Indian Institute of Technology Kanpur

CS496: Undergraduate Project Report

Discrete Logarithms, Elliptic Curves, and Index Calculus

Author: Shaurya Johari

Supervisor: *Dr. Nitin Saxena*

Department of Computer Science and Engineering

November 15, 2025

Abstract

The Elliptic-Curve Discrete Logarithm Problem (ECDLP) asks: given points P and Q find $k \in \mathbb{F}_q$ such that $kP = Q$. Today's public-key security on curves rests on the fact that no subexponential algorithm is known for well-chosen prime-field curves. In this talk, we will look at the best-known generic attack families (e.g., Baby-Step/Giant-Step and Pollard's rho variants) and contrast them with index-calculus methods that are highly effective for the classical finite-field DLP yet have not translated to prime-field elliptic curves. We will also look at an attempt for an algorithm capable of solving ECDLP in better time complexity.

Contents

Chapter 1

Introduction to the Discrete Logarithm Problem (DLP)

This chapter introduces the discrete logarithm problem (DLP), motivates its study, and sets the stage for the rest of the report. The presentation here expands the short introduction in the slides into a more detailed exposition so that subsequent chapters can assume this basic vocabulary and intuition.

1.1 What is the Discrete Logarithm Problem?

Let $(\mathbb{G}, \cdot)$ be a finite cyclic group of order n with generator g. For an element $h \in \mathbb{G}$, the *discrete logarithm problem* (DLP) asks for an integer $x \in 0, \ldots, n-1$ satisfying

$$g^x = h. \tag{1.1.1}$$

We then write $x = \log_g h$. The name stems from the analogy with the real logarithm: exponentiation is easy to compute in most algebraic groups, but inverting the operation (finding the exponent) appears computationally hard in many groups.

The DLP underpins many public-key cryptosystems such as Diffie–Hellman key exchange, ElGamal encryption, and the Digital Signature Algorithm (DSA). The assumed hardness of DLP in suitable groups provides the security guarantee for these protocols: an adversary who can solve DLP efficiently would break these systems.

1.2 Practical Hardness and Asymmetry

The operational asymmetry that makes DLP valuable for cryptography is this:

- Computing g^x given g and x is efficient (polynomial in the input size).

- Recovering x given g and g^x appears to require exponential work in the best-known generic algorithms for appropriately chosen groups.

1.3 Algorithms covered in this report

- **Brute force:** Try all x. As expected, takes $O(n)$ group operations.
- **Baby-Step Giant-Step (BSGS):** Requiring $O(\sqrt{n})$ time and $O(\sqrt{n})$ space.
- **Pollard's Rho:** Expected time $O(\sqrt{n})$ and negligible memory (parallelizable).
- **Index calculus and variants:** Subexponential algorithms in certain groups (notably $\mathbb{F}_p^\times$ and some extension fields) that exploit arithmetic structure.

The remainder of this report will expand these items where relevant and explain why index-calculus–type subexponential algorithms succeed in multiplicative finite fields but (in general) fail for prime-field elliptic curves.

1.4 Formal definition and notation

Let $\mathbb{G}$ be a finite cyclic group of order n with generator g. For $h \in \mathbb{G}$, the discrete logarithm problem asks for x with $g^x = h$. When we work in $\mathbb{F}_p^\times$ we will often identify exponents modulo $p-1$ and work in the additive group $\mathbb{Z}/(p-1)\mathbb{Z}$.

1.5 Known Suboptimal algorithms

1.5.1 Baby-Step Giant-Step

BSGS is a deterministic meet-in-the-middle algorithm. Write $n = \lceil\sqrt{n}\rceil$ and compute and store baby steps $g^j : 0 \le j < n$. Then compute giant steps hg^{-in} for $i = 0, 1, \ldots$ and search for a match. Once a collision $g^j = hg^{-in}$ is found we have $x = in + j$. The algorithm runs in $O(\sqrt{n})$ time and space.

1.5.2 Pollard's Rho

Pollard's rho constructs a pseudorandom sequence in the group using a partitioning function; it maintains triples (X, a, b) such that $X = g^a h^b$. Collisions $X_i = X_j$ give linear relations in a, b that reveal the discrete log. Expected time is $O(\sqrt{n})$ and memory is tiny (Floyd or Brent cycle detection suffices). It is easily parallelizable via distinguished points.

