# How to Detect Cryptocurrency Miners? By Traffic Forensics!

Vladimír Veselý[a,*], Martin Žádník[a]

[a]*Brno University of Technology, Božetěchova 2, Brno 612 66, Czech Republic*

## Abstract

Cryptocurrencies set a new trend for a financial interaction between people. In order to successfully meet this use-case, cryptocurrencies combine various advanced information technologies (e.g., blockchain as a replicated database, asymmetrical ciphers and hashes guaranteeing integrity properties, peer-to-peer networking providing fault-tolerant service). Mining process not only introduces new cryptocurrency units, but it has become a business how to generate revenue in real life. This paper aims at different approaches how to detect cryptocurrency mining within corporate networks (where it should not be present). Mining activity is often a sign of malware presence or unauthorized exploitation of company resources. The article provides an in-depth overview of pooled mining process including deployment and operational details. Two detection methods and their implementations are available for network administrators, law enforcement agents and the general public interested in cryptocurrency mining forensics.

*Keywords:* Bitcoin, cryptocurrency, mining pool, mining server, Stratum protocol, GetBlockTemplate protocol, GetWork protocol

## 1. Introduction

The motivation behind cryptocurrency is to introduce an alternative currency that is not controlled by a government (e.g., the central bank). Trustworthiness of such electronic cryptocurrency lies in the utilization of cryptographical algorithms to verify transactions and fair emission of new units into circulation. Dark web marketplaces utilize cryptocurrencies for their: a) nearly instant and free-of-charge payments; b) easily obtainable and changeable addresses; c) hard to trace transactions (thanks to their peer-to-peer nature). Several studies [1], [2], [3] investigate Bitcoin as the key component of any digital black marketplace because cryptocurrencies generally allow criminals to circumvent law enforcement agencies (LEAs) and regulators.

Of all cryptocurrencies, Bitcoin [4], [5] had become popular when it gained momentum at the end of 2013 after its exchange price skyrocketed. The current (at the August 2018)

[*] Corresponding author

*Email addresses:* `veselyv@fit.vutbr.cz` (Vladimír Veselý), `izadnik@fit.vutbr.cz` (Martin Žádník)

*URL:* `www.fit.vutbr.cz/~veselyv` (Vladimír Veselý), `www.fit.vutbr.cz/~izadnik` (Martin Žádník)

*Preprint submitted to The International Journal of Digital Forensics and Incident ResponseSeptember 2, 2018*

total number of Bitcoins (approx. 17.1 million) accounts for more than 139 billion USD [6]. Bitcoin is a peer-to-peer network with the distributed infrastructure of users and miners. A miner verifies ongoing transactions for a reward (either transaction fee or newly emitted Bitcoins). The reward is paid to the first miner who proves transaction by spending its computation power on this process. Other proof-of-work[1] cryptocurrencies also adopted the same mining concept. Anyone can join the solo mining process but the probability of earning a reward is low and the risk of wasted computational power without any profit too high. Therefore, miners form so-called mining pools. When the pool earns a reward, it is distributed by the pool operator among miners according to their contribution.

Apart from alternatives to Bitcoin (e.g., Litecoin, Ethereum, generally referred as *(alt)coins*), the cryptocurrency universe also contains *tokens*. Tokens (comparing to coins) represent digital asset or utility that leverages another's coin blockchain for being accounted. New tokens are generally not mined but distributed by their authors/owners. In the frame of this paper, we will focus only on the mining process behind coins and refer to them as "cryptocurrencies" interchangeably.

Any organization should be aware of running mining software on its hardware in its network due to at least two reasons: a) the mining activity is often caused by malware, therefore, the mining activity is an indicator of a compromise; b) the energy (e.g., electricity, cooling, CPU and GPU power) spent on mining is paid by the hosting organization, but the recipient of the reward is a malicious actor. This survey [7] speaks about various types of cryptocurrency malware dedicated to undercover mining on devices, desktops, and servers but also platforms like webcams, smartphones or network attached storages. Universities [8] or technological centers [9], [10] are typical examples of energy exploitation because they offer free computational resources (i.e., servers, network) to academics, researchers and students. Nevertheless, it is possible to start a mining operation in any organization including subsidized accommodation for Czech members of parliament [11].

The malicious actor might exploit these assets in such environment resulting in the increased energy bill, depleted resources, endangered work processes and other users. For instance, Bitcoin mining has a severe impact on electricity comparable to the energy consumption of Ireland [12] in 2014. Another report [13] provides a more in-depth analysis of how to estimate Bitcoin's hunger for energy concluding that it may reach 7.67 gigawatts (comparable with Austria) during 2018. The reader must take into account that Bitcoin mining is just a tip of the iceberg, which consists of all proof-of-work cryptocurrencies.

In this paper, we focus on the detection of devices participating in the mining pools. Cryptocurrency mining is the only option how users may obtain freshly minted currency units. Moreover, mining is still the prevailing form of how to earn cryptorcurrencies with the existing equipment.

We propose two approaches how to detect cryptocurrency miners in the network:

- The first approach is based on a mix of passive and active traffic monitoring. The

[1] In case of *proof-of-work* mining, the probability of finding a new block is directly proportional to a computational power invested in mining. While for *proof-of-stake* mining, the probability is directly proportional to a number of units owned by a miner.

passive monitoring is based on the analysis of IP flow records, while the active monitoring is based on probing. The detection method as a whole slowly learns a list of mining servers which subsequently reduces the need for the active monitoring. Since anyone can set up own mining pool or even mining server, the resulting list of publicly known mining servers cannot be considered complete. However, it may be employed as a baseline for miner detection by any network operator.

- The second approach can be described as a catalog of mining pools. We have created a publicly available web application that stores metadata about existing mining pools. Any user may query our system to check whether a given FQDN[2], IP address or port number is a part of known pool configuration.

The contribution of this article involves: a) an overview of the current cryptocurrency mining technology; b) two detection methods to detect network traffic related with cryptocurrency mining; c) open-access data samples; and d) publicly available service cataloging mining servers.

The rest of the paper is organized as follows. Section 2 informs about related work on cryptocurrency mining. Section 3 brings details about currently used mining architecture and involved protocols. Section 4 describes passive/active traffic monitoring (the first approach how to detect miners), which also includes its validation and verification. Section 5 explains the implementation and operation of the mining server catalog (the second approach). The article is summarized in Section 6, which also outlines our future work.

## 2. Related Work

This section summarizes knowledge from the selected articles relevant to cryptocurrency mining. We try to motivate miners detection in a frame of known cryptocurrency issues and research of others.

We consider Courtois et al. [14] work as a great introductory source explaining Bitcoin mining. Despite focusing on Bitcoin mining process improvement, authors provide theoretical background explaining bindings between employed cryptography and cryptocurrency mining. Moreover, this work and other ones mentioned in this section allow us to skip thorough the cryptographic description of the mining process. Instead of it, we will focus only on protocols and messages exchanged between miner and pool.

Kroll et al. [15] and Lawenberg et. al [16] provide an economical point of view on Bitcoin mining. They try to model the mining process as the game-theory problem. Eyal and Sirer [17] discuss Bitcoin security and mining incentive-compatibility. All of these articles introduce interesting attacks that might disrupt any cryptocurrency mining process. We will briefly mention mining protocol "flaws" that may be used to identify miner and connect its identity with a real person.

[2] Fully qualified domain name (FQDN) is complete host identification within a Domain Name System (DNS) tree hierarchy.

Several studies [18], [19], [20], [21] mention ways and means how cryptocurrencies are being employed in monetizing and as a platform for unlawful activities. Examples include ransomware attacks, botnet command and control operations, private keys thefts, spam advertisements, pay-per-click or pay-per-install scams and others. Our research complements these studies by targeting the illicit mining of cryptocurrencies.

Huang et al. [22] provide a comprehensive study of cryptocurrency mining malware. Authors developed methods, which correlate the mining bot with its mining pool. Moreover, authors were able to estimate the number of infected devices, generated revenue and duration of botnet infection. We consider this paper as a great stimulation for our work because it shows how successful discovery of miners can be crucial not only for proper network operation but also for significant reduction of botnets contagions. There is a connection between (unintentional) cryptocurrency mining and exploitation of resources.

D'Herdt [23] analyzed captured traffic samples and suggested to look for well-known ports and IP addresses of mining servers. Besides that, he derived that the communication of miners with mining server is sparse but often cyclic between 30-100s. Although it is possible to capture all the network traffic even on a high-speed link [24] so that the raw network data can be analyzed, it is a resource-expensive way of network monitoring from a long-term perspective.

Therefore, various meta-data collecting approaches are utilized, among others IP flow monitoring represented by several generations of NetFlow protocols [25] and IPFIX [26] is widely deployed. The intrinsic characteristic of flow monitoring is a loss of information in comparison with the full packet capture. The flow data analysis is, therefore, less reliable than the packet analysis itself. Nevertheless, flow analysis has found many applications such as network monitoring, application classification, host profiling, accounting and billing [27].

The research in flow analysis has come up with simplistic as well as complex approaches ranging from statistic methods to machine learning (ML) approaches. As an example, Hofstede et al [28] propose a simple statistical method based on the same number of packets per flow in multiple flows to detect attacker trying to brute-force SSH server. More complex methods have been proposed to detect anomalies, e.g., Silveira at al [29] proposes a complex statistic model to detect strong correlations in simultaneous flow volume changes.

To the best of our knowledge, there has been no work focusing on detection of cryptocurrency miners utilizing flow data. However, from the perspective of methodology, the closest to our work are methods based on supervised machine learning such as [30]. The authors of [30] select descriptive features that should be extracted from the flow data, prepare an annotated data-set and train a classification model which is used to recognize specific events in the flow data.

Due to the inherent loss of information in case of flow monitoring and due to the heuristic nature of the analysis methods, there are often false positive results. If the number of false positives is very low, then the heuristic can be deployed. Otherwise, administrators start to ignore the analysis results after several false positives due to a loss of trust in the method. We address the issue of false positives by combining the results of passive detection and active probing. Such an approach has not been investigated in the previous research.

## 3. Mining Background

This section provides a theoretical background (mostly based on Bitcoin use-case). However, explanation of the whole mining process for all cryptocurrencies is far beyond the scope of this article. Hence, only parts relevant to the miner detection are captured. The first subsection lays out the basic theory for any cryptocurrency operation. The second subsection familiarizes the reader with the state-of-the-art of cryptocurrency mining software and hardware. The third subsection provides a deeper description of existing mining protocols.

### *3.1. Theory*

*Transaction* encapsulates transfer of cryptocurrency units between parties, where a single transaction may contain multiple inputs and also outputs. To prevent fake or malicious transaction (e.g., double spending problem), a given user needs to validate the transaction history. Hence, transactions are chained together, where outputs of the previous transaction serve as inputs of the next transaction. Transactions are grouped into *blocks*, which vouches for the validity of contained transaction with timestamps and cryptographic hashes. Blocks are periodically recorded into public ledger dubbed as *blockchain.* Blocks are bound together in blockchain as a unidirectional list, where each item (i.e., block) has the pointer to its predecessor. The inception of cryptocurrency is done by starting its history with the first *genesis block.* Blocks are formed and their content verified by *miners*, who compete between themselves in the process (so-called *mining*) of appending new blocks to the blockchain. The winning miner earns reward in the form of newly minted coins (called *coinbase transaction*) as an incentive to participate on cryptocurrency peer-to-peer network operation. The winner is the miner, which would successfully solve the certain cryptographic task (e.g., compute a hash with certain properties from given inputs and nonces) of variable *difficulty* (which acts as a feedback mechanism guaranteeing deterministic time of block creation). Miners are grouped in *pools* in order to increase their chance of successful mining and thus to cash the reward.

