N is composite may therefore be detected in an early step, when there exists ℓ (not dividing the allowed cofactor c) such that $N \equiv 0 \pmod{\ell}$.

It is also possible to benefit from this early detection while still producing a proof of composition for the order of E. Namely, if $N \equiv 0 \pmod{\ell}$ then the counting algorithm finds a polynomial f, of degree $(\ell-1)/2$, whose roots are the abscissae of points of order ℓ of E. We may then recover one of these points by computing a root of f: using the Cantor-Zassenhaus polynomial factorization algorithm, we find that the complexity of this operation is approximately the same as that of computing $N \pmod{\ell}$, so that computing this extra information has a small impact on computation time.

The certificate in this case is the list (ℓ, P), where P is a point of order ℓ.[1]

We also note that, if the list of conditions include the primality of the twisted curve, then both methods above should be used simultaneously for the curve and its quadratic twist.

2.2.2 Adequate curve order

When the algorithm finds that the value N is prime or only includes the allowed small cofactor c, a certificate is as follows: $(N/c, G, \Pi)$, where N is the number of points, c is the expected cofactor, G is a point of order N/c (for example, any random non-zero point if $c = 1$ and N is prime), and Π is a proof of primality of N/c (which may be left empty if N/c is small enough that proving its primality directly is easier than using a certificate). The validation program then checks that $G = 0$, $(N/c) \cdot G = 0$, adequate points of small orders exist for the cofactor c (as c is

[1] It is also possible, although slightly less efficient, to conform to the preceding certificate format by using the list $(2\ell, 2, P)$, since $a = 2$ is a witness of composition for 2ℓ.

expected to be very small, this is a cheap operation), and $N \geq (\sqrt{p}-1)^2$. If this is the case, then N is the curve order.

For the sizes involved in elliptic curve cryptography, proving the primality is practical using a test such as the APR-CL test [1, 23]. In the case where the validation program runs under strong contraints, it is also possible to write a primality certificate in a form such as ECPP [3], which has a reasonable size ($O(\log^2 p)$ bits) and is verifiable in a short time ($O(\log^2 p)$ field operations).

We expect that all curve orders including more than the expected cofactor c will be rejected by a pseudoprimality test such as the Miller-Rabin test. Therefore, the generating program will have to run the primality proof only once, for the last curve.

2.2.3 Refreshing the base field

We saw in paragraph 1.2.3 that the rarity of secure and twist-secure elliptic curves depends on the cardinality of the base field, with a variation of a factor ≈ 10 for a given size of prime numbers. This means that, in the case where a pseudo-random base field is preferred, we suggest that the sampling function for elliptic curves change the base field for each new curve, to avoid being stuck at a "bad" prime. This averages out the probabilities and leads to a speed-up of ≈ 4 compared to the worst-case expectation. This has no apparent security implication since the final prime number is pseudo-random anyway; the only performance penalty is the cost of generating new primes, which is negligible comparde with the cost of computing curve orders.

2.3 Discriminant and class group

The fundamental discriminant is given by the factorization of the discriminant $D_\phi = t^2 - 4p$. While computing this factorization is quite expensive, (and asymptotically dominates the whole generation process), it will generally be performed only once, on the final curve, and validating the factorization is extremely easy. The same is true for the subsequent computation of the exact endomorphism ring $\mathcal{O}_E$ and its discriminant Δ_E in case the curve does not have prime order (recall that if the curve has prime order then $\mathcal{O}_E = \mathcal{O}_K$ and no additional computation has to be performed).

The sole knowledge of the fundamental discriminant also gives a lower bound on the class number of the endomorphism field of E and so on that of its endomorphism ring $\mathcal{O}_E$. However, this bound is only approximately $p^{1/4}/\log p$.

In some cases, a better (higher) lower bound B might be required. We can prove that the class number is greater than B in the following simple way: since the class group is expected to be almost cyclic [22], it is enough to produce an element g of order $\geq B$. However, checking this requires computing the B multiples $g, 2g, \ldots, Bg$, and is therefore exponential.