Chapter 2

Elliptic Curves and the Structure of Finite Groups

2.1 Elliptic curves: equations and group law

Over a field K with $\mathrm{char}(K) \neq 2, 3$, an elliptic curve E/K has the short Weierstrass form

$$E : y^2 = x^3 + ax + b, \quad a, b \in K,\ 4a^3 + 27b^2 \neq 0.$$

The set $E(K)$ of K-rational points together with the point at infinity O_∞ forms an abelian group. The group law admits explicit algebraic formulas for addition and doubling which are used in cryptographic implementations.

2.2 Group structure

For an elliptic curve $E(\mathbb{F}_p)$, the finite abelian group $E(\mathbb{F}_p)$ is isomorphic to

$$E(\mathbb{F}_p) \cong (Z)/n_1(Z) \times (Z)/n_2(Z), \quad n_1 \mid n_2,\ n_1 \mid (p-1).$$

The Weil pairing forces the divisibility condition $n_1 \mid (p-1)$. In many cryptographic designs $E(\mathbb{F}_p)$ is made cyclic (take $n_1 = 1$) or one works with a large cyclic subgroup.

2.3 Hasse's theorem and Frobenius

The Frobenius endomorphism $\pi : (x, y) \mapsto (x^p, y^p)$ acts on E and its characteristic polynomial on the ℓ-adic Tate module $T_\ell(E)$ is

$$X^2 - tX + p, \quad t \in \mathbb{Z}.$$

Then $\#E(\mathbb{F}_p) = p + 1 - t$ and Hasse's bound asserts $|t| \leq 2\sqrt{p}$. The eigenvalues α, β of π satisfy $\alpha + \beta = t$, $\alpha\beta = p$, and $|\alpha| = |\beta| = \sqrt{p}$.

2.4 Torsion and Tate modules

For m prime to p, the m-torsion subgroup satisfies

$$E[m] \cong (\mathbb{Z}/m\mathbb{Z})^2$$

over an algebraic closure. The inverse limit over ℓ-power torsion for a fixed prime $\ell \neq p$ yields the ℓ-adic Tate module $T_\ell(E)$ which is a free $(Z)_\ell$-module of rank 2 and on which Frobenius acts linearly. The action of Frobenius encodes point counts and arithmetic information used in point-counting algorithms.

2.5 Arithmetic Over Elliptic Curves

Chapter 3

Counting Points: Schoof's Algorithm

Schoof's algorithm is used to count the number of points on an elliptic curve over a finite field.

3.1 Schoof's idea

Schoof's algorithm computes the Frobenius trace t modulo small primes $\ell \neq p$ by examining the action of Frobenius on the ℓ-torsion $E[\ell]$ and then reconstructs t via the Chinese Remainder Theorem. Choosing primes ℓ such that $N = \prod \ell > 4\sqrt{p}$ suffices to recover t exactly.

Fix an odd prime $\ell \neq 2, p$. The problem reduces to determining

$$t_\ell \equiv t \pmod{\ell}.$$

If (x, y) lies in the ℓ-torsion subgroup

$$E[\ell] = \{P \in E(\overline{\mathbb{F}}_q) \mid \ell P = O\},$$

then the Frobenius map φ satisfies

$$qP = \bar{q}P,$$

where $\bar{q}$ is the unique integer with

$$q \equiv \bar{q} \pmod{\ell}, \quad |\bar{q}| < \ell/2.$$

Note that $\varphi(O) = O$ and, for any integer r, we have

$$r\varphi(P) = \varphi(rP).$$

Thus $\varphi(P)$ has the same order as P.

For $(x, y) \in E[\ell]$, we also have

$$t(x^q, y^q) = \bar{t}(x^q, y^q) \quad \text{if } t \equiv \bar{t} \pmod{\ell}.$$

Hence, the problem reduces to solving the equation

$$(x^{q^2}, y^{q^2}) + \bar{q}(x, y) \equiv \bar{t}(x^q, y^q),$$

where $\bar{t}, \bar{q}$ are integers in the range

$$[-(\ell - 1)/2,\ (\ell - 1)/2].$$

Matrix formulation. Choose a basis $\{P_1, P_2\}$ of the ℓ-torsion $E[\ell]$, so that $E[\ell] \simeq (\mathbb{Z}/\ell\mathbb{Z})^2$ and every $P \in E[\ell]$ has unique coordinates $v_P \in (\mathbb{Z}/\ell\mathbb{Z})^2$ with $P = v_P^1 P_1 + v_P^2 P_2$. With respect to this basis the Frobenius endomorphism φ acts linearly on $E[\ell]$ and is represented by a 2×2 matrix

$$M \ = \ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{Mat}_2(\mathbb{Z}/\ell\mathbb{Z}),$$

so that for every $P \in E[\ell]$ with coordinate column vector v_P we have

$$\varphi(P) \longleftrightarrow M\, v_P,$$

i.e. the coordinates of $\varphi(P)$ are given by Mv_P modulo ℓ.

