MASARYK
UNIVERSITY

Faculty of Informatics

Analysis of standard elliptic curves

Ph.D. Thesis Proposal

Mgr. Vojtěch Suchánek

Brno, Fall 2021

MASARYK
UNIVERSITY

Faculty of Informatics

Analysis of standard elliptic curves

Ph.D. Thesis Proposal

Mgr. Vojtěch Suchánek

Advisor: prof. RNDr. Václav Matyáš, M.Sc., Ph.D.

Department of Computer Science

Brno, Fall 2021

Signature of Thesis Advisor


FACULTAS ARTIS INFORMATICÆ
UNIVERSITAS MASARYKIANA

Abstract

Elliptic curve cryptography has repeatedly proven to be an essential part of the cryptographic ecosystem. Whether we are considering classical ECC or pairing-based cryptography, we are forced to select an appropriate elliptic curve. Although elliptic curves are used due to the difficulty of the discrete logarithm problem on them, the ones that are useful for cryptography are difficult to find. Hence, we have almost always no other option than to use a curve picked by a third party. Distrust and dissatisfaction with the system of curve selection have emerged based on documented instances of standard manipulation and backdoor insertion. Curves that have been selected for public use – standard curves, are the topic of this thesis proposal.

We propose methods for the security analysis of standard curves used in classical ECC as well as in pairing-based cryptography. The aim is to increase the level of trust in standard curves using a systematic large-scale analysis. As selecting elliptic curves is a problem of balance between security and efficiency, we also formulate certain directions for the improvement of the efficiency of elliptic curves.

Contents

1 Introduction

Since the introduction of elliptic curve cryptography (ECC) by Koblitz [Kob87] and Miller [Mil85], elliptic curves have proven to be an essential tool for public-key cryptography offering key exchange or digital signature algorithms. All of the ECC schemes utilize the group of points of an elliptic curve and, in particular, the discrete logarithm problem (DLP) in this group. One of the main reasons why the discrete logarithm problem on elliptic curves (ECDLP) preserved its difficulty throughout decades is its genericity. Indeed, for the vast majority of curves, the best known algorithms for breaking the ECDLP are generic exponential algorithms for DLP that do not utilize the elliptic curve structure. In contrast, subexponential attacks have been found on the DLP in the finite field [Gor93] (the core of the DSA protocol [FIP94]), which use the rich, specific, non-general group structure of the finite field. However, not all elliptic curves have a difficult ECDLP, and therefore, care must be taken when choosing a curve. There are two main areas of ECC: classical ECC and pairing-based cryptography[1].

Classical ECC includes schemes such as ECDH [BJS07], ECDSA [JMV01], or EdDSA [JL17]. Applications of these schemes can be found in the SSH protocol, the TLS protocol (ECDH for key establishment, ECDSA for authentication), or in the implementation of the Bitcoin cryptocurrency (ECDSA authentication). All instances of these protocols use a small fixed set of elliptic curves that have been preselected by third parties and standardized. The first standard that approved ECDSA was FIPS 186-2 [FIP00] with a short commentary, a reference to the X9.62 standard [ANSI98], and recommended elliptic curves in an appendix. This was historically the first standardization of elliptic curve parameters. Since then, several other elliptic curves standards have been published [LM10; PLK06; Cer; Wir00; IEEE00]. However, following the documented instance of a standard being manipulated – the Dual EC DRBG incident [Hal13; BLN15; Che+], the cryptographic community has expressed distrust and dissatisfaction with the process of standardization of elliptic curves [Sch13; Loc+14; Ber+15; CLN15].

1. We exclude here isogeny-based cryptography, as its security is not based on the ECDLP, but rather on the isogeny problem.

The second area of ECC is pairing-based cryptography. Pairing-based cryptography abandons the idea of an elliptic curve as a generic abelian group and focuses on a special class of curves that admit an efficiently computable bilinear map called pairing. Pairings were firstly introduced to cryptography as part of the MOV attack [MOV93]. This attack aimed at a specific class of curves with efficiently computable pairing by reducing the ECDLP to the DLP in a finite field. Joux [Jou09], Boneh and Franklin [BF01] later realized that pairings could be a feature, not a problem, and introduced the first practical implementation of identity-based cryptography. Since then, several other applications have been found, including the short BLS signatures [BLS01], broadcast encryption [BGW05] or the zero-knowledge proofs in cryptocurrencies such as Zcash [Hop+16].

Curves suitable for pairing-based cryptography (pairing-friendly curves) have such specific properties that sophisticated algorithms for their generation based on complex multiplication have been developed. The ISO/IEC 15946 [ISO17] standard was the first to standardize pairing-friendly curves for public use. The general direction in the field has been the improvement of the efficiency of pairing computations, forcing further conditions on pairing-friendly curves. This turned out to come at the cost of reducing the security [KB16]. As a result, pairing-based cryptography is undergoing a standardization process from the IRTF Crypto Forum Research Group [Sak+21].

This research proposal analyzes the security and the related efficiency of standard elliptic curves in both classical ECC and pairing-based cryptography. In particular, it aims to analyze the balance between elliptic curve efficiency and security, which appears to be related to their genericity: elliptic curves that are as general as possible avoid all known attacks, while curves with specific properties have the advantage of efficiency.

The aim of this thesis proposal is to present a research plan that will conclude with a dissertation thesis. In Chapter 2, we review the current state of the art relevant to our study of standard elliptic curves. In Chapter 3, we describe the aims of the thesis, including the time schedule and publication venues. Finally, Chapter 4 presents already achieved results with an attached submitted paper in the appendix.

2 State of the Art

This chapter summarizes the existing research on selected topics of ECC. The main sections of this chapter, 2.1 and 2.2, focus on the security of standard curves in classical ECC and pairing-based cryptography. Additionally, in sections 2.3 and 2.4, we introduce the scalar decomposition for fast scalar multiplication, and we give an overview of existing techniques for efficient prime field arithmetic in the context of standard elliptic curves.

2.1 Classical curves

For this section, denote E an elliptic curve over the prime field $\mathbb{F}_p$ and let N be the number of points over $\mathbb{F}_p$, i.e. $N = \#E(\mathbb{F}_p)$. Parts of this section are adopted from [SSD].

When using elliptic curves for ECDLP-based cryptosystems, both sides of the scheme must agree on a choice of a particular curve. The ECDLP on this curve must be difficult enough for our desired security level. The SafeCurves website [BL] presents a good overview of known curve vulnerabilities (and discusses implementation-specific vulnerabilities as well). In short, the difficulty of ECDLP is directly determined by the number of points N on the curve. Based on the existing attacks on ECDLP, the integer N should satisfy:

- The largest factor of N, denote it n, should be large enough, as it determines the complexity of the Pohlig-Hellman and Pollard-ρ attack (256-bit n for 128-bit security) [PH78; Pol74]; ideally, the same should hold for the quadratic twist of the curve (to avoid attacks on implementations);

- $N \neq p$ otherwise the Semaev-Satoh-Araki-Smart additive transfer attack applies [Sma99; Sem98; SA+98];

- The order of p in $\mathbb{F}_n^\times$ (the embedding degree of E) should be large enough (e.g., at least 20), otherwise the MOV attack based on multiplicative transfer using the Weil and Tate pairings applies [Sem96; FR94; MOV93].

- The absolute value of $(p+1-N)^2 - 4p$ (i.e., the absolute discriminant of the field of endomorphisms) should not be too low (SageCurves [BL] suggests at least 2^{100}), otherwise Pollard's ρ attack can be speeded up due to the presence of efficiently computable endomorphisms of the curve [GLV01a]. This does not pose a serious threat for the moment though, as the limits of the speedup are reasonably well understood.

The only approach for finding a curve that satisfies the mentioned conditions, including a sufficiently large CM discriminant, is the point-counting method, which repeatedly selects curve parameters and checks the mentioned conditions on N. The main bottleneck of this method is computing the number of curve points. We can use the polynomial SEA algorithm [Sch95], but in practice, it is not fast enough to allow on-the-fly ECC parameter generation. Although the SEA algorithm has a polynomial complexity, the density of secure curves satisfying the mentioned conditions is too low [BK98; GM00].

Several standardization organizations have therefore generated and proposed elliptic curve parameters for public use. To define a curve, one has to specify the prime p, the form of the curve (Weierstrass, Montgomery or twisted Edwards) and the parameters of the specified form. In the following, we provide a brief overview of the main approaches the standards have chosen for generating the curve parameters:

Unknown or ambiguous origin. Multiple standards recommend curves for public use with no or not sufficient explanation of their process of generation. These include the 256-bit curve `FRP256v1` recommended by the French National Cybersecurity Agency (ANSSI) in the Official Journal of the French Republic [Age11], curve by the SM2 standard [SSL14] published by the Chinese Office of State Commercial Cryptography Administration (OSCCA) and the seven standardized curves by the Russian GOST R 34.10 standard [PLK06; Dol10]. Further notable example is the Standard for Efficient Cryptography Group (SECG) [Cer] that recommends so-called Koblitz curves[1], which possess an efficiently computable endomorphism. However, the standard only

1. Analyzing these curves is of great importance due to the usage of `secp256k1` in Bitcoin [Nak08].

vaguely states *"The recommended parameters associated with a Koblitz curve were chosen by repeatedly selecting parameters admitting an efficiently computable endomorphism until a prime order curve was found."* As Bernstein et al. [Ber+15] demonstrate, proposing a curve without explanation of their generation enables the designer to input a secret vulnerability.

Verifiably pseudorandom curves. Aiming to re-establish the trust in ECC, several standards have picked the coefficients a, b in the Weierstrass form $y^2 = x^3 + ax + b$ in a *verifiably pseudorandom* way. The details differ, but the common idea is that the parameters are generated from a seed and the seed is made public, which limits the curve designer's freedom to manipulate the curve. However, this might not be sufficient, as [Ber+15] show that one could still iterate over many seeds (and potentially also over other "natural" choices) until they find a suitable curve. If a large enough proportion of curves (say, one in a million) is vulnerable to an attack known to the designer but not publicly, this offers an opportunity to insert a backdoor. In particular, we should be suspicious of curves whose seed's origin is unknown.

Two major standards recommending verifiably pseudorandom curves are the Brainpool and the FIPS 186 standards. In particular, the skepticism about the FIPS 186, created by NIST in collaboration with NSA, has been expressed multiple times [CLN15; Ber+15; Sco99]. Another notable paper considering the verifiably pseudorandom approach is "The Million dollar curve" which proposed a new source of public entropy for the seeds, combining lotteries from several different countries [Bai+].

Rigid methods. As a reaction to the NIST standardized curves, an interest in faster curves generated with rigid and verifiable methods not depending on a choice of initial seed has emerged. These include curves from an Internet draft "(ECC) Nothing Up My Sleeve (NUMS) Curves" [Bla+14], Bernstein's Curve25519 and Ed25519 [Ber06] and Ed448-Goldilocks [Ham15]. These curves aim at higher security levels while maintaining good performance. Curve25519, defined over an efficient prime $2^{255} - 19$, has been shown to perform elliptic curve operations at least twice as fast as the original NIST curves [Ber06] while satisfying stricter security conditions (including the twist security). Both Curve25519 and Ed448 have been included in the Transport Layer Security v1.3 standard [Res18].

2.2 Pairing-friendly curves

Pairing-based cryptosystems utilize non-degenerate bilinear maps from elliptic curves called pairings. A pairing maps points of subgroups $\mathbb{G}_1, \mathbb{G}_2$, of an elliptic curve over $\mathbb{F}_p$, to a subgroup of elements of finite field $\mathbb{G}_T \subset \mathbb{F}_{p^k}$.

$$\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T \subset \mathbb{F}_{p^k}$$

In order to have a cryptosystem secure, we need to have the discrete logarithm (ECDLP in $\mathbb{G}_1, \mathbb{G}_2$ and DLP in $\mathbb{G}_T$) difficult enough, while at the same time retaining a certain level of computational efficiency in the groups. Elliptic curves that satisfy these conditions are called pairing-friendly curves. More precisely, a pairing-friendly curve [FST10] is a curve over the field $\mathbb{F}_p$ satisfying:

- The order $N = \#E(\mathbb{F}_p)$ has a prime factor n of size at least $\sqrt{p}$. This is often expressed as $\rho \leq 2$ where $\rho = \frac{\log p}{\log n}$.
- The embedding degree k (the order of p in $\mathbb{F}_n^*$) is less than $\log_2(n)/8$.

The first condition guarantees efficient arithmetic on the elliptic curve and the second in the finite field $\mathbb{F}_{p^k}$. The security of the pairing-friendly curve is tightly connected to ρ and k by

$$\rho k = \frac{\log p^k}{\log n},$$

where $\log p^k$, $\log n$ determine the security of DLP in $\mathbb{F}_{p^k}$ and $E(\mathbb{F}_p)$, respectively. If we want to set, for example, 128-bit security, then the complexity of the attacks force n to have 256 bits (Pollard-ρ attack) and p^k roughly 4000 bits (number field sieve algorithm) [FST10]. This means that $k\rho \approx 16$ bits. Hence we have to balance the upper bounds for both k and ρ to force the set security levels. Both conditions from the definition have been chosen with this aim [FST10].

Generating pairing-friendly curves is considerably more difficult than generating curves for classical ECC. The additional condition on the embedding degree makes the desired pairing-friendly curves

too sparse and consequently the point-counting method (that has been used for NIST standardization) practically hopeless [BK98; LS06; ULS12]. Due to this, special generation methods must be employed. These are all based on the complex multiplication (CM) method with the exception of supersingular curves (curves with trivial p-torsion). Supersingular curves are very restrictive in terms of the embedding degree, but there are no conclusive arguments for their inferiority to ordinary curves [KM05]. Yet we omit them here as they were not included in standards for pairing-friendly curves [ISO17; Sak+21].

Complex multiplication method enables the construction of an elliptic curve if given the number of points N on the curve and the size of underlying prime field p [AM93]. The main limitation of the CM method is the condition that the CM discriminant $(p+1-N)^2 - 4p$ cannot be too large (roughly 50 bits [Sut12]). On the high level, there are two approaches to the CM method in the context of pairing-friendly curves. Firstly, there are generic methods including Cocks-Pinch [CP] and the Dupont-Enge-Morain [DEM05] algorithms. The second approach are the constructions of so-called families of pairing-friendly curves. In the rest of this section, we will briefly review both of these approaches and report their role in the standards of pairing-friendly curves (namely the ISO/IEC 15946 [ISO17] and IETF draft [Sak+21]).