We do not expect to ever have to prove that a curve was rejected because its class number is smaller than B. We know no efficient way to prove such a fact in general. However, it might be enough to prove that a few deterministically generated elements of the class group all have order $\leq B$: while this does not prove that $h \leq B$, it proves that the generating program was unable to prove that $h > B$ and that the curve should therefore be rejected.

The same considerations apply to the smoothness of the class number. If the class number is not smooth, then verifying it is a quadratic computation. If the class number is smooth, then the generating program cannot prove it, but it can prove that it was unable to prove that the class number is not smooth.

2.4 Embedding degree

The embedding degree of E is the smallest integer e such that q divides $p^e - 1$ (that is the multiplicative order of p modulo q). Computing exactly this embedding degree requires factoring $q-1$ and is an expensive computation. However, this factorization, which is the only expensive task in the computation of the embedding degree, is easy to include in the certificate. (If one only wants to check that the embedding degree is larger than a moderate bound B, brute-force might be sufficient.)

We note that the embedding degree of the quadratic twist is distinct from that of the curve; therefore, if twist security is required, then this computation will need to be performed twice.

2.5 Choice of the sampling function

The previous algorithms provide, to the best of our knowledge, a certifiable way to transparently generate a cryptographic elliptic curve from the input of some conditions and a sampling function. In the case where the generating function is considered as pseudo-random and the normality conditions of Section 1.3 are included, we feel confident that the resulting elliptic curve will not have any particular weakness. However, a malicious generating program, given enough computing resources, might be able to run the generating algorithm for a large family of seeds of a pseudo-random function until a suitable elliptic curve is found (see [9]).

Even though we think that a curve satisfying the normality conditions of Section 1.3 will generally be as good as possible for cryptographic use, and using a highly constrained seed (such as zero) would be sufficient in practice, we could imagine as a supplementary precaution against this manipulation to first fully specify the generating protocol in all its details, and to put the seed choice out of reach of the generating entity. Several examples come to mind. For example, several entities could contribute to the seed, each one of them generating its own secret share and publicly committing it before all shares are revealed. Another possibility is committing in advance to using the result of some future, publicly verifiable observation expected to be out of reach of manipulation, such as the observation of sunspots, a public physical random number such as the result of a lottery drawing, or a number derived from stock market or sports results.

2.6 A concrete example

2.6.1 Certificate format

We suggest that the certificate should be separated in three parts.

First, a header declares which choices were made: the sampling function, the seed (if applicable), as well as the subset of conditions retained from part 1 with the numerical values of these criteria. The validation program is then able, upon reading the header and prior to any computation, to determine if it accepts the included criteria.

The second part of the certificate is the final, "good" curve, together with a proof for all the criteria.

The last part is the list of sampled curves, each one accompanied by a proof for its rejection. The certificate should also include enough information about the internal state of the sampling function to be able to retrace its execution.

2.6.2 A toy example

We chose the following sampling function as an example of "pseudo-random" curves. Starting from a seed s, we define p as the smallest prime greater than s and g as the smallest generator of the multipliative group $(\mathbb{Z}/p\mathbb{Z})^\times$. We then iterate over the curves with $a = -3$ (1.4.1) of the form $y^2 = x^3 - 3x + b$, where $b = g^n$ for $n = 1, \ldots$, until a suitable curve is found.[2]

We also include the following conditions:

- the discriminant $2^4 \cdot 3^3 \cdot (4 - b^2)$ is non-zero (1.1.1);
- the orders of the curve (1.1.2) and of its quadratic twist (1.2.3) are prime;
- the trace is non-zero (1.1.3);
- the embedding degree of the curve and of its quadratic twist (1.1.4) are at least $p^{1/4} \approx 7$;
- the class number (1.3.2) is at least $p^{1/4}$.

As a seed, we use the current year 2015.[3] The next prime number is $p = 2017$, and the smallest generator is $g = 5$.

The resulting certificate, in pseudo-code, is as follows. (Given the size of the parameters, all proofs of primality have been left empty).