### *3.2. Hardware and Software*

A wide range of different mining hardware/software exist that is mostly differentiated by an employed hashing algorithm and *hashrate* (i.e., computational performance in a number of hashes per second, abbreviated as hash/s). Depending on a given cryptocurrency, the user chooses the appropriate combination of hardware and software that impacts mining operation. Mining hardware capabilities pose an upper-bound limit for a maximum available hashrate. Nevertheless, the choice of mining software may optimize and automatize the mining operation. To generalize it, successful establishment of cryptocurrency mining consists of several steps.

1. Select cryptocurrency - It is important to decide which cryptocurrency to mine if the miner speculates on the future price. Hence, it is necessary to take into account: a) trend of exchange price; b) cryptocurrency viability; c) possible increase of mining difficulty; and d) ever-changing total hashrate of the peer-to-peer network. There is no business perspective to mine cryptocurrencies if overall expenses exceed potential

income. Due to the very volatile exchange rates between cryptocurrencies and fiat money, the risks are high.

2. Choose a pool - Participation in the pool (compared to solo mining) offers a more predictable generation of revenue, which is proportional to work done by a miner. It is important to choose a stable pool (in terms of Internet connectivity and denial-of-service protection) with trustworthy pool operator (who will not embezzle earnings or submitted shares for own profit). Available strategies of how are miners rewarded (e.g., Pay Per Share, Pay Per Last N Shares, Shared Maximum PPS, Capped Maximum PPS With Recent Backpay) also affects the selection of the mining pool. Comparison of the most popular mining pools is available at following website [31].
3. Assemble mining rig - Overall power consumption of the mining rig goes hand to hand with its hashrate performance. Any mining rig dissipates heat that needs to be ventilated out. Improper cooling may lead to unnecessary outages or hardware failures that prolong or even challenges return of investment for mining rig cost. Professional mining rigs are out-of-the-box solutions (packed with GPUs or ASIC-based) that contain industrial air-vents producing a lot of noise. All previously mentioned facts can be used as indicators how to locate mining rig in corporate environment physically.
4. Configure mining software - Mining is controlled and managed either by official cryptocurrency client or dedicated software. Mining pool suggests to its members mining software for supported cryptocurrencies. Moreover, the pool provides a personalized configuration which helps miners to start quickly and easily their mining operation. Mining software needs a low-bandwidth, but constant connection to the Internet since it periodically exchanges work packages with the server.

Figure 1 outlines usual deployment scenarios between miners and mining pools. A user may control multiple mining devices (rigs, sometimes also referred to as workers). Each mining device may rotate mining operation (i.e., in a round-robin or fall-back fashion) between multiple pools. Each mining device is connected to a single mining server that belongs to the pool, thou switching to a secondary mining server is quite common in case of the outage of the primary one. The connection between miner and server can be: a) direct without any middle-box (although, it reveals IP address of miner); b) proxied by centralizing communication with mining server via middle-box that may relay or even alter mining protocol data; c) overlayed via VPN[3], TOR[4], I2P[5] or any similar service.

Cryptocurrency mining has evolved through the following five generations:

1. CPU mining - CPU software mining is the initial way how to verify transaction for the most of young cryptocurrencies when consensus about mining is known, but there is no parallelization or dedicated hardware support yet. Mining on general CPU offers just a fraction of a potential hashrate, but it is easy to develop software for CPU miner.

[3] Virtual Private Network. For more, see https://en.wikipedia.org/wiki/Virtual_private_network

[4] The Onion Router. For more, visit https://www.torproject.org/

[5] Invisible Internet Project. For more, check https://geti2p.net/en/

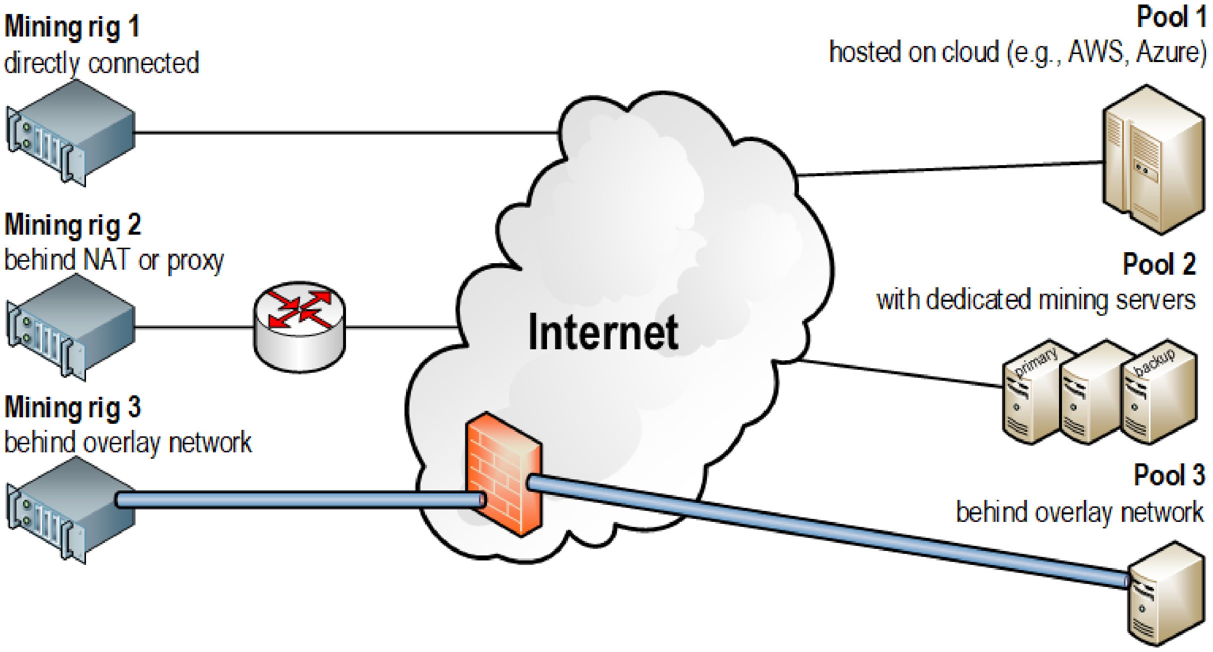


Figure 1: Deployment scenarios for miners and pools

2. GPU mining - Current generations of graphical cards are supercomputers capable of massive parallel computations. GPUs generally have higher hashrate than CPUs, but their effectiveness also depends on employed hashing algorithm (for instance the same GPU card would mine Bitcoins in order of Mhash/s, but Litecoins in khash/s). There is a major difference in hashrate between the two biggest GPU vendors. AMD cards are generally more effective because they have better Arithmetic Logic Unit support comparing to nVidia cards, which are focused on Floating Point Unit operations.
3. FPGA mining - Field-programmable gate array chips (FPGAs) have far less power consumption comparing to GPUs, but they do not significantly increase hashrate. FPGAs just like ASICs are available for a limited number of cryptocurrencies because development and manufacturing of these platforms require significant investments and there must be the business reasoning behind.
4. ASIC mining - Application-specific integrated circuits (ASICs) are the last stage of the mining hardware evolution. ASICs offer the highest mining speeds and the best ratio between hashrate and power consumption. Currently, ASICs availability is limited to doubled SHA-256 [32], Scrypt [33] and X11 [34] hashing algorithms (and accompanied cryptocurrencies). However, ASICs pose a threat to the stability of any cryptocurrency system because of uneven distribution of hashrate, where the minority of ASICs users usually represent the most powerful mining group that may lead to successful 51% attack [35]. Probably due to the appealing return of investment, the history of ASIC-miner manufacturing has lots of examples of false promises [36], [37] or even scams [38], [39]. Moreover, a limited number of companies survived this business clash and

| Hardware | Generation | Hashrate |
|---|---|---|
| Intel i7 Core 3930k | CPU | 66 Mhash/s |
| AMD Radeon 7970 | GPU | 710 Mhash/s |
| nVidia GeForce GTX 590 | GPU | 190 Mhash/s |
| BitForce SHA256 Single | FPGA | 830 Mhash/s |
| ModMiner Quad | FPGA | 800 Mhash/s |
| ButterFly Labs Single SC | ASIC | 30 Ghash/s |
| Avalon1 A3256 Miner | ASIC | 66 Ghash/s |

Table 1: A comparison of Bitcoin mining hardware relevant to 2013

(at least Bitcoin) ASIC vendor market is dominated by a single company (Bitmain[6]) with nearly none competition (i.e., Canaan Creative[7]).

5. Cloud mining - The setup of the successful mining operation requires non-trivial IT skills, appropriate software installation and investments in the hardware. Hence, the entry level for mining is too steep for the majority of the population. Therefore, mining rental service has got more and more popular since 2013. Using this service, the customer buys a mining contract for a limited period. A remote miner is paid with the fixed rent, and the customer receives all coins earned by a miner (thus, speculating for a future price of a selected cryptocurrency). For a duration of mining contract, the customer may even switch the configuration to mine a different cryptocurrency (as long as it uses the same hashing algorithm). Following website [40] compares different cloud mining services.

To compare generations between each other (and show their impact on the market), we will use Bitcoin and doubled SHA-256 hashrate as the example illustrated in Table 1. Since mining technology evolves fast, we list average hashrates (based on community shared [41], [42] user experience) of devices available in the year 2013, which was the last year when all five generations coexisted together in Bitcoin mining ecosystem.

According to a chosen cryptocurrency, miner needs to install and configure special software which coordinates mining task between mining hardware and the pool. Mining software responsibility is: a) to communicate with the mining server using mining protocol; and b) to relay mining work package for hardware processing. A plethora of consensus methods exists differentiated mostly by employed hashing algorithms. Following Table 2 summarizes some of the existing mining software supporting GPU mining for a selected number of cryptocurrencies:

The mining software configuration specifies URL of mining server (including port number), username and password for miner's authentication, hardware-related setup (e.g., preferred GPU kernels, graphical card processor, and memory overclocking). Recommended configuration parameters are always available on pool's website (usually together with a

[6] Namely a line of products called Antminers, for more see https://shop.bitmain.com/?lang=en

[7] Namely a line of products called AvalonMiners, for more see https://canaan.io/shop/

| **Cryptocurrency** | **Algorithm** | **Mining software** |
|---|---|---|
| Bitcoin | SHA-256d | cgminer<br>BFGMiner |
| Litecoin<br>Dogecoin | Scrypt | cgminer<br>BFGMiner |
| Dash | X11 | SGMiner<br>ccMiner |
| Ethereum<br>Ethereum Classic | Dagger-Hashimoto | ethminer<br>Claymore's Dual Miner |
| ZCash | Equihash | Silent Army ZCash Miner<br>Claymore's ZCash Miner |
| Monero | Cryptonight | Wolf Miner<br>ccMiner |
| Vertcoin | Lyra2RE | SGMiner |

Table 2: Overview of some mining software

mining guide for beginners, see for example Appendix A) in order to guarantee a smooth and user-friendly setup of the mining operation.

### *3.3. Protocols*

A mining pool and its members are using dedicated protocols to coordinate distribution of mining process. There exist three general mining protocols supported by a majority of cryptocurrencies:

- *GetWork* was the first mining protocol ever. Comparing to its descendants, GetWork is a simple request-response scheme protocol, where server assigns work package and miner blindly conducts mining task. Due to its simplicity, GetWork allows double-spent transactions in the case of corrupt pool operator. GetWork messages with JSON[8] syntax are carried inside HTTP[9]. GetWork supports a limited number of protocol extensions using additional HTTP header lines.

- *GetBlockTemplate* is official mining protocol developed in the frame of Bitcoin community but also adopted by other cryptocurrencies. GetBlockTemplate was codified in BIP[10] 22 [43]. GetBlockTemplate is more decentralized by offloading block creation process onto miners instead of pools. GetBlockTemplate increases potential work package size and reduces mining protocol overhead to support performance delivered by ASIC miners. Moreover, BIP 23 [44] standardizes extensions and ways how to flexible improve GetBlockTemplate without any major protocol redesign or non-conformant HTTP header hacks.

