Routing Attacks on Cryptocurrency Mining Pools

Muoi Tran
ETH Zurich

Theo von Arx
ETH Zurich

Laurent Vanbever
ETH Zurich

***Abstract*—Mining pools have been the driving force for ensuring the security of multiple proof-of-work (PoW) cryptocurrencies. Under the *de facto* protocol Stratum, pools allow miners to collaborate, discover new blocks, and earn rewards collectively. Recently, the blockchain community has been promoting the adoption of a more secure Stratum protocol known as Stratum V2. In this paper, we introduce EROSION, a novel network-level attack that applies to *both* Stratum and Stratum V2 protocols. The essence of the EROSION attack lies in its ability to disrupt connections between miners and a targeted mining pool, significantly impairing the miners' contributed PoWs and reducing the victim's mining power. We also discover a vulnerability in the Stratum V2 protocol that allows the adversary to persistently disrupt a connection by tampering with a *single* packet, thus enhancing the attack's stealthiness. Our survey shows that the EROSION adversary can readily execute attacks against a significant majority (e.g., 91%) of mining pools across the top ten cryptocurrencies. We also observe an extreme mining centralization that enables EROSION adversaries to simultaneously target multiple pools and cryptocurrencies. Furthermore, our focused evaluation of pooled mining in Bitcoin reveals that thousands of different adversaries can gain control over the majority of Bitcoin mining power, with one potentially malicious Autonomous System capable of taking down 96% of the total mining power.**

1. Introduction

Bitcoin [46] and various other cryptocurrencies leverage proof-of-work (PoW) consensus algorithms as the cornerstone of their blockchain systems for maintaining global transaction records. Expanding a PoW blockchain involves specialized participants, known as miners, engaging in the mining process, i.e., adding random data to a new block until its hash meets predefined criteria, typically requiring a specific number of leading zeros. Once miners discover new blocks, they broadcast these blocks to other participants via a peer-to-peer (P2P) network as evidence of completed work and earn rewards of newly generated cryptocurrencies.

As blockchain systems oversee increasing stakes, reaching into the billions of US dollars, they have increasingly become attractive targets for attacks in recent years. Notably, several attacks [7], [34], [63], [64], [66], [75], [76] have demonstrated that the P2P networks, notably of Bitcoin, can be partitioned such that nodes cannot exchange blockchain data (e.g., transactions, blocks) with nodes in other partitions. In these so-called partitioning attacks, the adversaries typically target mining nodes, wasting their mined blocks and corresponding rewards by preventing them from synchronizing with the latest blockchain states. Moreover, by disrupting the mining power of a targeted blockchain system, these attacks render its consensus unreliable and enable severe follow-up exploits, e.g., selfish mining [22] and the 51% attack [46]. The adversaries may also be motivated to attack a cryptocurrency to tarnish its reputation, thereby capitalizing on alternative avenues such as short selling to generate profits.

Large-scale attacks against blockchain systems commonly involve network-level adversaries, such as malicious Autonomous Systems (ASes) [7], [66], [75], [76]. Particularly, these malicious ASes disrupt the naturally-intercepted P2P connections or deliberately attract the traffic destined to targeted P2P nodes by launching Border Gateway Protocol (BGP) hijacks against their prefixes [9], effectively targeting many P2P nodes simultaneously. Handling these network-level attacks is, unfortunately, challenging. When the adversaries disrupt the connections they already intercept, localizing the disruptions and attributing them to the perpetrators remains an open problem [33]. When requiring prefix hijacking, the adversaries can minimize the attack's exposure by strategically evading the BGP monitors [41]. Even when BGP hijacks are traced back to the adversaries, there is no known legal consequence against them, possibly due to the high frequency of hijacks in practice (e.g., created by network misconfigurations) [16]. For these reasons, network-level attacks have been an alarming threat in blockchain systems and beyond (e.g., Federated Learning [69], Tor [72], and Public Key Infrastructure [10]). They have also been exploited outside of the academic realm (e.g., BGP hijacks against KLAYSwap [70], MyEtherWallet [14], and several Bitcoin mining pools [74]).

However, prior network attacks against blockchains have several limitations when targeting the P2P nodes. First, the adversaries can only target nodes with a public IP address (e.g., Bitcoin reachable nodes), which typically account for the minority of the P2P networks. The attack surface has shrunk further in recent years, as P2P nodes increasingly hide their addresses, e.g., almost two-thirds of the reachable Bitcoin nodes are now only accessible via Tor [79]. Second, due to the distributed nature of blockchain P2P networks, the adversaries often need huge Internet topological advantages (e.g., being Tier-1 or large Tier-2 ASes [75], [76]) or a

large number of prefixes hijacked simultaneously (e.g., 80 prefixes hosting 50% of Bitcoin nodes [66]). Therefore, only a restricted number of ASes can launch these attacks, which may not last long enough [32]. Third, no reliable methodology exists to identify P2P nodes with mining capabilities, which are the primary attack targets. Particularly, side-channel attacks for identifying mining nodes [18], [40] are already patched, and assuming mining nodes propagate their blocks earlier than other P2P nodes is shown to be error-prone [8]. In short, targeting P2P nodes for disrupting blockchain protocols appears to be impractical.

Instead, this paper identifies *mining pools* as attractive targets for network attacks against (PoW) blockchains. Indeed, pooled mining has become a crucial foundation for many blockchains nowadays, accounting for nearly all of their new blocks. For instance, known mining pools generate approximately 99% of recent Bitcoin blocks (see Section 4.2). At a high level, mining pools are collaborative groups where miners combine their hash rate to increase their chances of mining blocks and earning rewards more frequently than if mined individually. By default, most mining pools use the Stratum protocol [13], in which pool operators (or pools in short) coordinate easier mining tasks among participated miners. While mining pools do not disclose their P2P nodes' IP addresses, they commonly maintain a few Stratum servers that are publicly accessible so that miners across the globe can easily connect to them. For the same reason, mining pools usually attach their names to the blocks they found, revealing their miners' collective hash power to the adversaries. Furthermore, early measurement studies on Bitcoin [39], [65] show that the mining power was concentrated into a couple of mining pools, suggesting that attacking today's mining pools may require less resources than attacking P2P nodes (e.g., fewer number prefixes to be hijacked).

To that end, this paper presents the EROSION[1] attack that allows a network adversary (i.e., having the same attack capabilities as previous work [7], [59], [66], [75], [76]) to slash the hash power of one or more mining pools. At a high level, the EROSION attack involves a network adversary identifying the Stratum servers of a targeted mining pool, intercepting mining connections of those servers using known techniques such as BGP hijacking, and then impairing the miners' works submitted via those connections. We illustrate an example of the EROSION attack in Figure 1. Instead of targeting the victim's P2P nodes, the EROSION adversary directly disrupts the connections between the victim's pool servers (e.g., hosted in AS V) and its participating miners; see the dashed red lines. In this example, the malicious AS wastes the miners' hash power by manipulating packets on the connections that she already intercepts naturally (e.g., from miners in AS A and AS C) or the connections that she attracts after executing a BGP hijacking attack (e.g., from miners in AS B).

The EROSION attacks apply to both the mainstream version of the Stratum protocol [13] (or Stratum V1 in the rest of this paper) as well as the imminent Stratum V2 protocol [57]. Over the last decade, most mining pools have defaulted to utilizing the Stratum V1 protocol despite its lack of encryption in pool-miner communication. This inherent vulnerability of the Stratum V1 protocol renders mining pools susceptible to network attacks (e.g., [59]) and was the primary motivation to replace it with Stratum V2. To disrupt a mining pool running the Stratum V1 protocol, an adversary drops all packets in the intercepted pool-miner connections. Regarding the Stratum V2 protocol, we discover a vulnerability in its cryptography protocol specification that allows the EROSION adversary to cause the same persistent disruption by tampering with a *single packet*. Subsequently, this new vulnerability enables the EROSION attack to be stealthier, as packets are not dropped, and more sustainable, as maintaining traffic interception is not needed.

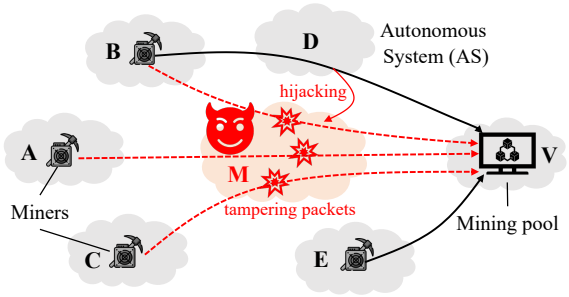


Figure 1: High-level overview of the EROSION attack. The malicious AS M wastes the hash power of a victim pool by tampering with the packets on the connections that she naturally intercepts (e.g., from A and C). To disrupt even more hash power of the victim, the adversary may hijack more mining connections (e.g., from B).

The EROSION attack proves to be highly destructive to the pooled mining landscape despite its simplicity. Our extensive study on cryptocurrency mining pools reveals that an alarming 91% (i.e., 50 out of 55) of the pools across the top ten cryptocurrencies are susceptible to the EROSION attack. Moreover, we observe a significant concentration of hash power within a small number of mining pools and the corresponding ASes hosting them. For instance, a single AS hosts eight distinct mining pools, accounting for over 50% of the hash rate in four cryptocurrencies. This multi-level centralization sets the stage for a large-scale attack, where the EROSION adversary can simultaneously disrupt multiple mining pools and cryptocurrencies. During our extensive evaluation of the attack effectiveness on Bitcoin mining pools, we discovered that there is almost always at least one potentially malicious AS capable of wasting over half of each pool's hash power, while the majority of pools can be destroyed entirely by almost all ASes. Furthermore, we identified over 1,300 different ASes that can individually control the majority of Bitcoin's hash power by employing BGP hijacks against mining pools. Even more concerning, a potentially malicious AS can disrupt up to 96% of the entire Bitcoin network's hash rate!

1. In geology, *erosion* is the process of wearing away stratum.

In summary, we claim the following contributions:

- We present the EROSION attack that enables network adversaries to waste the hash power of cryptocurrency mining pools. This attack applies to the default mining protocol, namely Stratum, as well as its next generation, namely Stratum V2. We particularly discover a vulnerability in the Stratum V2 protocol, enabling stealthier and more sustainable attacks.
- We find most mining pools are vulnerable to EROSION attacks after surveying 55 major pools across the top ten cryptocurrencies. Our in-depth evaluation of the attack on Bitcoin mining pools reveals thousands of different ASes that can disrupt the majority of the total hash power with an exceptional case in which a single AS can waste 96% of it.
- We discuss and evaluate several short-term and long-term defenses to the EROSION attack, some of which are readily deployable.

2. Background

To provide context for our routing attacks against mining pools, we briefly overview pooled mining in proof-of-work (PoW) blockchains. We start by introducing blockchains, particularly those utilizing PoW consensus, and discussing the concept of mining pools (§2.1). Additionally, we provide an overview of the widely used Stratum protocols for pooled mining (§2.2).