Cocks-Pinch. The Cocks-Pinch and the Dupont-Enge-Morain algorithms enable the construction of curves of arbitrary embedding degree k, but they tend to produce curves with $\rho \approx 2$, thus requiring to take twice as large base field of the curve for given security level. Both algorithms essentially solve the following equations in n, t, p, y, D:

$$\Phi_k(t-1) = 0 \pmod{n}, \quad (t-2)^2 = Dy^2 \pmod{n}, \quad \frac{t^2 + Dy^2}{4} = p$$

As long as both n and p are prime, we can can use the CM method to construct a curve with embedding degree k. However, the equations show that we should expect $\rho \approx 2$. Although curves produced using these algorithms are, in general, less efficient, they are very flexible in terms of the input and output parameters. They allow arbitrary values of D, n, and embedding degree k. Their genericity follows the usual recommendations in ECC to choose curves as general as possible in

order to avoid any potential attacks. Moreover, recent work has shown the potential in improving their efficiency [GMT20]. The Cocks-Pinch algorithm has been included in the ISO/IEC 15946 [ISO17] standard for generating pairing-friendly curves.

Families of curves. First family of pairing-friendly curves was introduced in [MNT01] and has been since then called Miyaji-Nakabayashi-Takano (MNT) curves. The idea is similar to the Cocks-Pinch method, as the MNT construction also solves the equation $t^2 - 4p = Dy^2$, but instead of solving it in integers it looks for polynomial solutions. More precisely, the MNT method writes t, n and p as polynomials $t(x)$, $n(x)$ and $p(x)$ with parameter x. For example, for the embedding degree $k = 6$ the MNT method forces

$$t(x) = 1 \pm 2x, \quad p(x) = 4x^2 + 1, \quad n(x) = 4x^2 \pm 2x + 1.$$

The generation of an MNT curve is then done by essentially looking for the right parameter x. Notice that the degrees of n and p are the same, which causes $\rho \approx 1$. This makes them very efficient. Following on this result, further families of pairing-friendly curves have been found including the Barreto-Lynn-Scott (BLS) family [BLS02], the Barreto-Naehrig (BN) family [Per+10], the Freeman family [Fre06] and the Kachisa-Schaefer-Scott family [KSS08]. The MNT, BN and Freeman families have been included in the ISO/IEC 15946 standard.

The main disadvantage of the families of curves is the special form of their parameters. It has been repeatedly shown that discrete logarithm in finite fields over primes of a special form is subject to special types of number field sieve attacks [Gor92; Sem02; WD98]. The complexity of the number field sieve algorithm is $L_{p^k}(1/3, c) = \exp(c(\log p^k)(\log\log p^k)^{2/3})$, where c depends on the type of the sieve algorithm, but in our context $1.526 \leq c \leq 2.20$. [Sem02] showed that the frequently considered complexity for prime fields, given by $c = 1.9229$ [Sch93], lowers to $c = 1.526$ if p is of the form $ab^t + d$ for appropriate a, b, t, d. Recently, the tower number field sieve algorithm was significantly improved [KB16] for extension fields over primes of a special form. The complexity lowered to $c = 1.526$ for extension fields used for pairings as long as the underlying prime is of a small polynomial form and the exponent k in $\mathbb{F}_{p^k}$ is of the form $k = ij$, $gcd(i, j) = 1$, $i, j > 1$. The basic idea of their extended tower number

field sieve (exTNFS) algorithm is the simple equality $\mathbb{F}_{p^{ij}} = \mathbb{F}_{(p^i)^j}$ and the fact that they can target $\mathbb{F}_{(p^i)^j}$ with the same complexity as $\mathbb{F}_{p^j}$, where $P \approx p^i$. This led to a significant drop in the security of the BLS and BN families (the security of BN254 reduced from 128 bits to 100 bits). As a reaction to these attacks, trust in these curves lowered, and work on the standardization of pairing-friendly curves by the IETF Crypto Forum [Sak+21] has started. The draft of this standard currently recommends the BLS and BN curves with higher security levels.

2.3 Scalar decomposition

The bottleneck of all schemes in ECC is the scalar multiplication: given a scalar $k \in \mathbb{Z}$ and a point P on a curve, compute $k \cdot P = P + \cdots + P$. Existing methods for speeding up the scalar multiplication can be divided into two categories. First category consists of algorithms for general finite groups which treat the elliptic curve as a generic group and don't utilize its structure in any further way, for example the window [HMV04] or comb [LL94] methods. The second category of algorithms uses special elliptic curve properties. However, these methods are applicable to only certain classes of curves. Putting aside curves over base fields with efficient arithmetic (e.g., Mersenne prime base fields), the most prominent classes of such curves with efficient scalar multiplication are the Koblitz[2] curves [GLV01b].

The Gallant-Lambert-Vanstone (GLV) method [GLV01b] on Koblitz curves utilizes an efficiently computable endomorphism for speeding up the scalar multiplication. Instead of directly computing $k \cdot P$, on Koblitz curves, we can make a decomposition

$$k \cdot P = (k_0 + k_1\lambda) \cdot P = k_0 \cdot P + k_1 \cdot \phi(P), \tag{2.1}$$

where $k_0, k_1 \approx \sqrt{q}$, q is the size of base field $\mathbb{F}_q$ and ϕ is an appropriate endomorphism that acts as $\lambda \in \mathbb{Z}$ on $\langle P \rangle$. Endomorphism is a morphism of the curve to itself[3]

$$\phi : E(\overline{\mathbb{F}}_q) \to E(\overline{\mathbb{F}}_q).$$

2. Some authors use the term Koblitz curve only for curves over binary fields. Here we stick to the generalization to an arbitrary base field $\mathbb{F}_q$.
3. The curve must be considered over the algebraic closure $\overline{\mathbb{F}}_q$.

In general, endomorphisms are difficult to compute. However, Koblitz curves are special exceptions. An example of a Koblitz curve is given by $y^2 = x^3 + ax$ over $\mathbb{F}_p$ with $p = 1 \pmod 4$, which admits the nontrivial endomorphism $\phi : (x, y) \mapsto (-x, \alpha y)$ that requires only one multiplication in $\mathbb{F}_p$. The GLV decomposition (2.1) can lead up to at least 50% speed up compared to the best methods for general multiplication [GLV01b; Bru15]. However, several issues need to be considered when using the GLV decomposition (2.1):

- Koblitz curves are very rare, need to have a very low discriminant, and must be constructed through special methods. It has been shown that the existing efficient endomorphism can improve the Pollard's ρ attack [GLV01a; DGM99], but only by a factor of at most 2.
- The scalars k_0, k_1 in (2.1) must be found. Since we are essentially looking for a short vector (k_0, k_1) in the preimage of k through the map $(x, y) \mapsto x + y\lambda \pmod n$, lattice reduction techniques are usually used.
- The method improves simple scalar multiplication only if computing ϕ costs less than $(\log_2 n)/3$ point doublings and when an appropriate multi-scalar multiplication algorithm is used.

Authors of [GLV01b] proposed two possible extensions of their technique. Firstly, instead of randomly choosing the scalar k and looking for appropriate decomposition, we can randomly choose the decomposed parts k_1, k_2. However, this can introduce a bias to the generation of k [Ara+14].

The second proposition by the authors extends the method to higher dimensions. The idea is to extend the decomposition (2.1) to

$$k \cdot P = k_0 \cdot P + k_1 \cdot \phi(P) + k_2 \cdot \phi^2(P) + \cdots + k_m \phi^m(P),$$

with $k_i \approx q^{1/m}$. The problem here is to find the short vector $(k_0, \ldots, k_m)$ as there are no efficient reduction algorithms for higher dimensions in general. The authors shortly stated that the extension to dimension $m = 3$ has a potential as long as certain conditions on the characteristic polynomial of ϕ are met. Although the 2-dimensional case has been extensively studied [KL02; Par+02; SCQ02], progress in

higher dimensions have been exclusive to curves over extension fields. Namely, [Zho+10] studies the 3-dimensional case and [GLS11] the 4-dimensional case in extension fields. However, no further work has been done on this matter over prime fields since the original [GLV01b] work.

The GLV method on curves over the extension field $\mathbb{F}_{p^r}$, often called the Galbraith-Lin-Scott (GLS) method, was proposed in [Iij+02; GLS11]. The advantage of the extension field was the nontrivial Frobenius endomorphism. The Frobenius endomorphism π is an efficiently computable endomorphism, but has no use on curves over prime fields as its action is trivial. However, π is a nontrivial endomorphism over extension fields and can be used with an appropriate isogeny for the decomposition (2.1). While this is not relevant for any standard curve used in classical ECC as they are defined over prime fields, pairings of elliptic curves are defined over $\mathbb{F}_{p^r}$. It has been shown that in the context of pairing-friendly curves, the impact can be significant for high performance [HLX12; LG10].

2.4 Efficient prime field arithmetic

When designing an elliptic curve, one of the factors most affecting the efficiency of the curve operation is the choice of the base field. Considering only curves over prime fields, there are essentially two options for the design of the underlying prime. Either we pick a random prime of desired bit length, or we take a prime of a special form, usually of a polynomial form.

Random primes offer two advantages. Firstly, they are not affected by any hypothetical attacks on base-field primes of a special form. For example, special primes have been shown to cause vulnerabilities in implementations of additive scalar blinding [FRV14]. Secondly, since random primes do not require any specific implementation methods, compared to special form primes, they can be interchanged between platforms without problems. This is relevant for high-security software on embedded devices and hardware, where the change of forms of prime and consequently the implementation of arithmetic requires a redesign of the hardware. The Brainpool standard [LM10] followed

these ideas and designed elliptic curves over pseudorandom primes generated from known constants to be verifiable and transparent.

Special form primes, as mentioned, offer high-performance field arithmetic. Their main advantage is that they enable fast reduction step in the most used multiply-then-reduce approach to modular multiplication in the finite field. Optimal choices for primes with an efficient reduction are the Mersenne primes $M_k = 2^k - 1$. Modular multiplication modulo Mersenne primes can be done by integer multiplication, which results in an integer of the form $U \cdot 2^k + L$. The reduction step is then simple subtraction $U - L \pmod{M_k}$. However, these primes are too rare. The only Mersenne prime which is usable for the current security levels in ECC is $2^{521} - 1$. The NIST curve P-521 is defined over this prime [KD13].

Several generalizations of Mersenne primes have been proposed, which do not necessarily offer as efficient arithmetic as Mersenne primes, but are much more frequent. Here we mention all the currently used special form primes. NIST curves are defined over the Generalized Mersenne primes suggested in [Sol+99]. These are of the form $2^k \pm 2^l \pm 1$ and offer little overhead compared to the Mersenne primes [Bro+01]. Pseudo-Mersenne primes (Crandell primes) [Sol11] are of the form $2^k - c$ for small c. These offer very good performance no matter the constant c as long as it is small, which allows some flexibility in c. Their efficiency and simplicity for implementation make them a very popular choice and have been therefore used for Curve25519 and some of the NUMS curves. Special Montgomery primes $2^k c - 1$ have been chosen for the NUMS curves as well, since they allow speedup for the Montgomery multiplication algorithm. Further forms have been proposed, such as the Granger-Moss primes utilizing cyclotomic polynomials [GM13]. Summarizing this list, there are currently several options for high-performance finite field arithmetic depending on the choice of algorithms and their implementation.

3 Aims of the Thesis

The aim of my research plan is to examine selected topics in ECC, which can be summarized in the following objectives:

1. Revisit generation methods of pairing-friendly curves with the focus on the differences between general methods for ordinary curves and methods for families of curves. The aim is to raise the trust in the standard curves using large-scale statistical analysis.

2. Analyze multi-dimensional scalar decomposition algorithms and the optimal lattice reduction algorithms. This could potentially increase the efficiency and security of scalar multiplication on elliptic curves.

3. Increase efficiency of finite field arithmetic in the context of elliptic curve operations both in classical ECC and pairing-based cryptography.

The rest of this chapter will explain the research objectives in more detail. The chapter ends with proposed publication venues and a time schedule.

3.1 Generating pairing-friendly curves

There are multiple methods for generating elliptic curves for pairing-based cryptography. We attempt to answer the question: What are the differences between curves produced by these methods? The plan is to deploy a large-scale generation and statistical comparison of a wide range of properties using the DiSSECT tool. For this purpose, we will develop an extension of DiSSECT for pairing-friendly curves. This will include:

1. Implementation of all standard algorithms for generating pairing-friendly curves and generation of a large number of curves from each family.

2. Design of additional trait functions for DiSSECT that will cover all currently missing algebraic properties of pairing-friendly

curves, including characteristics related to isogenies of elliptic curves.

3. Improvement of current tools of automatic analysis of DiSSECT, which will include appropriate clustering methods. This will also require theoretical analysis of the statistical distributions of the pairing-friendly curve properties.

The goal is to statistically compare the families of pairing-friendly curves through the implemented traits, identify and comprehensibly list all currently known and unknown differences between these standards. In particular, we will compare the properties of families of pairing-friendly curves and properties of curves produced by the generic Cocks-Pinch and Dupont-Enge-Morain methods [CP; DEM05].

It is possible that no unexpected property of pairing-friendly curves and no previously unknown difference between standards will be discovered. In this case, this large-scale analysis will support the hypothesis that the standardized curves do not contain any hidden weaknesses.

3.2 Scalar decomposition

The efficiency of scalar multiplication on elliptic curves can be significantly increased if a scalar decomposition with an appropriate multi-scalar multiplication is used. Our aim is to analyze multi-dimensional decomposition on elliptic curves over prime fields.

To be more precise, we will study what is the optimal dimension n in the scalar decomposition

$$k = a_0 + a_1\psi + \cdots + a_{n-1}\psi^{n-1},$$

as well as an appropriate choice of fast endomorphism ψ and scalars a_i. In particular, we will study the effect of algebraic properties of a_i (e.g., smoothness) on the efficiency of the decomposition. This will require extensive research into lattice reduction algorithms that might be relevant for these decompositions. It is probable that any current lattice reductions algorithms will not be appropriate or efficient enough and we will need to modify them for the specific needs of these decompositions or possibly the individual elliptic curves and base fields.

