

Friend or Foe?

Anomalous Peers in Monero's P2P Network

Yannik Kopyciok 

Technische Universität Berlin, Germany

Stefan Schmid 

Technische Universität Berlin, Germany

Friedhelm Victor 

TRM Labs, San Francisco, CA, USA

Abstract

Monero, the leading privacy-focused cryptocurrency, relies on a peer-to-peer (P2P) network to propagate transactions and blocks. Growing evidence suggests that non-standard nodes exist in the network, posing as honest nodes but are perhaps intended for monitoring the network and spying on other nodes. However, our understanding of the detection and analysis of anomalous peer behavior remains limited. This paper presents a systematic methodology for detecting anomalous peer behavior in P2P cryptocurrency networks, which we apply in a forensic case study of Monero's network. We collected and analyzed over 240 hours of network traffic captured from five distinct vantage points worldwide. We further present a formal framework which allows us to analytically define and classify anomalous patterns in P2P cryptocurrency networks. Our detection methodology defines detection rules directly applicable to real-time monitoring. Our analysis reveals the presence of non-standard peers in the network where approximately 14.38% (1,877 of 13,050) of the observed peers exhibit anomalous behavior. Of these, at least 1,582 peers show patterns consistent with control by a single entity. These peers exhibit behavioral patterns consistent with a Sybil attack coupled with self-reinforcing peer list poisoning, which establishes the preconditions for degrading Monero's privacy guarantees and undermining network decentralization. To support reproducibility and enable network operators to protect themselves, we release our examination pipeline to identify and block suspicious peers based on newly captured network traffic.

2012 ACM Subject Classification Networks → Network privacy and anonymity

Keywords and phrases network forensics, security, cryptocurrency networks

Digital Object Identifier 10.4230/LIPIcs.AFT.2026.31

Related Version *Full Version:* <https://doi.org/10.48550/arXiv.2509.10214>

Supplementary Material *Software:* <https://github.com/ykpyck/monero-traffic-analysis>
archived at `swb:1:dir:a1913201f79277bf2bd8f1107ef12c594ca5910d`

Funding Supported by German Research Foundation (DFG), SPP 2378/ReNO-2 (grant 511099228), 2025-2029.

1 Introduction

The security and privacy of cryptocurrencies critically depends on the underlying P2P networks as adversaries exploiting network-level vulnerabilities can bypass security and privacy mechanisms applied at the protocol and application layer, risking the network's integrity even for privacy-focused cryptocurrencies. Ten years after its release, Monero continues to be the dominant privacy coin. With a 24-hour trading volume exceeding



© Yannik Kopyciok, Stefan Schmid, and Friedhelm Victor;
licensed under Creative Commons License CC-BY 4.0

8th Conference on Advances in Financial Technologies (AFT 2026).

Editors: Aggelos Kiayias and Maria Kyropoulou; Article No. 31; pp. 31:1–31:22

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

100 million USD¹, Monero demonstrates that users increasingly demand financial privacy comparable to physical cash in traditional monetary systems. Simultaneously, empirical evidence has demonstrated a notable migration from Bitcoin to Monero in illicit online marketplaces [2].

While general blockchain analysis platforms exist [11] Monero’s privacy techniques have demonstrated robust protection against sophisticated blockchain analysis [9, 14]. Consequently, rather than breaking cryptographic primitives, adversaries appear to be increasingly targeting the underlying P2P network layer [4, 5] to exploit network-level vulnerabilities that undermine claimed privacy guarantees. In 2023, community developers identified a few highly saturated /24 subnets, each hosting close to the full 254-host range of peers, opening many short-lived connections in the Bitcoin network as well as the Monero P2P network, with this pattern continuing into 2024². In September of 2024, an independent community analysis³ followed, discussing the potential intrusion of non-standard monitoring nodes within the P2P network. Such monitoring nodes would be positioned to observe RPC calls, potentially linking invoked transactions to real-world IP addresses, a capability that directly undermines Monero’s privacy guarantees. This capability not only defeats Monero’s privacy promise but may also lower the bar for targeted attacks such as focused eclipse attacks against specific users, although eclipse feasibility depends on the details of peer discovery and connection management and several concrete vectors have since been closed. By late 2024, these concerns prompted Monero developers to issue an official warning⁴ and recommend node operators make use of a protective ban list. However, no public explanation was provided of how this ban list was compiled or what criteria were used to identify malicious nodes, creating a significant gap in the understanding of this network-level threat.

A variety of previously studied attacks on blockchain P2P networks [1, 4, 5, 10, 18, 21] show that whilst attacks are feasible, an adversary first needs to infiltrate the network with peers frequently diverting from standard configurations. Detection of such non-standard peers remains limited (Sec. 2), and our work fills this gap by systematically characterizing these subliminal network intrusions.

Contributions. We investigated Monero’s P2P network for about six months with repeated captures of network traffic and analysis cycles. This preliminary analysis revealed that detection cost differs sharply across anomaly classes. Some deviations are visible in the first message a peer sends and others only in the timing of a single connection over its first minutes. Others emerge only once compared against one another. Our final measurement design accommodated the slowest of these. This paper presents a forensic case study, based on a two-day, five-vantage-point traffic capture from June 2025. To our knowledge, this is the first work to systematically detect passive, in-protocol monitoring nodes in a cryptocurrency P2P network, rather than the disruptive attackers targeted by prior detection systems. Our detection angles are tailored to the cryptocurrency setting and exploit Monero’s presumed software homogeneity, under which any systematic deviation becomes a detectable fingerprint. Our systematic examination framework introduces a novel classification: protocol-level anomalies (Sec. 4.1) categorized into syntactic violations, identifying evident deviations from protocol specifications; semantic inconsistencies, going beyond structural validation

¹ <https://coinmarketcap.com/currencies/monero/>

² <https://b10c.me/blog/013-one-year-update-on-linkinglion>

³ <https://www.digilol.net/blog/chainanalysis-malicious-xmr.html>

⁴ <https://github.com/monero-project/meta/issues/1124>

and analyzing logical consistency; and behavioral deviations, including a temporal context separating standard state sequences from anomalous behavior. Furthermore, we advance this peer-based taxonomy by including network structural patterns (Sec. 4.2) and node attributes (Sec. 4.3) commonly found in P2P environments. Despite Monero's P2P network presuming homogeneous node software deployment across all participating peers, our examination (Sec. 5) revealed that approximately 14.38% of peers observed during our measurement period exhibited anomalous behavior, with one entity estimated to control at least 1,582 of the observed nodes. These nodes distort Monero's peer discovery mechanism to gain strategic network positions, enabling an effective first-spy estimator. Our anomaly detection approach complements community efforts by providing a comprehensive framework with empirical evidence for identifying non-standard nodes, raising concerns about Monero's privacy guarantees and network decentralization. To ensure reproducibility and to facilitate follow-up work, we release our GitHub repository and dataset, allowing every Monero node operator to replicate our results and analyze their own network traffic, identify unusual peers, and block them if desired.

2 Related Works

Prior work on Monero and related P2P layers spans topology inference, transaction deanonymization, and eclipse attacks, alongside analogous vulnerabilities in Bitcoin and Ethereum.

Monero's Network Attacks. Cao et al. [5] demonstrated a topology learning attack by exploiting `last_seen` timestamps in Handshake and Timed Sync responses, achieving 97.98% recall for inferring node connections across crawled data. In response, developers removed the timestamp⁵ from Peer Lists (PLs), neutralizing the attack while leaving network centralization unaddressed. Shi et al. [20] later correlated transactions with IP addresses by injecting malicious Monero Tor hidden service nodes and watermarking Tor circuits to link onion and real addresses. Shi et al. [21] then demonstrated the first practical eclipse attack against Monero, combining peer-discovery flooding, priority-list population, and double-spend-induced connection drops to eclipse a victim within minutes using 1,020 IPs. A 2024 update addressed the reset vector⁶, though new vectors may emerge.

