Security Analysis of Elliptic Curves over Sextic Extension of Small Prime Fields

Robin Salen, Vijaykumar Singh, Vladimir Soukharev

research@toposware.com

Toposware, Inc.

Abstract. In this report we investigate how to generate secure elliptic curves over sextic extension of prime fields of size roughly 64 bits to achieve 128-bit security. In particular, we present one of such curves over a 64-bit prime field, which we named Cheetah, and provide its security parameter. This curve is particularly well-suited for zero-knowledge applications such as FRI-based STARK proving systems, as its base prime field has the property of having a large two-adicity, necessary for FFT-related operations and at the same time it is used for elliptic curve-based signatures. We also provide a prototype implementation of this curve in Rust, featuring constant-time arithmetic and no use of the Rust standard library for WebAssembly support.

Keywords: elliptic curves, digital signatures, zero-knowledge proofs, stark, blockchain, field extension

1 Introduction

The hardness of the elliptic curve discrete logarithm problem (ECDLP) is the basic building block of several cryptographic constructions, including signature schemes based on elliptic curves. In the case of elliptic curves over prime fields, the NIST recommends elliptic curves over 256 bit prime fields to achieve 128 bits of security.

Elliptic curves defined over extension fields of characteristic $p > 3$ have received little interest over the years, despite their potential, due to their additional algebraic structure possibly leading to attacks [Die03,AMNS04,JV12] faster than the generic ones over prime field-based curves of equivalent size. [CL15] designed an efficient complete Edwards curve over a quadratic extension of the Mersenne prime field $\mathbb{F}_p$ with $p = 2^{127} - 1$, able to compete with existing curves over large prime fields. Unfortunately, neither the base field or the scalar field of this curve are FFT-friendly, a common requirement for zk-SNARK proving systems.

Motivated by applications to zk-SNARKs, we provide a detailed state-of-the-art cryptanalysis of elliptic curves based on sextic extensions, and then introduce Cheetah, an elliptic curve over a sextic extension of a 64-bit prime field tailored for zero-knowledge applications, along with its security analysis.

2 Our Results

Unlike previous zk-SNARK constructions based over algebraic groups (usually, a pairing-friendly elliptic curve), the zk-STARK proving system of Ben-Sasson et al. [BBHR18] only requires a prime field to work with. For the purpose of efficiency, this field can be of size much smaller than usual base finite fields of elliptic curves, around 64 bits compared to around 256 bits for the usual case (381 bits in the case of the BLS12-381 curve designed for the Sapling protocol [HBHW22]). At the same time, typical blockchain layer-2 zk-rollup solutions require some digital signature algorithm to order token transfers. [KCLM21] explored the aggregation of hash-based signatures which, if using STARK-friendly hash functions internally, could help scaling of post-quantum blockchains relying on such signature schemes. However, [KCLM21] focuses mostly on one-or-few-times hash-based signatures, and while their approach can be applicable to many-time signatures, performances and signature sizes of such schemes can slow down their adoption in the blockchain ecosystem, which favours elliptic curve digital signatures (such as EdDSA, ECDSA) constructed on elliptic curves defined over base fields of large prime order, like the Bitcoin's curve Secp256k1 or Curve25519 [Ber06], at the loss of quantum-resistance.

In this paper, we designed a new curve, with a sextic extension of a prime finite field of size 64 bits as base field, benefiting from the massive efficiency of FRI-based STARK proving systems induced by the small prime field. We adopted the so-called *Goldilocks* prime used in a number of STARK-related projects [Zer21a,Mid22,Nov22], of size of 64 bits and took it to an appropriate sextic extension, to ensure that we achieve a security level of approximately 128 bits. The motivation for using this prime also comes from the possibility to use it with the Cairo framework [GPR21], which requires a prime $p > 2^{63}$ for efficient instructions representation.

The resulting curve was generated to ensure that it is not vulnerable to common attacks as well as special attacks relevant to curves over sextic extension fields. A prototype library has been implemented in Rust at Toposware/cheetah, featuring no use of the Rust standard library, making it easily compilable for WebAssembly environments, and constant-time arithmetic for data-sensitive operations.