The characteristic polynomial of M equals the characteristic polynomial of φ on $E[\ell]$, hence

$$\chi_M(X) = X^2 - (\operatorname{tr} M)X + \det M = X^2 - t_\ell X + q \in (\mathbb{Z}/\ell\mathbb{Z})[X],$$

so in particular

$$\operatorname{tr} M \equiv t \pmod{\ell}, \quad \det M \equiv q \pmod{\ell}.$$

Equivalently the matrix M satisfies the matrix equation (Cayley–Hamilton)

$$M^2 - t_\ell M + qI_2 \equiv 0 \pmod{\ell},$$

where I_2 is the 2×2 identity matrix.

Writing this out gives the explicit matrix relation

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^2 - t_\ell \begin{pmatrix} a & b \\ c & d \end{pmatrix} + q \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \equiv \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \pmod{\ell}.$$

Expanding the entries yields four scalar congruences in $\mathbb{Z}/\ell\mathbb{Z}$ from which $t_\ell = a + d \pmod{\ell}$ can be read off (once a matrix representative of Frobenius is computed).

Relation to the pointwise equation. If $P \in E[\ell]$ has coordinates v_P, the identity

$$\varphi^2(P) - t_\ell \varphi(P) + qP = O$$

is equivalent (under the basis identification) to the vector equation

$$(M^2 - t_\ell M + qI_2)\, v_P \equiv 0 \pmod{\ell}.$$

Thus solving the pointwise equation

$$(x^{q^2}, y^{q^2}) + \bar{q}\,(x, y) \equiv \bar{t}\,(x^q, y^q)$$

for all $(x, y) \in E[\ell] \setminus \{O\}$ is equivalent to finding $\bar{t}$ for which the matrix relation $M^2 - \bar{t}M + \bar{q}I_2 \equiv 0$ holds; checking this relation on a basis (or on two independent ℓ-torsion points) determines $\bar{t}$ modulo ℓ.

3.2 Pseudocode (See Next Page)

Input: An elliptic curve $E : y^2 = x^3 + Ax + B$ over $\mathbb{F}_q$ ($q = p^b$, $p \neq 2$), field size q
Output: Number of points of E over $\mathbb{F}_q$
Choose odd primes S not containing p with $N = \prod_{\ell \in S} \ell > 4\sqrt{q}$;
if $\gcd(x^q - x,\ x^3 + Ax + B) \neq 1$ **then**
  $t_2 \leftarrow 0$;
else
  $t_2 \leftarrow 1$;
end
Work in $\mathbb{F}_q[x, y]/(y^2 - x^3 - Ax - B)$;
$\ell \in S$ Compute division polynomial ψ_ℓ;
Work in $R = \mathbb{F}_q[x, y]/(y^2 - x^3 - Ax - B, \psi_\ell)$;
Let $\bar{q} \in (-\frac{\ell}{2}, \frac{\ell}{2})$ with $q \equiv \bar{q} \pmod{\ell}$;
Compute (x^q, y^q), (x^{q^2}, y^{q^2}), and $\bar{q}(x, y) = (x_{\bar{q}}, y_{\bar{q}})$;
if $x^{q^2} \neq x_{\bar{q}}$ **then**
  Compute $(X, Y) = (x^{q^2}, y^{q^2}) + \bar{q}(x, y)$;
  For$\bar{t} = 1$ **to** $(\ell - 1)/2$ **if** $X = x_{\bar{t}}^q$ **then**
    if $Y = y_{\bar{t}}^q$ **then**
      $t_\ell \leftarrow \bar{t}$;
    else
      $t_\ell \leftarrow -\bar{t}$;
    end
else if q *is a square modulo* ℓ **then**
  Find w with $q \equiv w^2 \pmod{\ell}$;
  Compute $w(x^q, y^q)$;
  if $w(x^q, y^q) = (x^{q^2}, y^{q^2})$ **then**
    $t_\ell \leftarrow 2w$;
  else if $w(x^q, y^q) = (x^{q^2}, -y^{q^2})$ **then**
    $t_\ell \leftarrow -2w$;
  end
  else
    $t_\ell \leftarrow 0$;
  end
else
  $t_\ell \leftarrow 0$;
end
Use CRT to obtain t mod N from $t \equiv t_\ell \pmod{\ell}$ for all $\ell \in S$;
return $q + 1 - t$;