Network-Level Attacks on Bitcoin and Ethereum. Eclipse attacks are well studied here. Heilman et al. [10] monopolized a target's peer list and waited for a restart to achieve isolation; Tran et al. [22] introduced Erebus, a stealthier AS-level attack replacing TCP connections without direct occupation; and Marcus et al. [13] forced restarts and flooded incoming connections in a low-resource variant. These works highlight the persistent vulnerability of P2P networks to network-layer isolation.

Anomaly Detection in P2P Networks. Sybil attacks were formalized by Douceur [6], who showed that without a central identity authority they are unpreventable in open systems, and have since been measured in deployed P2P networks. Wang and Kangasharju [23] identified large-scale Sybil deployments in BitTorrent's Mainline DHT precisely by their partial protocol implementation, and Neudecker and Hartenstein [15] survey the corresponding network-layer design space for permissionless blockchains. These works detect Sybils through address concentration, active crawling, or honeypots. We instead fingerprint per-packet and per-connection protocol behavior observed passively from peers that connect to us.

⁵ <https://github.com/monero-project/monero/pull/5682>

⁶ <https://github.com/monero-project/monero/pull/9218>

3 Background

Monero differs from generic P2P networks as there is effectively one maintained client. Within its P2P network, Monero aims for homogeneous-looking traffic as core fungibility design goal, any systematic deviation requires deliberate code modification. Throughout this paper, we use three related but distinct terms with increasing evidential burden. We call a node non-standard if it deviates from the Monero specification at any layer, including protocol fields, message timing, or software configuration. We call a node's behavior anomalous when it produces observable patterns that deviate from what a standard node would produce. Finally, we reserve the terms malicious and adversarial for cases where corroborating evidence supports an intentional, hostile interpretation of non-standard behavior.

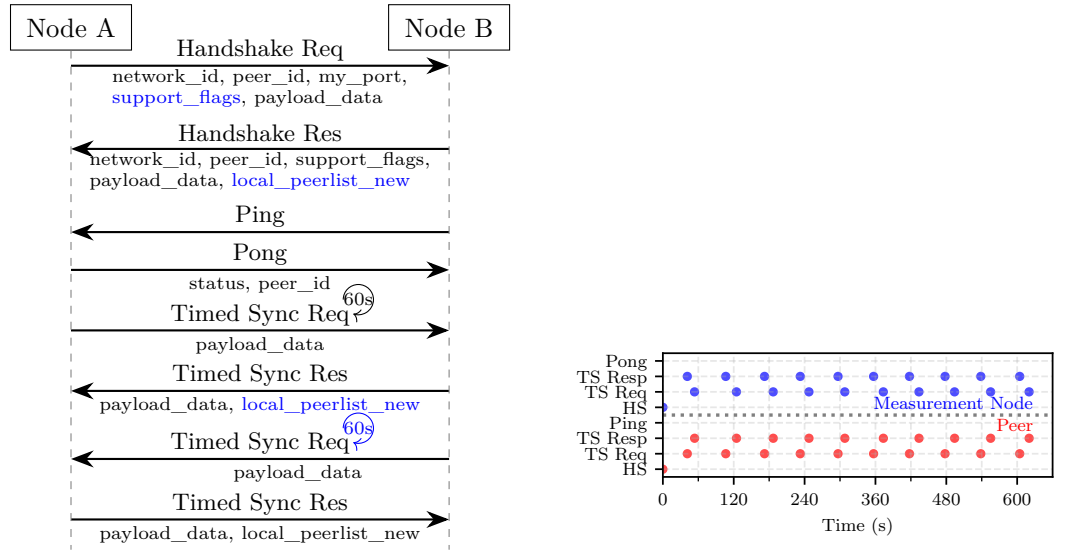
In the following, we will present an overview of relevant functionalities inside Monero's P2P network. Monero's P2P layer uses a custom application protocol known as the Levin protocol [17], most importantly consisting of a Handshake sequence and a continuous peer discovery mechanism using Timed Sync messages as visualized in Figure 1a and seen for a real sample peer in Figure 1b.

Handshake Sequence. To initialize a connection, node A sends a Handshake request to node B, sharing a unique network ID, node ID, and support flags. Since 2018, the support flag is 1 by default⁷, indicating the usage of fluffy blocks, where nodes send compact block representations that contain mostly transaction IDs rather than full transaction data. If B has capacity, B responds with its own node and payload data and additionally a PL containing a maximum of 250 peers as part of Monero's peer discovery mechanism.

Peer Discovery Mechanism. Monero's peer discovery combines local PL management and PL exchanges between neighbors. An initial PL exchange occurs as part of the Handshake, after which both peers enter a 60-second loop that schedules Timed Sync messages to request and send PLs. Each Timed Sync response contains up to 250 white list entries, avoiding duplicates during a continuous exchange. Peer entries include IP address, port, network ID, and optionally pruning seed, RPC port, and RPC cost in credits per hash. Local PL management employs a white list for up to 1,000 validated peers and a gray list for up to 5,000 unvalidated peers. Received peers are initially added to the gray list. Every 60 seconds, the node randomly selects a gray list peer, tests reachability via Ping, and adds it to the white list upon validation. Similarly, incoming connections advertising an open port are tested via Ping and added to the white list if validated.

Offense Tracker. Monero applies a penalty mechanism to score peer behavior and drop connections when necessary [17]. However, the mechanism is limited and evaluates specific behaviors in the P2P communication like blockchain height discrepancies, transaction validity, or block verification failures. Each peer maintains a score that decreases when problematic behaviors are detected, such as sending invalid blocks, providing inconsistent blockchain data, or attempting double-spend transactions. When a neighboring peer's score falls below a critical threshold, the connection to this peer is terminated and it may be temporarily banned from reconnecting.

⁷ <https://github.com/monero-project/monero/pull/3191>



(a) Monero's Default Message Sequence. The figure shows a standard Monero message sequence where node A opens an outgoing connection to node B. The Handshake request initiates the connection. The Ping/Pong sequence verifies that A allows incoming connections and adds A to B's white list. The Timed Sync commands are individual loops to retrieve updated information about the network's topology.

(b) Standard Connection. The graph displays a real sample of a standard incoming connection with each peer sending Timed Sync requests approximately every 60 seconds.

Figure 1 The figures depict how a typical connection in Monero's P2P network should look like, with the default message sequence outlined on the left and a sampled peer connection on the right.

Transaction Propagation. Monero's transaction propagation is based on Dandelion++ [8]. Dandelion++ employs two phases to enhance network-level anonymity. In the stem phase, transactions are forwarded to one of two outgoing relay nodes that are periodically switched. Each intermediate node either continues forwarding as stem or transitions to the fluff phase where transactions are broadcast using standard diffusion. While the protocol provides improved privacy over naive flooding protocols, Sharma et al. [19] demonstrate significant weaknesses against sophisticated adversaries. Using a combination of coordinated graph learning and intersection attacks, even modest adversaries occupying 20% of the network can significantly reduce the set of potential originators [19], revealing a substantial gap between formal guarantees and practical security.

4 Examination Framework

This section introduces our methodology for identifying and categorizing anomalous patterns. The framework provides a comprehensive analytical approach that extends beyond simple protocol compliance checking to encompass content, behavioral, structural, and attribute-based detection methods that can be easily applied to a variety of P2P networks. We establish formal definitions for the core components of our anomaly examination. Let $G = (V, E)$ represent a P2P network where V is the set of peers and $E \subseteq V \times V$ is the set of (logically) bidirectional connections. Each connection $e \in E$ carries a time-ordered sequence of messages $\mathcal{M}_e = \langle m_1, m_2, \dots, m_k \rangle$, where each message has an associated type from the set $\mathcal{T}$ of all protocol message types.

4.1 Protocol-Level Anomaly Detection

Protocol-level anomalies are observable behavioral deviations that indicate non-standard implementations or configurations adapted from traditional signature-based and specification-based detection methodologies for distributed network environments. We define three distinct categories based on detection complexity and required domain knowledge.