[8] JavaScript Object Notation. For more, see https://tools.ietf.org/html/rfc7159

[9] Hypertext Transfer Protocol. For more, visit https://tools.ietf.org/html/rfc7230

[10] Bitcoin Improvement Proposal. See https://github.com/bitcoin/bips/blob/master/bip-0002.mediawiki

- *Stratum* protocol [45] was prototyped by M. Palatinus, inventor of pooled mining and operator of the oldest Slush pool [46]. Stratum development was motivated by a need to remedy design flaws of previous two protocols: a) by removing HTTP as a carrier, which reduces unnecessary protocol overhead; b) by removing long polling feature that posed scalability issue for load balancing of traffic between miner and server; and c) by adding *extranonce* field that allows miner to generate more hashes locally without bothering a mining server for a new batch of work. Stratum is JSON-RPC 2.0 [47] compatible protocol that operates directly above TCP.

All of these protocols leverage TCP[11] as the transport layer protocol. Comparing to official cryptocurrency peer-to-peer clients, mining protocols do not use any ”well-known” port number. It depends solely on the preference of mining pool administrator on which ports pool servers accept connections. Hence, port numbers 80, 443, and 25 are often used as a best practice how to bypass firewalls between mining device and its mining server.

The usual message exchange involves several steps. With the initial message, the miner connects to the mining server and provides authentication credentials. Authentication is necessary because based on credentials, mining pool correlates submitted shares with miner’s account and credit earnings. Two types of authentication are common:

- *registration-oriented* - Before establishing the mining operation, the user owning mining rig needs to sign up to the pool and create an account. A part of account administration involves workers (i.e., separate mining devices) setup. Authentication credentials inside mining protocol include username and password.

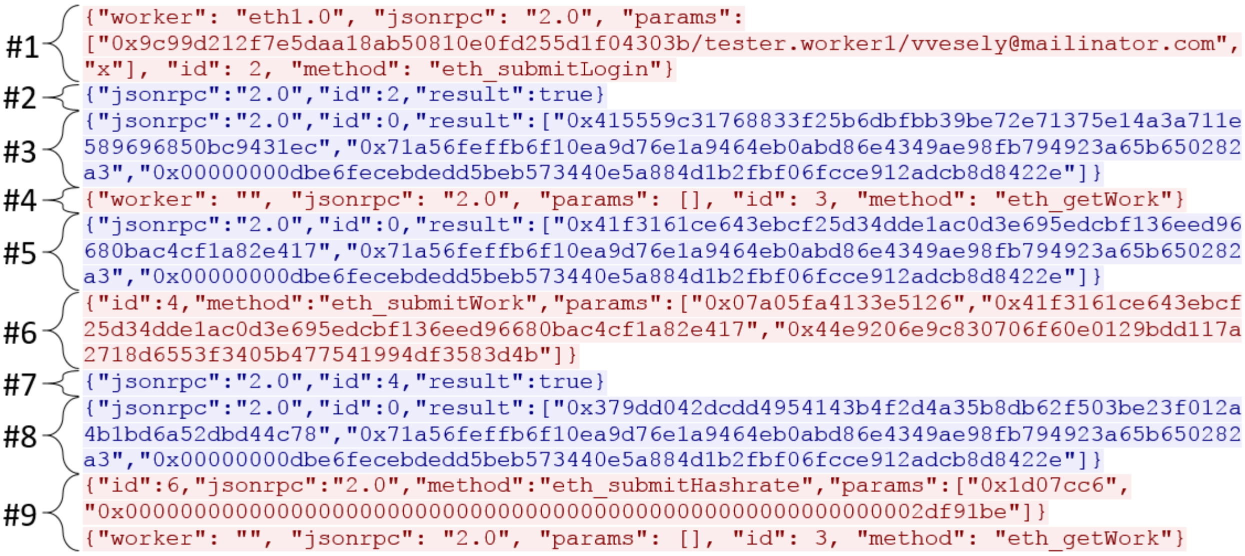
- *registration-less* - Some pools tender their services without any dedicated account registration. In that case, the miner usually provides just cryptocurrency address to inform pool where to send payments. This identifier substitutes username and is enough for authentication.

No matter on authentication type, the username may contain optional suffixes such as worker identifier (in order to distinguish different workers of the same user) or e-mail address (where the user is notified about any problems occurred during mining).

The next step in mining protocol communication is a recurrent assignment of work packages provided by the server. Each work package contains *data*, *target*, and *nonce* (other fields depend on cryptocurrency). A miner tries to find a hash (from combined data and nonce) that meets (i.e., is lower than) a target. Different cryptocurrencies are using distinct hashing algorithms - e.g., SHA-256d for Bitcoin, Scrypt for Litecoin, X11 for Dash. Miner either submits correct solution or restarts mining with different inputs upon receiving a new work package. Miner periodically announces its state to the server.

Figure 2 illustrates Stratum exchange from mining device to its server (with the red color) and vice versa (the blue color). We can observe the typical confluence of messages.

[11] Transmission Control Protocol. For more, see https://tools.ietf.org/html/rfc793



```
{"worker": "eth1.0", "jsonrpc": "2.0", "params":
["0x9c99d212f7e5daa18ab50810e0fd255d1f04303b/tester.worker1/vvesely@mailinator.com",
"x"], "id": 2, "method": "eth_submitLogin"}
{"jsonrpc":"2.0","id":2,"result":true}
{"jsonrpc":"2.0","id":0,"result":["0x415559c31768833f25b6dbfbb39be72e71375e14a3a711e
589696850bc9431ec","0x71a56feffb6f10ea9d76e1a9464eb0abd86e4349ae98fb794923a65b650282
a3","0x00000000dbe6fecebdedd5beb573440e5a884d1b2fbf06fcce912adcb8d8422e"]}
{"worker": "", "jsonrpc": "2.0", "params": [], "id": 3, "method": "eth_getWork"}
{"jsonrpc":"2.0","id":0,"result":["0x41f3161ce643ebcf25d34dde1ac0d3e695edcbf136eed96
680bac4cf1a82e417","0x71a56feffb6f10ea9d76e1a9464eb0abd86e4349ae98fb794923a65b650282
a3","0x00000000dbe6fecebdedd5beb573440e5a884d1b2fbf06fcce912adcb8d8422e"]}
{"id":4,"method":"eth_submitWork","params":["0x07a05fa4133e5126","0x41f3161ce643ebcf
25d34dde1ac0d3e695edcbf136eed96680bac4cf1a82e417","0x44e9206e9c830706f60e0129bdd117a
2718d6553f3405b477541994df3583d4b"]}
{"jsonrpc":"2.0","id":4,"result":true}
{"jsonrpc":"2.0","id":0,"result":["0x379dd042dcdd4954143b4f2d4a35b8db62f503be23f012a
4b1bd6a52dbd44c78","0x71a56feffb6f10ea9d76e1a9464eb0abd86e4349ae98fb794923a65b650282
a3","0x00000000dbe6fecebdedd5beb573440e5a884d1b2fbf06fcce912adcb8d8422e"]}
{"id":6,"jsonrpc":"2.0","method":"eth_submitHashrate","params":["0x1d07cc6",
"0x000000000000000000000000000000000000000000000000000000000002df91be"]}
{"worker": "", "jsonrpc": "2.0", "params": [], "id": 3, "method": "eth_getWork"}
```

Figure 2: Example of Stratum protocol message exchange

A connection to the pool is initiated with the first message (marked as #1), where we can see authentication details. The server confirms it with message denoted as #2. The server sends a work package (#3) that needs to be computed. Upon proper initialization of mining software, the miner asks for a new work package (#4), which the mining server gladly provides (#5). The miner successfully finds the hash and submits (#6) the complete solution back to the server. The server decides whether the miner's result is valid or not (in the case of #7, it is valid) and sends a new work package (#8). The miner starts a new task and meantime periodically updates server about its local computational speed (message marked as #9) so that server can dynamically adjust the size of subsequent work packages. If we focus on the forensic analysis of metadata related to mining protocol, then we can extract:

- IP addresses and port numbers - By inspecting IP addresses, we can geolocate both miner and mining server. Together with port numbers, we can account network traffic with NetFlow. Once we have NetFlow records available, we can answer questions such as for how long is mining operation active, how many mining devices are involved, etc.

- Pool information - GetWork or GetBlockTemplate protocol extensions may uncover other useful intel such as alternative mining servers including their IP addresses, fully-qualified domain names, and port numbers.

- Miner's username - Based on authentication type, username field may contain either nickname or account name of pool user or its cryptocurrency address. This information may be crucial for successful correlation of real-world person and its electronic identity.

- Miner's password - Authentication message of any mining protocol includes a password. However, it is seldom used for authorization or any purpose by a pool. The default value of password field for the most of mining software is ‘x’.

- Miner’s email - Some pools offer email notifications about the progress of mining operation. In case of any problem such as the miner outage, too many rejected shares or disconnection from the pool, the user is warned by email. The email address may be optionally part of mining protocol message filed, which may help to reveal user’s identity.

During the implementation phase of our miner’s detection methods, we needed relevant data. Hence, we recorded packets exchanged between our testing miner and various pools mining different cryptocurrencies. We offer these Wireshark packet captures as a data-set, which is publicly available at [48] to anyone interested in subsequent research related to mining protocols.

## 4. Traffic Monitoring

Active mining clients connect to the pool, ask for a job and deliver the results. Their communication with the mining server offers a possibility for passive detection. If the characteristics of the message exchange are specific enough, we may differentiate miners’ communication from the rest of the traffic. We intend to utilize basic IP flow records (e.g., NetFlow v5) as an input for the passive detection algorithm. This approach is generic and not dependent on the specific capabilities of the monitoring device, which offers more possibilities for broader deployment. On the other hand, we expect a lot of false positives caused by the limited set of input characteristics that are not unique enough. We address the problem by adding the second detection step, where an active connection attempt verifies the identity (whether it belongs to mining pool or not) of the server to which the suspicious client connects to. This active detection is done by probing the potential server with specific JSON message unique to Stratum protocol. In order to reduce the number of probes, we propose to employ a list of already probed servers. The list stores the probing result (either positive or negative), the destination port number and the timestamp of the probe.

### *4.1. Design of Cryptocurrency Network Traffic Detector*

We propose to apply a classifier built by a machine learning technique to detect candidate mining communication passively. But before it is possible to apply machine learning, there must be an annotated data-set that would allow for training and evaluation of the classifier.

Our process of data annotation is based on an iterative approach which allows continuous building of annotated data-set. Figure 3 provides the overall view of the process while individual components are elaborated later on. To start the annotation from scratch, we select features based on the observations of the traffic of local mining clients and the flow data of the well-known mining servers. Then we manually construct a simple classifier, which serves as a passive detector, and active probing verifies its results. Once enough data is annotated, we replace the simple classifier with the classifier trained by machine learning.