3 Preliminaries

In this section we first introduce the notations that will be used throughout the paper. We then review the background that is required to follow the contributions of this paper.

Notations. Let k denote the finite field $\mathbb{F}_p$, with prime p and let K/k denote the finite extension field of characteristic p and cardinality $q = p^n$.

Let $\mathcal{E}$ denote an elliptic curve in Weierstrass form over K defined by

$$\mathcal{E} : y^2 = x^3 + ax + b,$$

where $a, b \in K$.

The j-invariant of this elliptic curve is given by $j(\mathcal{E}) = 1728 \dfrac{4a^3}{4a^3 + 27b^2}$. We restrict to the non-singular cases, that is, $j(\mathcal{E}) \neq 0, 1728$.

We denote the size of elliptic curve group by $|\mathcal{E}(K)| = N$.

Background. The elliptic curve discrete logarithm problem (ECDLP) is the following computational problem:

Definition 1 (Elliptic Curve Discrete Logarithm Problem). *Given points $P, Q \in \mathcal{E}(K/k)$, find an integer r, if it exists, such that $Q = rP$.*

The hardness of this problem is an important cryptographic assumption which lies at the heart of many elliptic curve based schemes, in particular digital signature schemes. There are many known generic algorithms than can solve ECDLP; generic in the sense that they do not exploit the structure of the group type. We tabulate them below, along with asymptotic running times.

Attacks	Expected Time
Baby-step Giant-step [Sha71]	$O(\sqrt{N})$
Pollard rho/lambda algorithm [Pol78]	$O\left(\dfrac{\sqrt{\pi N}}{2}\right)$
Pohlig Hellman[PH78]	$O(\sqrt{N})$

Table 1: Generic attacks on ECDLP

These algorithms and their variants, even with parallelization, are still exponential and do not pose any threat to ECDLP.

However, considering only these attacks is not sufficient to be confident in the difficulty of solving ECDLP over a given curve. [MOV93] exhibited how one could reduce the discrete logarithm problem over an elliptic curve to the discrete logarithm problem over a finite field. If the finite field is not prohibitively large, solving the DLP there becomes much easier. To prevent such an attack, we usually require elliptic curves to have an embedding degree d which is sufficiently large (the SafeCurves criteria [BL22] recommend $d > 2^{200}$), ensuring that solving the DLP over the targeted finite field is not easier than solving the regular ECDLP.

4 Attacks on elliptic curves over sextic extensions

In this section, we will list all the known significant but specialized attacks on ECDLP that exploit the group structure.

4.1 Weil descent

Elliptic curves defined over the field extensions have a special structure. To exploit that, Frey [Fre98] originally suggested how to apply the Weil restriction of scalars for elliptic curves to solve the elliptic curve discrete logarithm problem, an approach which got generalized and optimized over the years. The idea of Weil restriction comes from the following result:

Theorem 1 (Weil Theorem). *Let K/k be a field extension of degree n and let $\mathcal{E}$ be an elliptic curve over K. Then there exists a unique abelian variety $W_K(\mathcal{E})$ of dimension n, called Weil restriction, such that the group of k-rational points over the Weil restriction $W_K(\mathcal{E})[k]$ is isomorphic (as abelian varieties) to the group $\mathcal{E}[K]$ of K-rational points over $\mathcal{E}$.*

The isomorphism between these two groups of rational points yields a direct equivalence between the discrete log problem (DLP) on $W_K(\mathcal{E})$ and on $\mathcal{E}$. Additionally, if one can find an algebraic curve $\mathcal{C}$, along with a map $\phi : \mathcal{C} \to W_K(\mathcal{E})$, then one can try lifting the DLP from $W_K(\mathcal{E})[k]$ to $Jac(\mathcal{C})[k]$, which may be faster to solve by index-calculus methods than directly tackling it on $\mathcal{E}$ with regular approaches such as Pollard-Rho.

This approach, called Weil descent, requires to find a suitable curve $\mathcal{C}$ to apply efficient index calculus methods. This was first addressed for even characteristic by Gaudry, Hess and Smart in [GHS02], and later generalized to odd characteristic by Diem [Die03]. Since the genus g of such a suitable $\mathcal{C}$ is small (it is either 2 or 3), the complexity of such an attack is $O\left(\left(g^2 \log^3 p\right) g!p + \left(g^2 \log p\right) p^2\right)$.