Chapter 4

Existing Methods to Break ECDLP

4.1 ECDLP statement

Let $E/(F)_p$ be an elliptic curve and let $P \in E(\mathbb{F}_p)$ be a point of large prime order n. Given P and $Q = kP$, the elliptic-curve discrete logarithm problem (ECDLP) asks to find k. The best known general algorithms are generic (BSGS, Pollard) and take $O(\sqrt{n})$ operations in the group.

4.2 Why elliptic curves are attractive for cryptography

Elliptic curves provide groups where the best known attacks are generic, which means that much smaller parameter sizes achieve the same security level compared to RSA or finite-field DH. For example, a 256-bit prime-order elliptic group yields comparable security to a 3072-bit RSA modulus.

4.3 Generic algorithms and their limitations

Generic algorithms (BSGS, Pollard's Rho) apply to any group given only the group operation. They require $\tilde{O}(\sqrt{n})$ group operations and set the baseline security for all DLP-based systems. For example, in an elliptic curve group of prime order $n \approx 2^{256}$, the estimated cost of Pollard's Rho is 2^{128} group operations.

4.4 Baby–Step Giant–Step (BSGS)

Baby–Step Giant–Step is a deterministic meet–in–the–middle algorithm for solving the discrete logarithm problem in groups of known order. Suppose we want to recover x from the equation $Q = xP$ in a group of size n. The main idea is to rewrite

$$x = im - j, \qquad m = \lceil \sqrt{n} \rceil,$$

so that the search for x can be converted into finding a collision between two precomputed lists:

$$Q + jP \quad \text{(baby steps)}, \qquad i(mP) \quad \text{(giant steps)}.$$

Baby steps. We compute and store the values

$$Q,\, Q + P,\, Q + 2P,\, \ldots,\, Q + (m-1)P,$$

each indexed by the corresponding j. All these values are inserted into a hash table for $O(1)$ lookup.

Giant steps. We precompute $M = mP$ and then compute the sequence

$$0 \cdot M,\ 1 \cdot M,\ 2 \cdot M,\ \ldots,\ (m-1) \cdot M.$$

For each value iM we check whether it appears in the baby-step table. If a match is found, say

$$iM = Q + jP,$$

then we have $imP = xP + jP$, so $x \equiv im - j \pmod{n}$.

Complexity. The algorithm performs $O(\sqrt{n})$ group operations and stores $O(\sqrt{n})$ group elements. Thus BSGS uses the optimal square-root running time for generic discrete logarithms, but at the cost of significant memory consumption. In elliptic-curve cryptography, where $n \approx 2^{160}$ or larger, memory becomes the bottleneck, which motivates the use of Pollard's Rho instead.

4.5 Pollard's Rho for ECDLP

Pollard's Rho is the most practical and widely used generic attack on elliptic-curve discrete logarithms. It achieves the same $\Theta(\sqrt{n})$ expected running time as BSGS but requires only *constant memory*. Instead of a meet–in–the–middle table, Pollard's Rho performs a pseudorandom walk over the group until two different iterations collide.

Consider again the DLP instance $Q = xP$ in a group of order n. Every point encountered in the walk will be expressed in the form

$$X = aP + bQ,$$

with coefficients tracked modulo n. To randomize the walk, the group is partitioned into a small number of subsets $S_1, S_2, S_3, \ldots$, usually according to a few bits of the x-coordinate. For each region S_k we define a different update rule; a classic choice is

$$X \mapsto \begin{cases} X + P, & X \in S_1, \\ X + Q, & X \in S_2, \\ 2X, & X \in S_3. \end{cases}$$

The coefficients (a, b) are updated accordingly.

Collision detection. The walk looks for a collision $X_i = X_j$ with $i \neq j$. Since the iteration behaves like a random function, the expected time to find a collision is $O(\sqrt{n})$ by the birthday paradox. Cycle detection is performed using Floyd's "tortoise and hare" method or Brent's variant, requiring only a constant number of stored points.