2.1. Pooled Mining in PoW Blockchains

Blockchain is a distributed ledger technology that records cryptocurrency transactions (e.g., Bitcoin [46]) across multiple nodes in a P2P network. It consists of cryptographically linked blocks, with each block containing confirmed transactions. Particular participants, often known as miners, are periodically selected to extend the blockchain, earning a rewarding combination of newly-created cryptocurrency and transaction fees.

In cryptocurrencies like Bitcoin, miners are selected as block creators under the PoW consensus. In particular, they compete to solve a cryptographic puzzle by repeatedly hashing the new block's data (e.g., using SHA-256). The puzzle requires finding a hash with specific properties, such as a certain number of leading zeros indicating its difficulty. Moreover, the puzzle's difficulty adjusts periodically to maintain a consistent block generation rate (e.g., roughly 10 minutes in Bitcoin). This difficulty is often measured in terms of hash rate, representing the blockchain network's computational power, i.e., the number of hashes computed per second. When a miner finds a valid solution (i.e., a new block), they broadcast it to the P2P network as proof of work. Other participants then verify the new block and add it to their copies of the blockchain if it is deemed valid.

The concept of pooled mining emerges along with the increasing hash rate of miners and a consistent block time,

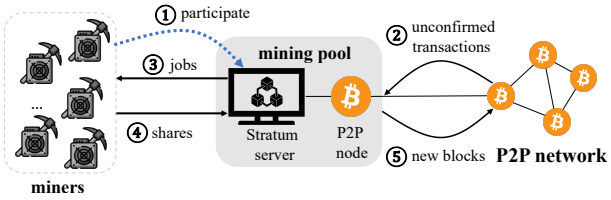


Figure 2: An illustration of pooled mining under Stratum protocols [13], [57], [62]. A mining pool gathers transactions into a block template, assigns jobs to participating miners, and verifies their submitted shares. Valid shares probabilistically become new blocks that are then broadcasted to the network.

causing the difficulty of generating new blocks to significantly escalate, e.g., creating a new Bitcoin block has become 16 million times harder over the last decade [62]. Consequently, miners may experience long periods without finding a block on their own, with the most powerful mining device taking approximately eight years to mine a single bitcoin [12]. To reduce such high reward variance, miners participate in mining pools, collaborating to create new blocks and sharing the rewards. Miners within the pool compute hashes to solve sub-puzzles and submit *shares*, which are partial solutions, to the pool. When a miner's share leads to a valid block solution, all participants receive split block rewards based on their contribution to the pool's hash rate or the number of shares they submitted. Pooled mining helps to reduce reward variance since the pool discovers blocks more frequently than individual miners.

2.2. Stratum Protocols

Stratum V1 [13] is a widely used pooled mining protocol for PoW-based cryptocurrencies designed to facilitate efficient communication between miners and their mining pools. We illustrate the workings of pooled mining under Stratum protocols in Figure 2. Specifically, the mining pool consists of one or more servers and a gateway node connecting to the P2P network. The mining pool typically openly accepts subscription connections from new participating miners; see step ①. Once the miner is authorized, the pool server sends a mining job to the miner, providing detailed information such as a block template consisting of unconfirmed transactions and the target difficulty for finding a valid block solution, see step ② and ③. Upon receiving the mining job, the miner starts performing the required computations. The miner, in turn, sends all solutions that satisfy the target difficulty back to the pool (step ④), which then verifies if they also fulfill the network difficulty to become new blocks (step ⑤). Though not part of the Stratum protocols, the pool server commonly adjusts the mining job's difficulty based on the miner's computational power (i.e., via the `VarDiff` algorithm). This practice ensures that the miner's share submissions align with the desired server load at the mining pool (e.g., one share received every few seconds).

The Stratum V1 protocol, however, was designed with little security in mind as it was initially used only for communication within a local network of miners (e.g., a mining farm). As the pools have started to accumulate hash rates from miners globally for better earnings, the unencrypted mining traffic traverses over the Internet, where malicious network adversaries can launch dangerous attacks (e.g., BiteCoin [59]). To address these security concerns, the blockchain community has recently developed Stratum V2, the next generation of the protocol [57], [62]. Stratum V2 incorporates encryption and authentication mechanisms, enhancing communication security between miners and mining pools. Thus, it prevents man-in-the-middle attacks and ensures that both parties can verify each other's identities.

3. Overview of the EROSION Attack

In this section, we present the overview of the EROSION attack. We first introduce the threat model of network adversaries considered in this paper (§3.1) and a few assumptions for clarifying the attack scope (§3.2). We then outline the main steps of the attack (§3.3).

3.1. Threat Model

We consider a network adversary that aims to cut down the hash rate of a victim pool that mines a targeted cryptocurrency. In particular, we consider the adversary controlling a single AS, called the malicious AS. This malicious AS can inspect and manipulate packets passing through its network (e.g., tampering, dropping, or delaying them). The adversary can further launch active routing attacks, e.g., hijacking an IP prefix via BGP manipulation. Previous attacks against cryptocurrencies (e.g., [7], [59], [66], [75]) also assume similar passive and active routing capabilities, which Internet Service Providers (ISPs), transit networks, or nation-state adversaries [38], [47] already own. Determined adversaries may also set up a new AS and buy transit from other ISPs at the expense of a few thousand dollars [73].

The main goal of the adversary is to slash all mining power of the targeted pool. At a larger scale, the adversary may also aim to reduce the network hash rate, i.e., the accumulated computational power of the entire blockchain network, by attacking one or more pools. Furthermore, the adversary can target multiple blockchains simultaneously.

3.2. Assumptions

We make a few realistic assumptions regarding the attack scope. First, we target only mining pools that produce new blocks for a proof-of-work (PoW) blockchain (e.g., listed in [43]). We exclude pools that operate under a different consensus (e.g., staking pools in Proof-of-Stake blockchains [25]) because they may not interact directly with their participating miners over the Internet.

Second, for simplicity, we assume the victim pool uses the Stratum V2 protocol [57]. The EROSION attack also applies to the Stratum V1 protocol following the same steps, albeit with a slight difference in manipulating intercepted packets. We consider the underused P2Pool mining protocol invulnerable to our attack, as it does not operate under the client-server paradigm but a separate P2P network of miners.

Third, the EROSION attack does not apply to private mining pools that hide their identities and restrict miners from joining them (i.e., commonly classified as unknown in several blockchain explorers [12]). As we will show in Section 4.2, those pools control only an insignificant portion of the network hash rate, and we thus ignore them.

Lastly, we assume only one adversary at any given time and leave a more comprehensive analysis of multiple EROSION adversaries for future work.

3.3. Main Attack Steps

At a high level, the EROSION adversary locates the Stratum servers of the targeted pool, intercepts the connections between the pool's miners and those servers, and then disrupts the miners' submissions of shares. We describe the three phases of the EROSION attack, illustrated in Figure 3, as follows.

[Step I] Identifying victim pool's servers (§4). In this reconnaissance step, the adversary aims to identify all Stratum servers of the targeted victim, along with their IP addresses, the prefixes, and the ASes hosting them. As illustrated in Figure 3, the adversary can leverage several sources of information to extract the victim's Stratum servers. While most servers are publicly accessible, some hide from non-participating miners, and some mining pools operate privately. We present in Section 4 that we can quickly learn most of today's Stratum servers and access them, including the hidden ones, suggesting that the EROSION attack is highly applicable.

[Step II] Intercepting mining traffic (§5). In this preparation step, the adversary aims to intercept as many connections between the victim's miners and the identified servers as possible. After learning the IP addresses of the pool servers, the adversary can already inspect the miner-pool connections that she naturally intercepts, such as the two right-most connections in Figure 3. To increase the connection interception, the adversary additionally launches BGP hijacking attacks against the prefixes hosting the pool servers (e.g., `1.2.3.0/24` in Figure 3) and subsequently attracts traffic in other connections. We show in Section 5 that a significant portion of prefixes hosting pool servers are not adequately protected by their ASes against BGP hijacks.

[Step III] Impairing miners' shares (§6). In this attack execution step, the adversary aims to waste the shares submitted by the miners in the intercepted connections. Figure 3 illustrates that the adversary can achieve this goal by tampering with a single packet in each connection, which renders the encapsulated share and all subsequent shares rejected by the pool servers. Section 6 describes our discovery of a new vulnerability in the Stratum V2 protocol that enables this exploit. We also demonstrate its feasibility in impairing miners' shares with controlled experiments in Section 6.

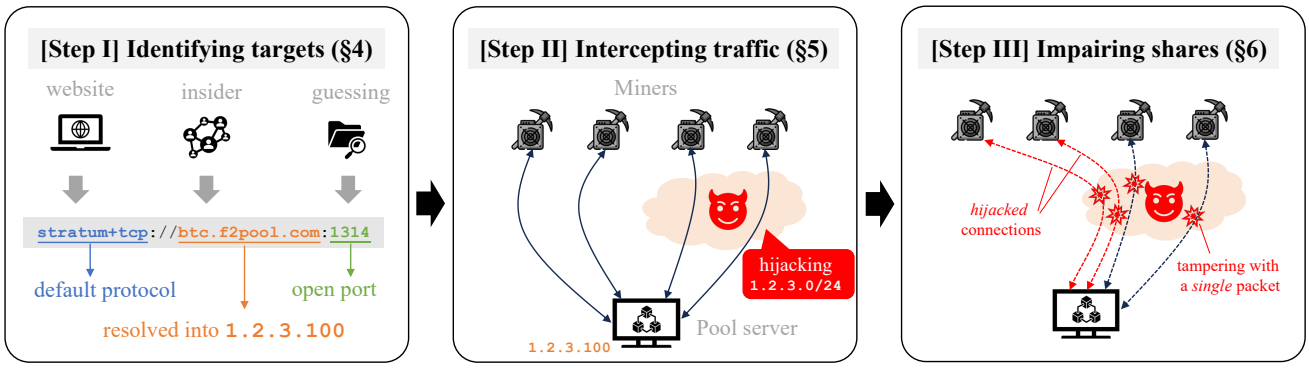


Figure 3: Overview of the EROSION attack. (I) The adversary first identifies the Stratum servers of the targeted mining pool. (II) The adversary then filters out mining communication from the traffic she intercepts (e.g., via hijacking the prefixes hosting the pool servers). (III) Finally, the adversary tampers a single packet in each intercepted mining connection.

4. Identifying the Victim's Pool Servers

In the reconnaissance step, the EROSION adversary identifies the Stratum servers of the victim mining pool. Since mining pools prioritize accumulating hash power for higher profits, they commonly configure their servers to be accessible to attract miners. Unfortunately, this practice enables adversaries to learn them with various methods (§4.1). We also show that the vast majority (e.g., 91%) mining pools have their servers exposed, making them vulnerable to the EROSION attack (§4.2).