4.2 Cover and Decomposition attack

Variants of the Index Calculus method to attack the ECDLP have been investigated from multiple angles [Sem04,Gau09]. However, they all face similar limitations as soon as the extension degree n or the considered genus g become too large. In particular, [JV12] emphasizes that all known approaches have hidden constants in their asymptotic complexity evaluations that make them impractical as soon as $n \geq 5$. Hence, instead of aiming to solve the DLP on the jacobian of a curve $\mathcal{C}$ defined over k, one can consider a tower of extension between k and K/k, and combine covers on each step of the tower to efficiently attack the DLP over $\mathcal{E}$. In particular, when $k = \mathbb{F}_p$ and $K = \mathbb{F}_{p^6}$, one can consider two extension towers from k to K:

$$\mathbb{F}_p \hookrightarrow \mathbb{F}_{p^2} \hookrightarrow \mathbb{F}_{p^6}$$

$$\mathbb{F}_p \hookrightarrow \mathbb{F}_{p^3} \hookrightarrow \mathbb{F}_{p^6}$$

Genus 2 cover The first extension tower $\mathbb{F}_p \hookrightarrow \mathbb{F}_{p^3} \hookrightarrow \mathbb{F}_{p^6}$ leads to the following:

Let $\mathcal{E}$ be an elliptic curve over $\mathbb{F}_{q^2}$, where $q = p^3$. If $\mathcal{E}$ admits a genus 2 cover $\mathcal{C}$, then $\mathcal{C}$ must be a hyperelliptic curve defined over $\mathbb{F}_q$. Such curves are extensively studied in [DS03,AMNS04]. The hyperelliptic covers in such cases are called Scholten forms and are defined by

$$y^2 = ax^3 + bx^2 + \sigma(b)x + \sigma(a)$$

where σ is the Frobenius automorphism of the extension $\mathbb{F}_{q^2}/\mathbb{F}_q$. Another characterization of the existence of Scholten form is that either

1. 2-torsion of $\mathcal{E}$ is defined over $\mathbb{F}_{q^2}$, or
2. $|\mathcal{E}(\mathbb{F}_{q^2})|$ is odd and $j(\mathcal{E}) \notin \mathbb{F}_q$.

Genus 3 cover The second extension tower $\mathbb{F}_p \hookrightarrow \mathbb{F}_{p^2} \hookrightarrow \mathbb{F}_{p^6}$ leads to the following:

Let $\mathcal{E}$ be an elliptic curve over $\mathbb{F}_{q^3}$, where $q = p^2$. If $\mathcal{E}$ admits a genus 3 cover $\mathcal{C}$ over $\mathbb{F}_q$, then $\mathcal{C}$ could be hyperelliptic or non-hyperelliptic. We will show below which curves suffer efficient covering and can be easily exploitable by the GHS method [GHS02] in both cases:

1. If $\mathcal{E}$ admits a genus 3 hyperelliptic cover $\mathcal{C}$, then $\mathcal{E}$ must be convertible to the form:
$$y^2 = h(x)(x-\alpha)(x-\sigma(\alpha))$$
where σ is the Frobenius automorphism of $\mathbb{F}_{q^3}/\mathbb{F}_q, \alpha \in \mathbb{F}_{q^3} \backslash \mathbb{F}_q$ and $h \in \mathbb{F}_q[x]$ is of degree 1 or 2. Such elliptic curves are easy to characterize, as their cardinality is divisible by 4. Also, the existing known algorithms using covers and GHS attack have(asymptotic) complexity of $\tilde{O}\left(p^{8/3}\right)$ with very high memory requirements $O\left(p^2\right)$, see [GTTD07].
2. If $\mathcal{E}$ admits a genus 3 non-hyperelliptic cover $\mathcal{C}$, then $\mathcal{E}$ admits the form:
$$y^2 = c(x-\alpha)(x-\sigma(\alpha))(x-\beta)(x-\sigma(\beta)),$$
where $c \in \mathbb{F}_{q^3}$ and either $\alpha, \beta \in \mathbb{F}_{q^3} \backslash \mathbb{F}_q$ or $\alpha \in \mathbb{F}_{q^6} \backslash \left(\mathbb{F}_{q^2} \cup \mathbb{F}_{q^3}\right)$ and $\beta = \sigma^3(\alpha)$.
In this case, the approximate running time calculation of the attack is [LL15] $\approx 1.23123 \cdot \log_2^2(q) \cdot q$

5 Cheetah: A STARK-friendly elliptic curve over a sextic extension base field

We now explicitly construct a STARK-friendly elliptic curve defined over a sextic extension of a 64-bit field.