In case that appropriate reduction algorithms reveal improvements for the scalar decomposition for $n > 2$, the next goal will be to design a curve that utilizes this speed up. We will have to take into consideration several other influences, including the type of the base field, the curve generation algorithm, the endomorphism ring, embedding degree, and others depending on the potential usage of the curves.

However, in case that no lattice reduction algorithms or their modifications will be efficient enough, other approaches will have to be considered for the speed up of the scalar decomposition, for example, to change the focus on curves over extension fields.

3.3 Efficient arithmetic over the base-field

All of the modern high-performance standard curves draw their efficiency from the appropriate choice of the type of the given base field prime. However, there is currently no universal type of prime that would be appropriate for all conditions on security and implementation. We plan to study the choice of the base field prime and the effects on the efficiency of the field arithmetic in the context of elliptic curves. A more detailed direction of the research will be specified after initial experiments and after we get a more in-depth understanding of the relevant algorithms. One possible direction is the pairing-friendly curves. Since the Cocks-Pinch and Dupont-Enge-Morain methods for generating ordinary pairing-friendly do not suffer from special form parameters, there is a potential for low-level optimization of base field arithmetic, including picking the appropriate type of prime. This could have a great impact as the main drawback of these curves is their lower efficiency. Moreover, depending on the result of this research, we may be able to extend its application to other types of standard curves, such as supersingular curves in isogeny-based cryptography.

3.4 DiSSECT

With the improvements of the DiSSECT tool during the research mentioned above, we will continue with the analysis of all standard curves and study any further relevant research questions that will originate from the results of this analysis. One of the results of DiSSECT de-

velopment is the discovery of the fact that the probability of finding an elliptic curve with prime order depends on the type of the base field. This seems like a not fully documented result that we will try to extend together with its application on elliptic curve generation and others.

3.5 Publication venues

We aim for some of the following publication venues:

- European Symposium on Research in Computer Security (ESORICS), CORE A rank.
- Annual International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT), CORE A rank.
- Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), CORE A* rank.
- Annual International Conference on the Theory and Applications of Cryptology (AFRICACRYPT), not ranked yet, acceptance rate 30%.
- International Conference on Practice and Theory of Public-Key Cryptography (PKC), CORE B rank.
- Conference on Selected Areas in Cryptography (SAC), CORE B rank.
- Journal of Cryptology, Q2.
- Mathematics of Computation, Q1.
- Design, Codes and Cryptography, Q1.

3.6 Schedule

- Winter 2022: The study of the distribution of prime order curves (Section 3.4).
- Spring 2022: Implementing the extension for pairing-friendly curves to DiSSECT. Analysis of initial results (Section 3.1).
- Fall 2022: Further work on the pairing-friendly curve analysis (Section 3.1). Research of the scalar decomposition (Section 3.2).
- Spring 2023: Design of an elliptic curve based on the previous results (Section 3.2). Initial experiments with the finite field arithmetic (Section 3.3).
- Fall 2023: Design of an optimal base-field prime (Section 3.3).
- Spring 2024: Writing the dissertation.

4 Achieved results

This chapter presents an overview of [SSD], an already achieved result in the area of elliptic curve cryptography and standard curves[1]. My contribution to this work is described at the end of the chapter.

In [SSD], we introduce what we believe is the first tool for a large-scale analysis of standard elliptic curves, called DiSSECT. The idea of DiSSECT is that if a hidden weakness is present in a curve, it will manifest itself via a statistical deviation when we compare the curve to a large number of generic curves. The whole work can be divided into three parts: data collection of elliptic curves, design of the statistical tool, and an analysis of its results.

Elliptic curve generation. In order to analyze standardized curves and their methods of generation, we have adopted six major standards for elliptic curve generation, including one for pairing-friendly curves. Proper implementation of all algorithms required extensive research of published standards in ECC from the past 25 years. The package of the implementations for curve generation is an open-source tool available for anyone in need of producing secure elliptic curves (e.g., for the mathematical study of curves as algebraic objects). We have managed to generate over 250 000 elliptic curves and for easier study of such a large amount of data, we have stored them in a database that is publicly available through an API on our website.

Traits. Functions characterizing properties of elliptic curves – which we call traits for brevity, are one of the main parts of DiSSECT. Traits take a curve as an input (sometimes with additional parameters) and return numerical results, thus reducing the elliptic curve, a complex mathematical structure, to a vector of numbers. The aim is to capture as many elliptic curve properties using these traits as possible. These include all the classical characteristics such as torsion points, low degree isogenies, embedding degree, or properties of the endomorphism algebra of the curve. We run these traits on all generated curves and store the results in our database. For the design of the traits, we had to systematically examine the state of the art of the theory of elliptic

1. Parts of this section are adopted from [SSD].

curves. DiSSECT contains 22 traits, but provides a framework for easy implementation of further traits.

Analysis of the results. DiSSECT offers options for graphical inspection of the results of traits. However, a manual analysis does not scale well, so we utilize automated approaches to identify suspicious curves. We ran the local outlier factor algorithm to identify outliers within each augmented dataset. If the approach reported a standard curve as an outlier, we inspected such results closer using the visualization tools.

Findings of DiSSECT. DiSSECT revealed that the generation process of three GOST curves described by Alekseev, Nikolaev, and Smyshlyaev [ANS18] is inconsistent with their properties. More precisely, the authors claim a verifiably pseudorandom generation which we prove to be improbable with probability 2^{-240}. Properly documenting such inconsistencies is crucial for establishing trust in the standard curves and the whole standardization system. We cannot expect users, developers, or policy-makers to notice even fairly obvious deviations as they often do not access the parameters directly.

During the generation of simulated curves, we have found a bias in the selection of secure curves due to the choice of prime field. This demonstrates the potential of statistical analysis of elliptic curves in general.

We found an interesting, previously undescribed property of the BLS12-381 curve that is caused by its generation method. Recent attacks [KB16] on special properties of pairing-friendly curves have proven that awareness of all properties caused by the pairing-friendly generation methods is crucial. In particular, our approach of isolation of individual properties in BLS generation has shown to have a lot of potential for future research of security of pairing-friendly curves.

Submitted conference paper

V. Sedláček, V. Suchánek, and A. Dufka. *DiSSECT: Distinguisher of Standard & Simulated Elliptic Curves via Traits*. Submitted to AFRICACRYPT 2022.

- My contribution: data collection (curve generation), design of test functions, analysis of results, writing – 35%.

Bibliography

[Age11] Agence Nationale de la Securite des Systemes d'Information. *Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français*. 2011. URL: https://www.legifrance.gouv.fr/download/pdf?id=QfYWtPSAJVtAB_c6Je5tAv00OY2r1-ad3LaVVmnStGvQ=.

[ANS18] E. K. Alekseev, V. Nikolaev, and S. V. Smyshlyaev. *On the security properties of Russian standardized elliptic curves*. 2018.

[ANSI98] *American National Standard X9.62-1998, Public key cryptography for the financial services industry: the elliptic curve digital signature algorithm (ECDSA)*. Tech. rep. Accredited Standards Committee X9, 1998.

[Ara+14] D. F. Aranha, P.-A. Fouque, B. Gérard, J.-G. Kammerer, M. Tibouchi, and J.-C. Zapalowicz. "GLV/GLS Decomposition, Power Analysis, and Attacks on ECDSA Signatures with Single-Bit Nonce Bias". In: *Advances in Cryptology – ASIACRYPT 2014*. Ed. by P. Sarkar and T. Iwata. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 262–281.

[AM93] A. O. L. Atkin and F. Morain. "Elliptic curves and primality proving". In: *Mathematics of computation* 61.203 (1993), pp. 29–68.

[Bai+] T. Baignères, C. Delerablée, M. Finiasz, L. Goubin, T. Lepoint, and M. Rivain. *Trap Me If You Can – Million Dollar Curve*. Cryptology ePrint Archive, Report 2015/1249. https://ia.cr/2015/1249.

[BK98] R. Balasubramanian and N. Koblitz. "The improbability that an elliptic curve has subexponential discrete log problem under the MOV algorithm". In: *Journal of cryptology* 11.2 (1998), pp. 141–145.

[BJS07] E. B. Barker, D. Johnson, and M. E. Smid. *SP 800-56A. recommendation for pair-wise key establishment schemes using discrete logarithm cryptography (revised)*. 2007.

[BLS02] P. S. Barreto, B. Lynn, and M. Scott. "Constructing elliptic curves with prescribed embedding degrees". In: *Interna-*

tional Conference on Security in Communication Networks. Springer. 2002, pp. 257–267.

[Ber06] D. J. Bernstein. "Curve25519: new Diffie-Hellman speed records". In: *International Workshop on Public Key Cryptography*. Springer. 2006, pp. 207–228.

[Ber+15] D. J. Bernstein, T. Chou, C. Chuengsatiansup, A. Hülsing, E. Lambooij, T. Lange, R. Niederhagen, and C. Van Vredendaal. "How to Manipulate Curve Standards: A White Paper for the Black Hat http://bada55. cr. yp. to". In: *International Conference on Research in Security Standardisation*. Springer. 2015, pp. 109–139.

[BLN15] D. J. Bernstein, T. Lange, and R. Niederhagen. *Dual EC: A Standardized Back Door*. Cryptology ePrint Archive, Report 2015/767. https://eprint.iacr.org/2015/767. 2015.

[BL] D. J. Bernstein and T. Lange. *SafeCurves: choosing safe curves for elliptic-curve cryptography*. https://safecurves.cr.yp.to/. (Visited on 08/17/2021).

[Bla+14] B. Black, J. Bos, C. Costello, P. Longa, and M. Naehrig. *Elliptic Curve Cryptography (ECC) Nothing Up My Sleeve (NUMS) Curves and Curve Generation*. Internet-Drafts are working documents of the Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-black-numscurves-02. 2014.

[BF01] D. Boneh and M. Franklin. "Identity-based encryption from the Weil pairing". In: *Annual international cryptology conference*. Springer. 2001, pp. 213–229.

[BGW05] D. Boneh, C. Gentry, and B. Waters. "Collusion resistant broadcast encryption with short ciphertexts and private keys". In: *Annual international cryptology conference*. Springer. 2005, pp. 258–275.

[BLS01] D. Boneh, B. Lynn, and H. Shacham. "Short signatures from the Weil pairing". In: *International conference on the theory and application of cryptology and information security*. Springer. 2001, pp. 514–532.

[Bro+01] M. Brown, D. Hankerson, J. López, and A. Menezes. "Software implementation of the NIST elliptic curves over prime fields". In: *Cryptographers' Track at the RSA Conference*. Springer. 2001, pp. 250–265.

[Bru15] B. B. Brumley. *Faster software for fast endomorphisms*. Cryptology ePrint Archive, Report 2015/036. https://ia.cr/2015/036. 2015.

[Cer] Certicom Research. *Standards for Efficient Cryptography Group*. https://secg.org/. (Visited on 08/13/2021).

[Che+] S. Checkoway, J. Maskiewicz, C. Garman, J. Fried, S. Cohney, M. Green, N. Heninger, R.-P. Weinmann, E. Rescorla, and H. Shacham. "A Systematic Analysis of the Juniper Dual EC Incident". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pp. 468–479.

[CP] C. Cocks and R. Pinch. "ID-based cryptosystems based on the Weil pairing, 2001". In: *Unpublished manuscript* ().

[CLN15] C. Costello, P. Longa, and M. Naehrig. "A brief discussion on selecting new elliptic curves". In: *Microsoft Research. Microsoft* 8 (2015).

[Dol10] V. Dolmatov. "GOST R 34.10-2001: Digital Signature Algorithm". In: *Independent Submission, Request for Comments* 5832 (2010), pp. 2070–1721.

[DEM05] R. Dupont, A. Enge, and F. Morain. "Building curves with arbitrary small MOV degree over finite prime fields". In: *Journal of Cryptology* 18.2 (2005), pp. 79–89.

[DGM99] I. Duursma, P. Gaudry, and F. Morain. "Speeding up the discrete log computation on curves with automorphisms". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 1999, pp. 103–121.

[FRV14] B. Feix, M. Roussellet, and A. Venelli. "Side-channel analysis on blinded regular scalar multiplications". In: *International Conference on Cryptology in India*. Springer. 2014, pp. 3–20.

[FIP94] P. FIPS. "186 National Institute of Standards and Technology". In: *FIPS PUB 186: Digital Signature Standard* (1994).

[FIP00] P. FIPS. *186-2. Digital Signature Standard (DSS). US Department of Commerce/National Institute of Standards and Technology*. 2000.

[Fre06] D. Freeman. "Constructing pairing-friendly elliptic curves with embedding degree 10". In: *International Algorithmic Number Theory Symposium*. Springer. 2006, pp. 452–465.

[FST10] D. Freeman, M. Scott, and E. Teske. "A taxonomy of pairing-friendly elliptic curves". In: *Journal of cryptology* 23.2 (2010), pp. 224–280.

[FR94] G. Frey and H.-G. Rück. "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves". In: *Mathematics of Computation* 62.206 (1994), pp. 865–874.

[GLS11] S. D. Galbraith, X. Lin, and M. Scott. "Endomorphisms for faster elliptic curve cryptography on a large class of curves". In: *Journal of cryptology* 24.3 (2011), pp. 446–469.

[GM00] S. D. Galbraith and J. McKee. "The probability that the number of points on an elliptic curve over a finite field is prime". In: *Journal of the London Mathematical Society* 62.3 (2000), pp. 671–684.

[GLV01a] R. P. Gallant, R. J. Lambert, and S. A. Vanstone. "Faster point multiplication on elliptic curves with efficient endomorphisms". In: *Annual International Cryptology Conference (CRYPTO 2001)*. Springer. 2001, pp. 190–200. ISBN: 978-3-540-44647-7.

[GLV01b] R. P. Gallant, R. J. Lambert, and S. A. Vanstone. "Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms". In: *Advances in Cryptology — CRYPTO 2001*. Ed. by J. Kilian. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 190–200.

[Gor92] D. M. Gordon. "Designing and detecting trapdoors for discrete log cryptosystems". In: *Annual International Cryptology Conference*. Springer. 1992, pp. 66–75.

[Gor93] D. M. Gordon. "Discrete Logarithms in GF(P) Using the Number Field Sieve". In: *SIAM Journal on Discrete Mathematics* 6.1 (1993), pp. 124–138.

[GM13] R. Granger and A. Moss. "Generalised Mersenne numbers revisited". In: *Mathematics of Computation* 82.284 (2013), pp. 2389–2420.