Header

sampling.function = `pseudo-random/powers`
sampling.seed = 2015
condition.cofactor = 1
condition.twist_prime = `True`
condition.embedding_degree = 7
condition.twist_embedding_degree = 7
condition.class_number = 7

Curve

$(2017, -3, 625)$
order = 2063, point = $(0, 25)$
twist_order = 1973
disc_factors = $\{6043\}$
class_number = 9, form = $(17, 3, 89)$
embedding_degree = 1031, factors = $\{2, 1031\}$
twist_embedding_degree = 493, factors = $\{2, 17, 29\}$

Rejected curves

$((2017, -3, 5)$, composite, 2065, witness, 1679, point, $(1, 258))$
$((2017, -3, 25)$, torsion_point, 3, point, $(448, 288))$
$((2017, -3, 125)$, torsion_point, 2, point, $(982, 0))$

Conclusion and suggestions

Some of the publicly announced elliptic curves, are provably sampled, in a way conforming to the presentation sketched out in Section 2. This is the case of the Brainpool family, where the sampling function generates curves defined over pseudo-random prime fields, with pseudo-random coefficients. The only slight reservation about this family would be that

[2] We use powers of g as a simple way to produce pseudo-random looking elements of $(\mathbb{Z}/p\mathbb{Z})^\times$ only for the sake of this toy example. An actual, working example should of course involve a more robust pseudo-random number generator.

[3] Note that this seed choice *was* manipulated, in order to produce a certificate short enough to fit in a few lines.

the generating process does not include the condition that the order of the quadratic twist be prime (1.2.3) which makes the curves more generic, but implementations more error-prone. This is also the case of some curves with small coefficients [7, 17], where the sampling function generates the coefficients in increasing order. However, since these latter curves are, by construction, defined over special prime fields and since they also have small coefficients, they may be threatened in the sense of Section 1.2 and exceptional in the sense of Section 1.3. Moreover, most of these curves also have a cofactor strictly greater than one.

We point out that, to our knowledge, there does not exist yet any public proposal of an elliptic curve, or of a family of elliptic curves, conforming both to the provable generation of Section 2 and to the maximal security criteria of Section 1 and in particular of 1.1, 1.2 and 1.3, i.e. with pseudo-random coefficients modulo a pseudo-random prime, with a secure twist, and with a cofactor equal to one. We therefore think it is advisable to standardize (a family of) such curves.

References

[1] Leonard M. Adleman. On distinguishing prime numbers from composite numbers. In *21st Annual Symposium on Foundations of Computer Science, Syracuse, New York, USA, 13-15 October 1980*, pages 387–406. IEEE Computer Society, 1980. http://dx.doi.org/10.1109/SFCS.1980.28.

[2] Diego F. Aranha, Paulo S. L. M. Barreto, C. C. F. Pereira Geovandro, and Jefferson E. Ricardini. A note on high-security general-purpose elliptic curves. *IACR Cryptology ePrint Archive*, 2013:647, 2013. http://eprint.iacr.org/2013/647.

[3] A. O. L. Atkin and F. Morain. Elliptic curves and primality proving. *Math. Comp.*, 61(203):29–68, 1993. http://dx.doi.org/10.2307/2152935.

[4] R. Balasubramanian and Neal Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes–Okamoto–Vanstone algorithm. *J. Cryptology*, 11(2):141–145, 1998. http://dx.doi.org/10.1007/s001459900040.

[5] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2014. http://dx.doi.org/10.1007/978-3-642-55220-5_1.

[6] Naomi Benger, Joop van de Pol, Nigel P. Smart, and Yuval Yarom. "Ooh aah... just a little bit" : A small amount of side channel can go a long way. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2014. http://dx.doi.org/10.1007/978-3-662-44709-3_5.

[7] Daniel J. Bernstein. Curve25519: New Diffie–Hellman speed records. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography - PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography, New York, NY, USA, April 24-26, 2006, Proceedings*, volume 3958 of *Lecture Notes in Computer Science*, pages 207–228. Springer, 2006. http://dx.doi.org/10.1007/11745853_14.