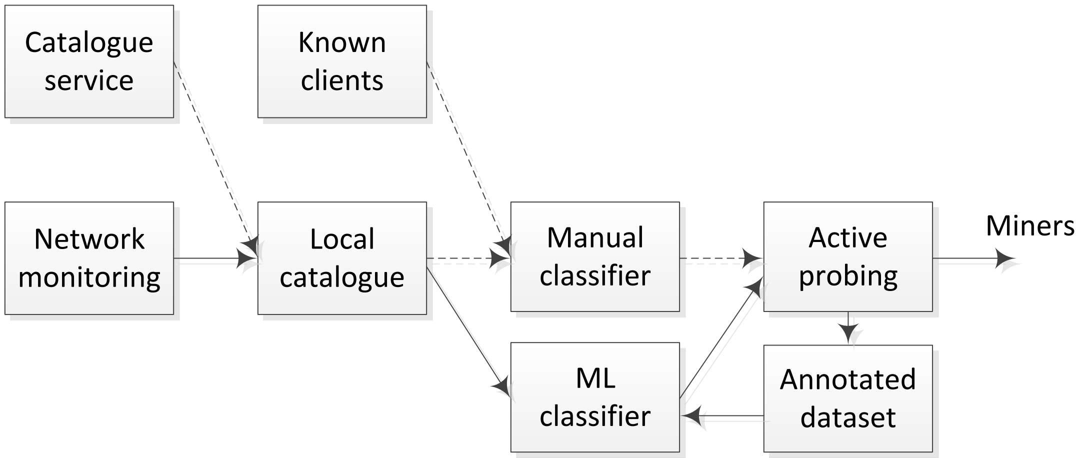


Figure 3: Data annotation process (dashed lines are utilized during the first iteration)

In order to derive the classification features, we analyzed several packet captures collected by observing communication of various mining softwares (e.g., cpuminer) and the traffic belonging to the well-known mining servers. According to our analysis, we can state that:

- Mutual communication between a miner and a mining server often lasts for several hours.
- Packets are generally small, often in the range from 40 to 120 bytes.
- Most flows contain TCP ACK+PUSH flag set.
- The destination port is either a well-known port of a different service or not well-known but definitely lower than the source port.
- Flows are generally long-lasting, often exported before its end.
- Communication is not disrupted, i.e., most flows do not contain RST flag set.

The detection itself starts with feature collection. For each triplet (source and destination IP addresses and destination port) following features are collected:

- an average number of bytes per packet derived from flows belonging to this triplet;
- an average number of packets per each flow;
- an average number of packets per minute;
- duration of a communication;

- a number of flows;
- percentage of flows with PUSH+ACK to all flows belonging to this triplet;
- percentage of flows with RST to all flows belonging to this triplet;
- percentage of flows with SYN to all flows belonging to this triplet;
- percentage of flows with FIN to all flows belonging to this triplet;
- percentage of flows with source port greater than destination port.

The triplet and its features are collected in a hash table in a memory. In order to limit the size of the hash table, triplets are evicted from the table if there is no update for more than a defined number of seconds (we set up this inactive timeout to be one hour). However, the passive detection itself is performed periodically, every 60 s.

In order not to flood the network with active probes right after the start when the list of the probed servers is almost empty (i.e., the list contains well-known mining servers only), the active probing is limited to 100 connections per second. As a result, it is not possible to rule out all false positives during the initial period. Since we want to report only the true positives, the monitoring scheme reports only verified results by the probe or by the list. Therefore false negatives may occur during the initial period. After the initial period (i.e., after the list is built), the probing becomes less intensive and reach a stable state which is below the limit on the number of probes per second.

The manually-constructed classifier is based on a cumulative score which must overcome a threshold $T$. The score is gradually increased by an increment of $1/n$, where $n$ is the number of satisfied conditions listed below:

- a number of bytes per packet per each flow is in the range 35 - 80 bytes or 105 - 110 bytes;
- a number of packets per each flow is out the range of 5 to 40;
- a number of packets per minute is in the range 2 - 8 or 40 - 5300;
- duration of communication is greater than 300 s;
- percentage of flows with ACK+PUSH to all flows is higher than 90%;
- percentage of flows with SYN to all flows is less than 5%;
- percentage of flows with RST to all flows is less than 1%;
- percentage of flows with FIN to all flows is less than 5%;
- percentage of flows with srcPort greater than dstPort is higher than 90%.

Unfortunately, it is not the case that each mining client meets all the conditions. Therefore, the detection must take into account communications satisfying only some of conditions.

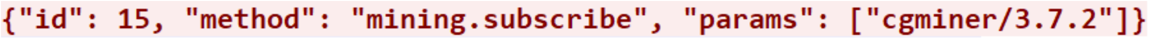
Therefore, the threshold $T$ must be set low enough so that the detector detects the majority of true positives. We trade off 100% true positive rate for the higher number of false positives since we know that the active probing will mitigate the false positives.

The machine learning detector is based on decision tree induction with a particular implementation of J48 in Weka [49]. The decision tree induction recursively selects features and their thresholds to maximize information gain contributed by the selected feature. This leads to a construction of a sub-optimal but well-performing decision tree.

No matter whether building manual classifier or training ML-classifier, it is important to bear in mind that any of them may misclassify non-miners for miners. However, the active probe mitigates these false positives. On the other hand, the number of probes must be reasonable (i.e., low enough) not to flood the network.

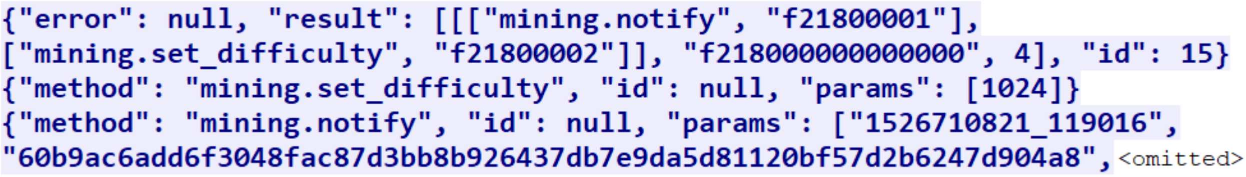
Active detection connects to a mining server pretending to be a regular miner asking for a job. If the server replies with an expected answer, then it is very likely that: a) the server is truly mining server; and b) clients connecting to this given server on this particular port are actually miners. The number of servers to check would be too high without the prior passive detection. Even our server running the detection scheme was reported by network monitoring tools as a scanner, while we were developing and testing this approach. Therefore, it is advisable to limit the number of active probes so that they fly under the radar. Based on the size of a network, this number could be 10 to 100 of probes per second.

The probe itself differs based on the mining protocol, where each probe consists of several queries targeting different cryptocurrencies (namely Bitcoin, Monero, Ethereum, Zcash). The following snippet is an example of a query for Stratum protocol:



```
{"id": 15, "method": "mining.subscribe", "params": ["cgminer/3.7.2"]}
```

The probing itself must run in parallel since it takes time a server to respond or time out. An example of a typical conforming response from a mining server is given below:



```
{"error": null, "result": [[["mining.notify", "f21800001"],
["mining.set_difficulty", "f21800002"]], "f218000000000000", 4], "id": 15}
{"method": "mining.set_difficulty", "id": null, "params": [1024]}
{"method": "mining.notify", "id": null, "params": ["1526710821_119016",
"60b9ac6add6f3048fac87d3bb8b926437db7e9da5d81120bf57d2b6247d904a8",<omitted>
```

To sum up, the active detection performs following tasks:

1. Check whether (*destination IP, destination port*)-tuple is on the triplet list of already probed servers.
2. If `false`, then probe (*destination IP, destination port*)-tuple and store the result in the triplet list together with the current timestamp.
3. If `true`, then check the timestamp
    - If the timestamp is younger than 7 days, utilize the stored result.

| | Flows | Packets | Bytes | Interval |
|---|---|---|---|---|
| Train.-eval. | 16M | 117M | 54G | 2018/02/02 14:00 - 14:15 |
| Real | 3.6G | 27.8G | 99.5T | 2018/02/09 14:00 - 20:00 |

Table 3: Volumes of utilized data-sets.

- Otherwise, probe the (*destination IP, destination port*)-tuple and update the triplet list including a new timestamp.

### *4.2. Evaluation*

#### *4.2.1. Data-sets*

The experiments were carried out on data collected in Czech National Research and Educational Network connecting more than 30 organizations (e.g., universities, labs, hospitals) including more than 400 thousand users altogether. However, only 3 subnets conforming to 3 large organizations (over 50 thousand users) were considered for training and evaluation of data-set in order to thin the amount of training data and to allow for manual verification of the results. The observation points are located on the peering links with internet exchange points or other national networks. Therefore, the communication between entities within the national network is not part of data.

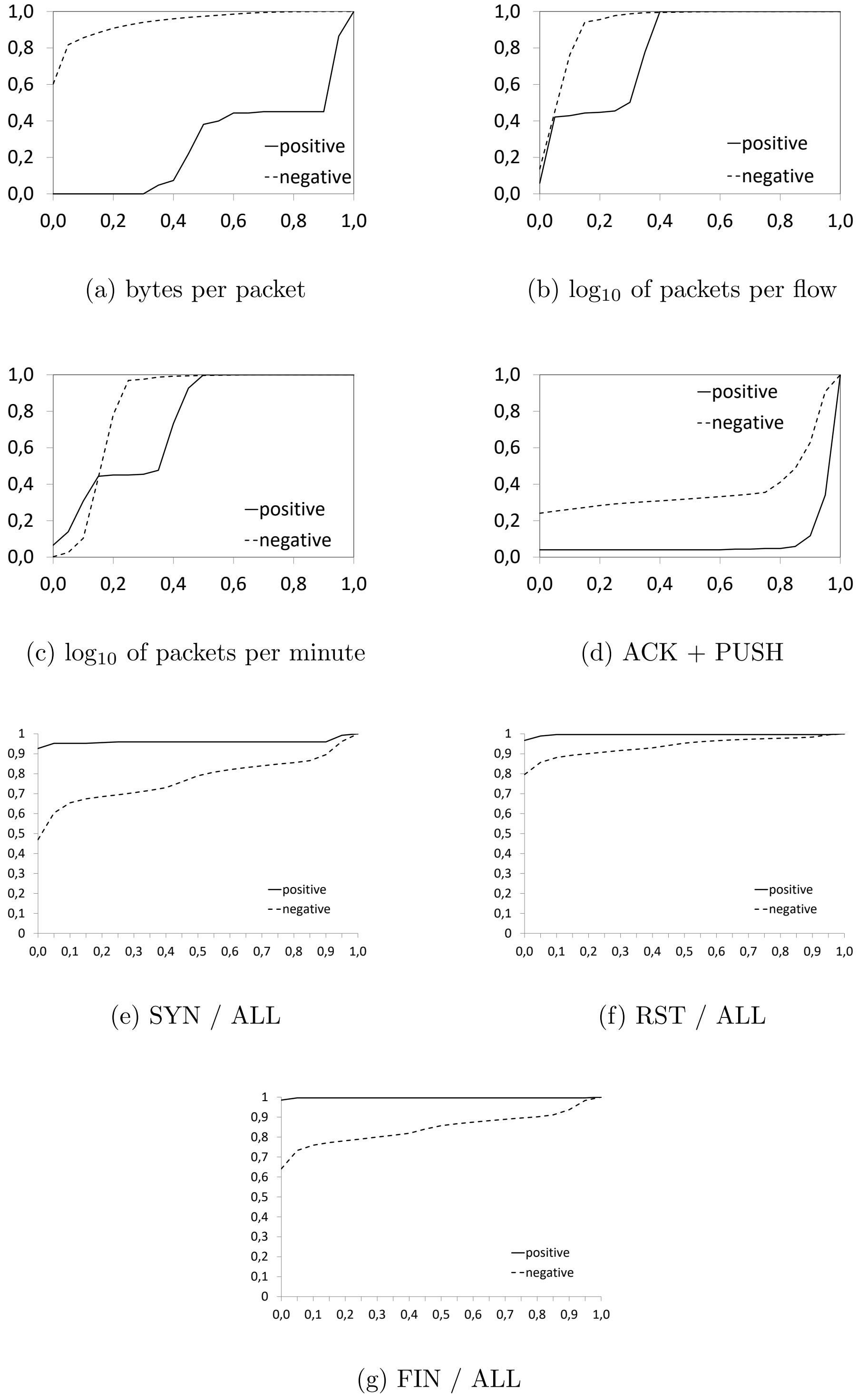
We created an offline data-set consisting of the feature vectors and their classification (i.e., mining/non-mining client). The data-set can be downloaded from the results reproduction page [48]. This data-set was created by streamwise[12] aggregation of IP flow records collected from the backbone described above. The set is annotated utilizing the proposed detection algorithm described above. Therefore, if the triplets are annotated as mining, we know they are true positives (later on referred to as positives only) while the rest (referred to as negatives) are either true negatives (i.e., negative verification by the probe) or potential false negatives (i.e., not verified by the probe).