4.1.1 Syntactic Violations

To identify syntactic violations, we systematically extract and validate all available protocol fields against current specifications, diverging from traditional signature-based approaches that target specific known attack patterns. Rather than searching for predetermined malicious signatures, our heuristic approach performs exhaustive field-level validation to identify any deviation from syntactic correctness. This includes:

- **Missing Required Fields:** Absence of mandatory protocol elements as defined in current specifications
- **Unexpected Field Presence:** Inclusion of deprecated or non-standard protocol elements
- **Malformed Message Structures:** Violations of defined message syntax and encoding standards

Through exhaustive field-level compliance checking against canonical protocol definitions, this approach identifies nodes running non-standard protocol implementations. While efficient, this methodology cannot detect more sophisticated anomalies or attacks that operate within syntactic correctness boundaries.

Syntactic Formalization. For each type $t \in \mathcal{T}$, let $\mathcal{F}_{req}(t)$ and $\mathcal{F}_{opt}(t)$ represent message type-specific required and optional fields. For a message m of type t , let $\text{dom}(\phi_m)$ denote the set of fields present in m . The message is syntactically anomalous, if:

$$\text{SynAnom}(m) = \mathcal{F}_{req}(t) \not\subseteq \text{dom}(\phi_m) \vee \text{dom}(\phi_m) \not\subseteq (\mathcal{F}_{req}(t) \cup \mathcal{F}_{opt}(t)) \quad (1)$$

4.1.2 Content Anomalies

Content Anomalies extend beyond structural validation to analyze logical content consistency. These subtle anomalies involve correctly formatted messages containing logically inconsistent or fabricated information that requires domain-specific knowledge for detection. Detection methodologies include:

- **Baseline Analysis:** Comparison of observed pattern with established expected content patterns
 - **Cross-Peer Comparison:** Comparing information consistency across multiple sources
- This approach effectively identifies content-based attacks that appear as legitimate protocol exchanges, particularly useful for detecting coordinated misinformation campaigns across multiple peers.

Content Formalization. For content anomalies we model validation through baseline comparison and cross-peer comparison. For a message m of type t with content c_m , the message is anomalous if:

$$\text{ContAnom}(m) = \delta_t(c_m, \mathcal{B}_t) \geq \tau_t \vee \rho_t(\{m\} \cup \mathcal{M}_{cross,t}^{(m)}) \notin [\tau_{c,min}^t, \tau_{c,max}^t] \quad (2)$$

where δ_t measures content deviation from the baseline against threshold τ_t for message type t . ρ_t computes cross-peer comparison scores for message set $\mathcal{M}_{cross,t}^{(m)}$, containing all messages from distinct peers of the same type. $[\tau_{c,min}^t, \tau_{c,max}^t] \subseteq [0, 1]$ defines the expected scoring range for type t .

4.1.3 Behavioral Pattern Deviations

Behavioral pattern deviations extend beyond static protocol validation to examine dynamic communication patterns, capturing anomalies that manifest through temporal and sequential deviations rather than explicit specification violations. It follows stateful protocol analysis principles [12], analyzing communication timing, frequency, and sequence patterns. These anomalies manifest through deviations from expected interaction patterns rather than explicit protocol violations. Key behavioral indicators include:

- **Timing and Frequency Anomalies:** Deviations from expected message timing and frequencies
- **Sequence Violations:** Non-standard state transitions
- **Connection Pattern Anomalies:** Unusual connection establishment and termination behaviors

Our methodology models expected protocol behavior, establishing definitive and statistical baseline patterns that enable detection of subtle deviations indicating reconnaissance activities, resource exhaustion attacks, or protocol manipulation attempts. This approach captures sophisticated attacks that operate within protocol specifications but exhibit anomalous temporal or sequential characteristics.

Behavioral Formalization. For behavioral anomalies we analyze communication patterns across connections and peers. For a connection $e \in E$, behavioral anomalies are detected through:

$$\text{BehavAnom}(e, v) = \text{sequence}(\mathcal{M}_e) \notin \mathcal{S}_{\text{standard}} \vee \delta(E^v, \mathcal{B}_{\text{conn}}) > \tau_{\text{conn}} \vee |\text{timings}(\mathcal{M}_e^t) - \text{timing}_{\mathcal{S}_{\text{standard}}}^t| > \epsilon_t \quad (3)$$

where $\text{sequence}(\mathcal{M}_e)$ extracts the sequence of message types from connection e , $\mathcal{S}_{\text{standard}}$ represents standard sequences, $\text{timings}(\mathcal{M}_e^t)$ extracts the timing frequency of messages of type t in connection e , $\text{timing}_{\mathcal{S}_{\text{standard}}}^t$ represents standard timing for message type t , ϵ is the deviation tolerance threshold for message type t , $\mathcal{B}_{\text{conn}}$ is the baseline for connection patterns, E^v denotes all connections by peer v , and δ measures deviation with threshold τ_{conn} .

4.2 Network Structure

We extend our examination by incorporating classic graph-theoretic concepts, including:

- **Centrality Anomalies:** Detecting nodes with unusual connectivity patterns
- **Community Detection:** Identifying unusual clustering patterns indicating coordinated malicious activity

P2P networks like cryptocurrency networks rely on decentralization and aim to fully decentralize the network. A certain degree of centralization can naturally occur due to geographically more active locations. Nonetheless, prior research by Zabla et al. and Beikverdi et al. reveal concerning trends toward increased centralization in major blockchain networks [3, 25]. In systems with topology obfuscation such as Monero, the true connection graph is not observable. We therefore apply these concepts to an observable proxy, the promotion topology defined in Sec. 5.2.3. We instantiate centrality as promotional influence but leave out community detection, as promotion edges would yield communities of advertisement rather than of connectivity.

4.3 Node Attributes

At last, we introduce a form of attribute fingerprinting based on various available attributes such as IP addresses, unique node identifier, RPC services, or pruning seeds for applicable cryptocurrencies. We include in our analysis:

- **Identity Correlation:** Correlating identities across the network
- **Geographic and ASN Analysis:** Analysis of infrastructure distribution patterns

Unlike protocol-level anomalies, network structure and node attribute assist our analysis in finding potential clusters by adding a network-wide perspective. We thus omit formal mathematical definitions. We now instantiate this framework on data collected from Monero's live network.

5 Case Study: Monero's P2P Network

We implement a comprehensive detection methodology to identify and quantify anomalous behavior within Monero's P2P network, applying the framework established in Sec. 4. The methodology leverages network traffic captured from unmodified P2P nodes participating in the presumably homogeneous network as standard pruned nodes with both incoming and outgoing connections. Our investigation spanned several months with repeated informal snapshots. In these preliminary observations, the same classes of anomalies recurred across capture cycles, which motivated a final measurement design of two consecutive 24-hour snapshots capturing this behavior at scale while keeping storage and computation feasible.

5.1 Data Collection

Our dataset constitutes a snapshot: five unmodified vantage points, two configurations each, captured over two days in June 2025. It is therefore a forensic record of the network state at that time, not a longitudinal characterization. All percentages reported in this section refer to peers observed during this window. Since anomalous infrastructure can be redeployed or withdrawn, current prevalence may differ. However, the released pipeline enables operators to repeat the measurement on newly captured traffic.

- **Locations:** San Francisco, Amsterdam, Bangalore, Singapore, and Sydney
- **Default Configuration:** 12 outgoing connections and no ban list enabled
- **Community Configuration:** 32 outgoing connections and a pre-loaded ban list⁸
- **Timeframe:** Network traffic captured June 2–4, 2025, with each configuration running for 24 hours

Following data collection, network packet analysis was conducted using Wireshark [24] to extract protocol information from identified Monero network traffic. We collected all available data fields, including command identifiers, node attributes, and payload data required for PL analysis. Each PL was systematically processed to extract complete peer entries with all associated metadata fields. This approach yielded a timestamped dataset enabling granular analysis of individual network connections, peer behaviors, and network representations.

5.2 Examination Results

In aggregate, the measurement nodes captured a total of 240 hours of network traffic. The collected dataset statistics are reported in Table 1 across all measurement nodes.