Extracting the discrete log. If $X_i = X_j$, then

$$a_i P + b_i Q = a_j P + b_j Q,$$

which rearranges to

$$(a_i - a_j)P = (b_j - b_i)Q = (b_j - b_i)xP.$$

Provided $b_i \not\equiv b_j \pmod{n}$, we solve for x via

$$x \equiv (a_i - a_j)(b_j - b_i)^{-1} \pmod{n}.$$

Complexity and features. Pollard's Rho uses only $O(1)$ memory and has expected running time

$$T = O(\sqrt{n}).$$

Improvements such as distinguished points, parallel walks, large partitions, and automorphism-based reductions can speed up the method substantially in practice. Because memory is nearly free and scaling is easy, Pollard's Rho is the attack of choice for ECDLP.

Chapter 5

Index Calculus and Number Field Sieve for DLP in Finite Fields

5.1 Index Calculus: main ideas

Index calculus relies on three phases:

1. *Factor base selection:* choose a set of small primes (or places) $\mathcal{B} = \ell_1, \ldots, \ell_m$.
2. *Relation collection:* find random group elements that factor completely over $\mathcal{B}$ and record the exponents, producing linear relations among the logarithms of ℓ_i.
3. *Linear algebra:* solve the resulting sparse linear system for the discrete logs of the factor base elements.

Once the factor-base logs are known, compute an individual logarithm of h by writing hg^k as a product of factor-base elements for a random k.

5.2 Number Field Sieve (NFS)

For large prime fields $\mathbb{F}_p$ the Number Field Sieve (NFS) provides the asymptotically best known algorithm for the DLP in $\mathbb{F}_p^\times$. It uses algebraic number fields for relation collection and achieves subexponential runtime in $L_p[1/3, c]$ for some constant c when carefully tuned. We sketch the structure:

- Select polynomials and an algebraic number field whose norms map to integers that split over chosen factor bases.
- Collect relations by sieving norms for smoothness.
- Solve the resulting large sparse linear system via Lanczos/Block Wiedemann.

5.3 Index Calculus Algorithm (Pseudocode)

Algorithm 1: Index Calculus for DLP in $\mathbb{F}_q^\times$

Input: $g \in \mathbb{Z}_q^*$ generator, $h \in \mathbb{Z}_q^*$, factor base $\mathcal{B} = \{-1, 2, 3, 5, \ldots, p_r\}$

Output: x with $g^x \equiv h \pmod q$

`relations` $\leftarrow \emptyset$;

For$k = 1, 2, \ldots$ Compute $g^k \bmod q$;

Try to factor over $\mathcal{B}$: $g^k \equiv (-1)^{e_0} 2^{e_1} 3^{e_2} \cdots p_r^{e_r} \pmod q$;

if *success* **then**

  store $(e_0, \ldots, e_r, k)$; add if independent;

if *at least* $r+1$ *relations* **then**

  break

end

Form matrix and solve for $\log_g(-1), \log_g(2), \ldots, \log_g(p_r)$;

For$s = 1, 2, \ldots$ Compute $g^s h \bmod q$ and try to factor over $\mathcal{B}$;

if *success* **then**

  return $x \equiv \sum_i f_i \log_g(p_i) - s \pmod{q-1}$

5.4 Complexity

Index calculus-type methods for $(F)_p^\times$ lead to subexponential running times, typically expressed in the L-notation. For the classical index-calculus we obtain runtimes of the form

$$L_p\left(\tfrac{1}{2}, c\right) = \exp\Big((c + o(1))(\log p)^{1/2}(\log\log p)^{1/2}\Big),$$

while the NFS improves to $L_p(1/3, c')$ for suitably large p and careful polynomial choices.

Chapter 6

Why Index Calculus Fails (in General) for ECDLP

6.1 Obstacles to porting index calculus to elliptic curves

Let's see what are the common problems involved while trying to expand Index Calculus to ECDLP.

1. **No norm map / size function.** Index–calculus depends on a multiplicative "size" or norm (e.g. the absolute value for integers or the ideal norm in number fields) that (i) measures the complexity of an element, (ii) decreases under factorisation, and (iii) makes the notion of "y-smooth" meaningful. There is no analogue of such a norm $\mathrm{N} : E(\mathbb{F}_p) \to \mathbb{Z}_{\geq 0}$ that interacts well with the additive group law of an elliptic curve, so one cannot develop a useful theory of smoothness probabilities for points.