4.1. Finding Addresses of Stratum Servers

The EROSION adversary can quickly identify the Stratum servers of a targeted mining pool, including the "hidden" ones. Notably, most mining pools publicly disclose the URLs and port numbers of the Stratum servers for each cryptocurrency they mine on their websites; see the example of F2Pool in [Step I] of Figure 3. Here, F2Pool uses the initial "btc" as the indicator for mining Bitcoin, and 1314 is the port opened for incoming connections from miners. A possible reason for publicly disclosing the server URLs is to ease the joining process for new miners.

When the servers' URLs are not publicly available, the adversary can rent mining devices from third-party services (e.g., NiceHash [51], Mining Rig Rentals [44]) for a short amount of time. Then, the adversary can join the victim pool and retrieve the servers' URLs as an insider.

When joining the victim pool as a miner is costly (e.g., requiring an industrial-scale mining farm), the adversary's last resort is guessing the URLs by combining the targeted cryptocurrency's initials (e.g., "btc" for Bitcoin) and the domain name of the victim pool (e.g., "foundryusapool.com"). The attacker then can identify the server's open ports using a port scanning tool (e.g., nMap [37], ZMap [19]). To ensure that the IP addresses and the open ports of the Stratum servers are correct, the adversary may want to verify the inferred addresses are indeed the victim servers. The attacker can quickly test that by sending a Stratum message (e.g., `mining.subscribe` in V1 or `SetupConnection` in V2) to those addresses or using a pool performance testing tool like Stratum Ping [2].

Once the adversary learns the URLs of the victim pool, she resolves the domain names to IP addresses. Using an IP address mapping (e.g., RouteViews dataset [53]), the attacker can further convert those IP addresses into corresponding prefixes and ASes for subsequent attack steps.

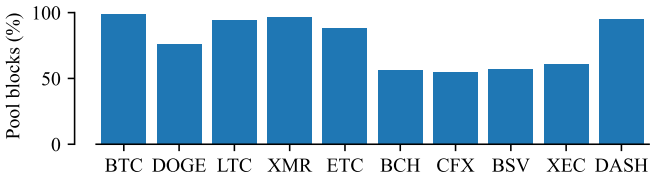
4.2. How Vulnerable Are Today's Mining Pools?

We consider a mining pool vulnerable to the EROSION attack if it mines a PoW-based cryptocurrency under Stratum protocols and has accessible pool servers. In the following, we describe our methodology for surveying major mining pools and present our findings.

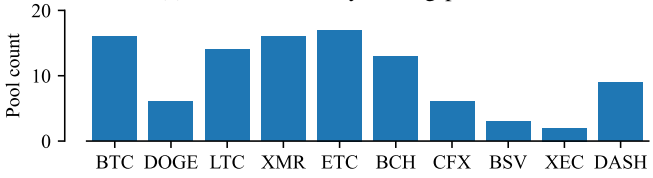
Methodology. We take a snapshot of all cryptocurrencies that allow PoW-based pooled mining reported by MiningPoolStats [43] on 20 May 2023. In this evaluation, we focus on the top ten cryptocurrencies with the most significant market capitalization, although the EROSION attack also applies to others. Particularly, we select Bitcoin (BTC), Dogecoin (DOGE), Litecoin (LTC), Monero (XMR), Ethereum Classic (ETC), Bitcoin Cash (BCH), Conflux (CFX), Bitcoin SV (BSV), eCash (XEC), and Dash (DASH). Note that the EROSION attack may apply to hundreds of other PoW cryptocurrencies beyond this evaluation because it does not explicitly target any blockchain system or consensus algorithm.

For each selected cryptocurrency, we dissect the latest 1,000 blocks to find the signatures of the mining pools usually encoded in the block data and assign these pools as the block creators accordingly.[2] We also assign the hash rate distribution proportionally to the number of blocks each mining pool finds. For each identified mining pool, we find the IP addresses of its Stratum servers by looking for its pub-

2. The duration for creating 1,000 blocks, shown by MiningPoolStats by default, ranges from eight minutes (CFX) to 10 days (BCH).



(a) Blocks mined by mining pools



(b) Number of active mining pools

Figure 4: (4a): The vast majority (55%–99%) of new blocks are created by mining pools. (4b): Bigger cryptocurrencies are more diversified in terms of active mining pools.

licly disclosed URLs or guessing them (cf., Section 4.1).[3] Finally, we associate each server address to the most specific IP prefix and the corresponding AS hosting it using the routing information bases collected by RouteViews in May 2023 [53]. Detailed information about our studied mining pools can be found in Appendix A.

Findings. We first confirm that pooled mining is essential for creating new blocks in PoW blockchains. In particular, Figure 4a shows that *known* mining pools find the vast majority of new blocks in the top ten cryptocurrencies (e.g., 99% in Bitcoin). We note that in some cryptocurrencies with lower ratios of pool-mined blocks (e.g., 55% in BCH), many block creators do not reveal themselves, and thus, we cannot confirm if they are mining pools. Figure 4b presents the number of known mining pools for each cryptocurrency. Mining pools are more diversified in the first five cryptocurrencies; e.g., about 16 pools are actively extending the blockchains. On the contrary, less than three pools are available for smaller cryptocurrencies, suggesting an increasing centralization for smaller blockchains. We also observe that pools often mine different cryptocurrencies. Out of 102 pools listed across all the pools, only 55 are unique, each mining 1.85 cryptocurrencies, on average. Some pools also offer merged mining, i.e., re-using PoWs for different cryptocurrencies having the same algorithm.

Second, we find that most of the mining pools in our study are vulnerable to the EROSION attack. Table 1 shows that 50 out of 55 mining pools have accessible Stratum servers. We also find five different mining pools that are not susceptible, as they run in private or under the decentralized P2Pool protocol [42]. Among 50 mining pools with accessible servers, we confirm that 44 pools openly accept new miners by responding to the `mining.subscribe` messages sent from the Stratum Ping tool [2]. The remaining six pools did not respond to our specific pings because they use different messages to communicate with miners running

3. Unlike those who possess mining power, we are unable to participate in the pools that conceal their servers and then, to learn their URLs.

Vulnerable?		No. of pools
Yes	Servers are publicly accessible	48
	Servers are hidden but accessible	2
No	Pools are private	2
	Pools run the P2Pool protocol	3
Total		55

TABLE 1: The EROSION attack applies to 91% of pools.

a customized Stratum client to mine Monero (XMR). We thus still consider them vulnerable to the EROSION attack.

Third, we observe a high concentration of hash rate in a few mining pools and the ASes and the prefixes hosting their Stratum servers. In Figure 5, we show the number of pools, ASes, and prefixes that accumulate certain portions of the total hash rate of each cryptocurrency. Note that we only consider the vulnerable pools when reporting the ASes and prefixes hosting them.[4] We see that in each cryptocurrency, the largest mining pool always possesses at least one-fourth of the total hash rate. Although there are a dozen active mining pools, 33% of the hash rate can be accumulated by only the top 2 pools across all cryptocurrencies. Also, each cryptocurrency's majority hash rate is controlled by only 2–6 mining pools. Figure 5 also points out that in 5 out of 10 studied cryptocurrencies, the majority of hash rate can be controlled by only *one* individual AS. Moreover, attacking <10 prefixes hosting the pool servers is enough to disrupt at least half of the total hash rate in most cases.

The heavy centralization of mining pools, especially from a routing perspective, is particularly worrying in light of the EROSION attack. When a few ASes or prefixes host a lot of hash rate, malicious ASes may naturally intercept many miner-to-pool connections and spend less effort hijacking the pool servers. The EROSION attack can become even more disruptive when targeting an AS or a prefix hosting multiple pools that mine various cryptocurrencies. For example, solely disrupting the mining connections toward AS13335 is enough to take down eight mining pools and more than 50% hash rate across four cryptocurrencies.

5. Intercepting Mining Traffic

After learning the victim's pool servers, the adversary maximizes the interception of connections between these servers and the victim's miners. Often, the naturally intercepted traffic is insufficient for a severe attack unless the adversary controls the ASes hosting the targeted pool or large ASes with huge Internet topological advantages. Alternatively, the adversary can hijack the prefixes hosting the pool servers via BGP manipulation (§5.1). We show that the majority of prefixes hosting pool servers are improperly secured, allowing the adversaries to easily hijack them (§5.2).

4. The largest BSV pool that accumulates >50% hash rate is private.

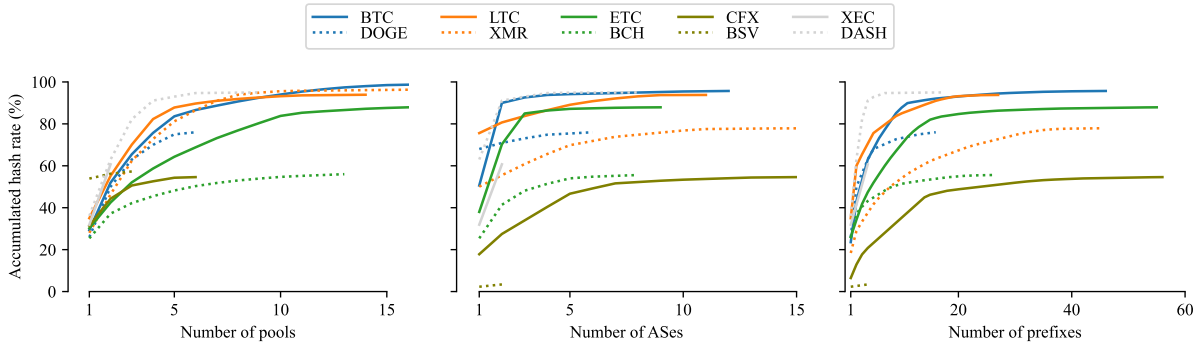


Figure 5: Attacking less than ten prefixes or less than three ASes is sufficient to disrupt at least 50% hash rate of almost all top ten cryptocurrencies.

5.1. Increasing Interception With BGP Hijacking

The EROSION adversary may naturally intercept some connections between the victim pool and its miners. To intercept more connections, the adversary can launch BGP hijacks [68] against the victim pool servers. In particular, the adversary inspects the AS V and prefix p hosting a targeted pool server before sending bogus BGP announcements from her malicious AS M to hijack its incoming traffic. Depending on whether V participates in the Resource Public Key Infrastructure (RPKI) and the length of p, the adversary follows one of four following cases.

(1) *V is RPKI-enabled and p is /24-long:* The adversary announces $[\{M, V\}\ p]$ to avoid being filtered out by RPKI-validated routers [28]. All traffic sources will choose between the legitimate AS path or the forged path that includes AS M. The forged path is likely longer than the legitimate one unless the traffic source is very close to the malicious AS.