#### *4.2.2. Experiments*

The feature selection was performed by the analysis of traffic belonging to a known mining client bearing in mind that only basic flow records allow for general deployment. The selected features were evaluated on the annotated data-set described in the previous section. Figure 4 depicts cumulative normalized distribution function (CDF) of the selected features belonging to the triplets as collected by during passive detection. Each feature is assigned two functions – one for samples annotated as miners (positive) and one for samples annotated as other (negative).

In Figure 4a we can observe that the distribution of positives and negatives differs significantly. Please note that the x-axis of this figure was shortened to display the detail. More than 50% of positives accounts for packets of size 105 to 110 bytes, and another 40% accounts for packets of average size from 35 to 80 bytes. On the other hand, 20% of negative

[12] Continuous stream of data rather than fixed length intervals as described in [50]



(a) bytes per packet

(b) $\log_{10}$ of packets per flow

(c) $\log_{10}$ of packets per minute

(d) ACK + PUSH

(e) SYN / ALL

(f) RST / ALL

(g) FIN / ALL

Figure 4: Cumulative distribution function of features

triplets consists of packets of a size smaller than 35 bytes, another 20% accounts for packets of size 80 to 105 bytes, and another 20% accounts for packets larger than 110 bytes.

Distribution of a number of packets per flow (in Figure 4b) and packets per minute (in Figure 4c) are correlated. Positive triplets belong to either a weak group with a low number of packets or a strong group with a high number of packets per flow as well as per minute. Moreover, in the case of negative triplets there exists less than 1% of instances with an extremely high number of packets.

Figure 4d displays the distribution of the rate of flows with push and ack flag set to all flows. The CDF of negatives shows that there is more than 20% of triplets without any ack and push flags set. The CDF of negatives slowly rises to 40% at the rate of 0.8. On the other hand, there are nearly zero positives with a rate lower than 0.8 while the majority of positives exhibits rate higher than 0.9. The opposite holds for distribution of other flags depicted in Figures 4e, 4f, 4g. Positives with zero rate account for the majority of its samples while negatives only for 50% in case of SYN, for 80% in case of RST and 65% in case of FIN.

Last but not least, the detection works upon requests from miners to the mining server. To this end, the feature capturing the rate between the number of flows with source port greater than destination port to all the flows aims at distinguishing between a prevalent request or response communication. By keeping the rate as one of the features and not an a priori condition we allow the detection algorithm to detect miners even in those triplets that aggregate responses if the other features recognize that triplet as potentially positive. This makes the algorithm more robust but for the price of more false positives and the higher number of triplets to work with. The distribution of positives as well as of negatives are almost identical.

Obviously, none of the features considered is good enough to directly distinguish between positives and negatives. Therefore, the detection algorithm must combine results of several features to improve detection results.

We evaluate two detection algorithms, one designed manually and one based on machine learning, both described in Section 4.1.

As the training and evaluation data-set contains a significantly lower number of mining communications (273 positive triplets) than of other communications (356,574 negative triplets), we apply *ClassBalancer* filter to balance the weight of both sets in order not to overtrain the detector on non-mining communications. The training process is set up with the following parameters 5-fold cross-validation, and the number of instances in a leaf must be at least 100 (other parameters are kept default).

The higher number of instances in a leaf, the less deeper tree is generated by the training process. And the less deeper tree, the faster is its evaluation as well as the tree is less overfitted to a particular data-set.

The confusion matrix of the resulting detector is depicted in Table 4. It shows ML-detector marks: a) mining communication as mining in most cases; b) another communication as other; c) except in 2.6% of cases where communication is misclassified as mining (i.e., false positive); and d) although the ML-detector misclassifies 4.7% of mining triplets, it is a significant improvement over the manually constructed classifier.

| Classified as | other | mining |
|---|---|---|
| other | 97.4% | 2.6% |
| mining | 4.7% | 95.3% |

Table 4: Confusion matrix of classified triplets by the decision tree.

| Classified as | other | mining |
|---|---|---|
| other | 91.4% | 8.6% |
| mining | 10.3% | 89.7% |

Table 5: Confusion matrix of classified triplets by the manual classifier.

The confusion matrix of the manual classifier is depicted in Table 5.

Although the manual classifier was utilized to collect the annotated training and evaluation data-set, it fails to correctly classify approx. 10% of each class. This discrepancy is caused by the mining servers list which is gradually built whenever passive detection is successful. Once this list is populated, the manual classifier also identifies such mining triplets that would otherwise not meet enough conditions of the classifier.

## 5. Catalog of Mining Pools

We were also looking for a more lightweight solution suitable even for small corporate networks lacking capacities to install dedicated probes performing our active/passive traffic monitoring employing machine learning. We want to offer conclusive detection results with a minimum set of input information.

Network administrator and law enforcement agent (i.e., our main actors for mining detection use-case) shall have basic NetFlow records of investigated device/network segment. These records contain at least source/destination IP addresses, source/destination ports and a protocol identifier. The reasoning behind our second approach is following. If we know IP address of mining pool server, then we can reliably distinguish between mining and non-mining connections. Moreover, if we are aware of the port number employed by a pool operator, then we can tell what cryptocurrency is being mined through the connection.

### *5.1. Design*

Both mining server's hostname and port number are publicly available (except mining malware cases) on pool's webpage because they are necessary for successful setup of the mining process. Without this vital information, the miner would not be able to configure mining software properly.

Based on these premises, we have decided to manually collect all mining software configurations announced by the biggest mining pools for several important cryptocurrencies. We gathered all these data in a database, which is accessible through a web application called sMaSheD (Mining Server Detector of cryptocurrency pools). In the rest of this subsection, we briefly outline some of the design choices that we have made during the sMaSheD development.

There are hundreds of coins (and tokens) available in cryptocurrency universe. In order to choose coins supported by sMaSheD, we did due diligence on ”the most popular” cryptocurrencies taking into account public news [51], dedicated reports [52] and consultations with our LEA partners. Bitcoin is dominating this ladder due to its importance (e.g., around 80 million USD worth of Bitcoins stolen from a hacked cloud mining service provider in December 2017 [53]). However, Monero becomes more and more used by malwares because of its anti-forensic features, which help to cover criminal’s tracks (e.g., nearly 5% of all Moneros in circulation worth of 175 million USD were mined using malware [54]). The third is Ethereum thanks to smart-contracts and popularity among token developers (e.g., 30 million USD worth of Ethereum stolen by a wallet breach in July 2017 [55]). No matter on the current set of cryptocurrencies, sMaSheD is designed to be a generic catalog of mining pools which should be easy to maintain and operate.

As a next step, we investigated mining distribution among available pools for each chosen cryptocurrency. The majority of pools add their signature into the freshly mined block. This marking allows to account the success rate of each participating pool. Moreover, pools are announcing their overall hashrate performance publicly. By combining these data, we receive quite a reliable overview about more and less important pools for every cryptocurrency. Anyone can obtain these data from dedicated web-pages, e.g. [56] for Bitcoins, [57] for Litecoin, and [58] for Ethereum.

Mining software configurations are collected by web scraping the content of pool web pages. This procedure is currently performed manually by sMaSheD administrators. However, we aim at the automation of this process in the near future. The following set of information is being collected for every pool (from web-pages similar to Appendix A):

- the name of the pool and its home URL;
- the list of pool servers identified by FQDN including ports associated with a mined cryptocurrencies;
- every mining server FQDN is resolved onto a list of IPv4/IPv6 addresses.

Nevertheless, some pools are private (e.g., Bitfury pool with roughly 2% hashrate share [59] run by a company[13] with the same name producing ASIC mining solutions). The operator of such pool does not maintain any publicly available web page, which makes any web scraping of configuration impossible. Hence, sMaSheD catalog does not contain a complete list of pools for a given cryptocurrency. Fortunately, private pools constitute a fraction of overall network hashrate.

Mining server FQDNs may include information about location, mined cryptocurrency (e.g., `eth-us2.dwarfpool.com`) or employed algorithm (e.g., `sha256.eu.nicehash.com`).

[13] For more, visit https://bitfury.com/

However, a single visit of a pool's web page does not take into account the changing nature of pool infrastructure (i.e., mining service availability on new/old servers). Thus, we conducted experiments with automated discovery of FQDNs. We tried to generate hostnames as permutations from a set of keywords, which includes cryptocurrency abbreviations, country codes and pool domains. Unfortunately, this approach: 1) generated way too many false hostnames; and 2) verification of generated hostnames is a time consuming process. DNS allows listing of all records (including hostname A and AAAA records) through zone transfer[14], but this is not applicable for our use-case.

Pool operators provide a server FQDNs, which are resolved by miners onto various IP address based on miner's geolocation. Based on deployment (see Figure 1), DNS may resolve a single FQDN onto many IP address (e.g., `stratum.slushpool.com`) in order to guarantee high-availability of mining service. sMaSheD tries to keep the list of these IP addresses as up-to-date as possible. It is a necessity especially for pools leveraging cloud deployment because cloud providers often rotate available IP addresses among customers' virtual machines. An IP address of mining server today can belong to a completely different machine tomorrow. Because of this changing nature and since a single FQDN may actually represent a set of load-balancing mining servers, sMaSheD periodically renews the list of IP addresses associated with each mining server within the system.

In order to provide more reliable results if a given IP address belongs to a mining server or not, we developed probing similar to one described in Section 4. This probing repeats for all known pools (and their mining servers) every 3 hours as a background task asynchronous to web application run. During every periodic check of mining server, sMaSheD sends crafted mining protocol message and waits for the response. If counterparty reacts properly (with a message containing work package), then it confirms that this device is really a pool's mining server.

Probing is supported for Stratum and GetBlockTemplate mining protocols. sMaSheD is probing single server for both of these protocols. GetWork is also implemented, but we were not able to test it since this protocol is deprecated and not employed by any pool within our system. There are three probing return codes:

- DOWN - Probing failed because connection had not been even established. This occurs when a port on the server is closed, or some middle-box is blocking the connection.
- LISTEN - The connection was accepted on a specified port, but the alleged server returns an empty response. This happens when a) server is using different mining protocol than tested one; b) port is opened but bound to a different application.
- UP - Probing succeeded because mining server responded with mining protocol message containing valid content. Message validity depends on employed mining protocol and consists of multiple value presence tests (e.g., *error*, *result* and other JSON fields). This validator can be easily extended to support changes or even new mining protocols.

[14] For more about DNS records and zones, please read RFC 1035 and related ones.

Probing return code is usually accompanied with a verbose result (i.e., destination unreachable, unknown method, mining subscribe). sMaSheD records each probing attempt, which creates a history of service availability for a given mining server. These temporal data can later prove that IP address was used by a mining server (at least from the perspective of sMaSheD).

### *5.2. Evaluation*

In order to validate probing results, we compared the behavior of sMaSheD with official mining software. We decided to use cgminer 3.7.2 [60] because it is well-established and supports all available mining protocols.

We tested both tools over the same set of mining servers and recorded communication into PCAP file. We compared connection success rate (based on textual console outputs) and messages exchanged between miner (either sMaSheD or cgminer) and mining server. We did not find any differences for detected mining servers when comparing sMaSheD and cgminer connection attempts. Both applications used the same set (1983 entries) of IP addresses and ports of alleged mining servers.

sMaSheD system does not send any authentication credentials towards a pool upon the check, the basic response for mining subscription message is enough to mark a device as mining server positively. This is illustrated in Wireshark message captures depicted in Figure 5 and Figure 6.