2. **The elliptic-curve group law is non-linear, so sums of points do not correspond to multiplicative factorizations.** In multiplicative groups (or in rings) knowledge of the factorisations of two elements gives information about the factorisation of their product. On an elliptic curve the group law uses rational functions (field inversions and cubic relations); a decomposition

$$P = P_1 + P_2 + \cdots + P_k$$

   does *not* translate into a decomposition compatible with a simple "prime" or "irreducible" building block. Thus there is no multiplicative-style algebraic machinery to turn many small relations into a global factorisation.

3. **Smoothness probabilities are far too small.** Index methods rely on a nonnegligible probability that a random element is a sum/product of "small" factors. The

usual smoothness heuristics (e.g. the Dickman–de Bruijn model for integers) have no effective counterpart for elliptic curve points: random points on $E(\mathbb{F}_p)$ do not exhibit analogous smoothness behaviour, and the probability that a random point decomposes as a sum of a bounded set of "small" points is exponentially smaller than what is needed for subexponential algorithms.

4. **No suitable factor base on a genus 1 curve.** A classical factor base is a large, dense collection of small primes or prime ideals. For elliptic curves (genus 1) over $\mathbb{F}_p$ there are very few low-degree places to serve as a factor base: the supply of small-degree places is too sparse (typically only $O(1)$ or at best $O(p^{\varepsilon})$ many) to produce the linear system of relations required by index calculus. Weil restriction or other geometric tricks do not produce a sufficiently dense and useful factor base for generic curves.

6.2 Special curves and exceptions

Certain families of curves are vulnerable because their arithmetic links to finite-field DLPs or higher-genus Jacobians. Important examples:

- **Anomalous curves:** When $\#E(F)_p = p$ (trace $t = 1$), the curve is called anomalous and ECDLP can be solved in polynomial time (Satoh–Araki reduction).
- **Binary field curves:** For curves over $(F)_{2^n}$ Weil descent sometimes maps ECDLP to higher-genus Jacobians where index-calculus is applicable (GHS attack and descendants).

Chapter 7

Does knowledge of one generator allow index calculus for ECDLP?

This section explains why knowing a single generator of $E(\mathbb{F}_p)$ (or of one cyclic direct summand) does *not* generally enable an index calculus style subexponential attack on the elliptic curve discrete logarithm problem (ECDLP). We give precise reductions that show what information an attacker gains from a known generator, why the usual index calculus pillars are missing for general elliptic curves, and we list the special curve families that must be avoided in practice.

Setup and notation

Write the finite abelian group of points in its invariant-factor decomposition

$$E(\mathbb{F}_p) \cong \mathbb{Z}/n_1\mathbb{Z} \times \mathbb{Z}/n_2\mathbb{Z}, \quad n_1 \mid n_2,\ n_1 \mid (p-1). \tag{7.0.1}$$

Choose basis points P_1, P_2 of orders n_1, n_2 respectively, so every point has a unique representation

$$R = xP_1 + yP_2, \quad x \in \mathbb{Z}/n_1,\ y \in \mathbb{Z}/n_2.$$

A typical ECDLP instance gives a public base point B and a target T and asks for an integer t with $T = tB$. We consider the attacker knowledge model where the attacker knows one basis element (say P_2) while P_1 is unknown.

What a known generator reveals

Suppose $T = xP_1 + yP_2$ and the attacker knows P_2. Multiplying by n_1 annihilates the P_1 component:

$$n_1T = n_1y, P_2. \tag{7.0.2}$$

Thus the attacker can compute n_1T and reduce the problem to a discrete log in the cyclic subgroup generated by n_1P_2. Since

$$\mathrm{ord}(n_1P_2) = \frac{n_2}{\gcd(n_1, n_2)} = \frac{n_2}{n_1} \quad \text{(because } n_1 \mid n_2\text{)},$$

recovering the discrete log of n_1T to the base n_1P_2 yields n_1y modulo n_2/n_1, i.e. it determines y only modulo n_2/n_1. No information about x (the coefficient on the unknown P_1) is obtained. In short: knowing one generator gives at best a *partial* coordinate recovery, not a full collapse to a single univariate discrete log.

Why index calculus ingredients are missing

Index calculus algorithms in multiplicative groups rely on three pillars:

1. a canonical notion of factorisation of group elements (e.g. integers factor into primes),
2. a favourable "smoothness" probability for random elements (they factor over a small factor base with noticeable probability), and
3. an efficient linear-algebra stage that pieces relations together to recover logarithms of factor-base elements.