[8] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted Edwards curves. In Serge Vaudenay, editor, *Progress in Cryptology - AFRICACRYPT 2008, First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008. Proceedings*, volume 5023 of *Lecture Notes in Computer Science*, pages 389–405. Springer, 2008. http://dx.doi.org/10.1007/978-3-540-68164-9_26.

[9] Daniel J. Bernstein, Tung Chou, Chitchanok Chuengsatiansup, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, and Christine van Vredendaal. SafeCurves: BADA55 elliptic curves, 2014. http://safecurves.cr.yp.to/bada55.html. Accessed 10 March 2015.

[10] Daniel J. Bernstein and Tanja Lange. SafeCurves: choosing safe curves for elliptic-curve cryptography, 2014. http://safecurves.cr.yp.to. Accessed 10 March 2015.

[11] Jean-François Biasse. Improvements in the computation of ideal class groups of imaginary quadratic number fields. *Adv. in Math. of Comm.*, 4(2):141–154, 2010. http://dx.doi.org/10.3934/amc.2010.4.141.

[12] Olivier Billet and Marc Joye. The Jacobi model of an elliptic curve and side-channel analysis. In Marc P. C. Fossorier, Tom Høholdt, and Alain Poli, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 15th International Symposium, AAECC-15, Toulouse, France, May 12-16, 2003, Proceedings*, volume 2643 of *Lecture Notes in Computer Science*, pages 34–42. Springer, 2003. http://dx.doi.org/10.1007/3-540-44828-4_5.

[13] Gaetan Bisson. Computing endomorphism rings of elliptic curves under the GRH. *J. Mathematical Cryptology*, 5(2):101–114, 2012. http://dx.doi.org/10.1515/jmc.2011.008.

[14] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Kilian [45], pages 213–229. http://dx.doi.org/10.1007/3-540-44647-8_13.

[15] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003. http://dx.doi.org/10.1137/S0097539701398521.

[16] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *J. Cryptology*, 17(4):297–319, 2004. http://dx.doi.org/10.1007/s00145-004-0314-9.

[17] Joppe W. Bos, Craig Costello, Patrick Longa, and Michael Naehrig. Selecting Elliptic Curves for Cryptography: An Efficiency and Security Analysis. Technical report, Microsoft Research, 2014. http://research.microsoft.com/en-us/projects/nums/.

[18] ECC Brainpool. ECC Brainpool Standard Curves and Curve Generation, 2005. http://www.ecc-brainpool.org/download/Domain-parameters.pdf.

[19] E. R. Canfield, Paul Erdős, and Carl Pomerance. On a problem of Oppenheim concerning "factorisatio numerorum". *J. Number Theory*, 17(1):1–28, 1983. http://dx.doi.org/10.1016/0022-314X(83)90002-1.

[20] Antoine Chambert-Loir. Compter (rapidement) le nombre de solutions d'équations dans les corps finis. *Astérisque*, Vol. 2006/2007(317):Exp. No. 968, vii, 39–90, 2008. Séminaire Bourbaki. http://arxiv.org/abs/math/0611584.

[21] Maurice-Étienne Cloutier, Jean-Marie De Koninck, and Nicolas Doyon. On the powerful and squarefree parts of an integer. *J. Integer Seq.*, 17(8):Article 14.8.6, 28, 2014. http://cs.uwaterloo.ca/journals/JIS/VOL17/Cloutier/cloutier2.html.

[22] H. Cohen and H. W. Lenstra, Jr. Heuristics on class groups of number fields. In *Number theory, Noordwijkerhout 1983 (Noordwijkerhout, 1983)*, volume 1068 of *Lecture Notes in Math.*, pages 33–62. Springer, Berlin, 1984. http://dx.doi.org/10.1007/BFb0099440.

[23] H. Cohen and H. W. Lenstra, Jr. Primality testing and Jacobi sums. *Math. Comp.*, 42(165):297–330, 1984. http://dx.doi.org/10.2307/2007581.