(2) *V is RPKI-enabled and p is shorter than /24:* If AS V registers p with its exact length in a Route Origin Authorization (ROA) registry, the adversary follows the same procedure as in scenario (1). In case p is a sub-prefix of a shorter prefix, which is registered in ROAs along with a `maxLength` attribute, p is still vulnerable to a forged-origin sub-prefix hijack [29]. Particularly, the adversary announces $[\{M, V\}\ p']$, where p' a sub-prefix of p that covers the address of the targeted pool server. This bogus BGP message is accepted by all routers, including the RPKI-validated ones. The adversary thus attracts all mining traffic forwarded to the victim pool server in this scenario.

(3) *V is not RPKI-enabled and p is /24-long:* The adversary announces $[\{M\}\ p]$ to claim herself as the owner of p. Note that the adversary cannot advertise a prefix longer than /24 since most ASes do not propagate it [71]. As a result, all Internet routers will forward traffic to M or V, preferably to the closer one in terms of AS path length if both paths are valley-free [31].

(4) *V is not RPKI-enabled and p is shorter than /24:* The adversary announces $[\{M\}\ p']$, claiming she is the owner of a sub-prefix p' (e.g., /24) of p. This BGP announcement is propagated to the entire Internet, and in turn, all traffic destined for the victim server, including traffic from the pool's miners, is forwarded to the malicious AS M.

In any case, the adversary maintains an available route from M to V so that she can forward the (tampered) mining traffic back to the victim pool servers. To achieve this, the adversary can leverage BGP communities [11], AS path poisoning [56], or selective neighbor announcements [31] when advertising bogus BGP messages. Since disrupting a mining connection requires intercepting, tampering, and forwarding back to the victim only a single packet (see Section 6), such BGP interception attacks can be short-lived without affecting the effectiveness of the EROSION attack.

From the intercepted traffic, the adversary can quickly identify the communication between the pool servers and the miners. In particular, the adversary filters relevant packets by applying a filter matching the pool servers' IP addresses, port numbers, and the protocol being TCP. We note that the adversary may intercept the communication only in one direction because the Internet routing is known to be asymmetric [72], that is, the AS path from a miner to its pool differs from the AS path from the pool to that miner. However, this restriction does not affect the EROSION attack as it works independently of the intercepted direction(s) (see Section 6).

5.2. How Susceptible Are Pools to BGP Hijacking?

We study the ASes and prefixes hosting mining pools of top cryptocurrencies, focusing on finding the required strategies to hijack them. In the following, we present our categorization methodology and report our results.

Methodology. We use the same snapshot from the measurements of mining pools (cf. Section 4.2). Thus, for each of the top ten PoW cryptocurrencies, we have the list of vulnerable mining pools, their servers' addresses, and the

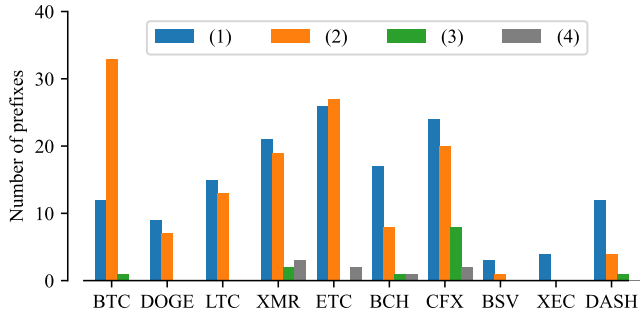


Figure 6: Case (1), (2), (3), (4) follow the categorization in Section 5.1. Most pool servers are hosted on RKPI-enabled ASes; see cases (1) and (2). Among them, 24%–72% of pool servers are hosted on insecure prefixes, see case (2).

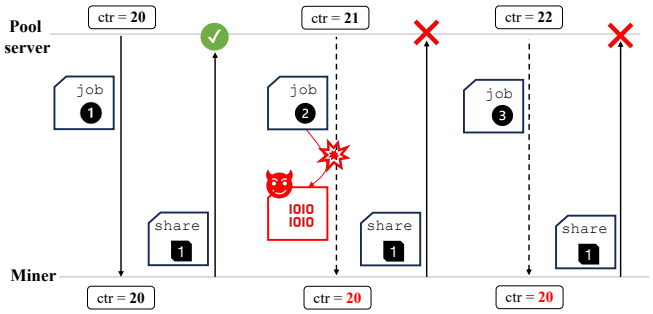


Figure 7: The adversary tampers the payload of a single packet to prevent the miner from decrypting new messages. The miner thus keeps working on outdated shares, which are rejected by the pool.

corresponding prefixes and ASes. To check if an AS enables RPKI-based protection for its prefixes (e.g., registering them in ROAs), we query it on the IRR Explorer website in May 2023 [35].

If a pool's AS does not participate in RPKI, the hijacking strategy will be either in case (3) or case (4) (cf. 5.1). We then check the length of the prefix, i.e., if it is /24-long, it belongs to group (3), and it belongs to group (4) otherwise.

If the pool AS participates in RPKI, we check the prefix length to see if it is /24-long, meaning it belongs to case (1). If the prefix is shorter than /24, we check its RPKI coverage with IRR Explorer, particularly testing if its length is registered with the same `MaxLength` attribute. In that case, the adversary must follow hijacking strategies in case (1). Otherwise, the prefix belongs to case (2).

Results. Figure 6 shows the number of prefixes hosting mining pools in each category. We first observe that most mining pools operate their servers on RPKI-enabled ASes (e.g., on cloud services) as the vast majority of prefixes belong to cases (1) and (2), see the blue and orange bars. However, we also see that there are in total more prefixes in group (2) than in group (1), meaning that the majority of prefixes are still susceptible to forged-origin sub-prefix hijacking attacks [29], despite having RPKI in place. Across the top ten cryptocurrencies, we report that there are 24%–72% of pool servers are hosted on such insecure prefixes. Our findings suggest that mining pools indeed consider protecting their servers against routing attacks (e.g., BGP hijacks); yet, they are not properly secured at the moment.

6. Impairing Miner's Shares

After intercepting the connections between a targeted pool and its miners, the EROSION adversary impairs the miners' shares submitted via those connections. Exploiting a new vulnerability that we have discovered, the adversary tampers with the intercepted packets, causing the pool to reject shares of affected miners (§6.1). Our experiments with the Stratum V2 protocol in a controlled environment demonstrate that tampering with a single packet indeed disrupts the entire mining connection (§6.2).

6.1. Tampering With Packets

We have discovered a vulnerability in the cryptography specifications of the Stratum V2 protocol. Particularly, the Stratum V2 protocol provides end-to-end encrypted communication (e.g., between the pool server and a miner) using the Noise Protocol Framework [55]. To establish a connection, two endpoints perform a handshake and agree upon a shared symmetric key. Subsequently, for each message, the sender encrypts it using this shared key and a changing cryptographic nonce. To ensure the nonce uniqueness, the Stratum V2 protocol implements nonce counters that increment on the sender side (respectively, the receiver side) after successfully encrypting (respectively, decrypting) a message. Note that nonce counters are uni-directional, meaning that each endpoint maintains separate counters for sending and receiving messages. This nonce synchronization allows decrypting the encrypted messages without the need for sending the nonce along. However, if a packet is corrupted in transmission, the receiver cannot decrypt the message and, thus, does not increment its nonce counter. Since the sender already incremented its counter, the two nonce counters are no longer synchronized. From our inspection of the sole available implementation of Stratum V2 [58][5], both the server and miner clients do not reset their TCP connection when this happens. Thus, the receiver cannot decrypt any subsequent messages from the sender. We present the root cause of this vulnerability in more detail in Appendix C.

The EROSION adversary exploits this vulnerability to persistently block the communication between the pool server and the miners. Specifically, in each intercepted pool-miner connection, the adversary tampers with a *single* packet (e.g., replacing its payload with random bytes of the same length). As a result, the nonce counters of the server and the miner are out of synchronization, preventing future messages from being decrypted. Therefore, the decryption failures persist even when the adversary no longer intercepts the mining traffic (e.g., when hijacked paths revert [68]).

5. The developers of this implementation also work on a closed-source mining firmware [54], potentially sharing the same vulnerability.

Regardless of the communication directions that the adversary tampers a packet with, the pool always rejects the miners' submitted shares. Particularly, if the tampered packet belongs to the miner-to-pool direction, the pool cannot decrypt any subsequent messages containing the shares (i.e., `SubmitShares`). If the tampered packet belongs to the pool-to-miner direction, the miners cannot decrypt messages containing the new assignments (i.e., `NewMiningJob`) and thus keep sending outdated shares that are rejected by the pool. We illustrate an example of tampering with a packet on the pool-to-miner direction in Figure 7. For simplicity, we omit the nonce counters for the miner-to-pool direction. Figure 7 shows that the miner successfully decrypts the message containing the job ❶, and the pool accepts its share for this job. Thereafter, the adversary corrupted the message containing the job ❷ with random bytes; see the red packet. Thus, the miner's nonce counter stops incrementing at 20 while the pool's counter increments to 21. Subsequently, the miner keeps producing shares on the job ❶, which are all rejected by the pool.

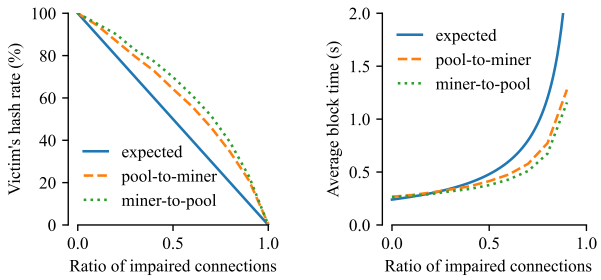
6.2. How Feasible is Tampering With Packets?

Here, we demonstrate the feasibility of the EROSION attack against Stratum mining pools. We first explain our setup for testing the attacks and then present our results.

Experiment setup. We launch controlled EROSION attacks against a mining pool running exclusively in a virtual environment for ethical reasons. Our experiment environment is a Bitcoin network consisting of one P2P node, one mining pool, and ten miners with equal hash power. We select the `regtest` mode for the Bitcoin network because this allows the minimum PoW difficulty, i.e., new blocks can be mined instantly, even with moderate CPUs.

We run the mining pool using the open-source Stratum V2 implementation [57]. This pool receives the latest blockchain information (e.g., the previous block hash) and sends the newly mined blocks to the blockchain via the P2P node. We connect ten miners to the mining pool and let them produce hashes using the CPU Miner program [26]. Since the CPU Miner program only allows Stratum V1 connections, we run translator proxies [57] to bridge the communication between the miners and the pool. To simulate an EROSION attack, we set up a man-in-the-middle proxy using `mitmproxy` [17] and tamper with a certain fraction of connections. In each targeted connection, we replace the payload of a specific TCP packet in either direction with random bytes of the same length as the original payload. After running our network for ten minutes, we count the number of collectively mined blocks.

