Formal Description of ECDLP Topology Analysis Research

Abstract

This document provides a comprehensive formal description of the research on the Elliptic Curve Discrete Logarithm Problem (ECDLP) with a focus on topology analysis for the secp256k1 curve. The study explores the topological structure of scalar multiplication operations in elliptic curve cryptography, particularly the double-and-add algorithm used for computing public keys from private keys. By analyzing the sequence of point doublings and additions inherent to the binary representation of the private key, we derive conditions that characterize unique topologies, count their occurrences, and demonstrate methods for private key recovery based on the intrinsic topological properties of the keys themselves. The research is grounded in mathematical formulations, with implementations tested on both a simplified "test" curve and the full "legacy" secp256k1 curve used in Bitcoin. This work aims to assist researchers and practitioners in understanding the combinatorial and structural aspects of ECDLP solvers.

Key contributions include:

- A formal definition of key topology and conditions derived from the private key's structure.
- Analytical formulas for counting unique topologies.
- Empirical statistics from key generation experiments.
- Methods for private key restoration based on topological patterns.

1. Introduction

The Elliptic Curve Discrete Logarithm Problem (ECDLP) is a foundational hard problem in elliptic curve cryptography (ECC). Given an elliptic curve E over a finite field F_p, a base point $G \in E(\mathrm{F}_p)$, and a point $Q = d \cdot G$ where d is a scalar (private key), the ECDLP requires computing d given Q and G. The secp256k1 curve, defined by parameters $p = 2^{256} - 2^{32} - 977$, $a = 0$, $b = 7$, and order $n \approx 2^{256}$, is widely used in cryptocurrencies like Bitcoin.

This research analyzes the *topology* of the scalar multiplication process, which reveals structural patterns directly from the binary representation of d and the operations performed. By modeling the double-and-add algorithm, we identify "conditions" (parameterized by a, b, c, d) that group keys with similar topologies. We also explore combinatorial counting of topologies and private key recovery from the topology inherent to the key itself, without relying on external information leakage.

The study uses two parameter sets:

- **Test mode**: A small curve with $p = 1,241,690,119$, $a = 1$, $b = 35$, $G = (311,072,572,523,565,415)$, $n = 1,241,630,743$, and $l = 29$ (number of fixed double points).
- **Legacy mode**: The full secp256k1 curve with $l = 254$.

2. Mathematical Background

2.1 Elliptic Curve Arithmetic

An elliptic curve E over F_p is given by the Weierstrass equation:

$$y^2 = x^3 + ax + b \pmod{p},$$

where $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$.

Point addition: For points $P = (x_1, y_1)$, $Q = (x_2, y_2)$, the sum $R = P + Q = (x_3, y_3)$ is:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod{p}, \quad x_3 = \lambda^2 - x_1 - x_2 \pmod{p}, \quad y_3 = \lambda(x_1 - x_3) - y_1 \pmod{p}.$$

If $P = Q$ (doubling):

$$\lambda = \frac{3x_1^2 + a}{2y_1} \pmod{p}, \quad x_3 = \lambda^2 - 2x_1 \pmod{p}, \quad y_3 = \lambda(x_1 - x_3) - y_1 \pmod{p}.$$

Scalar multiplication: $Q = d \cdot G$, where $d = \sum_{i=0}^{k-1} d_i 2^i$ (binary representation), is computed via the double-and-add algorithm:

- Start with $R = \mathrm{O}$ (point at infinity).
- For each bit d_i from MSB to LSB:
  - $R \leftarrow 2R$ (double).
  - If $d_i = 1$, $R \leftarrow R + G$ (add).

2.2 Double Points and Fixed Points

Double points are precomputed multiples: Let $D_i = 2^i \cdot G$ for $i = 1$ to l (where l is the bit length, e.g., 29 for test, 254 for legacy). These are fixed and can be indexed (1-based).

The topology of a key d is the sequence of operations during scalar multiplication, recorded as rows $[x_1, y_1, x_2, y_2, x_3, y_3, op]$, where $op = 0$ (double) or 1 (add).

3. Topology of Private Keys

The topology captures the chain of doublings and additions intrinsic to the binary structure of the private key d. For a private key d, the topology $T(d)$ is generated by simulating the double-and-add algorithm, tracking intermediate points. Additions occur only when a bit is 1, corresponding to adding a fixed double point D_j.

Key insight: The topology is determined solely by the positions of 1-bits in d's binary representation. The "added points chain" is the set of indices j where D_j is added, directly reflecting the Hamming weight and bit positions of d.

4. Conditions and Topology Identification

A condition is a string "$a\_b\_c\_d$" where:

- a: Index of the first (x_1, y_1) in the initial addition (1-based).
- b: Index of the first (x_2, y_2) in the initial addition.
- c: Total number of additions (Hamming weight of d).
- d: Index of the last (x_2, y_2) in the final addition.

Constraints:

- $a < b < c < d$
- $a + b + c < d$
- All parameters in $\{1, \ldots, l + 2\}$ for the original model.

These parameters uniquely identify topologies up to the combinatorial choices of addition positions, derived from the key's structure.

4.1 Identifying Condition from Topology

Given $T(d)$ and double points $D = [D_1, \ldots, D_l]$, the condition is extracted by:

- Finding the first addition: Indices of (x_1, y_1) and (x_2, y_2) in D.
- Counting total additions.
- Finding the last addition's (x_2, y_2) index.