5.1 Choice of base field and extension

Let p be the prime $2^{64} - 2^{32} + 1$. In particular, $\mathbb{F}_p$ enjoys a large multiplicative subgroup of order 2^ℓ, with $\ell = 32$ the two-adicity of $p-1$, and its particular shape ($p =$ 0xFFFFFFFF00000001) allows for efficient modular reduction when performing arithmetic operations in $\mathbb{F}_p$. This prime, known as a Goldilocks prime, was first used by the team of Polygon Zero in [Zer21a]. For a description of the benefits of using such prime for field arithmetic, we refer to [Zer21b].

Let the sextic field extension $\mathbb{F}_{p^6}$ of $\mathbb{F}_p$ be defined by the (irreducible over $\mathbb{F}_p$) polynomial $u^6 - 7$ as $\mathbb{F}_{p^6} = \mathbb{F}_p[u]$, where 7 is both a quadratic and cubic non-residue in $\mathbb{F}_p$.[1].

We then define Cheetah, an elliptic curve over $\mathbb{F}_{p^6}$:

$$\mathcal{E} : y^2 = x^3 + x + b,$$

where $b = u + 395$, with a 130-bit long cofactor h:

$$h = 2 \cdot 5 \cdot 29 \cdot 181 \cdot 155833 \cdot 86621679593707472449686472361$$

and a subgroup G of 255-bit prime order

$$\#G = \mathrm{0x7af2599b3b3f22d0563fbf0f990a37b5327aa72330157722d443623eaed4accf}$$

generated by the basepoint (in affine coordinates) $g = (g_x : g_y)$ with

$$\begin{aligned} g_x = {} & 12938930721685970739u^5 + 375185138577093320u^4 \\ & + 4830863958577994148u^3 + 10526511002404673680u^2 \\ & + 8599518745794843693u + 2754611494552410273 \end{aligned}$$

$$\begin{aligned} g_y = {} & 9990732138772505951u^5 + 13187678623570541764u^4 \\ & + 10708493419890101954u^3 + 14375303400746062753u^2 \\ & + 2774812795997841935u + 15384029202802550068 \end{aligned}$$

The above prime-order subgroup generator has been obtained by first applying the Simplified Shallue-van de Woestijne-Ulas hashing-to-curve algorithm to the integer representation of the ASCII encoding of the "Cheetah" string. The obtained point, not of prime order, has then been multiplied by the curve's cofactor h to send it to the prime-order subgroup G.

5.2 Security evaluation

We aimed for a curve with a close to 256-bit prime order subgroup, instead of looking for a prime order curve, because of efficiency considerations of scalar multiplications and cryptographic protocols based on scalar multiplications. This yields a security level for generic elliptic curve attacks of 127.3 bits for the Cheetah curve.

The choice of field extension, $\mathbb{F}_{p^6}$, is a very conservative one, since we can achieve a security level of at least $\frac{5}{3} \cdot size(p)$, by considering decomposition on $Jac(\mathcal{H})[\mathbb{F}_{p^2}]$ [JV12], the most efficient sextic-extension targeted attack while ignoring all hidden constants and logarithmic factors. In our case, p is 64-bit,

[1] The polynomial defining the extension $\mathbb{F}_{p^6}/\mathbb{F}_p$ has been chosen to be sparse with small constant coefficient to lower the cost of operations when multiplying by u during $\mathbb{F}_{p^6}$ multiplication.

which means we would have a security level at least equal to 106.67 bits. However, we would need to ideally achieve 127.3 bits of security, the estimated cost of running Pollard-Rho on the subgroup G.