[GMT20] A. Guillevic, S. Masson, and E. Thomé. "Cocks–Pinch curves of embedding degrees five to eight and optimal ate pairing computation". In: *Designs, Codes and Cryptography* 88.6 (2020), pp. 1047–1081.

[Hal13] T. C. Hales. "The NSA back door to NIST". In: *Notices of the AMS* 61.2 (2013), pp. 190–192.

[Ham15] M. Hamburg. *A note on high-security general-purpose elliptic curves*. Cryptology ePrint Archive, Report 2015/625. https://eprint.iacr.org/2015/625.pdf. 2015.

[HMV04] D. Hankerson, A. Menezes, and S. Vanstone. *Guide to elliptic curve cryptography*. Springer Professional Computing. Springer, 2004. ISBN: 0-387-95273-X.

[Hop+16] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox. "Zcash protocol specification". In: *GitHub: San Francisco, CA, USA* (2016).

[HLX12] Z. Hu, P. Longa, and M. Xu. "Implementing the 4 - dimensional GLV method on GLS elliptic curves with j-invariant 0". In: *Designs, Codes and Cryptography* 63.3 (2012), pp. 331–343.

[IEEE00] *IEEE Standard - Specifications for Public-Key Cryptography*. Standard. IEEE Std 1363-2000 Working Group, 2000.

[Iij+02] T. Iijima, K. Matsuo, J. Chao, and S. Tsujii. "Construction of Frobenius maps of twists elliptic curves and its application to elliptic scalar multiplication". In: *Proc. of SCIS*. 2002, pp. 699–702.

[ISO17] ISO/IEC 15946. *Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 5: Elliptic curve generation*. 2017.

[JMV01] D. Johnson, A. Menezes, and S. Vanstone. "The elliptic curve digital signature algorithm (ECDSA)". In: *International journal of information security* 1.1 (2001), pp. 36–63.

[JL17] S. Josefsson and I. Liusvaara. *Edwards-Curve Digital Signature Algorithm (EdDSA)*. RFC 8032. Jan. 2017. DOI: 10.17487/RFC8032. URL: https://rfc-editor.org/rfc/rfc8032.txt.

[Jou09] A. Joux. "Introduction to identity-based cryptography". In: *Identity-Based Cryptography* 2 (2009), p. 1.

[KSS08] E. J. Kachisa, E. F. Schaefer, and M. Scott. "Constructing Brezing-Weng pairing-friendly elliptic curves using elements in the cyclotomic field". In: *International Conference on Pairing-Based Cryptography*. Springer. 2008, pp. 126–135.

[KD13] C. F. Kerry and C. R. Director. "FIPS PUB 186-4 federal information processing standards publication digital signature standard (DSS)". In: (2013).

[KL02] D. Kim and S. Lim. "Integer decomposition for fast scalar multiplication on elliptic curves". In: *International Workshop on Selected Areas in Cryptography*. Springer. 2002, pp. 13–20.

[KB16] T. Kim and R. Barbulescu. "Extended tower number field sieve: A new complexity for the medium prime case". In: *Annual International Cryptology Conference*. Springer. 2016, pp. 543–571.

[Kob87] N. Koblitz. "Elliptic curve cryptosystems". In: *Mathematics of computation* 48.177 (1987), pp. 203–209.

[KM05] N. Koblitz and A. Menezes. "Pairing-based cryptography at high security levels". In: *IMA International Conference on Cryptography and Coding*. Springer. 2005, pp. 13–36.

[LL94] C. H. Lim and P. J. Lee. "More Flexible Exponentiation with Precomputation". In: *Advances in Cryptology - CRYPTO '94*. Ed. by Y. G. Desmedt. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 95–107.

[LM10] M. Lochter and J. Merkle. *Elliptic curve cryptography (ECC) brainpool standard curves and curve generation*. Tech. rep. RFC 5639, March, 2010.

[Loc+14] M. Lochter, J. Merkle, J.-M. Schmidt, and T. Schutze. *Requirements for Standard Elliptic Curves.* IACR Cryptology ePrint Archive, Report 2014/832. 2014.

[LG10] P. Longa and C. Gebotys. "Efficient techniques for high-speed elliptic curve cryptography". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. 2010, pp. 80–94.

[LS06] F. Luca and I. E. Shparlinski. "Elliptic curves with low embedding degree". In: *Journal of Cryptology* 19.4 (2006), pp. 553–562.

[MOV93] A. J. Menezes, T. Okamoto, and S. A. Vanstone. "Reducing elliptic curve logarithms to logarithms in a finite field". In: *IEEE Transactions on information Theory* 39.5 (1993), pp. 1639–1646.

[Mil85] V. S. Miller. "Use of elliptic curves in cryptography". In: *Conference on the theory and application of cryptographic techniques*. Springer. 1985, pp. 417–426.

[MNT01] A. Miyaji, M. Nakabayashi, and S. Takano. "New Explicit Conditions of Elliptic Curve Traces for FR-Reduction". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 84 (2001), pp. 1234–1243.

[Nak08] S. Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: *Decentralized Business Review* (2008), p. 21260.

[Par+02] Y.-H. Park, S. Jeong, C. H. Kim, and J. Lim. "An alternate decomposition of an integer for faster point multiplication on certain elliptic curves". In: *International Workshop on Public Key Cryptography*. Springer. 2002, pp. 323–334.

[Per+10] G. C. C. F. Pereira, M. A. S. Jr, M. Naehrig, and P. S. L. M. Barreto. *A Family of Implementation-Friendly BN Elliptic Curves*. Cryptology ePrint Archive, Report 2010/429. https://ia.cr/2010/429. 2010.

[PH78] S. Pohlig and M. Hellman. "An Improved Algorithm for Computing Logarithms over GF(p) and Its Cryptographic Significance". In: *IEEE Transactions on Information Theory* (1978), pp. 106–110.

[Pol74] J. M. Pollard. "Theorems on factorization and primality testing". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 76.3 (1974), pp. 521–528.

[PLK06] V. Popov, S. Leontiev, and I. Kurepkin. *Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms*. 2006. URL: https://rfc-editor.org/rfc/rfc4357.txt.

[Res18] E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Aug. 2018. DOI: 10.17487/RFC8446. URL: https://rfc-editor.org/rfc/rfc8446.txt.

[Sak+21] Y. Sakemi, T. Kobayashi, T. Saito, and R. S. Wahby. *Pairing-Friendly Curves*. Internet-Draft. Internet Engineering Task Force, July 2021. 54 pp. URL: datatracker.ietf.org/doc/html/draft-irtf-cfrg-pairing-friendly-curves-10.

[SA+98] T. Satoh, K. Araki, et al. "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves". In: *Rikkyo Daigaku sugaku zasshi* 47.1 (1998), pp. 81–92.

[Sch93] O. Schirokauer. "Discrete logarithms and local units". In: *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences* 345.1676 (1993), pp. 409–423.

[Sch13] B. Schneier. *The NSA Is Breaking Most Encryption on the Internet*. https://www.schneier.com/blog/archives/2013/09/the_nsa_is_brea.html. 2013. (Visited on 08/13/2021).

[Sch95] R. Schoof. "Counting points on elliptic curves over finite fields". In: *Journal de Théorie des Nombres de Bordeaux* (1995), pp. 219–254.

[Sco99] M. Scott. *Re: NIST annouces set of Elliptic Curves*. https://web.archive.org/web/20160313065951/https://groups.google.com/forum/message/raw?msg=sci.crypt/mFMukSsORmI/FpbHDQ6hM_MJ. 1999. (Visited on 08/13/2021).

[SSD] V. Sedláček, V. Suchánek, and A. Dufka. *DiSSECT: Distinguisher of Standard & Simulated Elliptic Curves via Traits*. Submitted to AFRICACRYPT 2022.

[Sem96] I. Semaev. "On computing logarithms on elliptic curves". In: *Discrete Mathematics and Applications* 6.1 (1996), pp. 69–76.

[Sem98] I. Semaev. "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p". In: *Mathematics of computation* 67.221 (1998), pp. 353–356.

[Sem02] I. Semaev. "Special prime numbers and discrete logs in finite prime fields". In: *Mathematics of computation* 71.237 (2002), pp. 363–377.

[SSL14] S. Shen, S. Shen, and X. Lee. *SM2 Digital Signature Algorithm*. Internet-Draft. 2014. URL: datatracker.ietf.org/doc/html/draft-shen-sm2-ecdsa-02.

[SCQ02] F. Sica, M. Ciet, and J.-J. Quisquater. "Analysis of the Gallant-Lambert-Vanstone method based on efficient endomorphisms: Elliptic and hyperelliptic curves". In: *International Workshop on Selected Areas in Cryptography*. Springer. 2002, pp. 21–36.

[Sma99] N. P. Smart. "The discrete logarithm problem on elliptic curves of trace one". In: *Journal of cryptology* 12.3 (1999), pp. 193–196.

[Sol+99] J. A. Solinas et al. *Generalized mersenne numbers*. Citeseer, 1999.

[Sol11] J. A. Solinas. "Pseudo-Mersenne Prime". In: *Encyclopedia of Cryptography and Security*. Ed. by H. C. A. van Tilborg and S. Jajodia. Boston, MA: Springer US, 2011, pp. 992–992. DOI: 10.1007/978-1-4419-5906-5_42. URL: https://doi.org/10.1007/978-1-4419-5906-5_42.

[Sut12] A. V. Sutherland. "Accelerating the CM method". In: *LMS Journal of Computation and Mathematics* 15 (2012), pp. 172–204.

[ULS12] J. Urroz, F. Luca, and I. Shparlinski. "On the number of isogeny classes of pairing-friendly elliptic curves and statistics of MNT curves". In: *Mathematics of Computation* 81.278 (2012), pp. 1093–1110.

[WD98] D. Weber and T. Denny. "The solution of McCurley's discrete log challenge". In: *Annual International Cryptology Conference*. Springer. 1998, pp. 458–471.

[Wir00] Wireless Application Protocol Forum. *Wireless Application Protocol Wireless Transport Layer Security Specification*. https://web.archive.org/web/20170829023257/https://www.wapforum.org/tech/documents/WAP-199-WTLS-20000218-a.pdf. 2000.

[Zho+10] Z. Zhou, Z. Hu, M. Xu, and W. Song. "Efficient 3 - dimensional GLV method for faster point multiplication on some GLS elliptic curves". In: *Information Processing Letters* 110.22 (2010), pp. 1003–1006.

A An appendix

The paper *DiSSECT: Distinguisher of Standard & Simulated Elliptic Curves via Traits* starts at the next page.

ACAT: Anonymous curve analyzer tool

Anonymous submission

Abstract. It can be tricky to trust elliptic curves standardized in a non-transparent way. To rectify this, we propose a systematic methodology for analyzing curves and statistically comparing them to the expected values of a large number of generic curves with the aim of identifying any deviations in the standard curves.
For this purpose, we put together the largest publicly available database of standard curves. To identify unexpected properties of standard generation methods and curves, we simulated over 250 000 curves by mimicking the generation process of four standards. We compute 22 different properties of curves and analyze them with automated methods to identify any unexpected deviations in standard curves, pointing to possible weaknesses.

Keywords: elliptic curves · standards · simulations · testing tool.

1 Introduction

Many prominent cryptographers [Sch13; Loc+14; Ber+15; CLN15; Sco99] criticize the selection of curves in all major elliptic curve standards [ANS05; Cer10; Age11; SSL14]. The lack of explanation of parameters used in cryptographic standards provides a potential space for weaknesses or inserted vulnerabilities.

This is a serious matter, considering that the popular NIST curves have been influenced by the NSA, which has been involved in a number of incidents: the Dual EC DRBG standard manipulation [Hal13; BLN16; Che+], Clipper chip backdoor [Tor94], or the deliberate weakening of the A5 cipher [BD00]. Bernstein et al. [Ber+15] justify such skepticism, showing that the degrees of freedom in the generation of elliptic curves offer a means to insert a backdoor. Relying on the supposed implausibility [KM16] of weak curves escaping detection for such a long time may prove catastrophic. As the usage of NIST curves increases with time [Val+18], the impact of any vulnerability would be extensive at this moment.

Even though newer, more rigidly generated curves like Curve25519 [Ber06], Ed448-Goldilocks [Ham15b] or NUMS curves [Bla+14] are on the rise, Lochter et al. [Loc+14] argue that *"perfect rigidity, i.e., defining a process that is accepted as completely transparent and traceable by everyone, seems to be impossible."* Thus, a thorough wide-scale analysis of the standard curves is important to establish trust in elliptic curve cryptosystems.

There is no clear way how to look for unknown vulnerabilities, especially within elliptic curve cryptosystems with such a rich and complex theory behind them. Our main idea is that if a hidden weakness is present in a curve, it can

manifest itself via a statistical deviation when we compare the curve to a large number of generic curves. We consider two cases: either the deviation stems from the generation method of the curve or from the choice of its parameters. Consequently, we compare the curve to two types of curves - random curves in the former case and curves generated according to the same standard in the latter case.

Core contributions. Our work provides the following contributions:

- We assemble the first public database of all standard[1] curves (to the best of our knowledge), with comprehensive parameter details.
- We develop an open-source, extensible framework ACAT[2] for generating curves according to known standards, and for statistical analysis of the standards. This includes a large number of test functions and visualization tools.
- We find properties of the standard GOST curves [PLK06; SF16] that are inconsistent with the claimed [ANS18] generation method and present unreported properties of the BLS12-381 [BLS02] curve.

In addition, we offer a new methodology for testing properties and potential weaknesses of standard curve parameters and their generation methods. By a systematic analysis of standard curves, we rule out multiple types of problems, raising the level of trust in most of the curves.

This paper is organized as follows: In Section 2, we briefly survey the major elliptic curve standards. Section 3 introduces our database and explains our methodology for simulating and distinguishing curves. Section 4 gives an overview of our proposed trait functions and presents the findings of the outlier detection. We report on the technicalities of our tool in Section 5 and draw conclusions in Section 6.

2 Background

Throughout the paper, we will use the following notation: $p \geq 5$ is a prime number, $\mathbb{F}_p$ is a finite field of size p, $E(a, b)\colon y^2 = x^3 + ax + b$ is an elliptic curve over $\mathbb{F}_p$, $n = \#E(\mathbb{F}_p)$ is the order of the group of points over $\mathbb{F}_p$, l the largest prime factor of n, $h = n/l$ is the cofactor, and $t = p + 1 - n$ is the trace of Frobenius of E. Parts of this section are adopted from [Sed22].