⁸ <https://github.com/monero-project/meta/issues/1124>

■ **Table 1 Individual and Aggregated Node Results.** The table shows the amount of total Monero packets sent and received. The IPs are represented as IPs we have had a direct connection with and additionally total IPs we have seen inside the collected PLs.

	SF	AMS	BLR	SG	SYD	Total
Monero Packets	3,573,738	4,807,042	1,902,680	3,004,744	3,606,164	16,894,368
PLs received	135,242	171,956	113,753	118,692	133,435	673,078
IPs connected (ASs)	6,573 (1,246)	6,697 (1,229)	6,226 (1,170)	6,396 (1,173)	6,507 (1,194)	13,050 (1,870)
IPs seen (ASs)	14,401 (1,668)	17,784 (1,833)	15,191 (1,694)	14,237 (1,630)	15,935 (1,695)	26,919 (2,480)

The extracted PLs contained multiple IPs that are either stale or didn't have a connection established with any of our measurement nodes. These IPs yield information on subnet and AS affiliation but cannot be classified further. The examination results are aggregated over the five nodes. Unless noted otherwise, all examination results derive from the default-configuration nodes, some community-configuration results are reported in Sec. 6.2. All data⁹ and the complete examination pipeline¹⁰ is publicly available to reproduce the results in this paper.

We present the results in order of increasing evidential weight rather than framework order, to build a chain from structural and content indicators toward behavioral and syntactic fingerprinting.

5.2.1 IP Address Affiliation

We begin by characterizing the global distribution of IP addresses, as part of the network structure (Sec. 4.2), across both direct connections and PLs, establishing a structural baseline against which anomalies are measured. As noted, the PLs include IP addresses with varying temporal validity, as PLs may contain stale entries from disconnected nodes or outdated IPs. This limits what PL-derived counts can support. We therefore use PL data only for structural claims about subnet and AS concentration, which are robust to individual entries being outdated, and never as a count of concurrently online peers. Our analysis reveals two concerning findings.

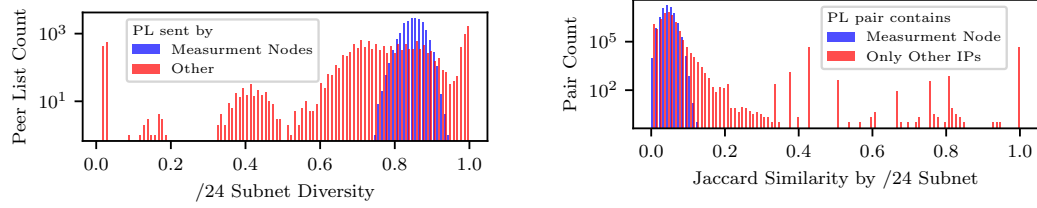
Subnet Saturation. We observed six /24 subnets fully saturated with 254 promoted IP addresses each. Additionally, Figure 2a shows that multiple subnets exhibit a population of ten or more peers.

AS Concentration. All six saturated subnets are assigned to a single AS, which contains an additional densely populated subnet, totaling 1,582 peers within AS LIONLINK-NETWORKS (ASLN), occupying a large fraction of the peers observed in our measurement.

A high number of peers inside a few ASs can be expected and follows an ongoing centralization trend in decentralized systems [5, 16, 18]. However, a high number of peers inside the same subnet is a common indicator that these peers share a single operator. Figure 2a shows the most represented subnets along with their respective AS organization. Whereas the median saturation of any subnet is at 1, the top ones are fully saturated, and the most saturated ones share the ASLN.

⁹ <https://zenodo.org/records/16947083>

¹⁰ <https://anonymous.4open.science/r/monero-traffic-analysis-2528>



(a) PL Diversity. Distribution of PL diversity scores of individual PLs. Blue bars show PLs sent by our default measurement nodes, red bars represent PLs sent by all other IPs. Diversity is the ratio of unique /24 subnets to total peers. A group of lists accumulates at the left side of the figure, representing less than seven subnets among 250 peers.

(b) PL Similarity. Distribution of pairwise similarities between all distinct PL pairs excluding same peer comparisons. IPs in the PLs are reduced to its /24 subnet. Blue bars represent comparisons where at least one PL is sent by a measurement node. The distribution shows noticeable exceptions with high similarity scores and hundreds of pairs reaching a perfect overlap are on the right side of the figure.

■ **Figure 3 PL Inconsistencies.** The figures display PL inconsistencies. The extreme outliers on the left side for the PL diversities and on the right side for PL similarities represent nodes that are part of the ASLN.

identify peers switching IDs¹¹, and adding affected IPs to their published ban list. Finally, one of the five identified clusters contains 109 unique IPs exclusively within the densely populated ASLN.

5.2.3 PL Anomalies

The peer discovery mechanism is central to Monero’s network health, independent of any blockchain or consensus logic, it governs how nodes locate and connect to peers. Here, we applied ContAnom (Sec. 4.1.2) to examine PL content with two approaches. First, we analyzed raw PL data at the individual IP address level. Second, we aggregated IP addresses by /24 CIDR subnet to identify potential clustering patterns. We selected /24 subnets as our aggregation unit because nodes within the same /24 subnet are likely to be controlled by a single entity, making subnet-level analysis a meaningful clustering indicator. Based on the different aggregation methods, we evaluated PLs individually by measuring their diversity and also performed cross-peer comparisons between distinct peers to identify cluster of peers sharing similarities. For each approach, we included only lists containing at least 225 entries (allowing a 10% variation) to establish a comparable baseline of similarly sized lists.

PL Diversity. We define diversity as the ratio of unique /24 subnets to total peers in a list. Under standard operation, random selection from the local white list is likely to produce PLs with high subnet diversity. A perfectly diverse list achieves a diversity index of 1.0, while lower values indicate subnet clustering. Modest clustering may occur naturally due to random selection from popular cloud providers, but extreme clustering represents a content anomaly that deviates from expectations of the protocol’s random selection specification.

Figure 3a shows the distribution of diversity scores for PLs sent by our measurement nodes (blue) and all other IPs (red). All of our measurement nodes consistently achieve diversity scores above 0.7, which aligns with 90% of all PLs. However, two notable outliers (left side) exhibit diversity scores below 0.028. These lists advertise up to 250 peers distributed across just six or fewer subnets. Empirically, we observed 554 unique IPs show this anomalous PL diversity, all residing within the ASLN.

¹¹<https://github.com/Rucknium/xmrnetscan/tree/main>

PL Similarity. We continued our analysis by measuring a pairwise similarity between PLs with IPs aggregated to its subnets. We computed standard Jaccard similarity coefficients, which capture the proportion of shared elements between two sets, to quantify the overlap between peer lists from independent IP sources. For a decentralized P2P network, PLs from independent peers should exhibit low similarity due to the randomized peer selection from distinct local white lists. However, our analysis revealed node pairs with suspiciously high similarity coefficients, including instances of over 90% and perfect overlap in subnet reduced representations. Figure 3b shows the distribution of pairwise similarities. With several peaks, especially, the peak showing a complete overlap in the subnet representation appears critical. Contrary, for similarity pairs where at least one of the PLs belongs to one of our measurement nodes, the similarity score never exceeds a score above 0.2.

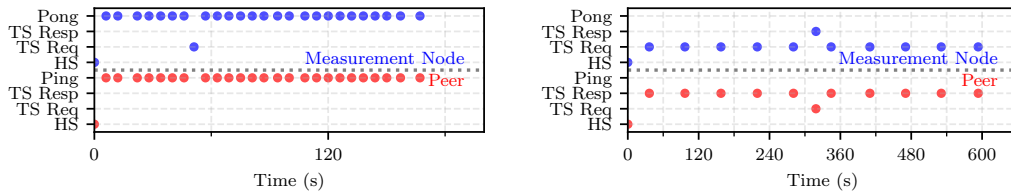
Empirically, 553 unique IPs achieve perfect subnet-level overlap in their peer lists, all residing within the ASLN. These peers exclusively advertise subnets in which they themselves reside, forming a promotional network spanning seven subnets. A peer that had itself been poisoned would predominantly advertise a foreign subnet. Self-referential advertisement, where a peer’s list is dominated by the subnet it resides in, is therefore more consistent with being the source of the promotion than its recipient.