We implement the above experiments in a Docker environment with separate containers for the P2P node, the mining pool, the translator proxies, and the CPU miners, respectively. The adversary runs in the same Docker container as the translator proxy. We ran the experiments on a system with the Ubuntu 22.04 OS, 24 cores running at 3.1 GHz each, and 32 GB of memory, and repeated every experiment ten times. With no attack, we observed that the ten running miners collectively produce around 2,500 blocks with this setup.



(a) Reduced hash rate. (b) Delayed blockchain growth.

Figure 8: With the increasing ratio of impaired connections, the hash rate decreases, and the block time decreases, independently of the direction of the tampered packets.

Results. We first confirm that no shares from affected miners are accepted after we tamper with the connections. We further observe that tampering with packets affects the attacked pool's hash rate and the blockchain growth speed. Figure 8a shows that the pool's hash rate decreases roughly linear with the ratio of attacked connections, both for tampering with packets on the pool-to-miner direction (see the orange dashed line) and the miner-to-pool direction (see green dotted line). We note slightly bigger hash rates than expected (see the blue solid line) due to more shares from non-affected miners being accepted when there are fewer competitors.

We also see in Figure 8b that wasting the hash rate of the pool increases the average time between two mined blocks proportionally, e.g., by impairing 50% of the connections, the average time between two mined blocks doubles. Again, the measured average block times are close to what we expected, regardless of the communication directions the adversary tampers packets with.

7. Evaluation

In this section, we focus on Bitcoin mining pools and evaluate the effectiveness of the EROSION attack against them with comprehensive and realistic simulations. We first describe our measurement methodology (§7.1). We then show that thousands of distinct EROSION adversaries can cause severe hash rate disruptions to individual mining pools (§7.2) as well as to the entire Bitcoin network (§7.3).

7.1. Data Collection and Methodology

We simulate the communications between Bitcoin mining pools and their miners and then calculate the hash rate intercepted or hijacked by all ASes on the Internet. As the inputs, the simulations require the ASes hosting the pool servers and miners, each miner's pool and hash rate, and the path between any two ASes.

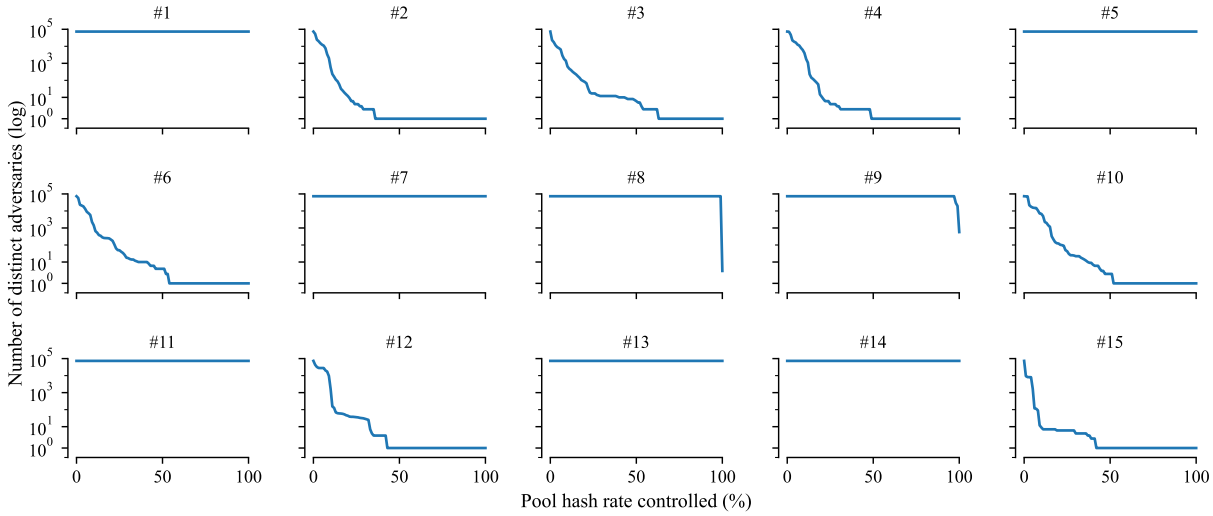


Figure 9: Number of distinct ASes that can disrupt a fraction of each mining pool's hash rate. At least one AS can destroy 50% of the hash rate of most pools. Almost all ASes can disrupt the entire mining power of eight out of 15 pools.

Regarding the ASes hosting the pools, we use the same snapshot as for studying vulnerable pools (cf. Section 4.2). We analyze the top 15 pools that account for 95.7% of the hash rate of the entire Bitcoin network. Their pool servers are hosted in 12 ASes, called the *pool ASes*.

Regarding the ASes hosting the miners, we use the latest representative Bitcoin mining map aggregated from major pools by CCAF [24]. To the best of our knowledge, this mining map is the *only* available data set about miners. Notably, it shows the estimated hash rate distribution of miners in each country and each state (or province) of the US (or China). For each location (e.g., a country or a state) in this map of miners, we distribute the hash rate equally to all ASes in that location using the iGDB data set [4]. After this step, we have an estimated hash rate distribution of $\approx 12,000$ ASes hosting miners (or *miner ASes* in short).

Knowing the hash rate distribution of each mining pool and all miner ASes, one can assign miners to a pool so that the sum of assigned miners' hash rate matches the pool's hash rate. This setting is essentially the subset sum problem, which is known to be NP-hard and too complex to brute-force when the number of elements (i.e., $\approx 12,000$ in this case) is enormous. We thus use a greedy-based heuristic to assign miners to pools; see the detailed algorithm in Appendix B. At a high level, we assign random miners to each mining pool ranked from smallest to biggest until the pool has no leftover hash rate.

Similar to existing works (e.g., [75]), we simulate the BGP-based routing decisions of all ASes to calculate the AS path between any two ASes. In particular, we use an AS-level topology composed of 70k ASes connected using customer-provider or peer-peer business relationships [15]. We then compute the AS path between any two ASes following the BGP decision process, namely: (1) preferring customers over peers and peers over providers for the next-hop AS; (2) preferring the shortest path; (3) preferring smaller numbers for on-path ASes in case of tie-breaking [30].

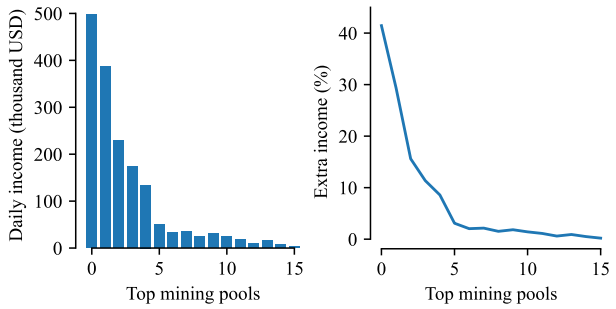
We also simulate BGP hijacks in which potentially malicious ASes hijack the prefixes hosting the pool servers. We consider an AS can hijack traffic from a miner to a pool if the path from the miner AS to that AS is preferred over the path from the miner to its pool AS, following the BGP policies mentioned previously. In case the malicious AS hijacks prefixes of an RPKI-enabled pool AS, the hijacked path from the miner must include the pool AS at the end. We re-use the RPKI status of ASes and the prefixes hosting pool servers queried in our previous study (cf. Section 5.2).

Lastly, we consider all ASes on the path from a miner to a pool can intercept that miner's hash rate. Similarly, we consider an AS to control a miner's hash rate if she can hijack the traffic destined for its pool.

7.2. Attacking Specific Mining Pools

We show the number of distinct ASes that can waste a fraction of each mining pool's hash rate in Figure 9. We first observe that the entire hash rate of the majority of pools (e.g., 8 out of 15) can be disrupted by almost all ASes on the Internet. The reason is that their pool servers are hosted in prefixes that are insecure against BGP hijackings. We also see that for the remaining pools, at least one malicious AS can reduce the pools' hash rates by more than 50%, and about 5–10 ASes can drop at least 10% of these pools' hash rates. We find that the servers of these 7 pools are hosted in more secure prefixes and ASes. Yet, the adversary can still hijack a non-negligible hash rate of miners closer to her malicious AS.

Implications. The adversary directly sabotages the victim's profit by wasting the hash rate of a targeted mining pool. Specifically, mining pools' primary income source comes from charging a fee from their miners in the form of commissions (e.g., 1%–5% rewards granted by found



(a) Potential profit loss. (b) Relative gain.

Figure 10: Disrupting a mining pool sabotages its income and grants extra income to other pools. (10a): A Bitcoin mining pool may lose thousands of USD daily for every percentage point of the hash rate lost. (10b): All mining pools (except the victim) receive 41% more revenue if the biggest pool is attacked.

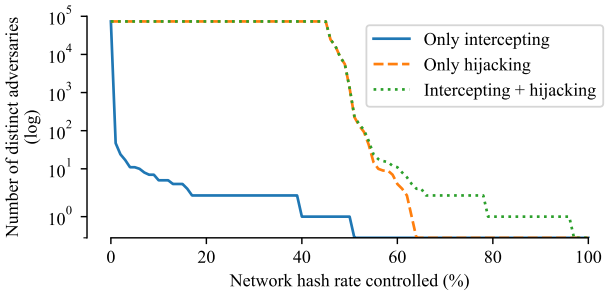


Figure 11: Number of distinct ASes that can disrupt a fraction of the network hash rate. There is one AS (respectively $\approx$ 1300 different ASes) that can waste 50% of the network hash rate by dropping only the naturally intercepted traffic (respectively only the hijacked traffic). Worse, AS13335 can disrupt *almost the entire* Bitcoin hash rate by dropping both intercepted and hijacked traffic.

blocks). To understand how much losing hash rate affects mining pools, we estimate the daily income of top Bitcoin pools listed in Appendix A. More specifically, we split the total mining revenue (about 57M USD/day as of this writing [12]) to pools based on their hash rate distribution (cf. Section 4.2) and then multiply it with a fixed commission rate of 3%. The estimated income of mining pools in Figure 10a shows that they can earn up to five hundred thousand dollars daily from the commissions alone, meaning that wasting every percentage point of their hash rate would strip a few thousand USD per day of their income.

On the other hand, mining pools other than the victim may gain extra revenue. Indeed, when the victim pool loses some hash rate, its share in the global hash rate decreases, and so does its relative revenue. This lost revenue is then re-distributed to other pools based on their hash rate distribution. We note that the difficulty of mining new blocks will eventually decrease as well. For instance, Bitcoin reduces the mining difficulty if miners take longer than 10 minutes on average to mine the last 2016 blocks. Yet, the relative revenue of each mining pool does not change because it is based on the hash rate distribution. We show the relative gain of all mining pools (except the victim) when each mining pool is attacked in Figure 10b. We observe that wasting the hash rate of bigger pools is more profitable than attacking smaller ones, e.g., the extra income can be up to 41% when disrupting the biggest pool.