[24] David A. Cox. *Primes of the form $x^2 + ny^2$*. Pure and Applied Mathematics (Hoboken). John Wiley & Sons, Inc., Hoboken, NJ, second edition, 2013. http://dx.doi.org/10.1002/9781118400722.

[25] William Dupuy and Sébastien Kunz-Jacques. Resistance of randomized projective coordinates against power analysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2005. http://dx.doi.org/10.1007/11545262_1.

[26] Iwan M. Duursma, Pierrick Gaudry, and François Morain. Speeding up the discrete log computation on curves with automorphisms. In Kwok-Yan Lam, Eiji Okamoto, and Chaoping Xing, editors, *Advances in Cryptology - ASIACRYPT '99, International Conference on the Theory and Applications of Cryptology and Information Security, Singapore, November 14-18, 1999, Proceedings*, volume 1716 of *Lecture Notes in Computer Science*, pages 103–121. Springer, 1999. http://dx.doi.org/10.1007/978-3-540-48000-6_10.

[27] Harold M. Edwards. A normal form for elliptic curves. *Bull. Amer. Math. Soc. (N.S.)*, 44(3):393–422 (electronic), 2007. http://dx.doi.org/10.1090/S0273-0979-07-01153-6.

[28] Andreas Enge and Jérôme Milan. Implementing cryptographic pairings at standard security levels. In Rajat Subhra Chakraborty, Vashek Matyas, and Patrick Schaumont, editors, *Security, Privacy, and Applied Cryptography Engineering - 4th International Conference, SPACE 2014, Pune, India, October 18-22, 2014. Proceedings*, volume 8804 of *Lecture Notes in Computer Science*, pages 28–46. Springer, 2014. http://dx.doi.org/10.1007/978-3-319-12060-7_3.

[29] Jean-Charles Faugère, Pierrick Gaudry, Louise Huot, and Guénaël Renault. Using symmetries in the index calculus for elliptic curves discrete logarithm. *J. Cryptology*, 27(4):595–635, 2014. http://dx.doi.org/10.1007/s00145-013-9158-5.

[30] Benoit Feix, Mylène Roussellet, and Alexandre Venelli. Side-channel analysis on blinded regular scalar multiplications. In Willi Meier and Debdeep Mukhopadhyay, editors, *Progress in Cryptology - INDOCRYPT 2014 - 15th International Conference on Cryptology in India, New Delhi, India, December 14-17, 2014, Proceedings*, volume 8885 of *Lecture Notes in Computer Science*, pages 3–20. Springer, 2014. http://dx.doi.org/10.1007/978-3-319-13039-2_1.

[31] Pierre-Alain Fouque, Reynald Lercier, Denis Réal, and Frédéric Valette. Fault attack on elliptic curve Montgomery ladder implementation. In Luca Breveglieri, Shay Gueron, Israel Koren, David Naccache, and Jean-Pierre Seifert, editors, *Fifth International Workshop on Fault Diagnosis and Tolerance in Cryptography, 2008, FDTC 2008, Washington, DC, USA, 10 August 2008*, pages 92–98. IEEE Computer Society, 2008. http://dx.doi.org/10.1109/FDTC.2008.15.

[32] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *J. Cryptology*, 23(2):224–280, 2010. http://dx.doi.org/10.1007/s00145-009-9048-z.

[33] Steven D. Galbraith, Xibin Lin, and Michael Scott. Endomorphisms for faster elliptic curve cryptography on a large class of curves. *J. Cryptology*, 24(3):446–469, 2011. http://dx.doi.org/10.1007/s00145-010-9065-y.

[34] Steven D. Galbraith and James McKee. The probability that the number of points on an elliptic curve over a finite field is prime. *J. London Math. Soc. (2)*, 62(3):671–684, 2000. http://dx.doi.org/10.1112/S0024610700001502.

[35] Robert P. Gallant, Robert J. Lambert, and Scott A. Vanstone. Improving the parallelized Pollard lambda search on anomalous binary curves. *Math. Comput.*, 69(232):1699–1705, 2000. http://dx.doi.org/10.1090/S0025-5718-99-01119-9.