Monero’s Promotion Topology. The PL anomalies identified above characterize individual peer behavior. We now examine how these behaviors manifest at the network level. Since Monero implements topology obfuscation to protect routing privacy, we cannot reconstruct the true connection graph. Rather than developing novel topology learning attacks, we examine Monero’s “promotion topology”, revealing structural anomalies in Monero’s peer discovery mechanism. We define the promotion topology as a directed graph with IP addresses as nodes. For each received PL, we construct directed edges from the source of the PL to every peer it has promoted within this PL. While this graph does not reflect the network’s actual topology at any given time, it captures promotional behaviors. Therefore, multiple edges from one node to another are allowed as multiple promotions naturally occur for extended timeframes. This view has limits: promotion edges reflect advertisement behavior, not established connections, and are biased toward peers that our vantage points happened to observe. It therefore supports claims about promotional prominence, not about actual link-level topology or global node degree.

We observed, that the in-degree distribution of the promotion topology is highly skewed, with a small number of peers receiving a disproportionate share of promotions, consistent with scale-free degree distributions observed in Internet topology [7]. At the IP level, individual ASLN addresses do not rank among the highest in-degree nodes. However, when promotions are aggregated by AS and within our captured PLs, ASLN dominates, receiving nearly three times as many promotions as the next largest AS (Hetzner Online GmbH) and more than hundred times as many as the top individual IP.

5.2.4 Behavioral Anomalies

Structural and content anomalies established the presence of non-standard peers. We now apply *BehavAnom* (Sec. 4.1.3) to examine behavioral deviations that further characterize these nodes. We measured timing, frequency, and sequential deviations across the P2P protocol. Monero’s message set comprises base commands that follow deterministic patterns and event commands that exhibit unpredictable timing characteristics. A standard connection sequence consists of two mandatory message exchanges, bidirectional Handshake and periodic Timed Sync commands, each in request-response format. Additionally, when a peer advertises



(a) Ping Flooding. A peer spamming Ping messages to our measurement node. After a successful Handshake, the peer spams Ping messages. The measurement node does not crash, treats the connection as usual, and starts at around 60 seconds with the first Timed Sync request. The connection is dropped as the peer does not respond to the request within 120 seconds.

(b) Throttled Timed Sync. Sample peer throttling its Timed Sync request messages. Contrasting to the measurement node using the fixed 60-second frequency for sending requests, the peer follows a non-standard frequency. With a single request within ten minutes while responding timely, the peer extremely deviates from the protocol standard.

■ **Figure 4 Behavioral Anomalies.** Both figures show connection-level deviations from Monero's protocol standard, as observed on our measurement nodes.

an accessible port for incoming connections, Ping messages are transmitted by the connecting peer to verify the advertised inbound connectivity capability. The Timed Sync exchanges occur at a defined frequency of 60 seconds, initiated by both peers in parallel. Figure 1 demonstrates the default and baseline behavior. The patterns represent the expected behavior between two standard Monero clients. To identify anomalous behavior, we focus on the three predictable command patterns. We adopt a connection grouping methodology, filtering our dataset by individual connections before conducting detailed analysis. Connections exhibiting TCP-level anomalies, likely attributable to Wireshark dissector limitations, were excluded from analysis due to their negligible frequency and potential for introducing analytical artifacts. We located:

- 181,091 total TCP connections.
- Connections last on average 364.61 (median: 0.54) seconds, and
- connections contain on average 93.14 (median: 4) commands.

The discrepancy between average and median is the result of many short-lived connections usually lasting less than one second.

Short-lived Connections. While brief connections may result from legitimate network conditions or operational factors, it becomes critical to determine whether specific peer populations exhibit disproportionate representation in short-lived connection occurrences. We therefore analyzed the distribution of IPs across short-lived connection anomalies, as repeating IPs indicate systematic scanning or probing activities. Empirically, we observed 85,288 connections lasting less than 1 second. Of these, 10 unique IPs have more than ten short-lived connections, across 7 unique ASs.

All of the observed IP connections completed the Handshake sequence and, in some cases, sent a single Ping or Timed Sync message before terminating. Notably, a single IP address accounts for 65,340 connections terminating in under one second. This pattern indicates systematic probing rather than transient network churn. We note that research and community crawlers produce a similar signature and this evidence alone does not distinguish scanning conducted for measurement from scanning conducted for reconnaissance.

Ping Flooding. We observed excessive Ping message transmissions beyond what normal peer verification requires. Figure 4a visualizes a sample connection exhibiting this bursty behavior. Within 24 hours, our measurement nodes received 2,900 Ping flooding incidents,

each delivering a burst of around 23 pings, from 501 distinct IPs, all residing in the ASLN. Notably, incoming connections should only respond to locally initiated Ping messages to prove their capability to as well accept incoming connections, showing a clear deviation from defined protocol sequences. These connections serve no other apparent protocol function, as evidenced by their uniform disregard for Timed Sync requests. They consistently ignore the first Timed Sync request, causing the connection to drop after the 120-second inactivity timeout, indicating that these nodes neither seek nor sustain normal protocol participation.

Throttled Timed Sync. Given the fixed protocol specification requiring Timed Sync requests every 60 seconds, we measured the frequency distribution of these messages across all connections. Figure 4b demonstrates a peer transmitting Timed Sync requests far off the specification. We initially developed a baseline by analyzing the network’s overall distribution of Timed Sync frequencies as some timing variations may be attributed to network latency or node computational load. We average the Timed Sync frequency on each connection and the median is at around 61.02 seconds, matching with the protocol specifications. However, we identified connections that consistently deviate from the defined 60-second interval. We set our threshold at 120 seconds, what corresponds to monerod’s timeout period after which a node connection drops due to peer unresponsiveness. Any peer with a Timed Sync frequency above 120 seconds must be responsive in general communication. In all cases where the Timed Sync frequency exceeds 120 seconds, peers respond to our requests without notable delay. While we cannot attribute throttled Timed Sync to a single specific attack vector, the behavior requires deliberate source code modification. Empirically, we observed 121 connections have a frequency above 120 and 27 IPs maintain connections with throttled Timed Sync frequencies, across 21 unique ASs.

5.2.5 Syntactic Anomalies

Beyond behavioral deviations, we applied *SynAnom* (Sec. 4.1.1) and identified three syntactic violations of Monero’s protocol standard, indicating outdated or incorrectly implemented protocol standards. While not immediately critical in isolation, they are consistent with intentional customization and correlate with other anomalies identified above.

Support Flags Omission. We identified nodes omitting the support flags field during Handshake exchanges. Support flags are enabled by default. These peers must have explicitly set this flag to 0 or falsely implemented a custom client. Beyond syntactic identification, this violation creates a unique command sequence pattern where support flags are explicitly requested through Support Flags request messages, followed by response messages, deviating from standard command sequence flows. Empirically, we observed 1,054 unique IPs omit the support flags, across 80 unique ASs.

These missing flags resulted in 177,853 additional support flags request sequences, deviating standard protocol flow and creating increased network workload.

Deprecated `last_seen` timestamps. During PL exchanges in Handshake responses or Timed Sync responses, we observed peers still including the deprecated `last_seen` timestamp field. As of our observation, the timestamp that these peers transmit appears to be placeholders fixed to a value, indicating an outdated custom implementation of a Monero node. Empirically, we observed that 288 unique IPs inside the ASLN continue to transmit the `last_seen` timestamp.

While these two anomalies represent syntactic violations, they have historically correct syntax and are still accepted in Monero’s current connection handling. Nonetheless, these outdated definitions date back longer than the last network upgrade, which should have enforced every node to update their software, raising questions on the legitimacy of node software running on servers showing these violations.

