THE RUSH FOR HASHPOWER

How the Integrity of the Proof-of-Work Cryptocurrencies Can Be Compromised by the Excessive Concentration of the Computational Power
# OUTLINE

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive Summary</td>
<td>1</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>2</td>
</tr>
<tr>
<td>2. Background</td>
<td>4</td>
</tr>
<tr>
<td>2.1. Transactions and Blocks</td>
<td>4</td>
</tr>
<tr>
<td>2.2. Mining</td>
<td>5</td>
</tr>
<tr>
<td>2.3. Mining Pools</td>
<td>7</td>
</tr>
<tr>
<td>3. Censorship</td>
<td>9</td>
</tr>
<tr>
<td>3.1. Blacklisting with the Punitive Forking</td>
<td>9</td>
</tr>
<tr>
<td>3.2. Blacklisting with Feather Forking</td>
<td>11</td>
</tr>
<tr>
<td>4. Pool Cannibalization and Block Withholding Attacks</td>
<td>13</td>
</tr>
<tr>
<td>5. Mining Strategies and Network Attacks</td>
<td>14</td>
</tr>
<tr>
<td>5.1. Selfish Mining</td>
<td>15</td>
</tr>
<tr>
<td>5.2. Stubborn Mining</td>
<td>15</td>
</tr>
<tr>
<td>5.3. Eclipse Attacks</td>
<td>17</td>
</tr>
<tr>
<td>6. Hashpower and Price</td>
<td>19</td>
</tr>
<tr>
<td>6.1 Historical Hashpower Distribution</td>
<td>19</td>
</tr>
<tr>
<td>6.2 Discussion</td>
<td>22</td>
</tr>
<tr>
<td>7. Conclusions</td>
<td>23</td>
</tr>
<tr>
<td>References</td>
<td>24</td>
</tr>
<tr>
<td>List of Figures</td>
<td>26</td>
</tr>
<tr>
<td>Teams</td>
<td>27</td>
</tr>
</tbody>
</table>
Blockchain is a technology that enables counterparties to transact without relying on financial intermediaries. This is achieved through substituting them by a public ledger. Users are willing to contribute their resources to support network store replicas of this ledger. In order to prevent double spending, such users have to achieve a consensus regarding the state of the copies of blockchain they store. There are various consensus algorithms that set rules which govern this process. Proof-of-Work (POW) consensus algorithm is the most popular one. It requires users to contribute their computational power by finding solutions to very hard mathematical problems. These contributors are referred to as miners in POW blockchains. The higher share in total network’s hashrate a miner has, the more powerful and influential she is. In this paper we use Bitcoin to show examples of how an adversary can use her hashrpower to disrupt the integrity of a POW network.

After analysing the particular kinds of censorship attacks, pool cannibalization and block withholding attacks, as well as selfish mining, stubborn mining and eclipse attacks, our team conducted a historical analysis of Bitcoin hashrpower distribution. The outcome of our hashrpower analysis is a table explaining the hashrate needed for conducting each type of the attack that we analyze in this paper.

We came to the conclusion that regardless of their country of residence, the major miners are not interested in disrupting the integrity of the Bitcoin network and abusing their dominant position to conduct attacks. The rationale behind our reasoning is that in the very end, such behavior will devalue Bitcoin and hurt investments, which these miners have already made in their hardware.

However, the Blockchain community has to face the fact that over 77% of its hashrpower is geographically concentrated in China, as we demonstrate in the Figure 4, Section 2.3. This means if the government of this country decides for any reason to influence the price of Bitcoin by using any of the abovementioned attack techniques — such interference is theoretically possible.

We close this paper with a list of questions, which we are planning to research quantitatively in the future. The answers to these questions will enable us to build more nuanced relationships between hashrpower concentration, Bitcoin price and the risk of future attacks on the Bitcoin network. Unfortunately, such attacks are certain to occur given the rising prominence of Bitcoin within the modern financial ecosystem.

This research is a truly international effort, brought about by a joint Ukrainian and American teams, based in Kyiv, Warsaw, Maryland and the Bay Area. Vladyslav Makarov of School 42, Fremont Campus and the member of Blockchain at Berkeley drafted sections 1 — 4. Adam Anderson of Gladius Network drafted sections 5, 6 and the conclusion. Artem Tabachuk developed and built the document layout and infographics. Hennadiy Kornev created the project and supervised it on behalf of Hacken.
1. INTRODUCTION

Society in the Information Age needs robust and secure method to make fast transactions without relying on financial intermediaries. One of the biggest challenges to overcome in the development of such methods is the so-called double spending problem. Essentially, the problem deals with the fact that if people can duplicate a digital asset at will, the value of such an asset is thereby reduced. This seems to necessitate the existence of a central authority keep track of which assets are valid originals and which are copies. In 2008 Satoshi Nakamoto managed to solve the double spending problem without relying on a trusted entity. For that he created the world's first peer-to-peer electronic cash embodied in decentralized network. In such a network, central entities are substituted by public ledger. The latter is represented by a data structure called the blockchain, which is maintained by the network's participants.

The absence of a central intermediary means that anyone can join the Bitcoin network. There are two types of participants: general users and miners. While the former just enjoy the network’s functionality, the latter dedicate their computational power to maintaining it (i.e. storing and updating replicas of blockchain). The amount of resources that one decides to allocate to network is measured in hashes per second and called hashrate or hashpower. The integrity of Bitcoin relies on miners since it is they who settle transactions and prevent double spending. To do this, they have to consider states of copies of the blockchain and reach consensus on which to accept. For that, they have to reach a consensus regarding the state of the copies of blockchain they store. There are different ways to reach a consensus, but POW consensus algorithm is used in the Bitcoin network. This algorithm requires miners to contribute their computational power by finding solutions to very hard mathematical problems. In exchange, the system rewards them with freshly minted coins. In addition users can choose to pay miners transaction fees. It is important to note that miners are not guaranteed to solve these puzzles, but their chances of doing so are equal to their share in the network’s total computational power. Miners are vital to the network’s existence. Nakamoto used cryptography and mathematics to incentivize them to be honest. However, deviation from default behaviour is sometimes more advantageous. Strategies that a malicious miner can use to control allowed transactions are described in section 3. In Section 4 a strategy for making extra profits is presented. Section 5 covers strategies that can be utilized for acquiring more effective hashpower.