None of these has a natural analogue for general elliptic curves over prime fields: the group law is additive (no canonical prime decomposition of points), definitions of "smoothness" for points yield combinatorial structures with low density, and constructing many independent relations requires work comparable to brute force. Consequently, there is no known general subexponential index calculus algorithm for ECDLP on random prime-field curves; the best attacks remain generic square-root algorithms (Pollard's rho and variants).

Generic-group lower bounds

In the generic-group model, where the adversary only performs group operations and equality tests on abstract encodings/ Any discrete-log algorithm requires on the order of $\sqrt{N}$ group operations (with $N = |E(\mathbb{F}_p)|$). Providing an explicit encoding of one generator does not circumvent this lower bound unless the concrete curve representation leaks extra algebraic structure.

Important exceptions

Non-generic, subexponential attacks do exist for curves with special algebraic structure; these are precisely the families avoided in cryptographic curve selection:

- **MOV / pairing reduction.** Curves with small embedding degree map ECDLP to the DLP in a finite-field extension via pairings, where index calculus is effective.
- **Anomalous curves.** Curves with $\#E(\mathbb{F}_p) = p$ are vulnerable to the Smart / SemaevSmartSatohAraki reductions.
- **Weil descent targets.** Certain curves defined over extension fields can be transferred to higher-genus Jacobians amenable to index calculus.

These vulnerabilities stem from the curve's structure (embedding degree, anomaly, descent behaviour), not from the attacker knowing a single generator.

Practical succinct statement for your report

> Knowing one generator of $E(\mathbb{F}_p)$ (or of a single cyclic summand) reveals at most a residue class of one coordinate and does not provide the multiplicative smoothness structure required for index calculus. Therefore, except for special curve families with additional algebraic structure, knowledge of a single generator does not enable a subexponential index calculus attack on the ECDLP.

References. For standard expositions see the treatments of the generic-group model and index calculus limits in and the original papers on MOV, Smart and Weil descent attacks.

Chapter 8

Other Attempts at ECDLP

This chapter surveys alternative approaches to solving the elliptic curve discrete logarithm problem (ECDLP) beyond generic group algorithms. We focus particularly on Weil–descent–based methods, their successes in small characteristic, and why similar ideas fail to yield subexponential algorithms over large prime fields.

8.1 Weil Descent and Attacks in Characteristic 2^n

When an elliptic curve E is defined over $\mathbb{F}_{p^k}$ with $k > 1$, one may express field elements relative to an $\mathbb{F}_p$-basis of $\mathbb{F}_{p^k}$. Writing

$$x = x_0 + x_1\alpha + \cdots + x_{k-1}\alpha^{k-1}, \quad y = y_0 + y_1\alpha + \cdots + y_{k-1}\alpha^{k-1},$$

transforms the curve equation into a system of multivariate polynomial equations over $\mathbb{F}_p$ in $2k$ variables. This process is called *Weil descent*. Applied to group relations in $E(\mathbb{F}_{p^k})$, it produces an algebraic variety whose $\mathbb{F}_p$-points correspond to valid points on E.

For curves over $\mathbb{F}_{2^n}$ this descent often produces systems that behave favourably:

- The resulting variety may have a moderately large genus, producing a curve of nontrivial genus, enabling index-calculus attacks on its Jacobian.
- Boolean polynomial systems arising after descent can sometimes be attacked by Gröbner-basis or SAT-style methods.
- Summation-polynomial relations (Semaev polynomials) descend to Boolean systems with surprisingly low first-fall-degree (FFD) in small or medium n, giving subexponential attacks in several parameter ranges.

Although promising in characteristic 2, these methods have not scaled to cryptographically relevant n (e.g. $n \geq 200$). Nevertheless, Weil descent remains the only direction

that has yielded non-generic attacks for certain structured curves and thus is the most significant non-generic threat in small characteristic.

8.1.1 Summation polynomials and Gröbner-basis complexity

Semaev's summation polynomials $S_m(x_1, \ldots, x_m)$ encode the condition $P_1 + \cdots + P_m = \mathcal{O}$ using only x-coordinates. After expressing each x_i in coordinates over $\mathbb{F}_p$, a Boolean polynomial system is obtained for $p = 2$.

Initial heuristic analyses suggested that the *first fall degree* (FFD) of these systems is very small (e.g. 3 or 4), implying fast Gröbner-basis elimination and hence subexponential solutions to ECDLP over $\mathbb{F}_{2^n}$. Later work, however, demonstrated that the more relevant parameter is the *last fall degree*, which is typically much larger. Consequently, these attacks lose practicality as n approaches cryptographic sizes, though they remain effective for some smaller parameters.