TCP Fragmentation. Messages in P2P traffic can get bigger than the Maximum Transmission Unit (MTU) size. Those messages naturally get fragmented and reassembled and the size of the fragmentation can differ depending on the network path’s MTU size. However, we identified IP addresses that consistently fragment packets in an unusual manner. For each reassembled message, the first packet only contains Monero’s 8 byte signature. This persists even for messages that could be transmitted in a single packet. We observed 1,123 unique IPs sending signature only packets, across 5 unique ASs.

Such a behavior is unlikely to occur naturally, as legitimate MTU-based fragmentation would create more varied fragment sizes based on network conditions. The concentration within a small number of ASNs, including the ASLN, is consistent with coordinated deployment.

Individually, several of these anomalies admit benign explanations. Their joint co-occurrence within a shared AS, however, is not explained by hosting, VPN aggregation, or churn, and is most parsimoniously attributed to a single coordinated actor. We make this interpretation explicit in Sec. 6.

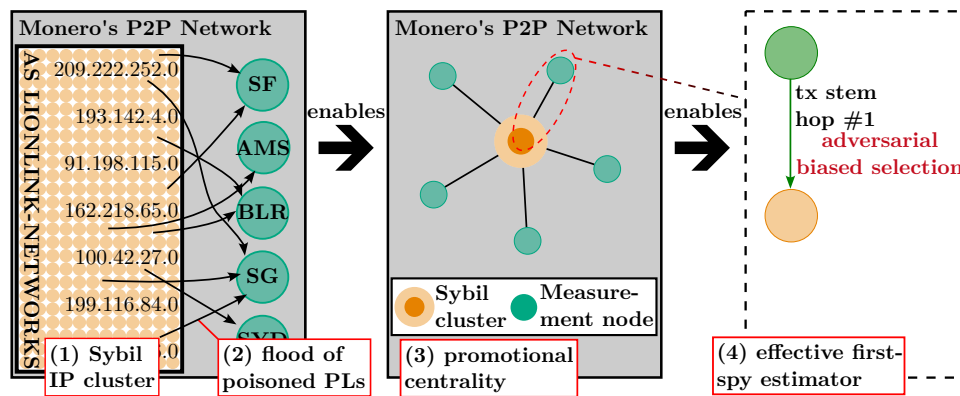
6 From Anomalies to Adversary

In this section, we first characterize the coordinated attack chain (Sec. 6.1), evaluate existing defenses (Sec. 6.2), and then propose mitigations (Sec. 6.3). Throughout this section, we use the term “adversary” to refer to the entity or entities responsible for the anomalies characterized in Sec. 5. We acknowledge in Sec. 7 that anomalous behavior does not constitute proof of malicious intent. However, the scale, coordination, and network positioning of the observed anomalies support an adversarial interpretation.

6.1 A Coordinated Attack Chain

The adversary’s operational footprint likely extends beyond ASLN. As shown in Figure 6a, the anomaly overlap reveals that many identified anomalies extend beyond ASLN into additional ASs. This suggests the adversary operates nodes with distinct IP addresses outside their primary infrastructure, either as auxiliary capacity or to reduce detectability. We focus our threat analysis on the ASLN cluster (1,582 nodes), where anomalies are most concentrated and corroborated, but the true operational footprint is likely broader. We break down the threat chain shown in Figure 5 by evaluating the identified anomalies and explaining the effect each enables.

(1) Sybil attack. At the center of our analysis lies a set of seven highly saturated subnets. The concentration of 254 peers within a single /24 subnet is strongly indicative of a Sybil attack, a conclusion further supported by several corroborating anomalies. First, these peers belong to large peer ID clusters characterized by shared and frequently rotated identifiers. Second, they exhibit a distinctive communication pattern consisting of signature-only packets. This uniform behavior suggests these IPs function as proxy or VPN endpoints load-balancing traffic for a limited number of actual Monero nodes.

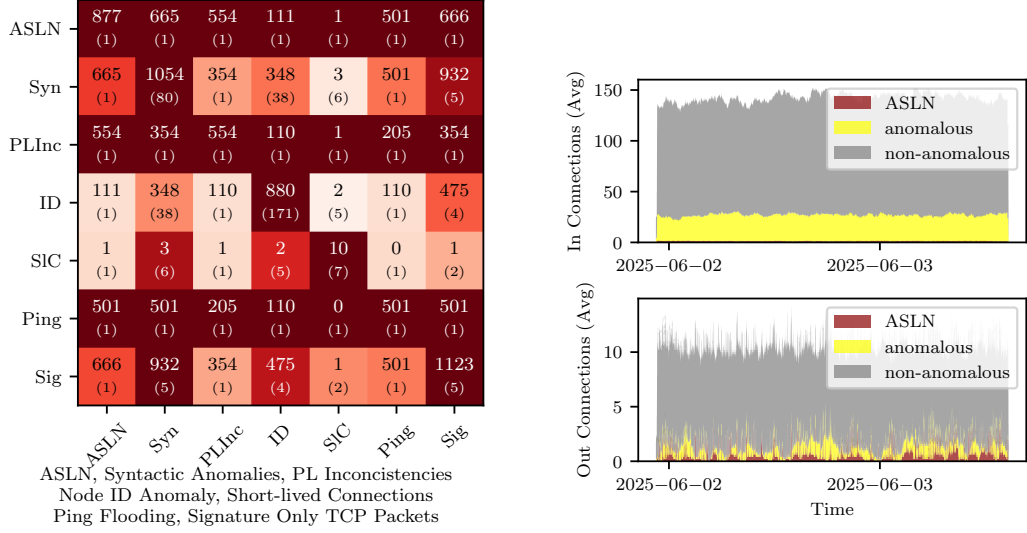


■ **Figure 5 ASLN Threat Chain.** AS LIONLINK operates seven densely populated subnets forming a Sybil cluster within Monero's P2P network. (1) The cluster's scale enables a Sybil attack, with the full saturation of /24 subnets suggesting that a small number of actual nodes operate behind proxy endpoints. (2) Peers across all seven subnets flood the network with poisoned PLs exclusively containing IP addresses from within these same subnets, as observed by each of our five monitoring nodes. (3) The combined effect of Sybil presence and PL poisoning translates into outsized promotional centrality, with adversarial peers consistently occupying a fraction of outgoing connections. (4) This central positioning enables an effective first-spy estimator: with high probability of being selected as an outbound relay during Dandelion++'s stem phase, the adversary can link transactions to their originating node.

(2) Peer List Poisoning. The semantic PL inconsistencies imply poisoning of the discovery mechanism to increase visibility and gain strategic positioning. Although nodes will prefer to choose outgoing connections to distinct /16 subnets, high amounts of nodes in the local white and gray lists will naturally increase the chance of picking a node in the ASLN. This visible PL manipulation occurs exclusively for the ASLN but appears to be very effective. The Sybil attack combined with the PL poisoning results in widespread exposure across the network. We examined the inferred direct exposure of each node by analyzing all collected PLs along their promoted peers. Our measurements reveal that the majority of PLs we received contain ASLN peers, and a peer's PL is on average occupied by 16.22% ASLN peers.

(3) Central Network Positioning. The promotion topology analysis in Sec. 5.2.3 establishes that ASLN achieves outsized promotional prominence in our captured PLs. To quantify how this translates into direct exposure on a node-level, we measured how many ASLN and anomalous peers occupy our measurements' nodes active connection pools over the observation period, averaged across all five nodes. As shown in Figure 6b, ASLN peers occupy 0.94% of incoming and 6.69% of outgoing connections on average. Additionally, 18.05% and 7.78% of incoming and outgoing connections are occupied by anomalous peers. This degree of saturation is not coincidental but a direct operational consequence of the Sybil attack and PL poisoning described in steps (1) and (2). By flooding the network with nodes and ensuring those nodes dominate peer lists, the adversary translates promotional influence into persistent connection-level presence. This central positioning is the precondition for the privacy attacks described next.