The key feature of the Bitcoin network is that all payments within the network are made in its native currency — Bitcoin. New users can buy it on cryptocurrency exchanges. There have been exhaustive ongoing deliberations on how to determine Bitcoin’s intrinsic value. However currently its price is set by an exchange rate that is established solely by market forces. The value of Bitcoin relies heavily on the network’s security and its ability to perform efficiently the role of borderless and censorship resistant peer-to-peer version of electronic cash.

Even though no critical vulnerabilities in source code of Bitcoin have been discovered yet, an adversary can use her hashpower to disrupt the network’s integrity. On top of that she can utilize Internet routing protocols for enhancing efficiency of her attacks. Attack vectors can be classified in the following manner:

1. **Censorship attacks.** As it discussed earlier, Bitcoin was created to be a decentralized type of electronic cash. However it turns out that malicious miners can use certain techniques to blacklist their victims. The choice of such a technique depends on the amount of computational power that a miner has at her disposal.

2. **Network attacks.** Malicious miners can perform DoS attacks on their competitors. A successful DoS attack allows an adversary to eliminate certain miners from the network by overwhelming them with traffic. Eclipse attacks fall into this category as well. It is performed to force network partition between a victim and its peers. If a malicious miner succeeds she will be able to make the victim work for him without knowing it (Heilman et al. 2015).

3. **Routing attacks.** An adversary may choose to attack the currency via the Internet routing infrastructure itself. This is done by manipulating routing advertisements (BGP hijacks) or by intercepting traffic. If an adversary succeeds, she will be able to break connections between peers and control what information victim receives from the network. An isolation of miners or delays in block propagation result in waste of significant amount of mining power, which leads to revenue losses and allows for a wide range of exploits such as double spending (Apostolaki, 2017).
Adversaries can combine censorship attacks and network attacks with different malicious mining strategies in order to acquire more power than their computational resources allow them to actually have. If one has enough computational power she can hinder network’s ability to perform its functions, or even worse, she can change the rules of the network and double spend her coins. This bears substantial economic and security risks that influence exchange rate of Bitcoin tremendously.

The well being of a blockchain network and people’s faith in it is usually best expressed in the value of its native cryptocurrency. The purpose of this paper is a) to describe mining strategies malicious agents can use to break rules that govern Bitcoin network and b) show what influence the level of concentration of hashpower has on Bitcoin ecosystem. For that we apply some data science techniques to examine the relationship between the level of concentration of computational power and Bitcoin’s price. To make our descriptions concrete, we imagine a scenario in which miners in the network are grouped into two pools — a malicious pool managed by an attacker and an honest pool managed by a protocol-following miner. Miners within both pools are free to choose which pool they wish to participate in.
In this section we describe concepts of transactions and blocks which are needed to comprehend the essence of malicious strategies that miners can stick to. In addition we outline the current landscape of mining industry. Afterwards we present a default workflow that miners follow and describe how they interact with each other within mining pools.

**Figure 1.** The Blockchain Block Composition

1. **Structure of a Block.** The block has a limited size of 1 MB, which means that it can contain up to 4000 transactions that results in a processing capacity of 7 transactions per second. Such limited processing power is the main issue that the Bitcoin’s community is trying to solve now.

2. **Structure of a Block Header.** The highlighted information stored in the Block Header is used by miners as inputs for cryptographic puzzles. Solutions to these puzzles enable them to generate blocks. Note that among these three highlighted inputs only nonce is unknown. A nonce is an arbitrary number that can only be used once for a specific task during encryption. It is the nonce that makes blockchain computationally infeasible to break or alter. Essentially, miners perform lots of computations in order to find such nonce that in combination with other two static variables yields a result that meets the criteria defined by the system.

3. **Merkle tree.** Merkle tree is a data structure that is used to encode all the transactions that are stored in a block into a single number. This number is usually referred to as a Merkle root. It is important to note that if the order of transactions is altered, Merkle root changes completely. The very first transaction is called a coinbase transaction. It is special since it assigns a certain amount of Bitcoins to a miner that manages to generate the block. This is how new Bitcoins are mined. Please, note that the coinbase transaction includes a nonce as well. This nonce is different from the one that is stored in Block Header. However both nonces are used for mining purposes.
The Rush for Hashpower

2.1 Transactions and Blocks

It is crucial to understand that Bitcoins do not exist either as physical objects, nor as digital files. Instead, they are represented by balances of users’ accounts. When a user wants to transfer her funds, she informs the rest of the network about her intentions. For that she sends messages to miners where she commands them to update their ledgers by writing off assets from her account and adding them to the receiver’s account.

Miners keep track of all transactions that take place in the network and make sure that no one spends more than each possesses. Another important point to consider is that the information about the balance of any Bitcoin address is not held at that address. It has to be reconstructed from the history of transactions related to that account. Such history can be accessed easily by referring to the blockchain, which makes Bitcoin transparent.

In any blockchain that utilizes the proof-of-work algorithm, transactions are not settled until mined. By being “mined” we mean included into a block that is successfully added to blockchain and accepted by the rest of the network’s participants. This means that settlement time approximately equals the time needed to generate a block. Each block points to its predecessor. Of course an attacker could substitute a valid block with a malicious one, where she double spends her coins, and then send a malicious copy to her peers. However, this will create inconsistency in the successive part of the chain and cryptography makes it easy for honest miners to detect this type of fraud. The structure of blocks that the blockchain is composed of is presented in Figure 1 “The Blockchain block composition”.

2.2 Mining

Computationally difficult POW algorithm is what the decentralized nature of Bitcoin is based on to ensure that the transaction history codified in the blockchain is valid (Satoshi, 2008). The algorithm requires miners to assemble pending transactions into a block. The one who solves an extremely difficult cryptographic puzzle first, will get a right to add her block to blockchain. The rest of the miners will check a newly added block for double spending transactions and if there are no such, they will accept it and get down to building the next block.

As new participants become miners, the Bitcoin network automatically adjusts the difficulty of the cryptographic puzzle so that each block takes an average of 10 minutes to generate. Computational power of miners is measured in hashes per second and their chances to solve the puzzle are equal to their share in the total network’s hashrate (i.e. computational power).