To reach this, we will show that Cheetah is resistant to all known sextic-extension targeted attacks. That is, despite these attacks, we are able to get at least 127 bits of security.

1. **Weil descent attack**
   The genus of the curve $\mathcal{C}$ is either 2 or 3 and p is a 64-bit prime. A Weil descent attack, running in time $O\left(\left(g^2 \log^3 p\right) g!p + \left(g^2 \log p\right) p^2\right)$, requires at least 2^{136} operations, ignoring hidden factors. This yields a security of at least 136 bits against Weil descents for Cheetah.
2. **Cover attacks**
   (a) **Genus 2 Covers** The cardinality of $\mathcal{E}(K)$ is not odd ($\#\mathcal{E}(K) = 2 \mod 4$). In addition, there is only one non-trivial two-torsion point defined over K, namely $P_2 = (X : 0 : 1)$ in projective coordinates with

$$\begin{aligned} X = 13601246634833184273u^5 + 3684827223278792538u^4 \\ + 6901070379872838024u^3 + 13396543320389503071u^2 \\ + 10762729315666779701u + 16464216994076148022 \end{aligned}$$

   hence the whole two-torsion subgroup of $\mathcal{E}$ cannot be fully defined over a strict subgroup of K when seen as a towered extension $\mathbb{F}_p \hookrightarrow \mathbb{F}_{p^3} \hookrightarrow \mathbb{F}_{p^6}$. Hence, the genus 2 cover attacks do not apply.

   (b) **Genus 3 Covers**
      i. **Hyperelliptic curves** As stated above, the cardinality of $\mathcal{E}(K)$ is not divisible by 4 ($\#\mathcal{E}(K) = 2 \mod 4$). Therefore, there is no hyperelliptic curve of genus 3 that cover $\mathcal{E}$.
      ii. **Non-hyperelliptic curves** If $\mathcal{E}(K)$ admits a non-hyperelliptic cover of genus 3 with $q = p^2$, then the running complexity $1.23123 \cdot \log_2^2(q) \cdot q$, which gives at least 135 bits of security in that case.

5.3 Implementation and Performances

We implemented a prototype of the Cheetah curve in Rust, publicly available at [Top22a]. The implementation does not rely on the Rust standard library, and is aimed at providing constant time field and group arithmetic operations, and will be extended in the future to support additional optimizations regarding field operations and scalar multiplication computations.

The Sagemath search algorithm that yielded the Cheetah curve is available at [Top22b]. Running it in sequential mode ensures a deterministic output[2]. The Cheetah curve is the first curve with a sufficiently high expected level of security to be outputted by the deterministic algorithm with a Goldilocks prime

[2] For search purposes, multi-threading is supported to enable faster analysis of potential candidates.

$p = 2^{64} - 2^{32} + 1$ as base prime field. [Top22b] also provides a verification script in Sagemath for the described Cheetah curve.

We emphasize that, while the search algorithm enforces that twist security holds for the outputted curves (the largest prime factor of Cheetah's twist has 287 bits) this is not strictly necessary as known single-coordinate ladders for scalar multiplication of short Weierstrass curves [BJ02] are less efficient than standard scalar multiplication methods. In addition, verifying that a given point belongs to the original curve is a relatively inexpensive operation.

We present below the running times of our Cheetah prototype implementation [Top22a]. Benchmarks are run with an Intel-Core i7-9750H CPU at 2.60GHz, on an Ubuntu 18.04 LTS 64-bits operating system, with the command:

```
# RUSTFLAGS="-C target-cpu=native" cargo bench
```

We denote by $\mathbb{F}_p$ the base prime field of the curve, and $\mathbb{F}_q$ its scalar field. Group operations are always performed in projective coordinates, except for mixed-addition which involves a point in projective coordinates and a point in affine coordinates.