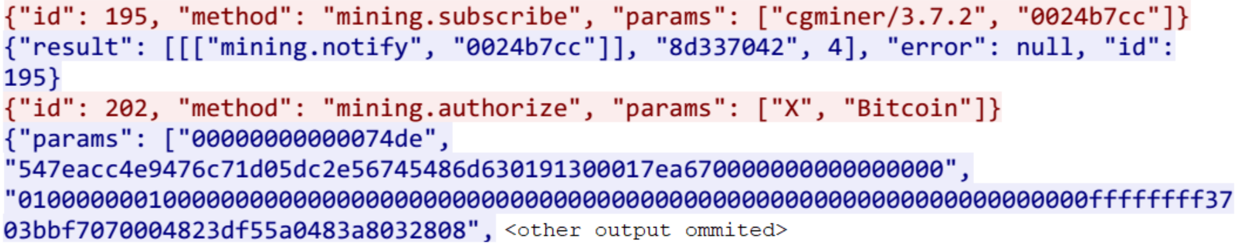
sMaSheD is coded as a web application employing PHP framework Laravel 5.6 with front-end based on Bootstrap 3. The system is operated in a Docker container on CentOS 7. All source codes are available on [61].

Figure in Appendix B explains relations between data available in the system. All data can be easily obtained in JSON via REST API. The database of sMaSheD currently (in August 2018) contains:

- 13 cryptocurrencies mined on 18 various port numbers;
- 56 pools operating 184 servers with 479 addresses;
- 2284 probing associations (dubbed as mining properties) and it takes approximately 20 minutes to them (i.e., check all IP-port tuples of all known mining servers for both Stratum and GetBlockTemplate).

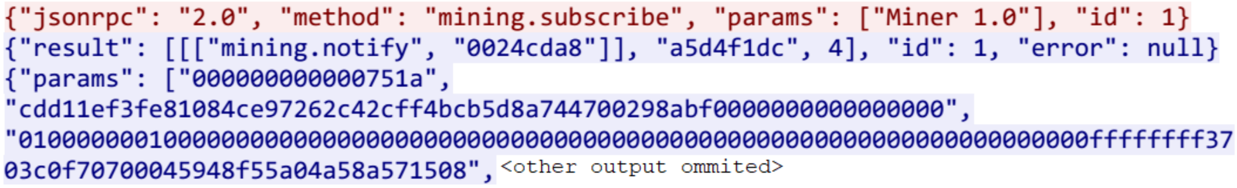
Mining pools catalog sMaSheD (implemented as the second approach solution for miners detection):

- offers access to all data in JSON format through REST calls;
- periodically probes available pools' servers whether they provide mining services and accounts the result;
- allows privileged users to update database according to the current situation (i.e., add newly established pools).



```
{"id": 195, "method": "mining.subscribe", "params": ["cgminer/3.7.2", "0024b7cc"]}
{"result": [[["mining.notify", "0024b7cc"]], "8d337042", 4], "error": null, "id":
195}
{"id": 202, "method": "mining.authorize", "params": ["X", "Bitcoin"]}
{"params": ["00000000000074de",
"547eacc4e9476c71d05dc2e56745486d630191300017ea670000000000000000",
"01000000010000000000000000000000000000000000000000000000000000000000000000ffffffff37
03bbf7070004823df55a0483a8032808", <other output ommited>
```

Figure 5: Message exchange between cgminer (red) and pool (blue)



```
{"jsonrpc": "2.0", "method": "mining.subscribe", "params": ["Miner 1.0"], "id": 1}
{"result": [[["mining.notify", "0024cda8"]], "a5d4f1dc", 4], "id": 1, "error": null}
{"params": ["000000000000751a",
"cdd11ef3fe81084ce97262c42cff4bcb5d8a744700298abf0000000000000000",
"01000000010000000000000000000000000000000000000000000000000000000000000000ffffffff37
03c0f70700045948f55a04a58a571508", <other output ommited>
```

Figure 6: Message exchange between sMaSheD (red) and pool (blue)

## 6. Conclusion

In this paper, we provided an in-depth analysis of cryptocurrency mining operation. We designed and implemented sMaSheD catalog and passive-active detection approaches to detect mining devices within the network. We tested the feasibility of these approaches on real-life data as well as published data-sets utilized in this article under open access policy. We conclude that catalog and passive-active approach are complementary.

The results of passive-active detection approach show that although there is a high number of false positives after the passive detection, it is sufficiently low to enable active verification of the results. In comparison to the pure catalog approach, passive-active detection is capable of discovering emerging or deliberately hidden pools. As such it should serve Security Operation Centers, CSIRT, and network security service providers to populate their cyber threat intelligence systems.

The goal of our sMaSheD system is to become a tool as valuable for network administrators and LEA operatives as what is ExoneraTor [62] application for TOR overlay network. sMaSheD prototype including a large data-set is available at [63] (see Appendix C for demo screenshots). Moreover, anyone can deploy own installation from sources [61] and feed it with custom pools. Online catalog offers a curated list of the most popular pools and their servers. Data available in sMaSheD offer a neat solution for following use-cases:

- create an access control list that will block unwanted traffic mining traffic (based on IPs and ports known to sMaSheD);
- detect the presence of miners via inspection of their DNS queries (based on FQDNs and IPs contained in DNS requests and answers);
- data-retention proof about mining service availability for a given IP+port tuple.

We would like to automatize metadata collection for sMaSheD by a periodic scraping of relevant web pages. Currently, the information provided by our catalog is updated manually, which makes the system less dynamic than we would appreciate. Regarding additional future work, we also consider other strategies on how to probe and positively identify pool servers based on different mining protocol messages. Last but not least, we are constantly adding new cryptocurrencies, pools, and servers as they appear in publicly disclosed announcements of illicit mining activities.

This article has been supported by the Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science (no. LQ1602). Authors also want to acknowledge work done by Jakub Kelečeni, Erik Šabík, and Martin Cagaš, the students of Brno University of Technology.

## References

[1] R. Raeesi, The Silk Road, Bitcoins and the Global Prohibition Regime on the International Trade in Illicit Drugs: Can this Storm Be Weathered?, Glendon Journal of International Studies/Revue d'études internationales de Glendon 8 (1-2) (2015) 20. doi:http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.960.6973.

[2] R. Grinberg, Bitcoin: An innovative alternative digital currency, Hastings Sci & Tech LJ 4 (2012) 159–208.

[3] B. Johnson, The advantages and disadvantages of the deep web, tor network, virtual currencies and the regulatory challenges thereof, Master's thesis, Utica College, USA (2014).

[4] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, https://bitcoin.org/bitcoin.pdf, (published 31 October 2008, accessed 30 May 2017).

[5] Bitcoin.org, Bitcoin - Open source P2P money, https://bitcoin.org/en, (accessed 11 August 2018).

[6] CoinMarketCap.com, Bitcoin (BTC) — CryptoCurrency Market Capitalizations, https://coinmarketcap.com/currencies/bitcoin, CoinMarketCap OpCo, LLC, (accessed 30 May 2017).

[7] S. T. Ali, D. Clarke, P. McCorry, Bitcoin: Perils of an unregulated global p2p currency, in: Cambridge International Workshop on Security Protocols, Springer, 2015, pp. 283–293.

[8] A. Hern, Student uses university computers to mine dogecoin, https://www.theguardian.com/technology/2014/mar/04/dogecoin-bitcoin-imperial-college-student-mine, The Guardian, (published 4 March 2014, accessed 31 July 2018).

[9] D. Nield, Student secretly used harvard's supercomputer to mine dogecoin, https://www.digitaltrends.com/computing/student-secretly-used-harvards-supercomputer-mine-dogecoin/, (accessed 31 July 2018).

[10] BBC.com, Russian nuclear scientists arrested for 'bitcoin mining plot', https://www.bbc.com/news/world-europe-43003740, (published 9 February 2018, accessed 31 July 2018).

[11] K. Frouzová, J. Zelenka, Pirát ve služebním bytě těžil kryptoměny, sněmovnu zaskočil účet za elektřinu. byla mi zima, hájí se!, https://zpravy.aktualne.cz/domaci/pirat-ve-sluzebnim-byte-tezil-kryptomeny-snemovnu-zaskocil-u/r 5c8095665ea211e8b19b0cc47ab5f122/, Aktualne.cz, (published 23 May 2018, accessed 31 July 2018).

[12] K. O'Dwyer, D. Malone, Bitcoin mining and its energy footprint, in: IET Conference Proceedings, The Institution of Engineering & Technology, 2014, p. 6.

[13] A. de Vries, Bitcoin's growing energy problem, Joule 2 (5) (2018) 801–805.

[14] N. T. Courtois, M. Grajek, R. Naik, The unreasonable fundamental incertitudes behind bitcoin mining, arXiv preprint arXiv:1310.7935.

[15] J. A. Kroll, I. C. Davey, E. W. Felten, The economics of bitcoin mining, or bitcoin in the presence of adversaries, in: Proceedings of WEIS, Vol. 2013, 2013, p. 11.

[16] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, J. S. Rosenschein, Bitcoin mining pools: A cooperative game theoretic analysis, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 919–927.

[17] I. Eyal, E. G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, in: International conference on financial cryptography and data security, Springer, 2014, pp. 436–454.

[18] A. Juels, A. Kosba, E. Shi, The ring of gyges: Investigating the future of criminal smart contracts, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 283–295.

[19] N. Hampton, Z. A. Baig, Ransomware: Emergence of the cyber-extortion menace, in: The Proceedings of 13th Australian Information Security Management Conference, Edith Cowan University, 2015, pp. 47–56.

[20] S. T. Ali, P. McCorry, P. H.-J. Lee, F. Hao, Zombiecoin: powering next-generation botnets with bitcoin, in: International Conference on Financial Cryptography and Data Security, Springer, 2015, pp. 34–48.

[21] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, E. Kirda, Cutting the gordian knot: A look under the hood of ransomware attacks, in: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, 2015, pp. 3–24.

[22] D. Y. Huang, H. Dharmdasani, S. Meiklejohn, V. Dave, C. Grier, D. McCoy, S. Savage, N. Weaver, A. C. Snoeren, K. Levchenko, Botcoin: Monetizing stolen cycles., in: Proceedings of the 2014 Network and Distributed System Security Symposium (NDSS), 2014, p. 16. doi:https://doi.org/10.14722/ndss.2014.23044.

[23] J. D'Herdt, Detecting Crypto Currency Mining in Corporate, Tech. rep., SANS Institute (January 2015).
URL `https://www.sans.org/reading-room/whitepapers/threats/detecting-crypto-currency-mining-corporate-environments-35722`

[24] V. Pus, L. Kekely, M. Spinler, V. Hummel, J. Palicka, HANIC100G: Hardware accelerator for 100 Gbps network traffic monitoring, Tech. rep., CESNET (February 2014).
URL `https://www.cesnet.cz/wp-content/uploads/2015/01/hanic-100g.pdf`

[25] B. Claise, Cisco Systems NetFlow Services Export Version 9, RFC 3954, IETF (October 2004).
URL `https://tools.ietf.org/html/rfc3954`

[26] P. A. B. Claise, B. Trammell, Specification of the IP Flow Information Export (IPFIX) Protocol, RFC 7011, IETF (September 2013).
URL `https://tools.ietf.org/html/rfc7011`

[27] B. Li, J. Springer, G. Bebis, M. Hadi Gunes, Review: A survey of network flow applications, J. Netw. Comput. Appl. 36 (2) (2013) 567–581. doi:10.1016/j.jnca.2012.12.020.
URL `http://dx.doi.org/10.1016/j.jnca.2012.12.020`

[28] R. Hofstede, L. Hendriks, A. Sperotto, A. Pras, Ssh compromise detection using netflow/ipfix, SIGCOMM Comput. Commun. Rev. 44 (5) (2014) 20–26. doi:10.1145/2677046.2677050.
URL `http://doi.acm.org/10.1145/2677046.2677050`

[29] F. Silveira, C. Diot, N. Taft, R. Govindan, Astute: Detecting a different class of traffic anomalies, in: Proceedings of the ACM SIGCOMM 2010 Conference, SIGCOMM '10, ACM, New York, NY, USA, 2010, pp. 267–278. doi:10.1145/1851182.1851215.
URL `http://doi.acm.org/10.1145/1851182.1851215`

[30] C. Livadas, R. Walsh, D. Lapsley, W. T. Strayer, Usilng machine learning technliques to identify botnet traffic, in: Proceedings. 2006 31st IEEE Conference on Local Computer Networks, 2006, pp. 967–974. doi:10.1109/LCN.2006.322210.