Note, that for our purposes we refer to hash as to an attempt to solve the cryptographic puzzle. Anyone can become a miner and start earning Bitcoins by maintaining the network. New miners have to download and setup mining software first. Then they recreate the whole blockchain by requesting historical blocks from their peers. After this process is finished they can get down to mining. For that they have to perform a number of specific tasks:

1. **Constantly update a stored replica of blockchain.** This is done by collecting new blocks from other miners and validating them before adding to their copy of blockchain.

2. **Collect transactions from users and validate them.** These transactions are sent in a form of cryptographically signed messages that specify recipient and the amount of funds that has to be transferred. At this stage, miners check that a sender does not spend more Bitcoins than she actually has. Miners are free to choose which transactions to include in their candidate block.

3. **Assemble blocks.** For that collected transactions are grouped together into data structure called Merkle tree (see Figure 1 “The Blockchain block composition”). Then certain calculations are done in order to derive at a single number that represents the state of the whole Merkle tree. Such a number is usually referred to as a Merkle root. It is important to note that if the order of transactions is altered, the Merkle root changes completely.

4. **Solve the puzzle.** This puzzle is solved by trial and error method. The goal of a miner is to find a hash (i.e. result of a cryptographic hash function) of her candidate block that has a predetermined number of leading zeroes. For that the miner will use information stored in the Block Header: hash of the previous block, Merkle root and nonce (see Figure 1 “The Blockchain block composition”). She will try to find such
a nonce that if used as an input to cryptographic hash function with the other 2 components will result in an output that has a number of leading zeros which is equal to or higher than the target. If a miner tries every single possible value for the nonce from the Block Header and none of them outputs the correct result, a miner will change the nonce from the Merkle tree, thus, changing the value of the Merkle root. Afterwards she will iterate again through every possible value of nonce from the Block Header. This process will be repeated until the solution is found.

Even if a miner becomes the first one to solve the puzzle there is no guarantee that the rest of the network will accept her block. There are a few reasons for that. Firstly, a miner can be situated in a remote area and the increased signal latency will hinder him in propagating her block quickly. Secondly, a competitor can mine another block simultaneously and the rest of the participants may not choose to include her block into their replica of blockchain. If two miners each mine new block with the same previous block, a block race occurs. Other miners will be split mining on the two versions of the blockchain until a new block is mined, in which case one branch in the fork will be longer. Whichever of the two miners has their block included in the longer chain is the winner of the block race. By default, the network will accept the longer version of the blockchain, orphaning the block belonging to the loser of the block race.

The behaviour of every participant in the Bitcoin network, including miners, is governed by certain rules. These rules are defined in the Bitcoin protocol that is implemented in the reference client — a piece of software that users install before being able to use the network. In general, however, miners are able to make certain decisions before they attempt to generate a new block. Considerations that miners usually account for, and their default behaviour, is presented in Figure 2 “The Default Strategy of a Miner”.

![Figure 2. The Default Strategy of a Miner](image)

<table>
<thead>
<tr>
<th>Choices</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which chain to support in case of a soft fork?</td>
<td>1. Extend the chain that they heard about first</td>
</tr>
<tr>
<td>Which block to mine on?</td>
<td>2. Mine on the longest valid chain that they know about</td>
</tr>
<tr>
<td>Which transactions to include?</td>
<td>3. Include any transaction which has a fee higher than certain threshold</td>
</tr>
<tr>
<td>When inform the network about a newly mined block?</td>
<td>4. Immediately announce about the block they generated (Eyal, 2013)</td>
</tr>
</tbody>
</table>

Miners are not obligated to follow this default — they are free to alter their strategy without any repercussions dictated by the protocol. For example, even though protocol states that the version of the blockchain that is accepted is the one which required the most computation, miners are free to choose on which version of the blockchain they wish to mine. Deviation from default behaviour may be used in attempts to earn abnormal profits and to perform different types of attacks.

### 2.3 Mining Pools

In general, the amount of time it takes an individual miner to mine a block is random. To reduce the variability in income created by this randomness, miners join together to form mining pools in which they share the profits. Members of such pools choose so-called pool managers. A pool manager is a miner who runs a Bitcoin node on behalf of participants, collects transactions and assembles them into a block. In addition she sets her address as a recipient in coinbase transaction. Then this block is sent to all members of the pool to work on (see Figure 3 “The Workflow of the Mining Pools”). If a participant solves the puzzle she sends her solution to the pool’s manager who propagates the block. The latter is the one to receive the mined Bitcoins which are to distributed among the members of the pool in proportion to the amount of work done.
So long as no single mining pool controls more than half of the computing power in the Bitcoin network, on average, new blocks will be distributed among many different mining pools. This insures that no single entity can gain control over which transactions are included or excluded from the blockchain. With control over the transaction history of the blockchain, malicious agents can nullify transactions, alter protocol, and orchestrate double-spending attacks (Decker, 2013).

Figure 3 “The Workflow of the Mining Pools” illustrates that the mining industry is very centralized. There are only 6 players who possess more than 5% in the network’s total hashrate. Even though there are 23 mining pools that actually manage to generate blocks, they are geographically located in only 5 countries. It is important to consider that mining requires tangible assets (i.e. hardware). One concerning possibility is that an authoritarian government can threaten to seize mining hardware within its jurisdiction unless miners obey its commands. Such government can try to destroy the network by performing attacks described in the paper. In order to evaluate the likelihood of this scenario we showed countries’ score and position in such international rankings as Ease of doing business, Corruption Perception Index and Democracy Index. It turns out that country with the lowest ranking in Corruption Perception Index and Democracy Index hosts 75% of the network’s hashrate. This makes political risk substantial. In the following sections we will describe what miners can do to the integrity of the system if they have too much hashpower.
Figure 4. The Geography of the Mining Pools

3. CENSORSHIP

Generally getting access to services provided by most modern financial firms requires that a user has a bank account. Even though banks do not own their clients’ money, they have control over its usage. When clients want access to their money, financial intermediaries decide whether to execute individuals’ requests or not. Moreover, they can freeze clients’ funds at any moment. For this reason, it is said that the modern financial system is based on trust. Creating a trustless electronic payment system was a primary motivation for the development of Bitcoin, which allows online payments to be sent directly from one party to another without going through a financial institution (Satoshi, 2008).

However, it turns out that it is possible to freeze victim’s funds in the Bitcoin network. There are two main techniques that an adversary may use to accomplish this: blacklisting with the punitive forking and blacklisting with feather forking.