Even though pairing-friendly curves and curves over binary and extension fields have found some applications, we focus mainly[3] on the prevalent category of prime field curves that are recommended for classical elliptic curve cryptosystems. We include Montgomery and (twisted) Edwards curves as well, though for unification purposes, we convert them to the short Weierstrass form, which is universal.

[1] In this work, we use the adjective "standard" for widely used curves (from actual standards and even that are/were a de facto standard).

[2] The name of the project is anonymized for review. ACAT is an acronym of the Anonymous curve analyzer tool.

[3] However, binary, extension and pairing-friendly curves are included in our database as well and have been included partly in our analysis.

Known curve vulnerabilities. When using elliptic curve cryptography (ECC) in the real world, both sides of the scheme must agree on a choice of a particular curve. We need curves where the elliptic curve discrete logarithm problem (ECDLP) is difficult enough. The SafeCurves website [BL] presents a good overview of known curve vulnerabilities (and discusses implementation-specific vulnerabilities as well). In short, to avoid the known mathematical attacks, curve should satisfy the following criteria:

- p should be large enough, e.g., over 256 bits if we aim for 128-bit security;
- l should be large enough, as this determines the complexity of the Pohlig-Hellman attack [PH78] (again, roughly 256 bits for 128-bit security); ideally, the same should hold for the quadratic twist of the curve (to avoid attacks on implementations[4]);
- the curve should not be anomalous (i.e., $n \neq p$), otherwise the Semaev-Satoh-Araki-Smart additive transfer attack applies [Sma99; Sem98; SA+98];
- the embedding degree of E (i.e., the order of p in $\mathbb{F}_l^\times$) should be large enough (e.g., at least 20), otherwise the MOV attack based on multiplicative transfer using the Weil and Tate pairings applies [Sem96; FR94; MOV93] – in particular, this rules out the cases $t = 0$ and $t = 1$;
- the absolute value of the CM field discriminant (i.e., the absolute discriminant of the field $\mathbb{Q}(\sqrt{t^2 - 4p})$ should not be too low ([BL] suggests at least 2^{100}), otherwise Pollard's ρ attack can be speeded up due to the presence of efficiently computable endomorphisms of the curve [GLV01]. This does not pose a serious threat for the moment though, as the limits of the speedup are reasonably well understood.

2.1 Overview of standard curve generation

The main approach for finding a curve that satisfies the mentioned conditions is repeatedly selecting curve parameters until the conditions hold. The main bottleneck here is point-counting: we can use the polynomial SEA algorithm [Sch95], but in practice, it is not fast enough to allow on-the-fly ECC parameter generation.

Several standardization organizations have therefore proposed elliptic curve parameters for public use. To choose a curve, one has to first pick the appropriate base field. Several standards also choose primes of special form for more efficient arithmetic, e.g., generalized Mersenne primes [Sol11] or Montgomery-friendly primes [CLN15]. When the field and the curve form are fixed, it remains to pick two[5] coefficients, though one of them is often fixed.

In this subsection, we survey the origin of all the standard curves. For easier orientation, we divide the generation methods of standard curves into three rough categories. Curves of unknown or ambiguous origin are hard to trust,

[4] This is often ignored in the standards though, except for the NUMS standard and individual curves such as Curve25519.

[5] At least for short Weierstrass, Montgomery and twisted Edwards curves.

while verifiably pseudorandom ones make use of one-way functions in hopes of addressing this. Other rigid methods go even further in attempts to increase their transparency.

Unknown or ambiguous origin. The following is a list of curves we analyze and whose method of generations is either completely unexplained or ambiguous.

- The 256-bit curve `FRP256v1` has been recommended in 2011 by the French National Cybersecurity Agency (ANSSI) in the Official Journal of the French Republic [Jou11]. The document does not specify any method of generation.
- The Chinese SM2 standard [SSL14] was published in 2010 [DY15] by the Office of State Commercial Cryptography Administration (OSCCA). The standard recommends one 256-bit curve but does not specify the generation method.
- In addition to the Russian GOST R 34.10 standard from 2001, six standardized curves were provided in [PLK06] and [SF16]. Again, no method of generation was specified in the original standard, although there were some attempts for explanation afterward [ANS18]. We discuss this in further sections.
- The Wireless Application Protocol Wireless Transport Layer Security Specification [Wir00] recommends 8 curves, 3 of which are over prime fields and are not copied from previous standards. Their method of generation is not described.
- Apart from verifiably pseudorandom curves, the Standards for Efficient Cryptography Group (SECG) [Cer] recommends so-called Koblitz curves[6], which possess an efficiently computable endomorphism. However, the standard only vaguely states *"The recommended parameters associated with a Koblitz curve were chosen by repeatedly selecting parameters admitting an efficiently computable endomorphism until a prime order curve was found."*

Verifiably pseudorandom curves. Aiming to re-establish the trust in ECC, several standards have picked the coefficients a, b in the Weierstrass form $y^2 = x^3 + ax + b$ in a so-called *verifiably pseudorandom* way, as Figure 1 demonstrates.

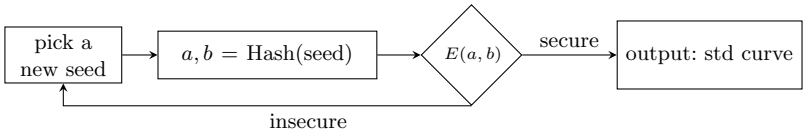

Fig. 1: A simplified template for generating verifiably pseudorandom curves over fixed $\mathbb{F}_p$.

The details differ and we study them more closely in Section 3.2. The common idea is that the seed is made public, which limits the curve designer's freedom to

[6] Analyzing these curves is of great importance due to the usage of `secp256k1` in Bitcoin [Nak08].

manipulate the curve. However, this might not be sufficient, as Bernstein et al. [Ber+15] show that one could still iterate over many seeds (and potentially also over other "natural" choices) until they find a suitable curve. If a large enough proportion of curves (say, one in a million) is vulnerable to an attack known to the designer, but not publicly, this offers an opportunity to insert a backdoor. In particular, we should be suspicious of curves whose seed's origin is unknown.

In our analysis, we will focus on two major standards containing verifiably pseudorandom generation procedures:

- The Brainpool curves, proposed in 2005 by Manfred Lochter and Johannes Merkle under the titles "ECC Brainpool Standard Curves and Curve Generation" [Brainpool] to focus on issues that have not been previously addressed such as verifiable choice of seed or usage of prime of nonspecial form.
- NIST in collaboration with NSA presented the first standardization of curves was in FIPS 186-2 in 2000 [Nat00]. However, the used generation method already appeared in ANSI X9.62 in 1998 [Com+98]. Other standards recommend these curves (e.g., SECG [Cer] or WTLS [Wir00]).

Another notable paper considering the verifiably pseudorandom approach is "The Million dollar curve" which proposed a new source of public entropy for the seeds, combining lotteries from several different countries [Bai+15].

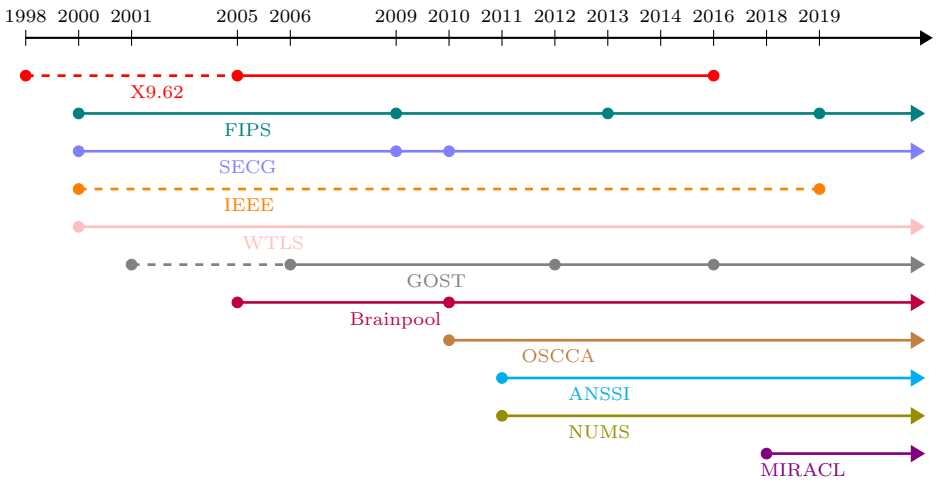

Fig. 2: Timeline of standardization of publicly available prime-field curves. Dashed line indicates that the source does not specify curve parameters. Individual curves (e.g., Curve25519) are omitted.

Other rigid methods.

As a reaction to the NIST standard curves, an interest in faster curves generated using rigid and verifiable methods has emerged:

- Three Weierstrass curves and three twisted Edwards curves were proposed by Black et al. [Bla+14] as an Internet draft called "Elliptic Curve Cryptography (ECC) Nothing Up My Sleeve (NUMS) Curves and Curve Generation" [Bla+14].
- MIRACL library [MIR18] combines approaches of NUMS and Brainpool by extracting seeds from well-known constants and iteratively incrementing one of the parameters of a curve in the Weierstrass form.
- Bernstein's Curve25519 and its sibling Ed25519 were created in 2006 [Ber06] and since then have been widely accepted by the cryptography community. The curve is in the Montgomery form and allows extremely fast x-coordinate point operations while meeting the SafeCurves security requirements. In 2013, Bernstein et al. [Ber+13] proposed the curve Curve1174 with an encoding for points as strings indistinguishable from uniform random strings (the Elligator map).
- To address higher security levels and maintain good performance, several authors developed curves of sizes in the 400-521 bit range such as the Edwards curve Curve41417 by Bernstein, Chuengsatiansup, and Lange [BCL14], Ed448-Goldilocks by Hamburg [Ham15a], E-3363 by Scott [Sco15] or M, E curves by Aranha1 et al. [Ara+13].

3 Methodology

Our ultimate goal is to enable assessment of security for standard curves. But since it is not clear how to look for unknown curve vulnerabilities, we now aim to pinpoint possible problems via identifying standard curves that deviate from other curves in specific aspects. To find deviations we employ one of the following strategies:

1. **Compare curves from a standard to corresponding simulated ones.** If the standard curves were generated using the defined processes without any hidden conditions, it would be unexpected to statistically distinguish them from our simulated ones, yet we try to do exactly that. We are looking for a standard curve achieving a value that is highly improbable when compared to the simulated curves.
2. **Compare curves from a standard to Random curves.** If the generation algorithm introduces a systematic bias, the first strategy will not allow us to find it – the problem might occur in both the standard and simulated curves. Thus this strategy tries to detect any sort of unexpected behaviour by comparing curves from a given standard to *Random*[7] curves.

To target specific curve properties, we describe and implement traits, which are functions that take a curve as an input (sometimes with additional parameters) and return numerical results. We run these traits on standard curves as well as simulated ones whenever it is computationally feasible and store the results in our database.

[7] In the remainder of this work, the word Random with capital R refers to the curves we generated by our own method to be as generic as possible.

3.1 Standard curve database

ACAT is, to the best of our knowledge, the first public database of all standard elliptic curves, divided into 18 curve categories by their source. The database includes:

1. verifiably pseudorandom curves (X9.62, NIST, SEC, Brainpool);
2. pairing-friendly curves: Barreto-Lynn-Scott curves [BLS02], Barreto-Naehrig curves [Per+11], Miyaji-Nakabayashi-Takano curves [MNT01]);
3. amicable curves: Tweedledee/Tweedledum [BGH19], Pallas/Vesta [Hop20];
4. rigidly generated NUMS curves and curves from the MIRACL library [MIR18];
5. Bernstein's high performance curves [Ber06] and M, E curves [Ara+13];
6. curves from the standards ANSSI [Jou11], OSCCA [SSL14], GOST [PLK06; SF16], OAKLEY [Orm98], WTLS [Wir00], ISO/IEC [ISO17] and others;

Although our analysis focuses mainly on prime-field curves, the database contains 31 curves over binary fields and one over an extension field. Currently, there are 188 standard curves in total. Note that we also include curves that were but are no longer supported by the standards, and curves that are not recommended for public use, but have been included in the documents for various reasons (e.g., curves for implementation checks). The database provides filtering by bit-length, field type, cofactor size, and curve form.

Additionally, our database contains five categories of simulated curves: $\text{X9.62}_{\mathsf{sim}}$, $\text{Brainpool}_{\mathsf{sim}}$, $\text{NUMS}_{\mathsf{sim}}$, $\text{Curve25519}_{\mathsf{sim}}$ and Random[8].

For each curve, we also precomputed usual properties such as the CM discriminant, the j-invariant, the trace of Frobenius t, and the embedding degree. This precomputation significantly speeds up computation of some traits.

Source	#	**Source**	#	**Source**	#	**Source**	#
X9.62	40	BARP	6	OSCCA	1	$\text{X9.62}_{\mathsf{sim}}$	120k
Brainpool	14	BLS	6	BN	16	$\text{NUMS}_{\mathsf{sim}}$	1.2k
NUMS	24	GOST	9	AMIC	4	$\text{Curve25519}_{\mathsf{sim}}$	804
SECG	33	ISO	4	DJB	10	$\text{Brainpool}_{\mathsf{sim}}$	12k
NIST	15	MNT	10	ANSSI	1	Random	250k
MIRACL	8	OAKLEY	2	WTLS	8	other	11

Table 1: Numbers of elliptic curves in our database grouped by their source.

3.2 Simulations

We have picked four major standards X9.62 [Com+98], Brainpool [Brainpool], NUMS [Bla+14] and Curve25519 [Ber06] for simulations, since their generation

[8] The set of Random curves and their trait results is currently still evolving.

method was explained in enough detail and can be easily extended for large scale generation. At a few points, the standards were a little ambiguous, so we filled the gaps to reflect the choices made for the actual standard curves whenever possible. We have simulated over 120 000 X9.62 curves, 12 000 Brainpool curves, 1 200 NUMS curves and 750 Curve25519 curves[9].

The aim of this part is not to undertake a thorough analysis of the published algorithms, rather explain our approach to the large-scale generation using the given methods. For the details of the original algorithms, see the individual standards. Although NUMS, Brainpool and Curve25519 provide a method for generating group generators, we are currently not focusing on their analysis.