Operation	**Running time**
addition	856.15 ps
subtraction	885.08 ps
doubling	743.42 ps
multiplication	1.3480 ns
squaring	1.3197 ns
exponentiation	810.68 ns
inversion	215.79 ns
square root	2.0639 us

Table 2: $\mathbb{F}_p$ arithmetic evaluation

Operation	**Running time**
addition	5.6031 ns
subtraction	5.6518 ns
doubling	5.0895 ns
multiplication	66.230 ns
squaring	33.805 ns
exponentiation	45.872 ns
inversion	798.30 ns
square root	62.778 us

Table 3: $\mathbb{F}_{p^6}$ arithmetic evaluation

Operation	**Running time**
addition	4.7137 ns
subtraction	4.4864 ns
doubling	4.5912 ns
multiplication	31.156 ns
squaring	22.201 ns
exponentiation	16.824 us
inversion	8.9155 us
square root	11.476 us

Table 4: $\mathbb{F}_q$ arithmetic evaluation

Operation	**Running time**
addition	864.22 ns
mixed-addition	796.62 ns
subtraction	858.47 ns
doubling	753.48 ns
scalar mult.	283.89 us
scalar mult. (basepoint)	74.094 us

Table 5: $\mathbb{G}$ arithmetic evaluation

We emphasize that the running times above are preliminary results, which would benefit greatly from additional optimizations not implemented yet in the Cheetah Rust library.

6 Conclusion

In this paper, we present a systematic method to search for elliptic curves designed over sextic extensions, targeting 128 bits of security, and introduce Cheetah, a STARK-friendly elliptic curve over a sextic extension of a 64 bits prime field. We discuss its security parameters and implementation, and provide the script that assisted its finding.

Acknowledgements. We thank Travis Alan Baumbaugh, Alonso González and Hamy Ratoanina, our colleagues at Toposware, for helpful discussions and careful review and suggestions for improving the quality of our work.

References

AMNS04. Seiko Arita, Kazuto Matsuo, Koh-ichi Nagao, and Mahoro Shimura. A weil descent attack against elliptic curve cryptosystems over quartic extension fields. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E89A, 10 2004.

BBHR18. Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. *IACR Cryptol. ePrint Arch.*, page 46, 2018.

Ber06. Daniel J. Bernstein. Curve25519: New diffie-hellman speed records. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography - PKC 2006*, pages 207–228, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

BJ02. Éric Brier and Marc Joye. Weierstraß elliptic curves and side-channel attacks. In David Naccache and Pascal Paillier, editors, *Public Key Cryptography*, pages 335–345, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

BL22. D.J Bernstein and T Lange. Safecurves: choosing safe curves for elliptic-curve cryptography. `https://safecurves.cr.yp.to`, 2022.

CL15. Craig Costello and Patrick Longa. Fourq: four-dimensional decompositions on a q-curve over the mersenne prime. In *Advances in Cryptology - ASIACRYPT, 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015.* Springer, December 2015.

Die03. Claus Diem. The ghs-attack in odd characteristic. *Journal of the Ramanujan Mathematical Society*, 18, 01 2003.

DS03. Claus Diem and Jasper Scholten. A report for the arehcc project, 2003.

Fre98. Gerhard Frey. How to disguise an elliptic curve, 1998.

Gau09. Pierrick Gaudry. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. *Journal of Symbolic Computation*, 44:1690–1702, 12 2009.

GHS02. Pierrick Gaudry, Florian Hess, and Nigel Smart. Constructive and destructive facets of weil descent on elliptic curves. *Journal of Cryptology*, 15, 03 2002.

GPR21. Lior Goldberg, Shahar Papini, and Michael Riabzev. Cairo - a turing-complete stark-friendly cpu architecture. Cryptology ePrint Archive, Report 2021/1063, 2021. https://ia.cr/2021/1063.

GTTD07. Pierrick Gaudry, Emmanuel Thomé, Nicolas Thériault, and Claus Diem. A double large prime variation for small genus hyperelliptic index calculus. *Mathematics of Computation*, 76:475–492, 11 2007.

HBHW22. Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. Zcash protocol specification. `https://zips.z.cash/protocol/protocol.pdf`, 2022.

JV12. Antoine Joux and Vanessa Vitse. Cover and decomposition index calculus on elliptic curves made practical - application to a previously unreachable curve over $\mathbb{F}_{p^6}$. In *EUROCRYPT*, 2012.

KCLM21. Irakliy Khaburzaniya, Konstantinos Chalkias, Kevin Lewi, and Harjasleen Malvai. Aggregating hash-based signatures using starks. Cryptology ePrint Archive, Report 2021/1048, 2021. https://ia.cr/2021/1048.