Regarding the victim pool, detecting the EROSION attack is challenging because its miners' hash rate can fluctuate in normal circumstances, e.g., when the miners switch to another pool or lie about their actual power. Indeed, the victim pool's miners may leave for another pool as soon as their rewards become noticeably inadequate, also fulfilling the adversary's goal of slashing the victim pool's mining power. Furthermore, the above implications suggest that a malicious mining pool can launch the EROSION attack against its competitors, effectively gaining extra profits as well as pulling their miners.

7.3. Attacking The Entire Network

We find that the EROSION attack launched by a single malicious AS can be disruptive to the *entire* network hash rate. More specifically, we show the number of different ASes that can disrupt a fraction of the entire Bitcoin network hash rate via only intercepting, only hijacking, or both strategies in Figure 11. By dropping only the naturally intercepted hash rate, five ASes can independently reduce the network hash rate by 10%. There also exists an AS that naturally controls the majority of the Bitcoin hash rate; see the solid blue line. Figure 11 also shows that hijacking the pool servers enables many more malicious ASes to control more significant fractions of the network hash rate. We find that $\approx 45\%$ of Bitcoin hash rate can be hijacked by *any* ASes in the Internet because the corresponding pool servers are hosted in unprotected or improperly protected prefixes. This practice allows more than 1300 different ASes to control the majority of hash rate after launching hijacking attacks; see the orange dashed line. Moreover, when combining both the naturally intercepted and hijacked traffic of all ASes, we notice that one particular AS (i.e., AS13335) can potentially destroy up to *96% of the Bitcoin network hash rate*. We further observe that two (respectively 11) different ASes can disrupt 70% (respectively 60%) of the total hash rate; see the green dotted line.

Implications. By wasting a fraction of the global hash power, an adversary can launch consensus attacks, such as the 51% attack [46] and the selfish mining attack [22], at a lower cost than initially required. For example, the EROSION attack enables adversaries with a smaller hash rate (e.g., 25%) to launch successful 51% attacks. To do so, the adversary creates new blocks that compete with latest blocks on the blockchain (i.e., creating a fork) and disrupts 50% of the network hash rate. Since the adversary controls the majority of the remaining hash power, the adversarial blocks will eventually be included in the main chain. Such

a consensus attack enables several follow-up attacks, such as double-spending and censoring transactions [75].

When the EROSION adversaries aim to disrupt the entire network hash rate, the attack effectiveness may fluctuate because miners may switch to another pool after a specific duration of losing rewards. However, adversaries may still intercept the connection between these miners and their new pools, causing them to keep switching pools until they receive rewards. Also, this process can be sluggish in practice because the typical reward payout duration is at least several hours and can be up to a week [52]. In the meantime, adversaries may already damage the entire network hash rate severely.

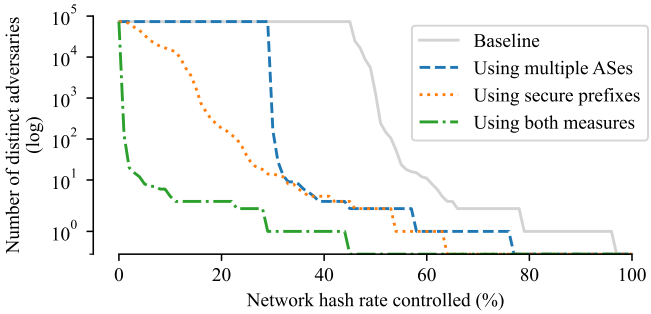


Figure 12: When using multiple ASes and exclusively secure prefixes, no adversary controls the majority of the hash rate.

8. Countermeasures

In this section, we discuss how to fix the discovered vulnerability in the Stratum V2 protocol (§8.1). Once this vulnerability is patched, the EROSION adversary can still drop all intercepted packets. Thus, we also propose several countermeasures, including readily deployable measures (§8.2) and long-term suggestions for pooled mining to be robust against network attacks like EROSION (§8.3).

8.1. Fixing the Stratum's Vulnerability

To handle our discovered vulnerability (cf. Section 6.1), the clients should raise a warning and restart the connection, similar to the well-studied and widely used TLS 1.3 [60]. As a result of our vulnerability disclosure, Stratum V2 clients now reset their connections upon decryption failures [1].

8.2. Short-term Measures

Better hosting of the pool servers. Mining pools should host their servers in multiple ASes and distribute miners across them so that no malicious AS can intercept a significant fraction of their hash rate. Note that this measure differs from a common practice in which miners from different regions would connect to the closest server provided by their pool because the servers are still hosted in the same AS. Towards a more distributed global hash rate, mining pools should also refrain from using exclusively an AS that hosts servers of many other pools.

Mining pools may prioritize hosting servers in an AS that participates in RPKI to minimize the mining traffic hijacked by EROSION adversaries. More importantly, they should also request the hosting AS to register in ROAs the prefix hosting pool servers with its exact length (cf. Section 5). Alternatively, hosting the servers in /24 prefixes also significantly reduces the effectiveness of the hijacking attacks [6].

We extend the experiments presented in Section 7.3 and evaluate the effectiveness of hosting pool servers in multiple ASes, in properly secured prefixes, and a combination of both. Particularly, we simulate pools hosting their servers in all 15 available pool ASes and their miners can connect to any of them, preferaly the closer ones with RPKI enabled. To enable hosting in secure prefixes, we simulate all pool ASes are RPKI-enabled, and the prefixes hosting pool servers are /24-long. We also test a combination of both measures, meaning the miners can connect to the closest pool AS with servers hosting in secure prefixes. Figure 12 depicts the effectiveness of hosting pool servers following these three options. We observe that individual measures reduce the number of adversaries that can control more than 50% of the hash rate from 1300 to less than 10 (see the blue dashed and the orange dotted line). We also see that when combining these two measures, only one AS can disrupt 30% of the hash rate while no adversary can control the hash rate's majority.

Hiding pool servers. Since the EROSION adversaries require addresses of pool servers for traffic filtering, hiding them would render the attack infeasible. For example, pools can run onion services and accept mining connections via the Tor anonymity network. A caveat is that the pool-miner communication may experience substantial delay, and thus, more submitted shares may become stale.

Hardening communication for mining. To reduce the attack risk from on-path ASes, miners can run a proxy that aggregates their shares in the same AS hosting the pool servers. Because communication within an AS is not routed by BGP, no EROSION adversary can hijack the connection between the pool and such a proxy. Moreover, when an attack happens, the victim pool and miners can immediately identify the perpetrator because the hosting AS handles their mining connections exclusively.

8.3. Long-term Measures

Decentralized mining. While highly efficient, the client-server paradigm of Stratum protocols makes them prone to denial-of-service attacks like EROSION. On the other hand, the less efficient decentralized mining protocols (e.g., P2Pool [42]) render such attacks obsolete because attacking a miner requires disrupting all of its P2P connections instead of a single connection with its pool server. Therefore, a long-term solution to the EROSION attack would be to embrace a decentralized mining protocol.

Routing-aware mining. To avoid specific ASes on the communication path, miners may incorporate AS-awareness when selecting pools or pool servers. However, additional protocol-level countermeasures may be required to compensate for its inaccuracies [76]. Mining pools can also collaborate with their miners to reveal the routing concentration of on-path ASes. If needed, pools can spawn more servers in new ASes (cf. Section 8.2).

Redundant mining. Mining pools can save miners' shares from being wasted by allowing them to be resubmitted to multiple servers in different locations simultaneously. However, this measure may require a more suitable data structure for less intensive pooled communication.

9. Related Works

We consider prior works related to whether they have the same target of pooled mining (§9.1) or the same capabilities of network adversaries (§9.2). We also review previous studies on the mining centralization in blockchains that are relevant to our studies in this paper (§9.3).

9.1. Security of Pooled Mining

Closest to our work is the BiteCoin attack that allows network adversaries to steal rewards of pooled miners [59]. Ahmed et al. propose a poisoning attack that exploits the lack of encryption in the Stratum V1 protocol to ban targeted pooled miners [3]. Unlike these attacks, the EROSION attack also applies to the Stratum V2 protocol.

The PoW mining model has been investigated thoroughly in several selfish mining and block withholding attacks [22], [23], [45], [49], [50], [61], [67]. A victim mining pool can also be targeted with an infiltration attack from competing pools [21]. The EROSION attack is orthogonal to these attacks and can be used to lower their cost.

The publicly accessible pool servers are frequently targeted by distributed denial-of-service (DDoS) attacks [77]. Johnson et al. use game theory to show that bigger mining pools are more incentivized to launch DDoS attacks than to improve their hash power [36]. The EROSION attack is similar in terms of aiming to disrupt the pool servers, yet, our packet tampering strategy is stealthier.

Numerous recent works aim to partition the blockchain P2P networks so that a portion of the global hash rate is wasted [7], [34], [48], [63], [75], [76]. As discussed in Section 1, targeting P2P nodes is impractical. The EROSION attacks target the pool servers directly and, thus, do not have this limitation.

9.2. Routing Attacks on Cryptocurrencies

Apostolaki et al. demonstrate that a malicious AS can partition the Bitcoin P2P network using BGP hijacking [7]. A recent work by Saad et al. shows that P2P nodes of multiple cryptocurrencies can be partitioned simultaneously by a network adversary hijacking the same set of prefixes [66]. Unlike these attacks, the EROSION adversaries use BGP hijacks to target the mining pool servers directly, thus requiring hijacking significantly fewer prefixes.

Our attack is more related to the study by Ekparinya et al., in which Ethereum mining pools are tested against BGP hijacks followed by double spending attacks [20]. The EROSION attack, however, can be effective without complete isolation of the pool servers. We also thoroughly evaluate the effectiveness of BGP hijacking (e.g., by simulating RPKI), which, unfortunately, is missing in these prior works.

Without sending bogus BGP announcements, network adversaries can still attack various aspects of cryptocurrencies, such as deanonymization of Bitcoin transactions [5], delaying block propagation in the P2P network [7], or linking off-blockchain transactions [78]. We assume the same attack capabilities for the EROSION attack.

9.3. Studies on Mining Centralization

CoinScope, one of the earliest studies on Bitcoin centralization, reveals that $\approx$ 100 P2P nodes originated three-quarters of the mining power in 2015 [40]. Gencer et al. show in 2018 that the majority of hash power in Bitcoin and Ethereum was controlled by eight and five mining pools, respectively [27]. In 2020, the number of pools needed to control the majority of Bitcoin mining power decreased to four [39]. From the routing perspective, the mining pools that accounted for 65% of the Bitcoin hash rate were hosted in only 3 ASes in 2019 [65]. We extend these prior works and show a worrying increase of the mining centralization on all three levels (i.e., pools, ASes, and prefixes) and in ten different cryptocurrencies (cf. Section 7).