(4) Transaction Privacy Degradation. The presence and network positioning of ASLN peers creates the preconditions necessary to compromise Dandelion++. The main capability to undermine privacy guarantees is based on the central position its nodes achieve through



(a) Anomaly Overlap Matrix. The matrix presents the overlap of IPs (ASs) showing anomalous behavior in the respective anomaly. Whereas many IPs are often affiliated with just a few ASs, some anomalies show higher AS overlap than IP overlap, suggesting coordinated deployment of peers with distinct functionalities.

(b) Proportion of Active Connection Saturation. The graph visualizes the proportion of active connections. Outgoing and incoming connections show on average around 7.78% and 18.05% are occupied by non-standard peers and 6.69% and 0.94% are occupied by adversarial peers.

Figure 6 Adversarial Reach. The overlap matrix (left) characterizes which anomalies share IP and AS affiliation. The connection saturation graph (right) quantifies how this translates into direct presence within an individual node's active connection pool.

effective PL poisoning and Sybil attacks. Dandelion++ applies an initial stem phase where the transaction is only forwarded to one of the node's chosen outbound relays. With the ASLN occupying on average 6.69% of each node's outgoing connection pool, the adversary consistently has a chance of being picked as an outbound relay, thereby being positioned to learn about new transactions and apply a first-spy estimator (a deanonymization technique linking the first node to relay a transaction as its likely originator). Assuming the ASLN successfully learns the network's topology, it can further limit possible originators beyond relying solely on the first-spy-estimator. According to Sharma et al. [19], an adversary that controls around 20% of the network can intercept around 70% of transactions and effectively reduce the anonymity set to 32 possible originators per transaction. However, we observed a node's connection pool to be occupied by 7.78% of anomalous peers. This scenario potentially constitutes a more critical threat model than conventional network distribution approaches. Widespread connection saturation may enable direct node identification or, as approached by Sharma et al., allow adversaries to systematically eliminate candidate nodes, constraining the anonymity set to a limited subset of potential originators.

Finally, in certain cases, the adversary might confidently identify a transaction's originator. If the adversary knows whether a node is reachable (i.e., whether it allows inbound connections) and receives a stem transaction from such an unreachable node, it can confidently identify this as the originator since the node cannot have relayed the transaction from elsewhere.

Eclipse Attacks. Another final attack mapping follows a combined approach of Handshake and Ping flooding as reported in two distinct vulnerability reports ¹². The observed patterns also align closely with the eclipse attack methodology described by Shi et al. [21]. Many distinct peers align around each measurement node, able to quickly fill a victim's gray and white lists once needed. The next step requires a connection reset attack. While the double-spend attack vector has been closed, we observed many attempts to perform Ping flooding attacks, which have a similar outcome but have also been disclosed and fixed. We did not observe a successful eclipse attack against our measurement nodes. Our observations are most consistent with a broad monitoring and Sybil deployment rather than a targeted eclipse campaign. A targeted eclipse attacker would concentrate resources on selected victims and would largely avoid simultaneous exposure at five unrelated vantage points. Our data therefore speaks to eclipse capability rather than to an attempt in progress.

6.2 Defense Mechanisms

As a potential countermeasure, Monero contributors advertise the usage of a community crafted ban list. We evaluate the effectiveness of this defense against the identified threats.

Ban List Effectiveness. To compare the ban list with our identified anomalous nodes, we merge all anomalous peers identified across measurement nodes, arriving at 1,877 anomalous IP addresses. Of these, 734 are not in the community proposed ban list. With an active ban list and an increased pool of outgoing connections, we observed no Ping flooding attacks and outgoing connections occupied by anomalous peers dropped to an average of 3.18%. The ASLN is fully blocked on the ban list.

Ban List Limitations. Although a ban list seems to increase privacy from an individual node's perspective, we observed ban-listed peers being promoted in the majority of PLs. This suggests that many nodes in the network have not enabled the recommended ban list. As such nodes will be neighbors of nodes having a ban list enabled, the adversary will most likely be able to map these neighboring nodes despite their active ban list. Extending this, even if a majority is reached that have enabled an effective ban list, updates must be maintained and monitored on an ongoing basis. The static nature of a ban list cannot adapt to a rapidly evolving adversary.

Furthermore, the results showed that even with an active ban list many distinct anomalies persisted. We continued to observe several distinct anomalous behaviors for each measurement node with an active ban list throughout the observation period. Extending ban list coverage to entire ASs would reduce exposure further, but an adversary with sufficient resources to deploy a diverse, multi-AS infrastructure can circumvent AS-level filtering by design. The fundamental limitation is architectural, as a static, externally-maintained list reacts to observed behavior rather than preventing it.

6.3 Mitigation Strategies

Protocol changes such as the removal of the `last_seen` timestamp or support flags date back to 2018, but the network does not yet punish false usage of either. The already existing offense tracker could be updated to account for syntactic incorrectness and act accordingly.

¹²<https://hackerone.com/reports/3185083>, <https://hackerone.com/reports/2858802>

The effectiveness of PL poisoning could be reduced through periodic pruning of a node's local white and gray lists, removing duplicated subnets or even ASs. In addition, a set of anchor nodes, potentially the seed nodes, could act as a regular peer, providing a healthy PL to help counteract the self-reinforcing nature of PL contamination.

However, these mitigations address symptoms rather than root causes, and each defense carries its own tradeoffs. Validating PL entries and pruning duplicated subnets raises the cost of PL poisoning, but favoring older or more stable peers risks introducing centralization and ossification, and constraining peer selection too far erodes the randomness of outgoing connections on which eclipse resistance depends. Reputation systems for peer behavior can themselves be accumulated and gamed by Sybil identities, shifting the attack surface rather than eliminating it. Any effective mechanism to detect and penalize coordinated Sybil networks must therefore be designed with these manipulation risks in mind. The sophistication of the observed adversary suggests that incremental improvements to existing defenses may prove insufficient without fundamental architectural changes to the peer discovery and connection management mechanisms, yet such changes must balance Sybil resistance against decentralization and openness. The community ban list remains a useful short-term measure but provides no durable security guarantee against the adversary characterized in this paper.

Complementing these protocol-level changes, our released pipeline lets operators act today: it takes a PCAP captured by their own node and outputs a per-anomaly breakdown, reporting each flagged IP alongside the specific anomaly classes it triggered. Operators can inspect *why* a peer was flagged rather than trusting an opaque list, and derive a local ban list from the output. As it is regenerated from current traffic, the breakdown adapts to redeployed infrastructure, addressing the static ban list's central limitation (Sec. 6.2). Flags remain indicators rather than proof of malice, so operators tune sensitivity against their review capacity.

7 Discussion

Our approach provides us with a verified behavioral baseline from five geographically distributed locations. Running the default and unmodified Monero software allows us to distinguish standard from anomalous behavior, and to infer non-standard implementations where the observed behavior cannot be produced by an unmodified Monero client. This consistent baseline behavior, combined with Monero's network architecture presuming homogeneous node software deployment, ultimately allows us to distinguish between standard and non-standard behavior.

Thresholds, Anomalies, and Malicious Intent. We identified many IPs showing anomalous signs, associated with either few or many distinct ASs. While syntactic anomalies are clear indicators, behavioral inconsistencies are more ambiguous. Node ID anomalies, throttled Timed Syncs, or PL similarities might indicate anomalous peers without malicious intent. Several benign explanations exist for individual anomalies in isolation. Shared VPN endpoints may cause multiple IPs to announce the same peer ID. Network congestion may account for throttled Timed Sync frequencies. Peering agreements between hosting providers could produce correlated peer lists over time. Short-lived connection bursts and elevated Ping rates are likewise characteristic of active measurement infrastructure. However, these explanations do not account for the joint co-occurrence of anomalies within a single AS, nor for the scale and uniformity of the observed behavior. While our node's data serves as a verified baseline for identifying anomalies, direct classification as malicious requires verified labels. However,

consistent deviation from network baseline remains significant for security contexts, regardless of ultimate intent. A sensitive threshold can produce many false positives requiring manual review but reducing missed threats. An insensitive threshold can prioritize precision at the cost of potentially overlooking subtle anomalies. In our examination, we focused on clear outliers in our dataset rather than defining specific thresholds, as optimal values depend on operational capacity and use case requirements.