[31] CryptoCompare, Compare bitcoin, ethereum and litecoin mining pools, `https://www.cryptocompare.com/mining/#/pools`, (accessed 25 August 2017).

[32] Q. H. Dang, Secure Hash Standard (SHS), Federal Information Processing Standards Publication FIPS

Pub 180-4, NIST (March 2012).
URL http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[33] C. Percival, S. Josefsson, The scrypt Password-Based Key Derivation Function, RFC 7914 (Aug. 2016). doi:10.17487/RFC7914.
URL https://rfc-editor.org/rfc/rfc7914.txt

[34] E. Duffield, D. Diaz, Dash: A Privacy-Centric Crypto-Currency, https://github.com/dashpay/dash/wiki/Whitepaper, GitHub, Inc., (published 5 June 2018, accessed 11 August 2018).

[35] J. Matonis, The bitcoin mining arms race: Ghash.io and the 51% issue, https://www.coindesk.com/bitcoin-mining-detente-ghash-io-51-issue/, CoinDesk, Inc., (published 17 July 2014, (accessed 24 August 2017).

[36] S. Higgins, Us government shuts down embattled mining firm butterfly labs, https://www.coindesk.com/us-government-shuts-embattled-mining-firm-butterfly-labs/, CoinDesk, Inc., (published 23 September 2014, accessed 31 July 2018).

[37] S. Higgins, Cointerra seeks out-of-court settlement to class action lawsuit, https://www.coindesk.com/cointerra-mediation-class-action-lawsuit/, CoinDesk, Inc., (published 24 June 2014, accessed 31 July 2018).

[38] J. I. Wong, Uncertainty builds as alpha technology misses another shipping deadline, https://www.coindesk.com/uncertainty-builds-alpha-technology-misses-another-shipping-deadline/, CoinDesk, Inc., (published 10 December 2014, accessed 31 July 2018).

[39] D. D. Pardo, $46k spent on bitcoin mining hardware: The final reckoning, https://www.coindesk.com/46k-spent-mining-hardware-final-reckoning/, CoinDesk, Inc., (published 27 July 2014, accessed 31 July 2018).

[40] CryptoCompare, Compare bitcoin, ethereum and litecoin mining contracts, https://www.cryptocompare.com/mining/#/contracts, (accessed 24 August 2017).

[41] Bitcoin.it, Mining hardware comparison - bitcoin wiki, https://en.bitcoin.it/wiki/Mining_hardware_comparison, (published 22 January 2018, accessed 11 August 2018).

[42] Bitcoin.it, Non-specialized hardware comparison, https://en.bitcoin.it/wiki/Non-specialized_hardware_comparison, (published 4 August 2015, accessed 24 August 2017.

[43] L. Dashjr, getblocktemplate - Fundamentals, BIP 22, Bitcoin Project (February 2012).
URL https://github.com/bitcoin/bips/blob/master/bip-0022.mediawiki

[44] L. Dashjr, getblocktemplate - Pooled Mining, BIP 23, Bitcoin Project (February 2012).
URL https://github.com/bitcoin/bips/blob/master/bip-0023.mediawiki

[45] M. Palatinus, Stratum Mining Protocol, https://slushpool.com/help/manual/stratum-protocol, Slushpool.com, (accessed 17 August 2018).

[46] M. Palatinus, Homepage – slushpool.com, https://slushpool.com, (accessed 17 August 2018).

[47] JSON-RPC Working Group, JSON-RPC 2.0 Specification, http://www.jsonrpc.org/specification, (published 4 January 2013, accessed 17 August 2017).

[48] V. Vesely, M. Zadnik, Pcap files and data-sets for digital investigation article, https://github.com/nesfit/DI-cryptominingdetection, GitHub, Inc., (published 11 August 2018, accessed 11 August 2018).

[49] E. Frank, M. A. Hall, I. H. Witten, The weka workbench, in: Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", 2016.

[50] T. Cejka, V. Bartos, M. Svepes, Z. Rosa, H. Kubatova, Nemea: A framework for network traffic analysis, in: 2016 12th International Conference on Network and Service Management (CNSM), 2016, pp. 195–201. doi:10.1109/CNSM.2016.7818417.

[51] O. Kharif, The criminal underworld is dropping bitcoin for another currency, https://www.bloomberg.com/news/articles/2018-01-02/criminal-underworld-is-dropping-bitcoin-for-another-currency, Bloomberg L.P., (published 2 January 2018, accessed 3 August 2018).

[52] Carbon Black, Cryptocurrency Gold Rush on the Dark Web, Tech. rep., Carbon Black, Inc. (June

2018).
URL https://www.carbonblack.com/wp-content/uploads/2018/06/Cryptocurrency_Gold_Rush_on_the_Dark_Web_Carbon_Black_Report_June_2018.pdf

[53] NiceHash.com, Official press release statement by nicehash, https://www.reddit.com/r/NiceHash/comments/7i0s6o/official_press_release_statement_by_nicehash/, note = H-BIT, d.o.o., (published 6 December 2017, accessed 3 August 2018).

[54] J. Grunzweig, The rise of cryptocurrency miners, https://researchcenter.paloaltonetworks.com/2018/06/unit42-rise-cryptocurrency-miners/, Palo Alto Networks, Inc., (published 11 June 2018, accessed 3 August 2018).

[55] W. Zhao, $30 million: Ether reported stolen due to parity wallet breach, https://www.coindesk.com/30-million-ether-reported-stolen-parity-wallet-breach/, CoinDesk, Inc., (published 19 July 2017, accessed 3 August 2018).

[56] Blockchain.info, Hashrate distribution an estimation of hashrate distribution amongst the largest mining pools, https://www.blockchain.com/en/pools?timespan=4days, Blockchain Luxembourg S.A, accessed 3 August 2018).

[57] Litecoinpool.org, Hash rate distribution (last 22 hours), https://www.litecoinpool.org/pools, accessed 3 August 2018).

[58] Etherscan.io, Ethereum top 25 miners by blocks, https://etherscan.io/stat/miner?range=7&blocktype=blocks, accessed 3 August 2018).

[59] Btc.com, Bitfury - pool - btc.com, https://btc.com/stats/pool/BitFury, Bitmain, accessed 3 August 2018).

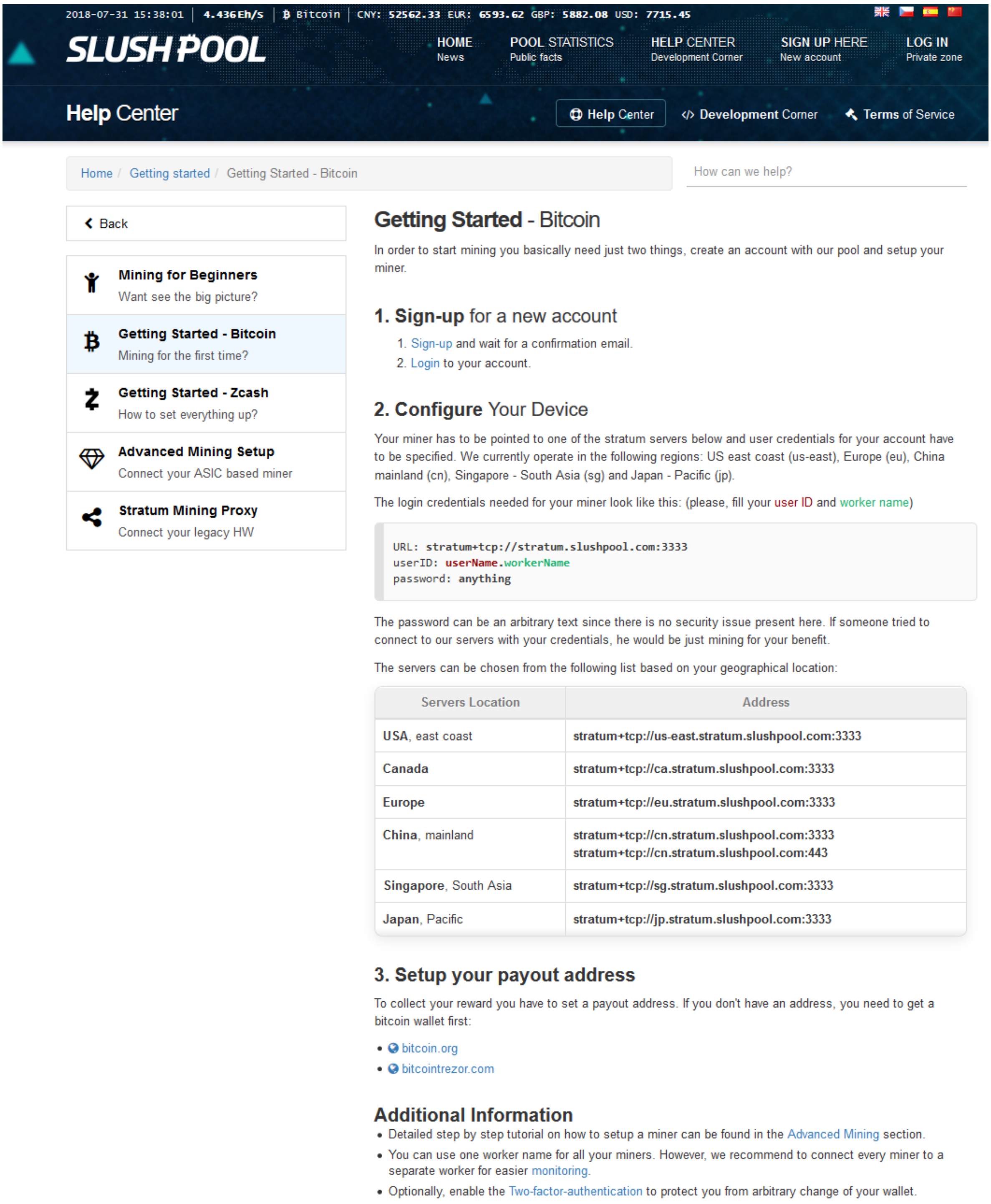
[60] C. Kolivas, Asic and fpga miner in c for bitcoin, https://github.com/ckolivas/cgminer, GitHub, Inc., (published 5 June 2018, accessed 4 August 2018).

[61] V. Vesely, Mining server detector of cryptocurrency pools, https://github.com/kvetak/sMaSheD/, GitHub, Inc., (published 4 August 2018, accessed 5 August 2018).

[62] The Tor Project, Inc., Exonerator, https://exonerator.torproject.org/, (accessed 11 August 2018).

[63] V. Vesely, smashed - online catalog of cryptocurrency mining pools, http://smashed.fit.vutbr.cz/, Brno University of Technology, accessed 5 August 2018).

# Appendix A. Mining software configuration



## Getting Started - Bitcoin

In order to start mining you basically need just two things, create an account with our pool and setup your miner.

### 1. Sign-up for a new account

1. Sign-up and wait for a confirmation email.
2. Login to your account.

### 2. Configure Your Device

Your miner has to be pointed to one of the stratum servers below and user credentials for your account have to be specified. We currently operate in the following regions: US east coast (us-east), Europe (eu), China mainland (cn), Singapore - South Asia (sg) and Japan - Pacific (jp).