LL15. Kim Laine and Kristin Lauter. Time-memory trade-offs for index calculus in genus 3. *Journal of Mathematical Cryptology*, 9(2):95–114, 2015.

Mcg06. John Mcgee. René schoof's algorithm for determining the order of the group of points on an elliptic curve over a finite field. Technical report, 2006.

Mid22. Polygon Miden. miden. `https://github.com/maticnetwork/miden`, 2022.

MOV93. A.J. Menezes, T. Okamoto, and S.A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.

Nov22. Novi. winterfell. `https://github.com/novifinancial/winterfell`, 2022.

PH78. S. Pohlig and M. Hellman. An improved algorithm for computing logarithms over<tex>gf(p)</tex>and its cryptographic significance (corresp.). *IEEE Transactions on Information Theory*, 24(1):106–110, 1978.

Pol78. John M Pollard. Monte carlo methods for index computation $\mod p$. *Mathematics of computation*, 32(143):918–924, 1978.

Sem04. Igor Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. 03 2004.

Sha71. Daniel Shanks. Class number, a theory of factorization, and genera. In *Proc. of Symp. Math. Soc., 1971*, volume 20, pages 41–440, 1971.

Top22a. Toposware. Cheetah. `https://github.com/ToposWare/cheetah`, 2022.

Top22b. Toposware. Cheetah evidence. `https://github.com/ToposWare/cheetah_evidence`, 2022.

Vol88. José Felipe Voloch. A note on elliptic curves over finite fields. *Bulletin de la Société Mathématique de France*, 116(4):455–458, 1988.

Wat69. William C. Waterhouse. Abelian varieties over finite fields. *Annales scientifiques de l'École Normale Supérieure*, Ser. 4, 2(4):521–560, 1969.

Zer21a. Polygon Zero. plonky2. `https://github.com/mir-protocol/plonky2`, 2021.

Zer21b. Polygon Zero. Plonky2: Fast recursive arguments with plonk and fri. `https://github.com/mir-protocol/plonky2/blob/main/plonky2.pdf`, 2021.

A Number of points of elliptic curves over $\mathbb{F}_{p^6}$ with coefficients in $\mathbb{F}_p$, $\mathbb{F}_{p^2}$ and $\mathbb{F}_{p^3}$

In the short Weierstrass equation, we can take a to be a square. This reduces the search space by half, as half the elements of $\mathbb{F}_{p^6}$ are quadratic residues. Following this by a birational map, we can restrict our attention to elliptic curves with $a = 1$ and iterate b over $\mathbb{F}_{p^6}$. Furthermore, we justify here our choice of iterating over sparse values of the type $\beta + i$ with β quadratic and cubic non-residue in $\mathbb{F}_{p^6}$ and $i \in \mathbb{F}_p$, for the b coefficient of the Weierstrass equation of our Cheetah curve, rather than sampling b from the subfield $\mathbb{F}_p$ directly.

Let p be a prime and $\mathcal{E}$ be any curve over $\mathbb{F}_{|}$. In particular, $\#\mathcal{E}(\mathbb{F}_p)= p+1-t$ where $|t| \leqslant 2\sqrt{p}$ is called the curve's trace.

Since $(t, p) = 1$ and $p > 4$, for every $t \in [0, 2\sqrt{p}]$ there exists $\mathcal{E}(\mathbb{F}_p)$ such that $\#\mathcal{E}(p) = p + 1 - t$ [Wat69,Vol88].

To compute the number of points of $\mathcal{E}$ on a sextic extension of $\mathbb{F}_p$, we recall the result of John McGee [Mcg06] below: $\#\mathcal{E}\left(\mathbb{F}_{p^n}\right) = p^n + 1 - s_n$ where s_n satisfies the recurrence relation

$$s_0 = 2, \quad s_1 = t, \quad s_{n+1} = ts_n - ps_{n-1}$$

Therefore following the relation, we obtain:

$$\begin{aligned}
s_2 &= ts_1 - ps_0 = t^2 - 2p \\
s_3 &= ts_2 - ps_1 = t\left(t^2 - 2p\right) - pt = t^3 - 3pt \\
s_4 &= ts_3 - ps_2 = t^4 - 3pt^2 - t^2p + 2p^2 = t^4 - 4pt^2 + 2p^2 \\
s_5 &= ts_4 - ps_3 = t^5 - 4pt^3 + 2p^2t - pt^3 + 3p^2t = t^5 - 5pt^3 + 5p^2t \\
s_6 &= t^6 - 5pt^4 + 5p^2t^2 - pt^4 + 4p^2t^2 - 2p^3 = t^6 - 6pt^4 + 9p^2t^2 - 2p^3
\end{aligned}$$