[36] Robert P. Gallant, Robert J. Lambert, and Scott A. Vanstone. Faster point multiplication on elliptic curves with efficient endomorphisms. In Kilian [45], pages 190–200. http://dx.doi.org/10.1007/3-540-44647-8_11.

[37] Pierrick Gaudry, Florian Hess, and Nigel P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *J. Cryptology*, 15(1):19–46, 2002. http://dx.doi.org/10.1007/s00145-001-0011-x.

[38] Louis Goubin. A refined power-analysis attack on elliptic curve cryptosystems. In Yvo Desmedt, editor, *Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings*, volume 2567 of *Lecture Notes in Computer Science*, pages 199–210. Springer, 2003. http://dx.doi.org/10.1007/3-540-36288-6_15.

[39] Robert Granger and Andrew Moss. Generalised Mersenne numbers revisited. *Math. Comput.*, 82(284), 2013. http://dx.doi.org/10.1090/S0025-5718-2013-02704-4.

[40] James L. Hafner and Kevin S. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. Amer. Math. Soc.*, 2(4):837–850, 1989. http://dx.doi.org/10.2307/1990896.

[41] Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français. Journal officiel de la République française, 2011. JORF n°0241 du 16 octobre 2011 page 17533, http://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000024668816.

[42] Antoine Joux. A one round protocol for tripartite Diffie–Hellman. *J. Cryptology*, 17(4):263–276, 2004. http://dx.doi.org/10.1007/s00145-004-0312-y.

[43] Antoine Joux and Vanessa Vitse. Elliptic curve discrete logarithm problem over small degree extension fields - application to the static Diffie–Hellman problem on $E(F_{q^5})$. *J. Cryptology*, 26(1):119–143, 2013. http://dx.doi.org/10.1007/s00145-011-9116-z.

[44] Michael J. Jacobson Jr., Neal Koblitz, Joseph H. Silverman, Andreas Stein, and Edlyn Teske. Analysis of the xedni calculus attack. *Des. Codes Cryptography*, 20(1):1–64, 2000. http://dx.doi.org/10.1023/A:1008312401197.

[45] Joe Kilian, editor. *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*. Springer, 2001.

[46] Chae Hoon Lim and Pil Joong Lee. A key recovery attack on discrete log-based schemes using a prime order subgroup. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 249–263. Springer, 1997. http://dx.doi.org/10.1007/BFb0052240.

[47] Manfred Lochter and Johannes Merkle. Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation. RFC 5639 (Informational), 2010. http://www.ietf.org/rfc/rfc5639.txt.

[48] Manfred Lochter, Johannes Merkle, Jörn-Marc Schmidt, and Torsten Schütze. Requirements for standard elliptic curves. *IACR Cryptology ePrint Archive*, 2014:832, 2014. http://eprint.iacr.org/2014/832.

[49] Stéphane Louboutin. Minorations (sous l'hypothèse de Riemann généralisée) des nombres de classes des corps quadratiques imaginaires. Application. *C. R. Acad. Sci. Paris Sér. I Math.*, 310(12):795–800, 1990. http://gallica.bnf.fr/ark:/12148/bpt6k63800716.

[50] Hideyo Mamiya, Atsuko Miyaji, and Hiroaki Morimoto. Efficient countermeasures against RPA, DPA, and SPA. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 343–356. Springer, 2004. http://dx.doi.org/10.1007/978-3-540-28632-5_25.

[51] Alfred Menezes, Edlyn Teske, and Annegret Weng. Weak fields for ECC. In Tatsuaki Okamoto, editor, *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, volume 2964 of *Lecture Notes in Computer Science*, pages 366–386. Springer, 2004. http://dx.doi.org/10.1007/978-3-540-24660-2_28.

[52] Peter L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Math. Comp.*, 48(177):243–264, 1987. http://dx.doi.org/10.2307/2007888.

[53] National Institute of Standards and Technology. FIPS 186–2. Digital Signature Standard. Technical report, NIST, 2000. http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf.

[54] Chinese Commercial Cryptography Administration Office. SM2: A group of ECC public key algorithms. Technical report, CCCAO, 2010. http://www.oscca.gov.cn/News/201012/News_1198.htm.