3.1 Blacklisting with the Punitive Forking

This censorship technique works only if an adversary controls more than 51% of the network’s hashrate. It is important to consider that mining requires tangible assets (i.e., hardware). An attacker potentially can take temporary control over such equipment by injecting malicious software. Another scenario is that mining equipment can be seized by the government. If this allows the latter to control 51% of the network, it can perform blacklisting with the punitive forking easily. Generally anyone who controls the majority of hashrate can perform this type of attack. In Figure 5 “Blacklisting with the Punitive Forking” we describe the attack in question in detail.

**Figure 5. Blacklisting with the Punitive Forking**

1. **Victim’s transaction is mined**

An adversary who wants to perform this attack will announce to the network that she will refuse to support any chain that contains transactions spending from victim’s address. In other words, the malicious actor threatens that she will create a hard fork if her blacklist is disobeyed.
2. A malicious miner always wins

Since she has most of the network’s computing power, she will be able to create a longer chain, which the other miners will accept as the legitimate version of the blockchain. Miners are free to decide which transactions to include in a block and which to ignore, so the adversary’s strategy does not violate any rules defined by the Bitcoin protocol.

3. Honest miners cease to include victim’s transaction

As long as the attacker continues to control more than 51% of the network’s hashrate, competing with him is waste of time and resources. Eventually other miners will stop trying to include transactions spending from blacklisted address and victim’s funds will be frozen.

This attack will end when transactions involving victim’s address are included into a block, which would occur when the adversary runs out of resources to maintain the necessary hashrate.

In the case that the attacker has 100% of the hashrate of network, she has total control over what transactions are included in the blockchain, so she can freeze her victim’s funds as long as she controls 100% of the hashrate. Because anyone can become a miner, maintaining this strategy may become costly as new miners join the network.
In the previous section of this paper blacklisting with the punitive forking was described. With that censorship technique, it is only feasible for an attacker to create a fork if they control more than 51% of hashpower. However blacklisting with feather forking makes it possible to blacklist victim’s address with less resources. The main idea behind this type of attack is to make Bitcoin network too costly for victim to use.

Essentially, a feather-fork is when a miner refuses to mine on any chain that includes a transaction it doesn’t like in the most recent several blocks. You can think of this as a very-soft-fork, or a weak form of blacklisting. Honest miners running the standard client will build on any valid block, regardless of its contents. A malicious miner that wants to enforce — at any cost — some additional condition (for example, to blacklist transactions from a particular address), might refuse to build on any chain containing a block it doesn’t like (Miller, 2013). Steps of an adversary who wants to orchestrate a feather forking attack against a particular victim are presented in Figure 6 “Blacklisting with Feather Forking”.

Figure 6. Blacklisting with Feather Forking

1. A malicious miner announces that she will attempt to fork
For this attack, the adversary will publicly announce that if a transaction involving victim’s address is mined, she will attempt to fork. However, if she does not have the computation resources to actually fork, she may give up after a while.

2. A malicious miner never wins
All honest miners will have to account for the attacker’s threat to fork when they decide which transactions to include into a block.

3. Honest miners require compensation for the additional risks
Honest miners bear in mind that there is a risk that the attacker will fork off if they include victim’s transaction into a block. If the adversary wins their resources will be wasted and they will bear losses. So they require victim to compensate them for that. Eventually the adversary will accept the chain with transaction spending from victim’s address.
Essentially, the victim’s fee has to include the risked loss of revenue caused by including her transaction in a block. This fee will act as insurance for honest miners for the situation where an attacker manages to generate longer chain.

Let us consider an example. As of the time of writing this paper the block reward was equal to 12.5 BTC and the price of 1 BTC was 7,500 USD. Let the commission received by miners from transactions be equal to 0.1 BTC per block. Let the attacker’s fraction of total hashrate be equal to 30%. The attacker announces to the network that if transaction spending from address X is mined, she will fork off the main chain. If she fails to mine a two sequential blocks, her fork will not be long enough to get other miners to join, so she will join the main chain. Now we will calculate the value of the transaction fee that has to be paid by the victim to make sure that her transaction is mined:

**Step 1. Calculate the odds of generating 2 sequential blocks by an attacker.**
Probability of generating a block equals to the attacker’s hashrate.

\[
\text{P(\text{Block1})} \times \text{P(\text{Block2})} = 0.3 \times 0.3 = 0.09 \quad \text{Odds of generating 2 sequential blocks.}
\]

\[
30\% \times 30\% = 9\% \quad \text{Chances of an attacker to successfully fork off after 2 blocks are mined.}
\]

Thus, there is a 9% chance that the attacker will generate 2 sequential blocks. If this happens, all resources spent by honest miners who stay on the main chain will be wasted.

**Step 2. Calculate the expected values of including and not including victim’s transaction to a block.**
Reward consists of fees and generated coins. The expected value (EV) of a reward equals to the likelihood of its receiving (i.e. mining a block).

\[
\text{EV(\text{including})} = 0.91 \times (12.5 + 0.1) + \text{Fee of Victim} \quad \text{The attacker refuses to mine, so chances for honest miners to find successive blocks are less than 100%. In addition they are able to get transaction fees from the victim.}
\]

\[
\text{EV(\text{including})} = 11.466 + \text{Fee of Victim}
\]

\[
\text{EV(\text{not including})} = 100\% \times (12.5 + 0.1) \quad \text{Since the victim’s transaction is not included the attacker will support honest miners. As a result the network’s hashrate will not be split.}
\]

\[
\text{EV(\text{not including})} = 12.6
\]

Victim has to incentivize honest miners to include its transaction to a block.

**Step 3. Solve equation for Fee of Victim.**
Now we can calculate the minimum fee of a victim easily.

\[
11.466 + \text{Fee of Victim} = 12.6 \quad \text{Equate EV(\text{including}) with EV(\text{not including}).}
\]

\[
\text{Fee of Victim} = 1.134
\]

So the victim should pay 1.134 BTC as transaction fee to compensate honest miners for taking the risk. This equals to 8505 USD per transaction as of time of writing this paper.
In 1970, Milton Friedman stated that the main purpose of a business is to maximize profits for its owners (Friedman, 1970). That can be achieved in various ways and investing is one of them. Like any other companies, mining pools constantly have to make investment decisions which are sometimes quite counterintuitive. In this section we describe why investing into rival miners is more profitable than reinvesting into own mining pool.