8.2 Lifting and Rational Factor Bases

A different line of attack considers lifting points from $E(\mathbb{F}_p)$ to $E(\mathbb{Q})$ with the goal of building relations using rational factor bases consisting of points with small-height coordinates. The idea is to factor heights in $\mathbb{Q}$ similarly to integer factorization. Practical obstacles include:

- not every reduction mod p lifts to a rational point with manageable height;
- the probability of a random multiple decomposing over a tiny factor base is negligible;
- ensuring consistency between reductions and rational factorizations is delicate.

As a result *rational-lifting* approaches to ECDLP over prime fields have not produced practical algorithms.

8.3 Index-Calculus via Summation Polynomials in Large Prime Fields

Summation polynomials also give a theoretical mechanism for index-calculus over $\mathbb{F}_p$ with p prime: one attempts to express a point P as a sum of points whose x-coordinates lie in a small factor base. However:

- the decomposition probability for random x-coordinates is extremely small;
- Gröbner computations explode in degree and number of variables;

- no subexponential complexity has been demonstrated for realistic prime-field curves.

Thus, while summation polynomials are central to Weil descent attacks in characteristic 2, they do not yield practical attacks on curves over large prime fields.

8.4 Summary

Weil descent in characteristic 2^n remains the most impactful non-generic strategy, producing real (if parameter-limited) attacks and deep connections to multivariate algebra. In contrast, lifting methods and index-calculus–style attempts over prime fields have not succeeded due to low decomposition probabilities and prohibitive algebraic complexity.

Chapter 9

Conclusion and Future Work

We surveyed the DLP and ECDLP, counting algorithms, index-calculus methods, and current obstacles to transferring index-calculus to prime-field elliptic curves. The high-level takeaway is that group structure matters: multiplicative finite fields admit a rich arithmetic infrastructure that index calculus exploits, whereas prime-field elliptic curves resist such decompositions.

Future directions include:

- Rigorous complexity analyses for summation-polynomial systems and a better understanding of last-fall-degree behaviour.
- Investigation of structured families of curves that may admit faster point-counting or discrete-log attacks (for the purpose of secure parameter generation, these families are avoided by standards).
- Practical improvements in point-counting (SEA optimisations), and further study of Weil descent and algebraic-geometry tools to either strengthen or find vulnerabilities in special curve families.

Bibliography

[1] Wikipedia Contributors. *Schoof's Algorithm*. `https://en.wikipedia.org/wiki/Schoof%27s_algorithm`.

[2] Joseph H. Silverman. *Advanced Topics in the Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics, Vol. 151, Springer, 1994. (Contains detailed discussion of Tate's Lemma and Tate modules.)

[3] Andrew M. Odlyzko. Discrete Logarithms: The Past and the Future. *Designs, Codes and Cryptography*, Vol. 19, No. 2–3, pp. 129–145, 2000.

[4] J. M. Pollard. Monte Carlo Methods for Index Computation (mod p). *Mathematics of Computation*, Vol. 32, No. 143, pp. 918–924, 1978.

[5] S. D. Galbraith, F. Hess, and N. P. Smart. Extending the GHS Weil Descent Attack. In *Advances in Cryptology – EUROCRYPT 2002*, Lecture Notes in Computer Science, Vol. 2332, pp. 29–44, Springer, 2002.

[6] Igor A. Semaev. Summation Polynomials and the Discrete Logarithm Problem on Elliptic Curves. In *Mathematical Cryptology and Cryptanalysis*, Contemporary Mathematics, Vol. 706, pp. 33–50, American Mathematical Society, 2018.

[7] P. Gaudry, F. Hess, and N. P. Smart. Constructive and Destructive Facets of Weil Descent on Elliptic Curves. *Journal of Cryptology*, Vol. 15, No. 1, pp. 19–46, Springer, 2002.

[8] Semaev, I. "New algorithm for the discrete logarithm problem on elliptic curves." IACR Cryptology ePrint Archive, Report 2015/310, April 10, 2015. `https://eprint.iacr.org/2015/310.pdf`

[9] Semaev, I. "Summation polynomials and the discrete logarithm problem on elliptic curves." IACR Cryptology ePrint Archive, Report 2004/031, 2004. `https://eprint.iacr.org/2004/031.pdf`