X9.62 - the standard. We focus on the generation method of the 1998 version [Com+98]. Its input is a 160-bit seed and a large prime p and the output is an elliptic curve in short Weierstrass form over the field $\mathbb{F}_p$, satisfying the following security conditions:

- "Near-primality": The curve order shall have a prime factor l of size at least $\min\{2^{160}, 4\sqrt{p}\}$. Furthermore, the cofactor shall be s-smooth, where s is a small integer (the standard proposes $s = 255$ as a guide).
- The embedding degree of the curve shall be greater than 20. The standard also specifies that to check this condition, we may simply verify that $p^e \neq 1 \pmod{l}$ for all $e \leq 20$.
- The trace t shall not be equal to 1.

Given a seed and a prime p, the standard computes a $\log(p)$-bit integer r using[10] the function (1). The next step is choosing $a, b \in \mathbb{F}_p$ such that $b^2 r = a^3$. This process is repeated until the curve satisfies the security conditions mentioned above.

$$H(\text{seed}) = \underbrace{\texttt{SHA-1}(\text{seed})}_{\text{discard}} \underbrace{||\texttt{SHA-1}(\text{seed}+1)|| \ldots ||\texttt{SHA-1}(\text{seed}+i)}_{\log(p)-\text{bit integer as output}}. \tag{1}$$

X9.62 – our approach. For each of the five bit-lengths 128, 160, 192, 224 and 256 bits we have fixed the same prime as the standard (hence all curves of the same bit-length are defined over the same field). Since there is no guidance in the standard how to pick the seed for each iteration, we have taken the published seed for each bit-length and iteratively incremented its value by 1. To pick a and b, we have fixed $a = -3$ (as was done for most X9.62 curves for performance reasons [CC86]) and computed b accordingly, discarding the curve if $b^2 = -27/r$ does not have a solution for $b \in \mathbb{F}_p$. We also restricted the accepted cofactors to 1, 2, or 4 as this significantly accelerated the point counting. This choice also conforms to with the fact that the standard curves all have cofactor 1 and the SECG standard (which overlaps with the X9.62 specifications) recommends the cofactor to be bounded by 4. The point counting – the main bottleneck of the computations – was done by an early-abort version of the SEA algorithm [Sch95].

[9] This took up to a week per standard on 40-core cluster of Intel Xeon Gold 5218.
[10] More precisely, H also changes the most significant bit of the output to 0.

For each of the five bit-lengths we have tried 5 million seeds, resulting in over 120 000 elliptic curves. Figure 3 captures a simplified overview of the algorithm.

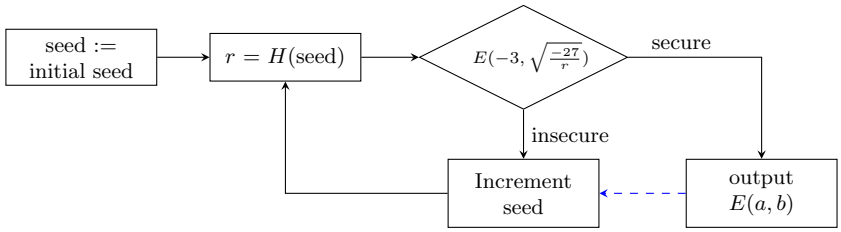

Fig. 3: X9.62 algorithm adjusted (indicated by the dashed line) for large-scale generation.

Brainpool – the standard. The Brainpool standard proposes algorithms for generating both the prime p and the curve over $\mathbb{F}_p$. Since in our simulations we have used the same finite fields as are in the recommended curve parameters we will skip the algorithm for prime generation.

Given a seed and a prime, the curve generation process outputs an elliptic curve in Weierstrass form that satisfies the following security conditions:

- The cofactor shall be 1, i.e. the group order n shall be prime.
- The embedding degree shall be greater than $(n-1)/100$.
- The trace t shall not be equal to 1. Technical requirements then state that $t > 1$.
- The class number of the endomorphism algebra of the curve should be larger than 10^7.

The algorithm itself follows similar idea as X9.62, but in more convoluted way, as can be seen in Figure 4. This time, the H function (1) is used to compute both a and b in pseudorandom way. Roughly speaking, a given seed is repeatedly incremented by 1 and mapped by H until an appropriate a is found and the resulting seed is used for finding b in a similar way. If the curve does not satisfy the condition, the seed is incremented and used again as the initial seed. See the standard for details of generation, GenA and GenB represent generation of a and b in Figure 4.

Brainpool – our approach. We have used the same approach as in X9.62 and used the published seed as initial seed for the whole generation, incrementing seed after each curve was found. Since generation of both a and b can in theory take an arbitrary number of attempts, the seed is incremented after each failed attempt. We have again used 5 million seeds for each of the four bit-lengths (160, 192, 224, 256) Brainpool recommends. The number of generated curves in total is over 12 000. The drop in the proportion of generated curves compared to X9.62 standard is caused by stricter conditions.

There is currently no known efficient method that computes the class number; instead we checked that a related quantity – the CM discriminant – is greater than 2^{100}, following the SafeCurves recommendations [BL].

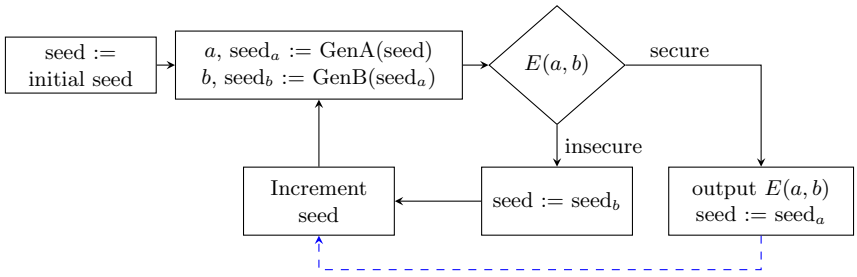

Fig. 4: Brainpool algorithm adjusted (indicated by the dashed line) for large-scale generation.

NUMS – the standard. The NUMS generation method, in accordance with its name, does not fall into the verifiably pseudorandom category. As Brainpool, NUMS proposes algorithms for generating both the prime field and the curve. The lowest recommended bit-length for prime fields by NUMS is 256. The method of prime generation works by starting with $c = 1$ and incrementing this value by 4 until $p = 2^s - c$ is a prime congruent to 3 mod 4.

Although NUMS proposes algorithms for generating both Weierstrass and Edwards curves, we have focused only on the Weierstrass curves. The curve is found by searching for $E(-3, b)$ satisfying the following security conditions by incrementing b, starting from $b = 1$:

- The curve order as well as the order of its twist shall be primes.
- The trace t shall not equal 0 or 1. This condition is further extended by requiring $t > 1$, supposedly for practical reasons.
- The embedding degree shall be greater than $(n-1)/100$ following the Brainpool standard.
- The CM discriminant shall be greater than 2^{100}, following the SafeCurves recommendations.

NUMS – our approach. We have used the same process, but this time iterating over 10 million values for b, for each of the four bit-lengths 160, 192, 224 and 256. Even though the lowest recommended bit-length for prime fields by NUMS is 256 we have generated 160, 192 and 224-bit primes using the proposed method to study curves of lower bit-lengths. This process produced over 1 200 curves, implying that the twist condition is strongly restrictive.

Curve25519 – the standard. Curve25519 was created in a similar rigid manner as NUMS curves by minimizing one coefficient of the curve equation. The prime $2^{255} - 19$ for the prime field was chosen to be close to a power of 2 and

with bit-length slightly below multiple of 32 for efficiency of the field arithmetic. Curve25519 is a Montgomery curve of the form $y^2 = x^3 + Ax^2 + x$ with the smallest $A > 2$ such that $A - 2$ is divisible by 4 and:

- The cofactor of the curve and its twist shall be 8 and 4 respectively.
- The trace t shall not equal 0 or 1.
- The embedding degree shall be greater than $(n - 1)/100$.
- The CM discriminant shall be greater than 2^{100}.

Curve25519 – our approach. Similarly to the NUMS approach we have iterated over A for bit-lengths 159, 191 and 255 where we have picked primes $2^{159} - 91$ and $2^{191} - 19$ for lower bitlengths. Iterating over 10 million values for A resulted in roughly 804 curves in total.

Random – our approach. Since all of the simulated standards contain certain unique properties (small b for NUMS, small A for Curve25519, bit overlaps in Brainpool, $rb^2 = -27$ for X9.62) and curves of the same bit-length share a base field, we have also generated curves for bit-lengths 128, 160, 192, 224, 256 using the method depicted in Figure 5 that aims to avoid such biases. We call such curves Random, as this term clearly describes our intention. However, for practical reasons, we made the process deterministic using a hash function. For each curve, we generated a prime for the base field by hashing (SHA-512) a seed and taking the smallest prime bigger than this hash when interpreted as an integer. Curve coefficients were chosen using the hash function as well, see fig. 5. We kept such a curve if it was not anomalous, had cofactor 1, 2, 4 or 8 and satisfied the SafeCurves criteria on embedding degree and CM discriminant. We tried 5 million seeds, resulting in over 250 000 Random curves.

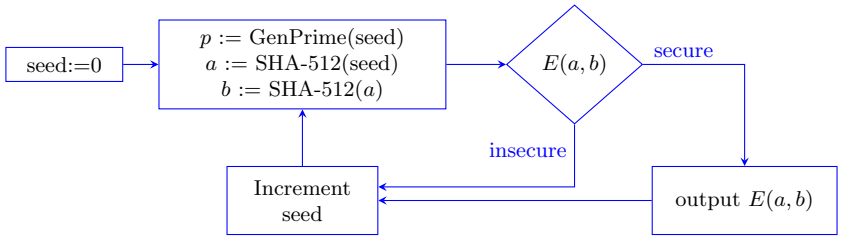

Fig. 5: Random simulation.

3.3 Outlier detection

Our framework offers options for graphical comparisons for described distinguishing strategies. However, manual inspection does not scale well, so we utilize automated approaches to identify suspicious curves. Since we do not have (and cannot have) a labeling that could be used for the typical supervised approaches, we had to resort to unsupervised methods, namely outlier detection.

We built several datasets of simulated curves according to the selected distinguishing strategies. Each dataset of simulated curves consists only of curves of the same bit-length that were generated by the same method. Table 2 shows numbers of curves contained within each dataset. The rows correspond to the used generation method and the columns to the bit-length of the selected curves.

	256 bits	224 bits	192 bits	160 bits	128 bits
$\text{X9.62}_{\text{sim}}$	18 502	22 211	18 836	27 780	36 126
$\text{Brainpool}_{\text{sim}}$	1 677	2 361	2 640	3 184	0
NUMS_{sim}	83	109	191	325	0
$\text{Curve25519}_{\text{sim}}$	160	0	366	278	0
Random	18 636	21 226	24 805	29 639	37 311

Table 2: Numbers of curves in our datasets of simulated curves.

Additionally, in cases where trait results are (mostly) independent of the curve bit-length, we analyzed curves of all bit-lengths together. This approach allowed us to analyze even curves that did not match any of the generated bit-lengths.

We augmented each dataset with standard curves according to distinguishing strategies, i.e., Random curves with each category of standard curves, and the other three simulated categories only with standard curves of the same type. From these augmented datasets, we derived feature vectors consisting of all computed trait results, as well as some of their subsets. The features were scaled using a min-max scaler to fit within the $[0; 1]$ range. In case some results were not computed, they were replaced with -1, signifying that the given value took too long to compute (e.g., the factorization of large numbers). Finally, we ran the local outlier factor algorithm [Bre+00] to identify outliers within each augmented dataset. If the approach reported a standard curve as an outlier, we inspected such results closer.

4 Traits

ACAT currently contains 22 trait functions[11] – traits for brevity – designed to test a wide range of elliptic curve properties, including mathematical characteristics of curves and properties connected to curve standards or implementations. We have divided them into five categories:

Potential attacks. These traits test the classical properties of elliptic curves relevant to known attacks (the order and cofactor of the curve; the quadratic twist; the embedding degree). However, we also cover lesser-known and threatening attacks, for example, the factorization of values of the form $kn \pm 1$ as a

[11] For full list of available traits see our website (anonymized for review).

generalization of [Che06]. To test scalar multiplication, essential to all ECC protocols, we inspect multiplicative orders of small values modulo the group order n. Another trait follows the idea from [Che02], with a possible connection of the discrete logarithm on ECC to factorization.

Complex multiplication. Although at this moment there is no serious attack utilizing any knowledge about the CM discriminant or class number, these values are the defining features of an elliptic curve. They determine the structure of the endomorphism ring, the torsion points, isogeny classes, etc. Multiple traits deal with the factorization of t, the size and the factorization of the Frobenius discriminant $D = t^2 - 4p$, as well as how D changes as we move the defining base field to its extensions. We also compute bounds on the class number using the Dirichlet class number formula [Dav80].

Torsion. We have designed several traits to directly analyze the torsion points of a given elliptic curve. One of the traits computes the degree of the extension $\mathbb{F}_{p^k}/\mathbb{F}_p$ over which the torsion subgroup is (partially) defined. Lower degrees might lead to computable pairings. Another trait approaches torsion from the direction of division polynomials and computes their factorization. We also consider the lift $E(\mathbb{F}_p) \to E(\mathbb{Q})$ and computes the size of the torsion subgroup over $\mathbb{Q}$ which might be relevant for lifting of ECDLP.

Isogenies. Both torsion and complex multiplication can be described using isogenies. Kernel polynomial of every isogeny is a factor of the division polynomial, and the special cases of isogenies, endomorphisms, form an order in an imaginary quadratic field of $\mathbb{Q}(\sqrt{D})$. One of isogeny traits computes the degree of the extension where some/all isogenies of a given degree are defined. Isogeny traits also analyze isogenies of ordinary curves and the corresponding isogeny volcanoes – their depth and the shape of so-called crater, and also the number of neighbors for a given vertex in the isogeny graph.

Standards and implementation. During our analysis of standard methods for generating curves, we have noticed unusual steps in the algorithms, so we designed traits to capture the resulting properties. The nature of the Brainpool standard (see Figure 4) causes overlaps in the binary representations of the curve coefficients in roughly half of the generated curves. The NUMS standard, as well as the Koblitz curves, use small curve coefficients by design. The X9.62 standard uses the relation $b^2r = a^3$ to determine the curve coefficients from r. The SECG standard recommends the curves `secp224k1` and `secp256k1` together with the group generators; the x-coordinate of half of both of these generators is the same small number. Most attacks on the ECDLP are aiming at specific implementation vulnerabilities. To address this, we propose traits that target possible irregularities in practical implementations. One trait follows the idea of [Wei+20], where the authors analyze the side-channel leakage caused by improper representation of large integers in memory. Based on [Bai+09], we designed a trait that analyzes the number of points with a low Hamming weight on a given curve.