Suppose you are a miner with 30% of the hashrate and the current block reward is 1 BTC. We will imagine that the computing resources in the network are broken up into 100 units of mining power so that you control 30 of the units. In this scenario, your expected value from mining activity is 0.3 BTC. Now, suppose you invest into enough additional mining equipment so that you control an additional 1 unit of the total network hashrate. When you do this, it might seem intuitive to expect that you would control 31% of total network hashrate. However, since the network has an additional unit of mining power, your chances of generating a block now are 31/101 = 30.69%. So you can mine 0.3069 BTC which is 0.0069 BTC revenue gain for 1% hashrate added.

However there is a smart way to increase the chances of mining a block by more than 0.69%. This technique is called Pool Cannibalization. You should evenly distribute additional hashrate among other miners who control the remaining 70% of the network. Now the odds of generating a block equal to 71/101 which translates into expected value of rewards to be 0.7029 BTC. However, you get only a fraction of that reward. It equals your share in the rest of the network which is 1/71 or 1.4%. At the end you get 1.4% of 0.7029 BTC which is 0.0098 BTC.

An adversary can combine Pool Cannibalization with Block Withholding attack. Once the attacker invested into acquiring additional hashrate and registered with the victim pool as a regular miner, she can start solving cryptographic puzzles for the victim. The trick is to constantly send wrong solutions to prove that the work is actually done, but when a valid solution is found, the adversary withholds it (see Figure 7 “The Block Withholding Attack”). The pool manager can detect that she is being attacked. However there is no way she can find the pool member who finds valid blocks and withholds them. As it was described in section 2 (see Figure 3 “The Workflow in the Mining Industry”) the pool members are rewarded accordingly to the amount of work they contribute but not the validity of solutions they send. That is why malicious miners will profit from a Block Withholding attack even if they fail to actually withhold a block.

Figure 7. The Block Withholding Attack

1. As usual, the pool manager assembles a block and disseminates it among pool members.
2. Like other miners the adversary sends solutions to prove that she is actually contributing resources.
3. If the adversary mines a block she withholds it to give her own pool more time to generate a block.
4. If an honest pool member manages to generate a block the adversary receives rewards as well.

This attack affects the revenues of the pools in several ways. The victim pool’s effective mining rate is unchanged, but its total revenue is divided among more miners. The attacker’s hashrate is reduced but it earns additional revenue from participation in victim’s pool (Eyal, 2014). In addition to revenue increasing the adversary gives its own mining pool more time to find a solution to the puzzle and, thus, generate a block.
5. MINING STRATEGIES and Network Attacks

In order to achieve the high hashpower needed to orchestrate the previously described attacks, malicious miners can apply strategies that encourage honest miners to give their hashpower to a malicious pool. Two important strategies to accomplish this are selfish mining and stubborn mining. It is important to mention that a miner who uses these strategies does not necessarily want to boost her profits in the short term (Buterin, 2013). Instead she tries to attract honest miners and thus increase cumulative hashrate of her pool. Research has shown that when an attacker uses selfish/stubborn mining, the dominant strategy of honest miners is to avoid having their blocks orphaned by joining the malicious miner’s pool.

To orchestrate the attack, an attacker creates a private copy of the blockchain in the hopes of mining ahead. If she reveals a longer private chain than the public chain, honest miners will accept her version as the legitimate blockchain. This may orphan blocks mined on the public version of the chain, meaning the miners who created those blocks wasted their computational resources.

In Figure 8 “Mining Strategies”, we see a simplified presentation of the selfish and stubborn mining strategies. At a high level, we can explain the strategies based on when the attacker decides to reveal her private copy of the blockchain, and when the attacker abandons her copy and accepts the version the honest miners are mining on. Generally there are five events that can occur to trigger decisions in these strategies.

Figure 8. Mining Strategies

1. Malicious miners win a block race
   When the malicious miner has one block on her private chain and the honest miners mine a block, a block race occurs. Event 1 is the event that an attacker wins a block race by successfully mining the next block on her private chain.

2. Malicious miners lead reducted from 2 to 1
   The attacker has two blocks on her private chain and the honest miners mine the next block.

3. Honest miners win a block race
   A block race occurs and the honest miners win by mining the next block on the public version of the blockchain.

4. Honest miners mine one block ahead
   The attacker has not yet mined any blocks on her private chain and the honest miners mine the next block.

5. Honest miners mine a second block ahead
   The attacker has not mined any blocks on her private chain and the honest miners mine two blocks ahead. For reasons described below, this can only occur during trail stubborn mining.
5.1 Selfish Mining

In a selfish mining attack, the attacker will keep blocks she mines on her private chain hidden from the honest miners unless she wins a block race (Event 1, Figure 8), her lead is reduced from two blocks to one (Event 2, Figure 8) or she gets behind the public chain (Event 3 and Event 4, Figure 8) (Eyal, 2013).

If the attacker ever has a private chain the same length as the public chain and she mines a new block, she will instantly publish her private chain to orphan any blocks mined on the public chain after when she created her private fork (Event 1, Figure 8). Additionally, if the attacker is 2 blocks ahead on her private chain and an additional block is mined on the public chain, she will publish her chain to avoid her own blocks being orphaned (Event 2, Figure 8). In both of these cases, the honest miners will receive a version of the blockchain longer than their own, which they will abandon the previous version of the blockchain to accept.

Since it is unlikely that an attacker will right away have enough hashpower to get very far ahead of the honest miners, she will have to decide when to accept the public version of the blockchain to create a new private fork. In a selfish attack, the attacker will abandon her private fork to start a new one with the updated public chain whenever the public chain is ahead of the private one. This can occur when the honest miners win a block race (Event 3, Figure 8) or when the honest miners mine ahead before the attacker is able to mine a block (Event 4, Figure 8).

5.2 Stubborn Mining

If we carefully consider the selfish mining strategy, it becomes apparent that there are several potential modifications one could try to make in order to further increase the revenue of the selfish pool relative to the honest pool. In general, when the honest miners announce a new block, the attacker can choose how many blocks to reveal from her private chain. Furthermore, if she is behind the public chain she can decide when to abandon the private chain in favor of the public one. Lastly, when the attacker finds a new block, she gets to decide when to reveal it to the honest miners. This enables three fundamental types of stubborn mining, which can be combined to produce more complex stubborn mining strategies (Nayak, 2016). They are described in Figure 9 “Stubborn Mining Attack Types” in detail.
Figure 9. Stubborn Mining Attacks

**Lead—Stubborn Mining.** Whenever the attacker has a lead and the honest miners mine a block, she will only ever reveal a single block from her private chain. This differs from selfish mining in the following way: in selfish mining, when the attacker’s lead is reduced from 2 to 1, she would publish her entire private chain. With a lead stubborn strategy, the attacker only publishes one block, causing a block race. This is why we do not see a green arrow from Event 2 in Figure 8. By doing this, the attacker creates the opportunity to orphan several of the honest miner’s blocks at a time.