10. Ethical Considerations

This work does not raise any ethical issues. Our study on cryptocurrency mining pools uses the already publicly available data. We also test the attack feasibility exclusively in a controlled environment (e.g., `regtest` network) that does not interact with the live network. We solely evaluate the effectiveness of the EROSION attack using simulations. We disclosed the discovered vulnerability to the Stratum developers in December 2023, which resulted in an immediate patch deployed into the Stratum V2 official repository [1].

11. Conclusion

Ironically, the security of numerous supposedly decentralized blockchains depends heavily on mining pools that centralize global hash power to just a few servers. This paper introduces EROSION, a novel network attack that undermines the security of targeted mining pools and cryptocurrency blockchains by wasting their mining power. Our real-world data analysis reveals that this attack can target 50 out of the 55 most prominent mining pools across the top ten cryptocurrencies, potentially disrupting up to 96% of the total Bitcoin mining power. We hope this paper will encourage discussions on strengthening the soon-to-be default mining protocol, Stratum V2, and advancing more decentralized protocols.

References

[1] Fix network-helpers close connection on invalid messages. https://github.com/stratum-mining/stratum/commit/15b2c2, 2023.

[2] 2miners.com. Stratum ping. https://github.com/2miners/stratum-ping, 2023.

[3] Mohiuddin Ahmed, Jinpeng Wei, Yongge Wang, and Ehab Al-Shaer. A poisoning attack against cryptocurrency mining pools. In *Proc. ESORICS CBT*, 2018.

[4] Scott Anderson, Loqman Salamatian, Zachary S. Bischof, Alberto Dainotti, and Paul Barford. iGDB: Connecting the Physical and Logical Layers of the Internet. In *Proc. ACM IMC*, 2022.

[5] Maria Apostolaki, Cedric Maire, and Laurent Vanbever. Perimeter: A network-layer attack on the anonymity of cryptocurrencies. In *FC*, 2021.

[6] Maria Apostolaki, Gian Marti, Jan Müller, and Laurent Vanbever. SABRE: Protecting Bitcoin against Routing Attacks. In *Proc. NDSS*, 2019.

[7] Maria Apostolaki, Aviv Zohar, and Laurent Vanbever. Hijacking Bitcoin: Routing attacks on cryptocurrencies. In *Proc. IEEE S&P*, 2017.

[8] Seungjin Baek, Hocheol Nam, Yongwoo Oh, Muoi Tran, and Min Suk Kang. Short Paper: On the Claims of Weak Block Synchronization in Bitcoin. In *FC*, 2022.

[9] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A study of prefix hijacking and interception in the internet. *ACM SIGCOMM CCR*, 2007.

[10] Henry Birge-Lee, Yixin Sun, Anne Edmundson, Jennifer Rexford, and Prateek Mittal. Bamboozling certificate authorities with BGP. In *USENIX Security*, 2018.

[11] Henry Birge-Lee, Liang Wang, Jennifer Rexford, and Prateek Mittal. Sico: Surgical interception attacks by manipulating bgp communities. In *Proc. ACM CCS*, 2019.

[12] Braiins. Mining Insights. https://insights.braiins.com/en, 2023.

[13] Braiins. Stratum V1 Docs. https://braiins.com/stratum-v1/docs, 2023.

[14] Russell Brandom. Hackers emptied ethereum wallets by breaking the basic infrastructure of the internet, 2018.

[15] CAIDA. AS Relationships Dataset, 2023.

[16] Shinyoung Cho, Romain Fontugne, Kenjiro Cho, Alberto Dainotti, and Phillipa Gill. Bgp hijacking classification. In *IEEE TMA*, 2019.

[17] Aldo Cortesi, Maximilian Hils, Thomas Kriechbaumer, and contributors. mitmproxy: A free and open source interactive HTTPS proxy. https://mitmproxy.org/, 2010.

[18] Sergi Delgado-Segura, Surya Bakshi, Cristina Pérez-Solà, James Litton, Andrew Pachulski, Andrew Miller, and Bobby Bhattacharjee. TxProbe: Discovering Bitcoin's Network Topology Using Orphan Transactions. In *FC*, 2019.

[19] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *Proc. USENIX Security*, 2013.

[20] Parinya Ekparinya, Vincent Gramoli, and Guillaume Jourjon. Impact of Man-In-The-Middle Attacks on Ethereum. In *Proc. IEEE SRDS*, 2018.

[21] Ittay Eyal. The Miner's Dilemma. In *IEEE S&P*, 2015.

[22] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, 2018.

[23] Chen Feng and Jianyu Niu. Selfish mining in ethereum. In *Proc. IEEE ICDCS*, 2019.

[24] Cambridge Centre for Alternative Finance. Bitcoin Mining Map. https://ccaf.io/cbnsi/cbeci/mining_map, 2023.

[25] Ethereum Foundation. Pooled staking. https://ethereum.org/en/staking/pools/, 2022.

[26] Jeff Garzik. CPU miner for Litecoin and Bitcoin. https://github.com/pooler/cpuminer, 2021.

[27] Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert Van Renesse, and Emin Gün Sirer. Decentralization in Bitcoin and Ethereum Networks. In *FC*, 2018.

[28] Yossi Gilad, Avichai Cohen, Amir Herzberg, Michael Schapira, and Haya Shulman. Are We There Yet? On RPKI's Deployment and Security. In *NDSS*, 2016.

[29] Yossi Gilad, Omar Sagga, and Sharon Goldberg. MaxLength Considered Harmful to the RPKI. In *Proc. ACM CoNEXT*, 2017.

[30] Phillipa Gill, Michael Schapira, and Sharon Goldberg. A survey of interdomain routing policies. *ACM SIGCOMM CCR*, 2013.

[31] Sharon Goldberg, Michael Schapira, Peter Hummon, and Jennifer Rexford. How secure are secure interdomain routing protocols. *ACM SIGCOMM CCR*, 2010.

[32] Jaehyun Ha, Seungjin Baek, Muoi Tran, and Min Suk Kang. On the sustainability of bitcoin partitioning attacks. In *FC*, 2023.

[33] Andreas Haeberlen, Ioannis C Avramopoulos, Jennifer Rexford, and Peter Druschel. Netreview: Detecting when interdomain routing goes wrong. In *USENIX NSDI*, 2009.

[34] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In *Proc. USENIX Security*, 2015.

[35] IRR explorer. https://irrexplorer.nlnog.net/, 2023.

[36] Benjamin Johnson, Aron Laszka, Jens Grossklags, Marie Vasek, and Tyler Moore. Game-theoretic analysis of DDoS attacks against Bitcoin mining pools. In *FC*, 2014.

[37] Gordon Fyodor Lyon. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure, 2009.

[38] Bill Marczak, Nicholas Weaver, Jakub Dalek, Roya Ensafi, David Fifield, Sarah McKune, Arn Rey, John Scott-Railton, Ron Deibert, and Vern Paxson. An Analysis of China's "Great Cannon". In *Proc. USENIX FOCI*, 2015.

[39] Sami Ben Mariem, Pedro Casas, Matteo Romiti, Benoit Donnet, Rainer Stütz, and Bernhard Haslhofer. All that Glitters is not Bitcoin – Unveiling the Centralized Nature of the BTC (IP) Network. In *Proc. IEEE/IFIP NOMS*, 2020.

[40] Andrew Miller, James Litton, Andrew Pachulski, Neal Gupta, Dave Levin, Neil Spring, and Bobby Bhattacharjee. Discovering Bitcoin's public topology and influential nodes. 2015.

[41] Alexandros Milolidakis, Tobias Bühler, Kunyu Wang, Marco Chiesa, Laurent Vanbever, and Stefano Vissicchio. On the effectiveness of bgp hijackers that evade public route collectors. *IEEE Access*, 2023.

[42] P2Pool: Decentralized mining pool protocol. P2Pool. http://p2pool.in/, 2023.

[43] Mining pool stats. https://miningpoolstats.stream/, 2023.

[44] Mining Rig Rentals. https://www.miningrigrentals.com/, 2023.

[45] Michael Mirkin, Yan Ji, Jonathan Pang, Ariah Klages-Mundt, Ittay Eyal, and Ari Juels. BDoS: Blockchain denial-of-service. In *Proc. ACM CCS*, 2020.

[46] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf, 2008.

[47] Gabi Nakibly, Jaime Schcolnik, and Yossi Rubin. Website-Targeted False Content Injection by Network Operators. In *Proc. USENIX Security*, 2016.

[48] Christopher Natoli and Vincent Gramoli. The Balance Attack or Why Forkable Blockchains are Ill-Suited for Consortium. In *Proc. IEEE/IFIP DSN*, 2017.

[49] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *Proc. IEEE EuroS&P*, 2016.

[50] Kevin Alarcón Negy, Peter R. Rizun, and Emin Gün Sirer. Selfish Mining Re-Examined. In *FC*, 2020.

[51] NiceHash. Nicehash: World's leading hashpower marketplace. https://www.nicehash.com/, 2023.

[52] NiceHash. When and how do you get paid? https://www.nicehash.com/support/mining-help/earnings-and-payments/when-and-how-do-you-get-paid, 2023.

[53] University of Oregon. Route views archive project. http://archive.routeviews.org/, 2023.

[54] Pavlenex. Stratum v2 (SRI) Roadmap - To infinity and beyond. https://stratumprotocol.org/blog/sri-roadmap-2023/, 2023.

[55] Trevor Perrin. The Noise Protocol Framework. http://noiseprotocol.org/, 2018.

[56] Alex Pilosov and Tony Kapela. Stealing the internet: An internet-scale man in the middle attack. *NANOG-44*, 2008.

[57] Stratum Protocol. Stratum v2. https://stratumprotocol.org/, 2023.

[58] Stratum Protocol. Stratum V2 Reference Implementation (SRI). https://github.com/stratum-mining/stratum/, 2023.

[59] Ruben Recabarren and Bogdan Carbunar. Hardening Stratum, the Bitcoin Pool Mining Protocol. *Proc. PETs*, 2017.

[60] Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3, 2018. RFC 8446.

[61] Meni Rosenfeld. Analysis of Bitcoin pooled mining reward systems. 2011.

[62] Rachel Rybarczyk. The Future of Bitcoin Mining Protocols: Making Every Watt Count. https://www.galaxy.com/research/insights/future-of-bitcoin-mining-protocols/, 2022.

[63] Muhammad Saad, Afsah Anwar, Srivatsan Ravi, and David Mohaisen. Revisiting Nakamoto Consensus in Asynchronous Networks: A Comprehensive Analysis of Bitcoin Safety and Chain Quality. In *Proc. ACM CCS*, 2021.

[64] Muhammad Saad, Songqing Chen, and David Mohaisen. SyncAttack: Double-spending in Bitcoin Without Mining Power. In *Proc. ACM CCS*, 2021.