The presented traits cover all SafeCurves ECDLP requirements. Unlike SafeCurves, our traits explore a much broader space of properties, providing a comprehensive tool in the hope of finding any interesting deviation, not necessarily directly connected to a known vulnerability.

4.1 Notable findings

GOST curves. The trait that analyzes the size of the coefficients in the Weierstrass form was motivated by the NUMS standard. However, our outlier detection also recognized two 256-bit GOST curves (`CryptoPro-A-ParamSet`, `CryptoPro-C-ParamSet`) in this trait, and a closer inspection in Figure 6 revealed that both curves have small b coefficients (166 and 32858). This contradicts Alekseev, Nikolaev, and Smyshlyaev [ANS18], who claim that all of the seven standardized GOST R curves were generated in the following way[12]:

1. Select p that allows fast arithmetic.
2. Compute r by hashing a random seed with the Streebog hash function.
3. For the generation of twisted Edwards curve $eu^2 + v^2 = 1 + du^2v^2$, put $e = 1, d = r$. For the generation of Weierstrass curve $y^2 = x^3 + ax + b$, put $a = -3$ and b equal to any value such that $rb^2 = a^3$.
4. Check the following security conditions:
   - $n \in (2^{254}, 2^{256}) \cup (2^{508}, 2^{512})$.
   - The embedding degree is at least 32 (resp. 132) if $n \in (2^{254}, 2^{256})$ (resp. if $n \in (2^{508}, 2^{512})$).
   - The curve is not anomalous.
   - The j-invariant is not 0 or 1728.

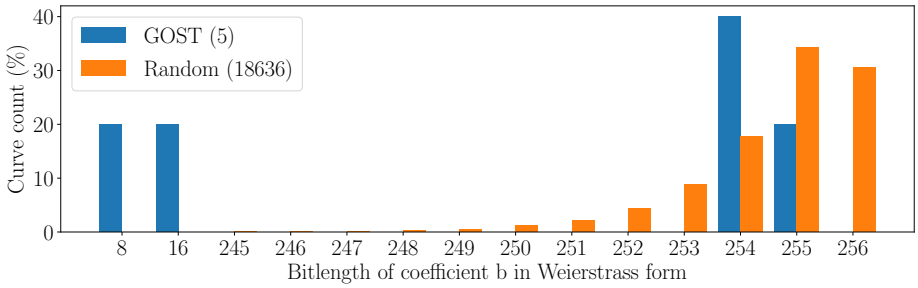

Fig. 6: Two GOST curves (`CryptoPro-A-ParamSet`, `CryptoPro-C-ParamSet`) exhibit particularly low bit-length of b parameter, even though it was claimed that they were generated pseudorandomly.

[12] The authors support their claims by providing seeds for two of the seven curves. We find it problematic that the seeds were not previously published for the general public.

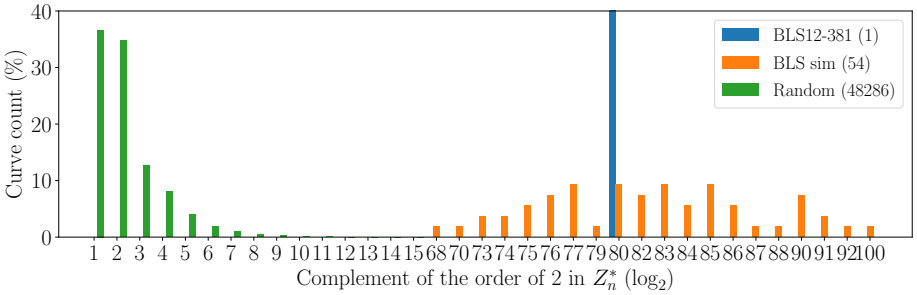

Fig. 7: BLS12-381 and BLS simulated curves exhibit unexpectedly large complement of the order of 2 in $\mathbb{Z}_n^*$ when compared to Random curves.

Thus the small size of b (which should be pseudorandom if r is) implies that it is very unlikely that they were generated with this claimed method. More precisely, since $b^2 = -27/r$ in $\mathbb{F}_p$ and p has 256 bits, then the probability for $b = 166$, resp. $b = 32858$ is 2^{-248}, resp. 2^{-240}, considering the bitlengths of the parameters. We hypothesise that the `CryptoPro-A-ParamSet` curve was generated by incrementing b from 1 until the GOST security conditions were satisfied. We have verified that $b = 166$ is the smallest such value with the added condition that the cofactor is 1 (otherwise, the smallest value is $b = 36$). The `CryptoPro-C-ParamSet` does not have this property, as its b coefficient 32858 is only the 80th smallest such value. However, there is an additional problem with the generator point $(0, \sqrt{32858})$. It was shown [Gou03], prior to the standardization of this curve, that there exist side-channel attacks utilizing such special points with $x = 0$. If we consider the existence of such point as another condition imposed on the curves, then 32858 is the 46th smallest value. We employed ACAT to distinguish this particular curve for from the rest of the 45 curves using the implemented traits in order to explain their choice, but without results.

Furthermore, the trait inspecting CM discriminant revealed that the third curve `CryptoPro-B-ParamSet` from [PLK06] has a CM discriminant of -619. Such a small value is extremely improbable, unless the curve was generated by the CM method [Brö06]. (The CM discriminant -915 of `gost256` is small as well, but this curve was used just as an example and there are no claims about its generation.)

The BLS12-381 Curve. The trait that measures $\phi(n)$ divided by the multiplicative order of 2 modulo n (low multiplicative orders translate to high values and vice versa) identified the BLS12-381 curve [Bow17] as an outlier. To further investigate this, we have adopted the generation method of the BLS12-381 curve to ACAT. Figure 7 shows the results of the trait on Random curves, the BLS12-381 curve and the BLS simulated curves.

Figure 7 clearly demonstrates that the unexpected detection of BLS12-381 as an outlier was not a coincidence. Rather it was caused by the generation method,

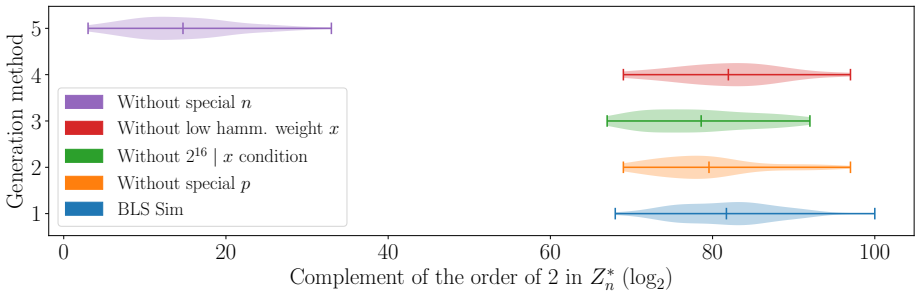

Fig. 8: Simulation of BLS curves while ignoring different properties have shown that the special construction of n caused the unexpected behavior.

and consequently the properties, of the BLS12-381 curve. To briefly summarize, the generation method works by finding an integer x satisfying:

- x has low hamming weight,
- x is divisible by 2^{16},
- $n = x^4 - x^2 + 1$ is a prime,
- $p = \frac{1}{3}n(x-1)^2 + x$ is a prime.

When such x is found, the CM method is used to construct a BLS curve[13] over $\mathbb{F}_p$ with cardinality $\frac{1}{3}n(x-1)^2$. We have identified exactly what conditions in the BLS generation method were behind the outlier detection of BLS12-381 curve by removing each condition and computing the trait for these modified curves. On one hand, Figure 8 shows that the conditions on x and special form of p have little or no impact on the results of the trait. On the other hand, we can see that the conditions that were the main cause of the unexpected results of BLS12-381 is the special form of the order n. Although this does not seem to cause any vulnerabilities in the BLS12-381 curve, it reveals an undocumented property of the generation method.

The Bitcoin curve. A trait that inspects the x-coordinate of inverted generator scalar multiples, i.e., x-coordinates of points $k^{-1}G$, where $k \in \{1, \ldots, 8\}$, was inspired by Brengel et al. [BR18], who reported an unexpectedly low value for $k = 2$ on the Bitcoin curve `secp256k1`. Furthermore, Maxwell [Max15] pointed out that `secp224k1` yields exactly the same result. Pornin [Por19] guesses that this was caused by reusing the code for Koblitz curves with the same seeds, together with poor documentation.

We analyzed the results of this trait in our visualization framework (Figure 9) and discovered that `secp256k1` and `secp224k1` are the only standard curves for which the x-coordinate of $k^{-1}G$ is significantly shorter than the full bit-length.

Brainpool overlaps. The trait designed to detect bit-overlaps in the Weierstrass coefficients, revealed a structure in the Brainpool curves, already observed by Bernstein et al. [Ber+15].

[13] Iterating through 100 million values of x we have generated 54 BLS curves.

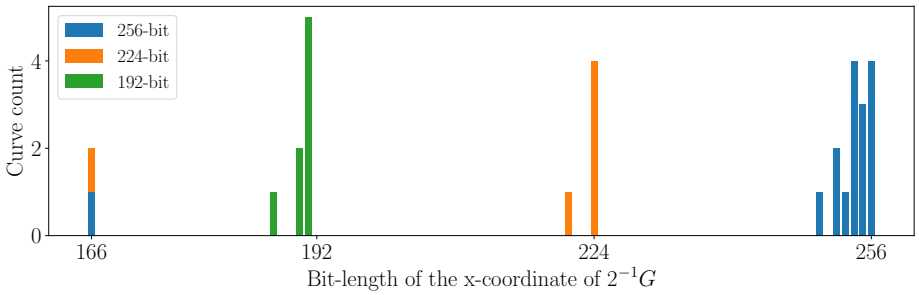

Fig. 9: Curves `secp256k1` and `secp224k1` exhibit significantly shorter x-coordinate of the point $2^{-1}G$ than expected. The other standard curves of given bit-lengths are also plotted.

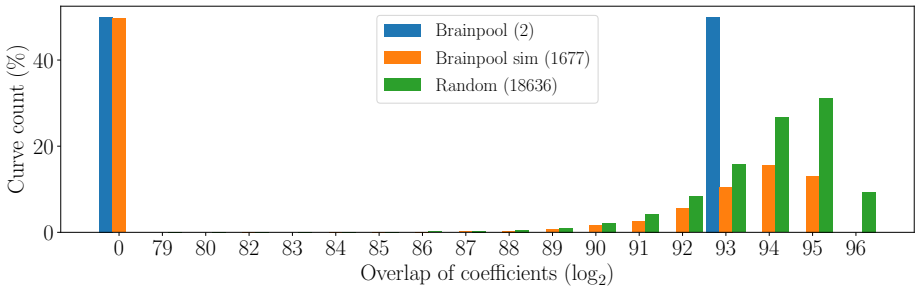

Fig. 10: Bit-overlap of 256-bit standard Brainpool, simulated Brainpool, and Random curves.

More precisely the trait compares coefficients a and b when stripped of 160 bits (SHA-1 output size) from the left and right respectively. Inspecting `brainpoolP256r1`, we can see the identical segments in a, b:

$a =$ `0x7d5a0975fc2c3057eef67530417affe7fb8055c126dc5c6ce94a4b44f330b5d9`

$b =$ `0x26dc5c6ce94a4b44f330b5d9bbd77cbf958416295cf7e1ce6bccdc18ff8c07b6`

Such overlaps occur during roughly half of the time during generation; e.g., for `brainpoolP{192,256,384}` but not for `brainpoolP{160,224,320}`. Figure 10 clearly demonstrates that the standard Brainpool curves behave as the simulated ones in this manner and both of them behave as Random curves roughly half of the time.

5 Our tool ACAT

ACAT is an open-source tool for generating elliptic curves according to our standard simulation methods, computing traits on elliptic curves, and analyzing data using automated approaches and an easy-to-use visualization environment.

Its code contains implementations of 22 traits described in the previous section, but it can be easily extended. Adding a new trait requires only a short description of its function and writing a few lines of Sage code that computes the new trait result for any given curve.

A Jupyter notebook environment can be used to analyze trait results. Alternatively, we offer an automated outlier detection. Both of these tools allow configuring the subset of analyzed curves, traits, and their parameters, and can access locally stored data as well as our publicly available database.

Our website[14] contains detailed information about curves in our database and corresponding trait results, and trait descriptions with statistics. For a more complex data inspection, the analysis environment configured with access to our database can be launched directly from the website. Additionally, the frontend provides an API for accessing the database to analyze the data without any need for its local storage.

6 Conclusions

Our framework ACAT aspires to survey all standard elliptic curves and to assist in identification of potential problems by comparing them to simulated ones and visualizing the results. We built it as a foundation of elliptic curve cryptanalysis for the cryptographic community and hope that more cryptographers and mathematicians will join the project. ACAT's code is available at our repository[15].

Our tool revealed two surprising types of deviations. We realized that the generation process of three GOST curves described by Alekseev, Nikolaev, and Smyshlyaev [ANS18] is inconsistent with the sizes of the b coefficient in two cases and with the size of the CM discriminant in the third one. Properly documenting such properties is crucial for re-establishing trust in the standard curves and the whole ECC ecosystem. We cannot expect its users, developers, or policy-makers to notice even fairly obvious deviations (e.g., small Weierstrass coefficients), as they often do not access the parameters directly.

We found an interesting, previously undescribed property of the BLS12-381 curve (related to smoothness) that is caused by its generation. Recent attacks [KB16] on special properties of pairing-friendly curves have proven that awareness of all properties caused by the pairing-friendly generation methods is crucial. In particular, our approach of isolation of individual properties in BLS generation has shown to have a lot of potential for future research of security of pairing-friendly curves.

Selected parts of ACAT could also be used to quickly assess new individual curves. This might be useful for implementations following the idea of Miele and Lenstra [ML15], trading standard curves for ephemeral on-the-fly generated ones. Besides cryptographic applications, ACAT might also be useful to number theorists by providing them with empiric distributions of various traits.

[14] Anonymized for review.
[15] Anonymized for review.