[55] P1363. Standard Specifications for Public-Key Cryptography. Technical report, IEEE, 2000. http://grouper.ieee/org/groups/1363/.

[56] Kenneth G. Paterson. ID-based signatures from pairings on elliptic curves. *Electronic Letters*, 38(18):1025–1026, 2002.

[57] Jérôme Plût. On various families of twisted Jacobi quartics. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 373–383. Springer, 2011. http://dx.doi.org/10.1007/978-3-642-28496-0_22.

[58] Certicom Research. Standards for Efficient Cryptography 2 (SEC 2): Recommended Elliptic Curve Domain Parameters. Technical report, Certicom Corp., 2010. http://www.secg.org/sec2-v2.pdf.

[59] Yasuyuki Sakai and Kouichi Sakurai. Simple power analysis on fast modular reduction with NIST recommended elliptic curves. In Sihan Qing, Wenbo Mao, Javier Lopez, and Guilin Wang, editors, *Information and Communications Security, 7th International Conference, ICICS 2005, Beijing, China, December 10-13, 2005, Proceedings*, volume 3783 of *Lecture Notes in*

Computer Science, pages 169–180. Springer, 2005. http://dx.doi.org/10.1007/11602897_15.

[60] Yasuyuki Sakai and Kouichi Sakurai. Simple power analysis on fast modular reduction with generalized Mersenne prime for elliptic curve cryptosystems. *IEICE Transactions*, 89-A(1):231–237, 2006. http://dx.doi.org/10.1093/ietfec/e89-a.1.231.

[61] René Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.*, 44(170):483–494, 1985. http://dx.doi.org/10.2307/2007968.

[62] René Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7(1):219–254, 1995. Les Dix-huitièmes Journées Arithmétiques (Bordeaux, 1993), http://jtnb.cedram.org/item?id=JTNB_1995__7_1_219_0.

[63] Joseph H. Silverman. The xedni calculus and the elliptic curve discrete logarithm problem. *Des. Codes Cryptography*, 20(1):5–40, 2000. http://dx.doi.org/10.1023/A:1008319518035.

[64] Joseph H. Silverman. The four faces of lifting for the elliptic curve discrete logarithm problem. In *11th Workshop on Elliptic Curve Cryptography*, 2007. http://www.math.brown.edu/~jhs/Presentations/ECC4FacesOfLifts.pdf.

[65] Joseph H. Silverman. Lifting and elliptic curve discrete logarithms. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, volume 5381 of *Lecture Notes in Computer Science*, pages 82–102. Springer, 2008. http://dx.doi.org/10.1007/978-3-642-04159-4_6.

[66] Joseph H. Silverman and Joe Suzuki. Elliptic curve discrete logarithms and the index calculus. In Kazuo Ohta and Dingyi Pei, editors, *Advances in Cryptology - ASIACRYPT '98, International Conference on the Theory and Applications of Cryptology and Information Security, Beijing, China, October 18-22, 1998, Proceedings*, volume 1514 of *Lecture Notes in Computer Science*, pages 110–125. Springer, 1998. http://dx.doi.org/10.1007/3-540-49649-1_10.

[67] Benjamin Smith. Families of fast elliptic curves from Q-curves. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 61–78. Springer, 2013. http://dx.doi.org/10.1007/978-3-642-42033-7_4.

[68] Benjamin Smith. The Q-curve construction for endomorphism-accelerated elliptic curves. *CoRR*, abs/1409.4526, 2014. http://arxiv.org/abs/1409.4526, to appear in J. Cryptology.

[69] Edlyn Teske. An elliptic curve trapdoor system. *J. Cryptology*, 19(1):115–133, 2006. http://dx.doi.org/10.1007/s00145-004-0328-3.

[70] Joop van de Pol, Nigel P. Smart, and Yuval Yarom. Just a little bit more. *IACR Cryptology ePrint Archive*, 2014:434, 2014. http://eprint.iacr.org/2014/434, to appear in the proceedings of CT-RSA 2015.