The login credentials needed for your miner look like this: (please, fill your user ID and worker name)

```
URL: stratum+tcp://stratum.slushpool.com:3333
userID: userName.workerName
password: anything
```

The password can be an arbitrary text since there is no security issue present here. If someone tried to connect to our servers with your credentials, he would be just mining for your benefit.

The servers can be chosen from the following list based on your geographical location:

| Servers Location | Address |
|---|---|
| **USA**, east coast | **stratum+tcp://us-east.stratum.slushpool.com:3333** |
| **Canada** | **stratum+tcp://ca.stratum.slushpool.com:3333** |
| **Europe** | **stratum+tcp://eu.stratum.slushpool.com:3333** |
| **China**, mainland | **stratum+tcp://cn.stratum.slushpool.com:3333**<br>**stratum+tcp://cn.stratum.slushpool.com:443** |
| **Singapore**, South Asia | **stratum+tcp://sg.stratum.slushpool.com:3333** |
| **Japan**, Pacific | **stratum+tcp://jp.stratum.slushpool.com:3333** |

### 3. Setup your payout address

To collect your reward you have to set a payout address. If you don't have an address, you need to get a bitcoin wallet first:

- bitcoin.org
- bitcointrezor.com

### Additional Information

- Detailed step by step tutorial on how to setup a miner can be found in the Advanced Mining section.
- You can use one worker name for all your miners. However, we recommend to connect every miner to a separate worker for easier monitoring.
- Optionally, enable the Two-factor-authentication to protect you from arbitrary change of your wallet.

Figure A.7: Example of mining software setup taken from SlushPool

# Appendix B. E-R Diagram

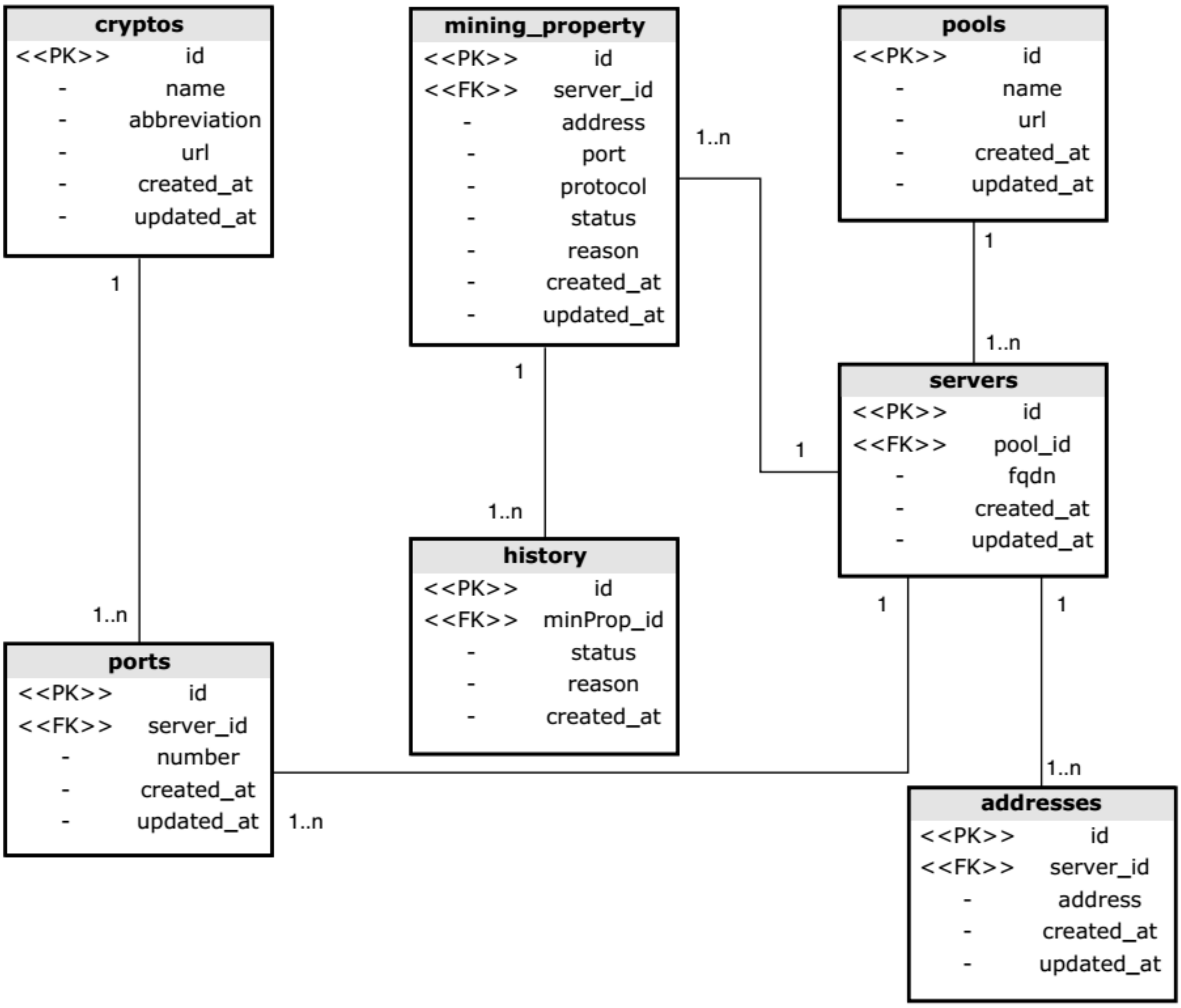
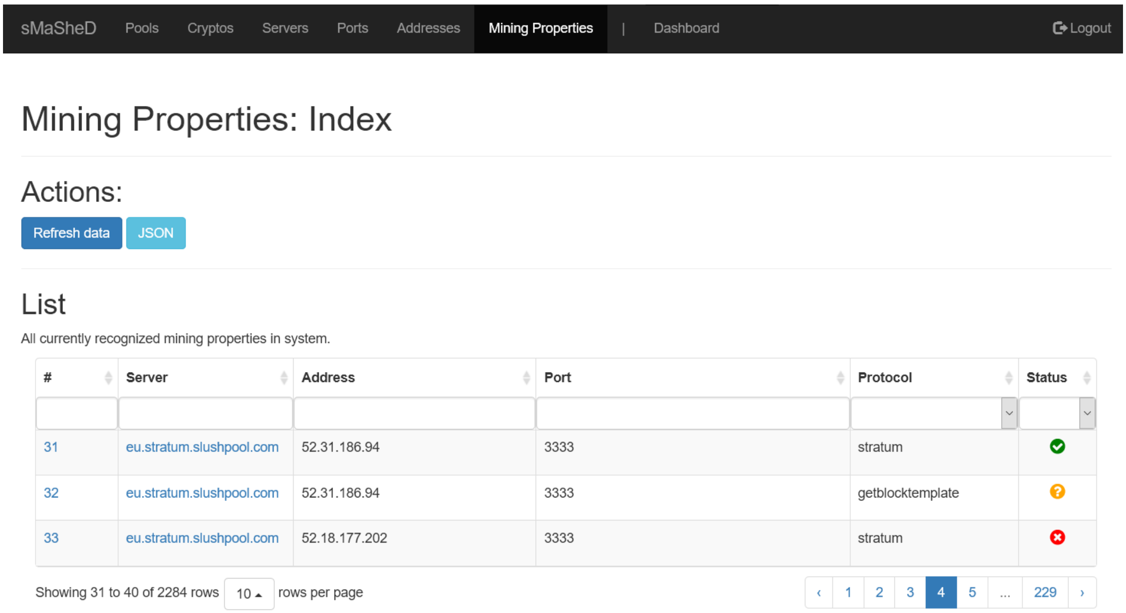


Figure B.8: Entity Relationship diagram for sMaSheD database

# Appendix C. sMaSheD Demo



## Mining Properties: Index

### Actions:

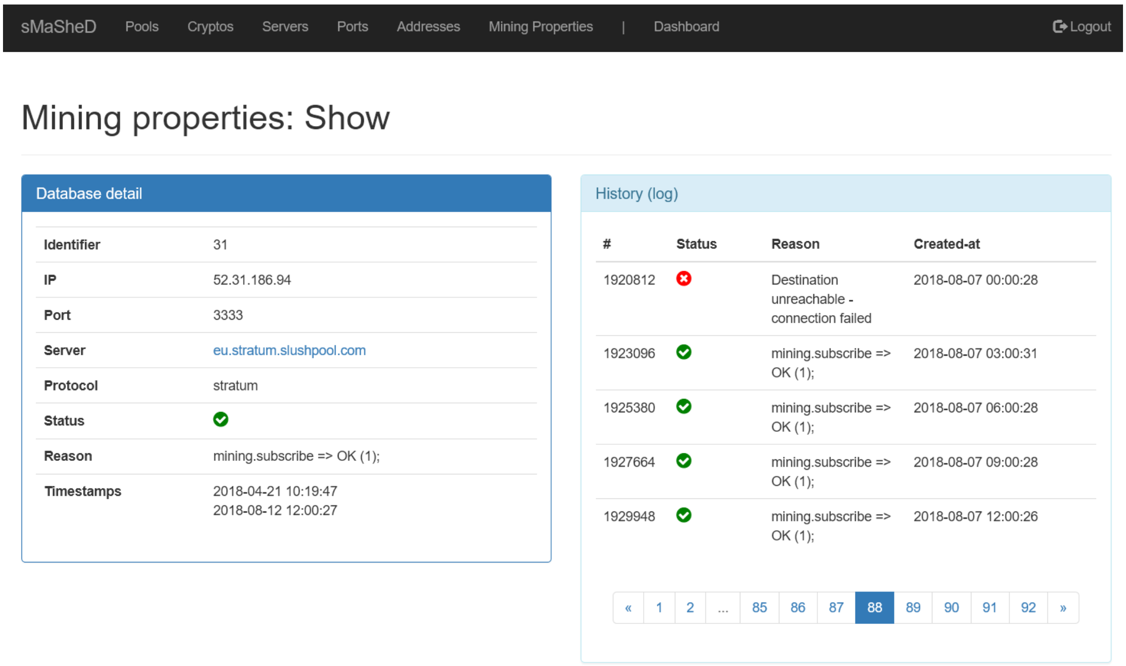
### List

All currently recognized mining properties in system.

| # | Server | Address | Port | Protocol | Status |
|---|---|---|---|---|---|
| 31 | eu.stratum.slushpool.com | 52.31.186.94 | 3333 | stratum | ✓ |
| 32 | eu.stratum.slushpool.com | 52.31.186.94 | 3333 | getblocktemplate | ? |
| 33 | eu.stratum.slushpool.com | 52.18.177.202 | 3333 | stratum | ✗ |

Showing 31 to 40 of 2284 rows 10 rows per page

Figure C.9: Probe results of selected subset from all available mining servers



## Mining properties: Show

Database detail

| | |
|---|---|
| **Identifier** | 31 |
| **IP** | 52.31.186.94 |
| **Port** | 3333 |
| **Server** | eu.stratum.slushpool.com |
| **Protocol** | stratum |
| **Status** | ✓ |
| **Reason** | mining.subscribe => OK (1); |
| **Timestamps** | 2018-04-21 10:19:47<br>2018-08-12 12:00:27 |

History (log)

| # | Status | Reason | Created-at |
|---|---|---|---|
| 1920812 | ✗ | Destination unreachable - connection failed | 2018-08-07 00:00:28 |
| 1923096 | ✓ | mining.subscribe => OK (1); | 2018-08-07 03:00:31 |
| 1925380 | ✓ | mining.subscribe => OK (1); | 2018-08-07 06:00:28 |
| 1927664 | ✓ | mining.subscribe => OK (1); | 2018-08-07 09:00:28 |
| 1929948 | ✓ | mining.subscribe => OK (1); | 2018-08-07 12:00:26 |

Figure C.10: Log of probe attempts for a given mining server