It is unrealistic to go through the whole space of trait results for different parameter choices and curve sets manually. Thus we employed an automated outlier detection method, which found all discrepancies we discovered manually. Still, there may be other outliers, and we believe it is an interesting open problem to statistically evaluate the results in a way that takes into account the inner structure of the data for a given trait.

References

[Age11] Agence Nationale de la Securite des Systemes d'Information. *Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français*. 2011. URL: https://www.legifrance.gouv.fr/download/pdf?id=QfYWtPSAJVtAB_c6Je5tAv00OY2r1-ad3LaVVmnStGvQ=.

[ANS18] E. K. Alekseev, V. Nikolaev, and S. V. Smyshlyaev. "On the security properties of Russian standardized elliptic curves". In: *Mathematical questions of cryptography* 9.3 (2018), pp. 5–32.

[ANS05] ANSI. *Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*. ANSI X9.62. 2005.

[Ara+13] D. F. Aranha1, P. S. L. M. Barreto, G. C. C. F. Pereira, and J. E. Ricardini. *A note on high-security general-purpose elliptic curves*. Cryptology ePrint Archive, Report 2013/647. https://eprint.iacr.org/2013/647. 2013.

[Bai+15] T. Baigneres, C. Delerablée, M. Finiasz, L. Goubin, T. Lepoint, and M. Rivain. "Trap Me If You Can-Million Dollar Curve." In: *IACR Cryptol. ePrint Arch.* 2015 (2015), p. 1249.

[Bai+09] D. V. Bailey, L. Batina, D. J. Bernstein, P. Birkner, J. W. Bos, H.-C. Chen, C.-M. Cheng, G. Van Damme, G. de Meulenaer, L. J. D. Perez, et al. "Breaking ECC2K-130." In: *IACR Cryptol. ePrint Arch.* 2009 (2009), p. 541.

[BLS02] P. S. Barreto, B. Lynn, and M. Scott. "Constructing elliptic curves with prescribed embedding degrees". In: *International Conference on Security in Communication Networks*. Springer. 2002, pp. 257–267.

[Ber06] D. J. Bernstein. "Curve25519: new Diffie-Hellman speed records". In: *International Workshop on Public Key Cryptography*. Springer. 2006, pp. 207–228.

[Ber+15] D. J. Bernstein, T. Chou, C. Chuengsatiansup, A. Hülsing, E. Lambooij, T. Lange, R. Niederhagen, and C. Van Vredendaal. "How to Manipulate Curve Standards: A White Paper for the Black Hat http://bada55. cr. yp. to". In: *International Conference on Research in Security Standardisation*. Springer. 2015, pp. 109–139.

[BCL14] D. J. Bernstein, C. Chuengsatiansup, and T. Lange. "Curve41417: Karatsuba revisited". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2014, pp. 316–334.

[Ber+13] D. J. Bernstein, M. Hamburg, A. Krasnova, and T. Lange. "Elligator: Elliptic-curve points indistinguishable from uniform random strings". In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 2013, pp. 967–980.

[BLN16] D. J. Bernstein, T. Lange, and R. Niederhagen. "Dual EC: A standardized back door". In: *The New Codebreakers*. Springer, 2016, pp. 256–281.

[BL] D. J. Bernstein and T. Lange. *SafeCurves: choosing safe curves for elliptic-curve cryptography*. https://safecurves.cr.yp.to/. (Visited on 08/17/2021).

[BD00] E. Biham and O. Dunkelman. "Cryptanalysis of the A5/1 GSM stream cipher". In: *International Conference on Cryptology in India*. Springer. 2000, pp. 43–51.

[Bla+14] B. Black, J. Bos, C. Costello, P. Longa, and M. Naehrig. *Elliptic Curve Cryptography (ECC) Nothing Up My Sleeve (NUMS) Curves and Curve Generation*. Internet-Drafts are working documents of the Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-black-numscurves-02. 2014.

[Bow17] S. Bowe. *BLS12-381: New zk-SNARK Elliptic Curve Construction*. https://electriccoin.co/blog/new-snark-curve/. 2017. (Visited on 09/16/2021).

[BGH19] S. Bowe, J. Grigg, and D. Hopwood. *Recursive Proof Composition without a Trusted Setup*. Cryptology ePrint Archive, Report 2019/1021. https://ia.cr/2019/1021. 2019.

[BR18] M. Brengel and C. Rossow. "Identifying Key Leakage of Bitcoin Users". In: *Research in Attacks, Intrusions, and Defenses*. Ed. by M. Bailey, T. Holz, M. Stamatogiannakis, and S. Ioannidis. Cham: Springer International Publishing, 2018, pp. 623–643.

[Bre+00] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. "LOF: identifying density-based local outliers". In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.

[Brö06] R. Bröker. "Constructing elliptic curves of prescribed order". PhD thesis. Thomas Stieltjes Institute for Mathematics, 2006.

[Cer10] Certicom Research. *SEC 2: Recommended Elliptic Curve Domain Parameters, Version 2.0*. 2010. URL: https://secg.org/.

[Cer] Certicom Research. *Standards for Efficient Cryptography Group*. https://secg.org/. (Visited on 08/13/2021).

[Che+] S. Checkoway, J. Maskiewicz, C. Garman, J. Fried, S. Cohney, M. Green, N. Heninger, R.-P. Weinmann, E. Rescorla, and H. Shacham. "A Systematic Analysis of the Juniper Dual EC Incident". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pp. 468–479.

[Che02] Q. Cheng. *A New Special-Purpose Factorization Algorithm. Citeseer*. 2002.

[Che06] J. H. Cheon. "Security Analysis of the Strong Diffie-Hellman Problem". In: *Advances in Cryptology - EUROCRYPT 2006*. Ed. by S. Vaudenay. Springer Berlin Heidelberg, 2006.

[CC86] D. V. Chudnovsky and G. V. Chudnovsky. "Sequences of numbers generated by addition in formal groups and new primality and factorization tests". In: *Advances in Applied Mathematics* 7.4 (1986), pp. 385–434.

[Com+98] A. S. Committee et al. "American National Standard X9. 62-1998". In: *Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA)* (1998).

[CLN15] C. Costello, P. Longa, and M. Naehrig. "A brief discussion on selecting new elliptic curves". In: *Microsoft Research. Microsoft* 8 (2015).

[Dav80] H. Davenport. *Multiplicative number theory*. 1980, pp. 43–53.

[DY15] L. Di and L. Yan. *Introduction to the Commercial Cryptography Scheme in China*. https://icmconference.org/wp-content/uploads/C23Introduction-on-the-Commercial-Cryptography-Scheme-in-China-20151105.pdf. 2015. (Visited on 08/23/2021).

[Brainpool] *Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation*. Tech. rep. IETF RFC 5639, 2010.

[FR94] G. Frey and H.-G. Rück. "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves". In: *Mathematics of Computation* 62.206 (1994), pp. 865–874.

[GLV01] R. P. Gallant, R. J. Lambert, and S. A. Vanstone. "Faster point multiplication on elliptic curves with efficient endomorphisms". In: *Annual International Cryptology Conference (CRYPTO 2001)*. Springer. 2001, pp. 190–200. ISBN: 978-3-540-44647-7.

[Gou03] L. Goubin. "A refined power-analysis attack on elliptic curve cryptosystems". In: *International Workshop on Public Key Cryptography*. Springer. 2003, pp. 199–211.

[Hal13] T. C. Hales. "The NSA back door to NIST". In: *Notices of the AMS* 61.2 (2013), pp. 190–192.

[Ham15a] M. Hamburg. *A note on high-security general-purpose elliptic curves*. Cryptology ePrint Archive, Report 2015/625. https://eprint.iacr.org/2015/625.pdf. 2015.

[Ham15b] M. Hamburg. *Ed448-Goldilocks, a new elliptic curve*. IACR Cryptology ePrint Archive, Report 2015/625. 2015.

[Hop20] D. Hopwood. *The pasta curves*. https://electriccoin.co/blog/the-pasta-curves-for-halo-2-and-beyond/. 2020. (Visited on 08/27/2021).

[ISO17] ISO/IEC 15946. *Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 5: Elliptic curve generation*. 2017.

[Jou11] Journal officiel de la republique francaise. *Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français*. 2011. URL: https://www.legifrance.gouv.fr/download/pdf?id=QfYWtPSAJVtAB_c6Je5tAv00OY2r1ad3LaVVmnStGvQ=.

[KB16] T. Kim and R. Barbulescu. "Extended tower number field sieve: A new complexity for the medium prime case". In: *Annual International Cryptology Conference*. Springer. 2016, pp. 543–571.

[KM16] N. Koblitz and A. Menezes. "A riddle wrapped in an enigma". In: *IEEE Security & Privacy* 14.6 (2016), pp. 34–42.

[Loc+14] M. Lochter, J. Merkle, J.-M. Schmidt, and T. Schutze. *Requirements for Standard Elliptic Curves*. IACR Cryptology ePrint Archive, Report 2014/832. 2014.

[Max15] G. Maxwell. *The most repeated R value on the blockchain*. https://bitcointalk.org/index.php?topic=1118704.0. 2015. (Visited on 09/09/2021).

[MOV93] A. J. Menezes, T. Okamoto, and S. A. Vanstone. "Reducing elliptic curve logarithms to logarithms in a finite field". In: *IEEE Transactions on information Theory* 39.5 (1993), pp. 1639–1646.

[ML15] A. Miele and A. K. Lenstra. "Efficient ephemeral elliptic curve cryptographic keys". In: *International Conference on Information Security*. Springer. 2015, pp. 524–547.

[MIR18] MIRACL UK Ltd. *Multiprecision Integer and Rational Arithmetic Cryptographic Library*. https://github.com/miracl/MIRACL. 2018.

[MNT01] A. Miyaji, M. Nakabayashi, and S. Takano. "New Explicit Conditions of Elliptic Curve Traces for FR-Reduction". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 84 (2001), pp. 1234–1243.

[Nak08] S. Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: *Decentralized Business Review* (2008), p. 21260.

[Nat00] National Institute of Standards and Technology. *FIPS Publication 186-2*. 2000. URL: https://csrc.nist.gov/publications/detail/fips/186/2/archive/2000-01-27.

[Orm98] H. Orman. *The OAKLEY Key Determination Protocol*. RFC 2412. Nov. 1998. URL: https://rfc-editor.org/rfc/rfc2412.txt.

[Per+11] G. C. Pereira, M. A. Simplício Jr, M. Naehrig, and P. S. Barreto. "A family of implementation-friendly BN elliptic curves". In: *Journal of Systems and Software* 84.8 (2011), pp. 1319–1326.

[PH78] S. Pohlig and M. Hellman. "An Improved Algorithm for Computing Logarithms over $GF(p)$ and Its Cryptographic Significance". In: *IEEE Transactions on Information Theory* (1978), pp. 106–110.

[PLK06] V. Popov, S. Leontiev, and I. Kurepkin. *Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms*. 2006. URL: https://rfc-editor.org/rfc/rfc4357.txt.

[Por19] T. Pornin. https://crypto.stackexchange.com/questions/60420/what-does-the-special-form-of-the-base-point-of-secp256k1-allow. 2019. (Visited on 09/16/2021).

[SA+98] T. Satoh, K. Araki, et al. "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves". In: *Rikkyo Daigaku sugaku zasshi* 47.1 (1998), pp. 81–92.

[Sch13] B. Schneier. *The NSA Is Breaking Most Encryption on the Internet*. https://www.schneier.com/blog/archives/2013/09/the_nsa_is_brea.html. 2013. (Visited on 08/13/2021).

[Sch95] R. Schoof. "Counting points on elliptic curves over finite fields". In: *Journal de Théorie des Nombres de Bordeaux* (1995), pp. 219–254.

[Sco99] M. Scott. *Re: NIST annouces set of Elliptic Curves*. https://web.archive.org/web/20160313065951/https://groups.google.com/forum/message/raw?msg=sci.crypt/mFMukSsORmI/FpbHDQ6hM_MJ. 1999. (Visited on 08/13/2021).

[Sco15] M. Scott. *A new curve*. https://moderncrypto.org/mail-archive/curves/2015/000449.html. 2015. (Visited on 09/01/2021).

[Sed22] V. Sedláček. "On cryptographic weaknesses related to elliptic curves". PhD thesis. Masaryk University, 2022.

[Sem96] I. Semaev. "On computing logarithms on elliptic curves". In: *Discrete Mathematics and Applications* 6.1 (1996), pp. 69–76.

[Sem98] I. Semaev. "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p". In: *Mathematics of computation* 67.221 (1998), pp. 353–356.

[SSL14] S. Shen, S. Shen, and X. Lee. *SM2 Digital Signature Algorithm*. Internet-Draft. Work in Progress. 2014. URL: https://datatracker.ietf.org/doc/html/draft-shen-sm2-ecdsa-02.

[Sma99] N. P. Smart. "The discrete logarithm problem on elliptic curves of trace one". In: *Journal of cryptology* 12.3 (1999), pp. 193–196.

[SF16] E. S. Smyshlyaeva and V. Fotieva. *Information technology. Cryptographic data security. Parameters of elliptic curves for cryptographic algorithms and protocols*. Federal Agency on Technical Regulating and Metrology. 2016.

[Sol11] J. A. Solinas. "Generalized Mersenne Prime". In: *Encyclopedia of Cryptography and Security*. Ed. by H. C. A. van Tilborg and S. Jajodia. Springer US, 2011.

[Tor94] C. E. Torkelson. "The Clipper Chip: How Key Escrow Threatens to Undermine the Fourth Amendment". In: *Seton Hall L. Rev.* 25 (1994), p. 1142.

[Val+18] L. Valenta, N. Sullivan, A. Sanso, and N. Heninger. *In search of CurveSwap: Measuring elliptic curve implementations in the wild*. Cryptology ePrint Archive, Report 2018/298. https://ia.cr/2018/298. 2018.

[Wei+20] S. Weiser, D. Schrammel, L. Bodner, and R. Spreitzer. "Big Numbers - Big Troubles: Systematically Analyzing Nonce Leakage in (EC)DSA Implementations". In: *USENIX Security Symposium*. 2020.

[Wir00] Wireless Application Protocol Forum. *Wireless Application Protocol Wireless Transport Layer Security Specification*. https://web.archive.org/web/20170829023257/https://www.wapforum.org/tech/documents/WAP-199-WTLS-20000218-a.pdf. 2000.