Recall, $\#\mathcal{E}\left(\mathbb{F}_q\right)$ is a subgroup of $\#\mathcal{E}\left(\mathbb{F}_{q^n}\right)$ which gives $\#\mathcal{E}\left(\mathbb{F}_{p^k}\right)$ divides $\#\mathcal{E}\left(\mathbb{F}_{p^n}\right)$ whenever k divides n. This relation can exploited to factorize the order $\mathcal{E}\left(\mathbb{F}_{p^6}\right)$ as follows:

1. $k = 1,\ n = 2$ gives $\#\mathcal{E}\left(\mathbb{F}_{p^2}\right) = (p + 1 - t)(p + 1 + t)$.
2. $k = 3,\ n = 6$ gives $\#\mathcal{E}\left(\mathbb{F}_{p^6}\right) = \left(p^3 + 1 - t^3 + 3pt\right)\left(p^3 + 1 + t^3 - 3pt\right)$

Combining 1 and 2 and $t \to -t$, yields the further factorisation

$$\#\mathcal{E}\left(\mathbb{F}_{p^6}\right) = (p{+}1{-}t)(p{+}1{+}t)\left(p^2 - p + 1 + tp + t + t^2\right)\left(p^2 - p + 1 - tp - t + t^2\right)$$

To understand the quadratic factors, consider

$$f(t) = t^2 + t + pt + p^2 - p + 1 = \left(t + \frac{p+1}{2}\right)^2 + \frac{1}{2}\left(p^2 - 4p + 1\right)$$

Since $p^2 - 4p + 1 > 0$, this polynomial is irreducible, which has no value of t, as a solution.

So, the largest prime that divides $\#\mathcal{E}\left(\mathbb{F}_{p^6}\right)$ can potentially come from this quadratic factor. However, $f(t)$ is an increasing function on $[0, 2\sqrt{p}]$, We can then conclude that the largest prime factor of the order of a curve $\mathcal{E}(\mathbb{F}_{p^k})$ with Weierstrass coefficients defined in $\mathbb{F}_p$ cannot be greater than $f(2\sqrt{p})$. For a 64-bit prime p, this yields a prime factor of at most 129 bits, which is too small for ECDLP to be hard to solve with generic attacks. Similarly, if we sample $b \in \mathbb{F}_{p^3}$, then $\#\mathcal{E}\left(\mathbb{F}_{p^6}\right) = (p^3 + 1 - t)(p^3 + 1 + t)$, where $t \in [0, 2\sqrt{p^3}]$, in which case the largest prime factor of the order cannot exceed 192 bits. The last case is when $b \in \mathbb{F}_{p^2}$. In that case, $\#\mathcal{E}\left(\mathbb{F}_{p^6}\right) = \left(p^6 + 1 - t^3 + 3p^2 t\right) = (p^2 + 1 - t)\left(p^4 - p^2 + 1 + tp^2 + t + t^2\right)$, where $t \in [0, 2p]$. The largest prime factor of the order cannot exceed 256 bits which can come from the quadratic factor. So for higher security the sampling must be done from $b \in \mathbb{F}_{p^6} \setminus \mathbb{F}_{p^2}$.

To find curves with sufficiently large prime-order subgroups while reducing the overhead of multiplication by $3B$ during group additions, we extended the space search to cover coefficients over $\left\{\beta + i \mid i \in \mathbb{F}_p, \beta \in \mathbb{F}_{p^6} \setminus S\right\}$, where S is generated by the roots of $X^6 - 7$, rather than over the whole $\mathbb{F}_{p^6}$ space.