5. Counting Unique Topologies

5.1 Original Model

Count topologies where $a, b, c, d \in \{1, \ldots, d_{\max}\}$ ($d_{\max} = l + 2$), with $a < b, a + b + c < d$.

Analytical formula (O($d_{\max}^2$) time):

$$N = \sum_{a=1}^{d_{\max}-2} \sum_{b=a+1}^{d_{\max}-1} \frac{m(m+1)}{2}, \quad m = \max(0, d_{\max} - 1 - (a + b)).$$

For test curve ($d_{\max} = 31$): $N = 14,665$.

For legacy curve ($d_{\max} = 256$): $N = 86,709,504$.

5.2 Restricted Ranges

Given ranges $[a_{\min}, a_{\max}]$, etc., count under strict order $a < b < c < d$ and sum constraint:

$$N = \sum_{a=a_{\min}}^{a_{\max}} \sum_{b=\max(b_{\min},a+1)}^{b_{\max}} \sum_{c=\max(c_{\min},b+1)}^{c_{\max}} \max(0, d_{\max} - \max(d_{\min}, a+b+c+1) + 1).$$

Validation ensures ranges are feasible (e.g., $a_{\max} < b_{\min}$ invalid).

6. Private Key Recovery

Given the set of added point indices $S = \{j_1, \ldots, j_k\}$ (1-based) derived from the key's topology, recover d by reconstructing its binary representation.

Algorithm:

- Let $m = \max(S)$.
- Initialize bitstring $b = 0^m$ (LSB to MSB).
- Set $b[j-1] = 1$ for each $j \in S$.
- $d = \int_b$ (binary to int).

Verification: Compute $Q' = d \cdot G$ and check against known Q.

Example (test): $S = \{1,3,4,5,6,7,9,16,17,18,22,23,26,27,31\}$ yields $d = 1,180,926,333$.

7. Empirical Results and Statistics

Key generation experiments (1M keys for test, 100K for legacy) produce statistics on condition frequencies.

Top conditions cover ~96-99% of keys with 5000 conditions. Total unique conditions: ~8400-8500.

Subset counting for recovery: For condition `a_b_c_d`, the number of possible addition subsets is $\binom{t}{s}$, where t is unknown points, s is additions.

8. Advantages Over Classical ECDLP Attacks

Classical attacks on ECDLP, such as Pollard's rho algorithm, Baby-Step Giant-Step (BSGS), or parallelized variants, face significant limitations due to the computational hardness of the problem. Pollard's rho requires approximately $\sqrt{n}$ operations (about 2^{128} for secp256k1), making it infeasible

with current hardware. These methods rely on collision searches in group operations without exploiting structural properties of the private key.

In contrast, the topological structure method offers several advantages:

1. **Combinatorial Reduction**: By grouping keys into topologies defined by conditions (a, b, c, d), the search space is partitioned into manageable subsets. For a given topology, the number of candidate keys is $\binom{t}{s}$ (where t is the range of possible addition points and s is the Hamming weight), often much smaller than n. Justification: Empirical data shows that top topologies cover 96-99% of keys with only ~5000 conditions, allowing targeted searches within high-frequency groups, reducing effective complexity from exponential to combinatorial.
2. **Intrinsic Key Analysis**: Unlike side-channel attacks (e.g., timing or power analysis) that require physical access or leakage, this method analyzes the inherent binary structure of d via its topology. No external information is needed; the topology is computable from Q and G through reverse-engineering addition chains. Justification: The double-and-add sequence is deterministic from d's bits, enabling recovery from topological parameters without runtime observations, bypassing limitations of classical attacks that treat the group as unstructured.
3. **Scalability for Subspaces**: Classical methods scale poorly for full n, but topology counting (e.g., 86M for legacy) allows precomputation of frequent patterns. Restricted range counting further optimizes for specific key distributions (e.g., low Hamming weight keys). Justification: Analytical formulas compute topology counts in $O(d_{\max}^2)$ or $O(a_{\max} b_{\max} c_{\max})$ time, far faster than $O(\sqrt{n})$, enabling efficient enumeration and brute-force within subsets.
4. **Integration Potential**: This approach can augment classical solvers by prioritizing topologies, e.g., running Pollard's rho within high-probability subsets. Justification: Statistical coverage (e.g., 99% with 5000 topologies) concentrates efforts, potentially reducing overall time by orders of magnitude compared to uniform searches.

These advantages stem from exploiting the algebraic and binary structure of scalar multiplication, which classical attacks overlook, providing a novel pathway for ECDLP analysis.

9. Prospects and Conclusions

This topology analysis reveals combinatorial structures in ECDLP, potentially aiding solvers via intrinsic key properties. However, the central core of the problem remains the prediction of the addition trajectory, specifically the positions within a given number of additions constrained by a given topology. To address this, the author of the research is developing new advanced methods in machine learning/deep learning (ML/DL) and fundamental mathematics aimed at this purpose.

Future work: Optimize counting for larger ranges, integrate with Pollard's rho, explore quantum implications.

Implementations in provided scripts validate the math.

References: NIST FIPS 186-4, Bitcoin secp256k1 specs.