Community Awareness. Monero’s community has been increasingly aware of potential surveillance and malicious activities within the network with an ongoing discussion and monitoring¹³ that partially overlap with our findings. Most prominently, repeated ID switching is used to fingerprint spy nodes, and affected IPs are added to a continuously updated ban list. Our anomaly detection approach complements these community efforts by providing a comprehensive framework with empirical evidence for identifying non-standard nodes. The Monero Research Lab is aware of our analysis and already is testing our examination pipeline on newly captured data, which supports the practical applicability of our approach despite the absence of formal ground truth data.

8 Conclusion & Future Work

In this work, we have proposed methods to identify anomalous nodes in Monero’s P2P network. We have found that approximately 14.38% of the observed peers in Monero’s P2P network exhibit anomalous behavior. At the time of measurement, this indicated that a well-resourced actor was active in the network, performing what appears to be PL poisoning, topology mapping, and connection saturation. Despite community mitigation efforts, including when adopting ban lists, anomalous peers still occupied 3.18% of outgoing connections, potentially undermining Monero’s privacy guarantees and creating a critical gap between theoretical protections and practical vulnerabilities.

First, we envision further study of the promotion topology as we have identified interesting patterns that might yield further insights. Second, another direction concerns the impact on Dandelion++ considering adversarial structures which go beyond random distributions in the network.

References

- 1 Maria Apostolaki, Aviv Zohar, and Laurent Vanbever. Hijacking bitcoin: Routing attacks on cryptocurrencies. In *2017 IEEE symposium on security and privacy (SP)*, pages 375–392. IEEE, 2017. doi:10.1109/SP.2017.29.
- 2 Katsiaryna Bahamazava and Rohan Nanda. The shift of darknet illegal drug trade preferences in cryptocurrency: The question of traceability and deterrence. *Forensic Science International: Digital Investigation*, 40:301377, 2022. doi:10.1016/j.fsidi.2022.301377.
- 3 Alireza Beikverdi and JooSeok Song. Trend of centralization in bitcoin’s distributed network. In *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 1–6, 2015. doi:10.1109/SNPD.2015.7176229.
- 4 Alex Biryukov and Sergei Tikhomirov. Deanonymization and linkability of cryptocurrency transactions based on network analysis. In *2019 IEEE European symposium on security and privacy (EuroS&P)*, pages 172–184. IEEE, 2019. doi:10.1109/EuroSP.2019.00022.

¹³<https://moneronet.info>

- 5 Tong Cao, Jiangshan Yu, Jérémie Decouchant, Xiapu Luo, and Paulo Verissimo. Exploring the monero peer-to-peer network. In *Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 Revised Selected Papers 24*, pages 578–594. Springer, 2020. doi:10.1007/978-3-030-51280-4_31.
- 6 John R. Douceur. The sybil attack. In Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*, pages 251–260, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. doi:10.1007/3-540-45748-8_24.
- 7 Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. *ACM SIGCOMM computer communication review*, 29(4):251–262, 1999. doi:10.1145/316188.316229.
- 8 Giulia Fanti, Shaileshh Bojja Venkatakrishnan, Surya Bakshi, Bradley Denby, Shruti Bhargava, Andrew Miller, and Pramod Viswanath. Dandelion++: Lightweight cryptocurrency networking with formal anonymity guarantees. *Proc. ACM Meas. Anal. Comput. Syst.*, 2(2), June 2018. doi:10.1145/3224424.
- 9 Nada Hammad and Friedhelm Victor. Monero traceability heuristics: Wallet application bugs and the mordinal-p2pool perspective, 2024. doi:10.48550/arXiv.2408.05332.
- 10 Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. Eclipse attacks on {Bitcoin’s}{peer-to-peer} network. In *24th USENIX security symposium (USENIX security 15)*, pages 129–144, 2015. URL: <https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman>.
- 11 Harry Kalodner, Malte Möser, Kevin Lee, Steven Goldfeder, Martin Plattner, Alishah Chator, and Arvind Narayanan. BlockSci: Design and applications of a blockchain analysis platform. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2721–2738. USENIX Association, August 2020. URL: <https://www.usenix.org/conference/usenixsecurity20/presentation/kalodner>.
- 12 Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24, 2013. doi:10.1016/j.jnca.2012.09.004.
- 13 Yuval Marcus, Ethan Heilman, and Sharon Goldberg. Low-resource eclipse attacks on ethereum’s peer-to-peer network. Cryptology ePrint Archive, Paper 2018/236, 2018. URL: <https://eprint.iacr.org/2018/236>.
- 14 Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, and Nicolas Christin. An empirical analysis of traceability in the monero blockchain, 2018. doi:10.48550/arXiv.1704.04299.
- 15 Till Neudecker and Hannes Hartenstein. Network layer aspects of permissionless blockchains. *IEEE Communications Surveys & Tutorials*, 21(1):838–857, 2019. doi:10.1109/COMST.2018.2852480.
- 16 Michael Polinski, Richard Jo, Kevin McAfee, and Fabián E. Bustamante. The centralization of a decentralized video platform - a first characterization of peertube. *SIGCOMM Comput. Commun. Rev.*, 54(4):25–35, February 2025. doi:10.1145/3717512.3717516.
- 17 Monero Project. Monero. <https://github.com/monero-project/monero>, 2024. offense tracker: src/cryptonote_protocol/ cryptonote_protocol_handler.inl:2792, levin protocol: docs/LEVIN_PROTOCOL.md., peer_id: src/p2p/net_node.inl:143 Accessed: May 06, 2025.
- 18 Muhammad Saad, Victor Cook, Lan Nguyen, My T. Thai, and David Mohaisen. Exploring partitioning attacks on the bitcoin network. *IEEE/ACM Transactions on Networking*, 30(1): 202–214, 2022. doi:10.1109/TNET.2021.3105604.
- 19 Piyush Kumar Sharma, Devashish Gosain, and Claudia Diaz. On the anonymity of peer-to-peer network anonymity schemes used by cryptocurrencies. *arXiv preprint arXiv:2201.11860*, 2022. doi:10.48550/arXiv.2201.11860.
- 20 Ruisheng Shi, Yulian Ge, Lina Lan, Zhiyuan Peng, Shenwen Lin, and Lin Li. Deanonymizing transactions originating from monero tor hidden service nodes. In *Companion Proceedings of the ACM Web Conference 2024*, pages 678–681, 2024. doi:10.1145/3589335.3651487.

- 21 Ruisheng Shi, Zhiyuan Peng, Lina Lan, Yulian Ge, Peng Liu, Qin Wang, and Juan Wang. Eclipse attacks on monero's peer-to-peer network. In *32nd Annual Network and Distributed System Security Symposium, NDSS 2025, San Diego, California, USA, February 24-28, 2025*. The Internet Society, 2025. URL: <https://www.ndss-symposium.org/ndss-paper/eclipse-attacks-on-moneros-peer-to-peer-network/>.
- 22 Muoi Tran, Inho Choi, Gi Jun Moon, Anh V. Vu, and Min Suk Kang. A stealthier partitioning attack against bitcoin peer-to-peer network. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 894–909, 2020. doi:10.1109/SP40000.2020.00027.
- 23 Liang Wang and Jussi Kangasharju. Real-world sybil attacks in bittorrent mainline dht. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 826–832. IEEE, 2012. doi:10.1109/GLOCOM.2012.6503215.
- 24 Wireshark Foundation. Wireshark 4.4.7. Network protocol analyzer. URL: <https://www.wireshark.org/>.
- 25 Philipp Zabka, Klaus-T. Förster, Christian Decker, and Stefan Schmid. A centrality analysis of the lightning network. *Telecommunications Policy*, 48(2):102696, 2024. doi:10.1016/j.telpol.2023.102696.