**Equal Fork Stubborn Mining.** Imagine the scenario where the attacker is one block ahead, but the honest miners mine a block. This means that both the public and private chains are of equal length and a block race is occurring. The selfish mining strategy dictates that if the attacker mines another block, she will immediately reveal it in order to avoid her private chain being orphaned (Event 1, Figure 8). However, the Equal Fork Stubborn Miner will not reveal the block found during the block race immediately so that they can try to orphan multiple honest blocks at once. This is why there is no yellow arrow from Event 1 in Figure 8.

**Trail Stubborn Mining.** When the attacker’s private chain falls behind the public chain, she may decide to continue mining on it anyway in the hopes of catching up and orphaning even more blocks on the public chain. A selfish miner would immediately give up if the public chain becomes longer than the private one (Event 4 and Event 5, Figure 8), but a trail stubborn miner can decide how many blocks behind she must be in order to abandon her private chain. A typical trail stubborn miner will give up when the public chain is two blocks ahead, hence the orange arrow from Event 5 on Figure 8.
If an attacker can control the flow of information in the Bitcoin network, she does not need as much hash-power in order to execute censorship and selfish/stubborn mining attacks. In this section, we describe how an attacker orchestrates what is known as an eclipse attack. We then discuss how this can be used to augment other attack strategies (Heilman et al. 2015).

When a new participant first joins the Bitcoin network, she gets information on other network participants from a server called a DNS seeder. The DNS seeder tells the new node the IP addresses of various nodes in the network so that the new participant can start communicating with them. Once the node has joined the network, its peers can send it information about even more nodes in the network. All nodes store two tables containing the IP addresses of other nodes: the Tried Table and the New Table.

**Tried Table** — Keeps track of peers to whom the node has successfully connected previously.

**New Table** — Keeps track of addresses the node has heard of but not yet connected to.

Nodes in the Bitcoin network accept incoming connections from unknown peers, who can send the addresses of other nodes in the network. The address of the newly connected node is stored in the Tried Table while the information that is received about other nodes is stored in the New Table. When a node restarts, it establishes a maximum of eight peers to form outgoing connections with using the addresses in the Tried Table and the New Table.

To gain control of the eight outgoing connections for a victim node, an attacker can exploit the fact that nodes accept addresses from unsolicited incoming connections. The attack progresses as follows:

1. The attacker has several malicious nodes connect to the victim and send junk IP addresses.
2. When the victim is connected to by the malicious nodes, it will store the malicious nodes’ addresses in its Tried Table.
3. Upon receiving junk addresses, the victim node will store them in its New Table.
4. When the victim node restarts and chooses new outgoing connections, it is highly likely that all eight connections will be to malicious nodes.

Because the victim’s New Table will be full of junk addresses, the victim will have to connect to an address stored in the Tried Table. Connections to recently connected peers are more likely to be established, so the victim is likely to have its connections monopolized by the attacker’s nodes (see Figure 10 “Eclipse Attack”).

**5.3 Eclipse Attacks**

In preparation for this attack an adversary populates Tried and New Tables with either junk ip addresses or malicious ones. Then she has to force victim to restart its system. When victim restarts it establishes outgoing connections with its peers using addresses from these two tables. With high probability, the victim will form all of its 8 outgoing connections with malicious IP addresses. If this happens, the attacker will be able to choose what information the victim receives and exploit victim’s hash-power for its own benefit.
Now, we consider how an eclipse attack can be combined with other attacks. Recall that in a censorship attack, the attacker’s goal is to prevent transactions involving a certain address to be included in blocks. If the attacker has eclipsed some miners, she can simply refuse to show them versions of the blockchain containing blocks with transactions she disapproves of. If the eclipsed miners manage to mine a longer chain, the network will accept the longer version of the blockchain, which will not contain the transactions.

Lastly, we consider how the attacker might use an eclipse attack to make it easier to execute a selfish or stubborn mining attack. Recall that in these attacks, the attacker maintains a private fork of the blockchain and she tries to mine ahead of the public fork. The attacker can force the eclipsed miners to help him mine blocks on her private chain. She can accomplish this by only allowing the eclipsed miners to see her private fork. In this strategy, the attacker’s effective hashpower is equal to her own plus that of the eclipsed miners.
Now, we examine how the concentration of hashpower can affect the price of a cryptocurrency. As we have seen, attackers with a large share of the total hashpower can execute double spending and censorship attacks, and there are many strategies such attackers can use to achieve higher hashpower.

To quantitatively measure the concentration of hashpower among Bitcoin miners, we consider the Herfindahl—Hirschman Index (HHI). The HHI is calculated by taking the percent hashpower each mining pool controls, squaring the percentages, and adding them up. For example, if there are three pools where the first has 60% of the hashpower, the second has 30%, and the third has 10%, the HHI is $60^2 + 30^2 + 10^2 = 4600$.

### 6.1 Historical Hashpower Distribution

In Figure 11 "Historical Distribution of Hashpower" we see a graph of how concentration of hashpower has changed over time. The background of the figure is an area plot representing the percent hashpower controlled by various mining pools over time, while the black line shows how the HHI. Note that the total hashpower in the graph never equals 100% because there are smaller mining pools and individual miners not shown in the graph.

On the far left of the graph, we see some of the highest HHI around January and June 2014. This corresponds to the controversial time that the mining pool GHash.IO reached 51% control of the Bitcoin network (Goo-din, 2014). We also see a relatively high concentration of hashpower in early 2016, especially among the mining pools AntPool (purple) and F2Pool (yellow). Both of these pools are managed from China, and each has been involved in some controversy. Miners contributing to AntPool complained in 2017 that the pool managers were taking large fees from miners (BitcoinTalk, 2017). F2Pool, which has also controlled over 25% of the Ethereum mining hashpower, once faced calls for a boycott by the Ethereum community for its policies (Trustnodes, 2017).
We visualize HHI, Bitcoin price, and trends in two metrics computed using the price for qualitative consideration — the discrete first difference and the volatility. Discrete first difference shows the week to week change in price while the volatility measures the standard deviation of trade prices averaged across the exchanges. During times where the Bitcoin price changed rapidly, the first difference and volatility increased. In Figure 12 “Hashpower Concentration and Price Metrics”, we see fluctuations in the HHI and price metrics during 2014 through the end of 2016.