[65] Muhammad Saad, Victor Cook, Lan Nguyen, My T Thai, and Aziz Mohaisen. Partitioning Attacks on Bitcoin: Colliding Space, Time, and Logic. In *Proc. IEEE ICDCS*, 2019.

[66] Muhammad Saad and David Mohaisen. Three Birds with One Stone: Efficient Partitioning Attacks on Interdependent Cryptocurrency Networks. In *Proc. IEEE S&P*, 2022.

[67] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal Selfish Mining Strategies in Bitcoin. In *FC*, 2017.

[68] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. ARTEMIS: Neutralizing BGP hijacking within a minute. *IEEE/ACM ToN*, 2018.

[69] Giorgio Severi, Matthew Jagielski, Gökberk Yar, Yuxuan Wang, Alina Oprea, and Cristina Nita-Rotaru. Network-level adversaries in federated learning. In *IEEE CNS*, 2022.

[70] Aftab Siddiqui. Klayswap – another bgp hijack targeting crypto wallets. https://www.manrs.org/2022/02/klayswap-another-bgp-hijack-targeting-crypto-wallets, 2022.

[71] Stephen Strowes. Visibility of IPv4 and IPv6 Prefix Lengths in 2019, 2019.

[72] Yixin Sun, Anne Edmundson, Laurent Vanbever, Oscar Li, Jennifer Rexford, Mung Chiang, and Prateek Mittal. RAPTOR: Routing attacks on privacy in Tor. In *Proc. USENIX Security*, 2015.

[73] Daryll Swer. How I set up my own Autonomous System. https://blog.apnic.net/2022/07/01/how-i-set-up-my-own-autonomous-system/, 2022.

[74] Andree Toonk. The Canadian Bitcoin Hijack. https://www.bgpmon.net/the-canadian-bitcoin-hijack/, 2014.

[75] 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 *Proc. IEEE S&P*, 2020.

[76] Muoi Tran, Akshaye Shenoi, and Min Suk Kang. On the Routing-Aware Peering against Network-Eclipse Attacks in Bitcoin. In *Proc. USENIX Security*, 2021.

[77] Marie Vasek, Micah Thornton, and Tyler Moore. Empirical analysis of denial-of-service attacks in the Bitcoin ecosystem. In *FC*, 2014.

[78] Theo von Arx, Muoi Tran, and Laurent Vanbever. Revelio: A Network-Level Privacy Attack in the Lightning Network. In *Proc. IEEE EuroS&P*, 2023.

[79] Addy Yeow. Reachable bitcoin nodes. https://bitnodes.io/, 2023.

Appendix A. Survey on Mining Pools

According to MiningPoolStats [43], nearly 700 PoW cryptocurrencies allow pooled mining as of May 2023. Our study focuses on the top ten cryptocurrencies, which accumulate over 99% of the total capitalization. We consider only the pools that mine one or more blocks in the last 1,000 blocks for each of the selected cryptocurrencies. We can find the creators of the many blocks as mining pools usually include their names in the blocks as a marketing tool for attracting more miners to join.

Table 2 shows that mining pools are commonly active in multiple cryptocurrencies. We also observe that 50 out of 55 studied pools are vulnerable to the EROSION attacks, see the red boxes. Five other pools are not vulnerable to the EROSION attack because they do not accept new miners or operate under Stratum protocols, see the green boxes. Since Stratum has been the de facto protocol for pooled mining in the last decade, many more pools are potentially vulnerable to the EROSION attack.

Appendix B. Algorithm for Mapping Miners to Pools

We simulate Bitcoin mining based on available data to evaluate the effectiveness of the EROSION attack. One crucial input to our simulations is the mining connections between miners and pools. Fortunately, we can estimate the hash rate distribution of miner ASes (i.e., the ASes hosting miners) and pool ASes (i.e., the ASes hosting the pool servers). The remaining challenge is assigning miners to pools so that the sum of miners' hash rate equals the pool's hash rate. This is the well-known subset sum problem with an NP-hard complexity; thus, it is hard to find the precise assignment by brute forcing. Instead, we implement a greedy-based heuristic for assigning miners to pools; see its pseudo-code in Algorithm 1.

At a high level, the heuristic considers the pools' hash rate as a budget and aims to assign as many miners within

	#	Pool name	Actively mined cryptocurrencies									
			BTC	DOGE	LTC	XMR	ETC	BCH	CFX	BSV	XEC	DASH
Vulnerable	1-10	2Miners				✓	✓					
		666pool							✓			
		Antpool	✓	✓	✓		✓	✓				✓
		Binance Pool	✓				✓					✓
		Braiins Pool	✓									
		BTC.com Pool	✓		✓		✓	✓				
		C3Pool				✓						
		CrazyPool					✓					
		DxPool			✓							
		EMCD	✓									
	11-20	Ethermine					✓					
		Ezil					✓					
		F2Pool	✓	✓	✓		✓					✓
		Flexpool					✓					
		Foundry USA Pool	✓					✓				
		GNTL Monero Pool				✓						
		GorillaPool								✓		
		HashVault				✓						
		HeroMiners				✓			✓			
		Hiveon Pool					✓					
	21-30	K1Pool					✓					
		Kryptex Pool				✓						
		KuCoin Pool	✓									
		LitecoinPool		✓	✓							
		Luxor Mining Pool	✓									✓
		Mining-Dutch		✓	✓			✓		✓	✓	✓
		Mining Pool Hub					✓					
		MoneroHash				✓						
		MoneroOcean				✓						
		Nanopool				✓	✓		✓			
	31-40	NiceHash	✓		✓			✓				
		PEGA Pool	✓									
		POOL-MOSCOW						✓				
		Poolflare							✓			
		Poolin	✓	✓	✓		✓	✓				✓
		Prohashing			✓	✓		✓	✓			✓
		SBICrypto Pool	✓									
		Sigmapool			✓							
		Skypool				✓						
		solomining.io						✓				
	41-50	SoloPool			✓	✓	✓	✓				
		SupportXMR				✓						
		Toomim						✓				
		Ultimus Pool	✓									
		ViaBTC	✓	✓	✓		✓	✓			✓	✓
		Volt mine				✓						
		WoolyPooly							✓			
		XMRPool				✓						
		Zergpool										✓
		ZULUPooL			✓							
Invulnerable	51-55	HyperDonkey			✓							
		MaraPool	✓									
		p2p-spb						✓				
		P2Pool				✓						
		TAAL								✓		

TABLE 2: The studied 55 mining pools and the top cryptocurrencies they actively mined. The EROSION attack applies to 50 pools. The remaining five pools are either private or running the P2Pool protocol.

the budget as possible. In particular, our algorithm iterates through all mining pools, ranked from smallest to biggest. For each mining pool, we find miners that can be assigned; that is, their hash rate is less than the pool's remaining hash rate, and they are not assigned to any pool yet; see lines 6–11. Then, we randomly assign a possible miner to that pool, update the set of assigned miners and the pool's leftover hash rate, and repeat the process until all the pool's hash rate is assigned or no possible miner is left (i.e., line 12–18). Finally, the algorithm outputs an allocation of miners to individual mining pools.

Algorithm 1 Assigning n miners to k pools.

Require: $M = \{m_1, m_2, \cdots, m_n\}$: Hash rate distribution of all miners.
$P = \{p_1, p_2, \cdots, p_k\}$: Hash rate distribution of all mining pools
$(0 \leq p_1 \leq p_2 \leq \cdots \leq p_k \leq 1)$.
Ensure: $Allocation_i$: An array of miners allocated to pool i $(i = 1 \cdots k)$.

procedure ASSIGNINGMINERS
  $AssignedMiners \leftarrow []$
  for $i \leftarrow 1$ to k **do** ▷ assign miners to smaller pools first.
    $Allocation_i \leftarrow []$
    while $p_i > 0$ **do** ▷ until no more hash rate to be assigned.
      $PossibleMiners \leftarrow []$
      for $j \leftarrow 1$ to n **do**
        if $(m_j \leq p_i) \wedge (j \notin AssignedMiners)$ **then**
          $PossibleMiners \leftarrow PossibleMiners + [j]$
        end if
      end for
      if $|PossibleMiners| = 0$ **then** ▷ no possible miner left.
        break
      end if
      $SelectedMiner \leftarrow PossibleMiners.random()$
      $p_i \leftarrow p_i - m_{SelectedMiner}$
      $Allocation_i \leftarrow Allocation_i + [SelectedMiner]$
      $AssignedMiners \leftarrow AssignedMiners + [SelectedMiner]$
    end while
  end for
  return $Allocation$
end procedure

Appendix C. Root Cause of the Vulnerability

In Section 6, we describe a vulnerability of the Stratum V2 protocol, in which a corrupted packet can cause persistent decryption failures at the receiver end. Here, we present the detailed root cause of this vulnerability, including the encrypted messages' format, the typical decryption scenario, and the buggy implementation that creates the vulnerability.

Figure 13 describes the format of the encrypted Stratum messages [57]. After Stratum clients finish a connection handshake, their subsequent messages are encrypted using the Noise Protocol Framework [55]. Each packet includes a header consisting of a 6-byte ciphertext of Stratum message header and 16-byte MAC and a payload containing 65535-byte encrypted chunks.

In normal circumstances, the message receiver first decrypts the message header (i.e., `ct_msg_hdr`) to learn the message payload length (i.e., `pld_len`) and then decrypts the payload (e.g., `ct_pld_1`) into plaintext (e.g., `pt_pld_1`) accordingly. Upon successful packet decryption, the receiver increments its nonce to decrypt the next packet.

The identified vulnerability comes from how Stratum clients handle decryption errors. Particularly, when receiving a tampered packet (e.g., containing random bytes with the same length or out-of-sync nonce counter), the client fails to decrypt the expected frame (e.g., a 6-byte encrypted header or a 65519-byte encrypted payload chunk). When a decryption error happens, the client *always* assumes that missing data in the stream due to network loss causes it and then tries to decrypt the next frame. As decryption failures

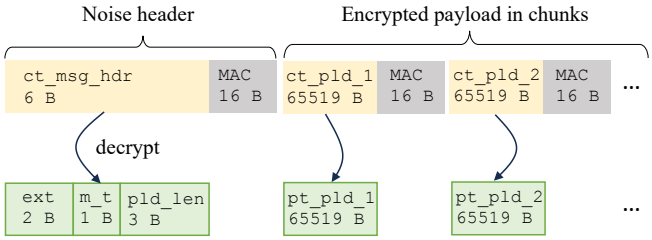


Figure 13: The framing of encrypted Stratum messages. Every packet consists of a 22-Byte Noise header followed by variable-length payload in chunks of 65535 Bytes.

repeat, the client cannot retrieve the Stratum message in this packet and all subsequent messages.

Fortunately, Stratum developers already fixed this vulnerability as of this writing. In particular, when a client fails to decrypt an arriving Stratum message, it shuts down the stream reader and resets the connection with the sender [1].