Note that the date range on the x—axis of the price metric graph only goes to January 2017. This is because including the rapid price increase during 2017 would require rescaling the graph such that features prior to 2017 would be difficult to see.
The Rush for Hashpower

Key for Event Drivers

1. On February 7, 2014, the MtGox exchange halted all withdrawals. On February 23, transactions were suspended. It was discovered that 650,000 coins were stolen in a hack that resulted in the exchange declaring bankruptcy on February 28, 2014 (Eichholz, 2017).

2. In April 2014, a slight price increase is curtailed when Chinese banks threaten to freeze accounts belonging to Bitcoin exchanges (Southurst, Banks, 2014).

3. During June 2014, the GHash.IO mining approached majority hashpower (Goodin, 2014). This resulted in the highest market concentration in the history of Bitcoin.


5. In October 2014, the oldest Bitcoin exchange BTCC (known then as BTC China) added a mining pool and merchant payment services (Southhurst, BTCC, 2014).

6. On January 6, 2015, the Bitcoin exchange Bitstamp was hacked for 19,000 Bitcoins. It suspended service, and Bitcoin price reached the bottom of a downtrend that had lasted through 2014. The price remained in the $200—300 range until September 2015.

7. In mid January 2015, low Bitcoin prices prompt CEX.IO, parent company of GHash.IO, to temporarily cease cloud mining operations (Higgins 2015).

8. On June 15, 2015, the BTC Guild mining pool announced that it would close at the end of the month (BitcoinTalk Forum, 2015).

9. On July 30, 2015, the Ethereum Frontier network is launched (Ethereum Homestead Documentation, 2017). Note that the Ethereum mining protocol was designed differently from that of Bitcoin so that established Bitcoin mining pools with dedicated hardware would not be able to instantly control the Ethereum blockchain (Hajdarbegovic, 2014).

10. On September 18, 2015, the US Commodity Futures Trading Commission determined that Bitcoin should be classified as a commodity (Clinch, 2015).

11. In April 2016, the BTCC Pool deployed new servers globally to allow miners to connect to the pool quickly from around the world (BTCC, 2017). The pool also launched a bilingual website to improve the engagement of international miners (Dob, 2016).

12. In June 2016, F2Pool stated publicly that their increased size was due to new members, not from new mining equipment owned by the pool (Torpey, 2016).

13. In June 17, 2016, a hacker stole Ethereum worth almost $80 million at the time, in the infamous DAO hack. The BTC price fell 6%, ending a price rally (Wong, 2016).

14. On July 9, 2016, the number of Bitcoins awarded to miners for mining a block was cut in half from 25 Bitcoins to 12.5 (Tepper).

15. At the end of 2016, the private mining pool BTC.TOP was established, which quickly shot up to 8% of the network’s total hashrate within the first few months of 2017 (Quentson, 2017).

16. In April 2017, BitFury signed a deal to integrate blockchain into Ukrainian government platforms in a deal called the ‘biggest blockchain government deal ever’ (Das, 2017). This may explain why we see the BitFury relative hashrate decrease, as the company had to focus on scaling to handle the contract.

17. On August 1, 2017, a fork split the blockchain into Bitcoin and Bitcoin cash. In the days that followed, the hashrate in the Bitcoin network fell 50% (Suberg, 2017).
6.2 Discussion

In the graphs in Section 6.1, markers for various events related to the Bitcoin community have been included. We see that some events coincide with significant fluctuations in the HHI or the price, while others do not appear to drive significant changes. For example, for Event 9, a high profile Ethereum hack brought the price of Bitcoin down and paused an uptrend. However, the original launch of Ethereum in Event 7 did not cause a significant change in the price or hashpower distribution. This is likely at least partially because Ethereum mining is not as efficient on hardware optimized for Bitcoin mining, so it would not necessarily be efficient for established miners to switch which currency they mine (Hajdarbegovic, 2014).

A particularly noteworthy occurrence is the massive spike in hashpower concentration occurring as Event 3 in June 2014. The mining pool GHash.IO approached majority hashpower, and it was reported that the pool reached 51% hashpower for a 12 hour period (Goodin, 2014). Recall that this is a tremendously dangerous situation for the Bitcoin community, as majority hashpower enables censorship and double spending attacks. When GHash.IO’s hashpower reached high levels, the Bitcoin community encouraged each other to leave the pool to prevent the possibility of 51% attacks (Borchgrevink, 2014). The pool’s proportion of hashpower decreased dramatically, and the pool eventually closed down in late 2016 (Danova, 2016). Interestingly, we see the price increase around this time, though we might expect investors to be cautious investing when such a potentially dangerous event occurs.

One event roughly coinciding with a large downtrend in hashpower concentration from April 2016 through 2017 is the halving of the block reward. By design, Bitcoin has a finite supply of 21 million coins. This is intended to ensure continual steady increase in Bitcoin’s purchasing power (Faggart, 2015). To gradually reduce the supply growth of Bitcoin, every few years, the number of coins a miner gets for mining a block is cut in half. This last occurred on July 9th, 2016, and is marked as Event 10. We note that the downtrend in hashpower cannot be solely attributed to this event.

Several interesting questions warrant quantitative investigation. To what degree does price influence hashpower, and to what degree does hashpower influence price? We might expect that a higher Bitcoin price encourages more people to set up computers for mining. Conversely, it might make sense to conjecture that price volatility is related to hashpower concentration, as higher concentration means a greater risk that a majority group gains control of the network. However, it remains to be analyzed quantitatively how the community reacts to high hashpower concentration.

Another important direction for research is modeling how hashpower changes in response to various events. Minimizing the possibility that an adversary can acquire majority hashpower in the Bitcoin network is vital to the health of Bitcoin as a currency, so identifying possible mechanisms for reducing hashpower concentration is an important research goal.
THE RUSH FOR HASHPOWER

7. CONCLUSIONS

As the popularity of Bitcoin has increased, we have seen more and more hashrate dedicated to mining on the Bitcoin network, from tens of GHashes/sec in 2010 to over ten EHashes/sec today — a one billion-fold increase in seven years. During that time, many different attack vectors against the Bitcoin network and POW cryptocurrencies in general have been hypothesized. Censorship attacks allow an adversary to control who is allowed to make transactions in the network. Pool Cannibalization and Block Withholding enable an attacker to earn abnormal profits at the expense of other miners in the network. Selfish and Stubborn mining attacks can be used to incentivize honest miners to join an adversary’s pool thus giving the adversary more hashrate, while Eclipse attacks allow the attacker to increase his effective hashrate.

The Figure 13 “Summary of Strategies” summarizes what proportion of the total network hashrate an adversary needs to execute various attacks. For attacks where the percentage necessary depends on the structure of the network, we use a tilde (~) and give approximate numbers.

Over time as cryptocurrencies have become profitable to mine, we have seen a reduction in the percent hashrate controlled by a single pool. However, as of August 2017, one organization managing multiple mining pools controls approximately 29% of the Bitcoin hashrate (Wong, 2017). Is this concentration of hashrate a cause for concern? As was shown in Figure 11 and Figure 12, when the GHash.IO mining pool approached majority hashrate, the community reacted by moving computational resources into other pools (Rizzo, 2014). This implies that the community is willing to undertake precautionary measures to prevent an entity from consolidating majority hashrate.

Miners have a vested interest in the stability of the Bitcoin network, which is in part dependent on the community’s trust in the network. If participants in the Bitcoin network think that prominent miners will violate protocol, they may decide to use alternative cryptocurrencies and abandon Bitcoin altogether. This would devalue Bitcoin, hurting miners who have invested in mining hardware.

One potential cause for concern is the geographic concentration of hashrate. A government interested in controlling the network may be more willing to accept a devaluation in Bitcoin than the companies operating mining pools. We saw in Figure 4 that hashrate is geographically concentrated, with a tremendous amount of the network hashrate consolidated in China, in part due to cheap energy costs enabling mining hardware manufacturers to set up massive data centers for mining (Peck, 2017).

Moving forward, we have identified some interesting questions to investigate quantitatively with future research. How can we model relationships in price fluctuations and measures of hashrate concentration like the HHI? Can we predict the reaction of the Bitcoin community to a high level of hashrate concentration? Precisely how do different events influence the level of hashrate concentration? These questions may offer insight into strategies for preserving the health of the the Bitcoin ecosystem as cryptocurrencies become more widely accepted and utilized.

Figure 13. Summary of Strategies

<table>
<thead>
<tr>
<th>Attack Strategy</th>
<th>Hashrate needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blacklisting with punitive forking</td>
<td>&gt;50%</td>
</tr>
<tr>
<td>Blacklisting with feather forking</td>
<td>&lt;50%</td>
</tr>
<tr>
<td>Pool cannibalization</td>
<td>Any</td>
</tr>
<tr>
<td>Selfish mining</td>
<td>~20%</td>
</tr>
<tr>
<td>Stubborn mining =&gt; Lead + Equal Fork</td>
<td>~20%</td>
</tr>
<tr>
<td>Stubborn mining =&gt; Trail + Equal Fork</td>
<td>~45%</td>
</tr>
<tr>
<td>Stubborn mining =&gt; Trail</td>
<td>~30%</td>
</tr>
</tbody>
</table>
REFERENCES


LIST OF FIGURES

Figure 1. The Blockchain Block Composition ................................................. 4
Figure 2. The Default Strategy of a Miner ..................................................... 6
Figure 3. The Workflow of the Mining Pools .................................................. 7
Figure 4. The Geography of the Mining Pools ............................................... 8
Figure 5. Blacklisting with the Punitive Forking .......................................... 9
Figure 6. Blacklisting with Feather Forking ................................................ 11
Figure 7. The Block Withholding Attack ...................................................... 13
Figure 8. Mining Strategies ......................................................................... 14
Figure 9. Stubborn Mining Attacks .............................................................. 16
Figure 10. Eclipse Attack ........................................................................... 17
Figure 11. Historical Distribution of Hashpower ......................................... 20
Figure 12. Hashpower Concentration and Price Metrics ............................. 20
Figure 13. Summary of Strategies ............................................................... 23
TEAMS

The **Hacken Ecosystem** is a community-based business organization consisting of the HackenProof bug bounty marketplace, Zero-day Remuneration Platform, Hacken Accelerator and Cybersecurity Analytics Center. The Hacken Ecosystem utilizes its own cryptocurrency HKN — a dedicated cryptocurrency for white hat hackers. HKN token incentivizes community members to interact with our ecosystem. Hacken also hosts HackIT — one of the major cybersecurity conferences in Eastern Europe.

Hacken’s vision is to launch a movement that in several years will become one of the main driving forces deterring and countering international cybercrime. On November 30, 2017 Hacken successfully completed token sale for HKN, selling 4 400 000 tokens and raising roughly $7M in fiat equivalent (based on Mid-December 2017 ETH exchange rate). Currently the development of HackenProof and other solutions comprising the Hacken Ecosystem proceeds according to the product roadmap. For further media inquiries please contact media@hacken.io

The **Gladius Network** is a decentralized, Blockchain-driven platform designed to protect against DDoS attacks and accelerate websites. The **Gladius platform** allows users to rent out spare bandwidth to help fight cybercrime and load content faster. Users who rent out their bandwidth earn Gladius tokens (GLA) which can then be exchanged for fiat currency at a later time. The process is peer-to-peer (P2P) and uses the power of the blockchain in order to make a simple and systemized use and reward protocol.

The vision of Gladius is to build a technology and marketplace which lets people all across the world rent/ use their spare and underutilized bandwidth to help safeguard (DDoS protection) and accelerate websites by participating in a global CDN -- a decentralized and more effective Cloudflare/Akamai. The bandwidth will be collected from people around to world and sent to pools/nodes around the world and then funneled to websites under DDoS attack/distress. Users can also cache content on their device (and get paid for it in GLA) and thus act as a mini CDN, that way we’ll be building a much more scalable and distributed CDN with mini “data centers” around the world. For further media inquiries please contact contact@gladius.io

www.hacken.io  www.gladius.io
THE RUSH FOR HASHPOWER