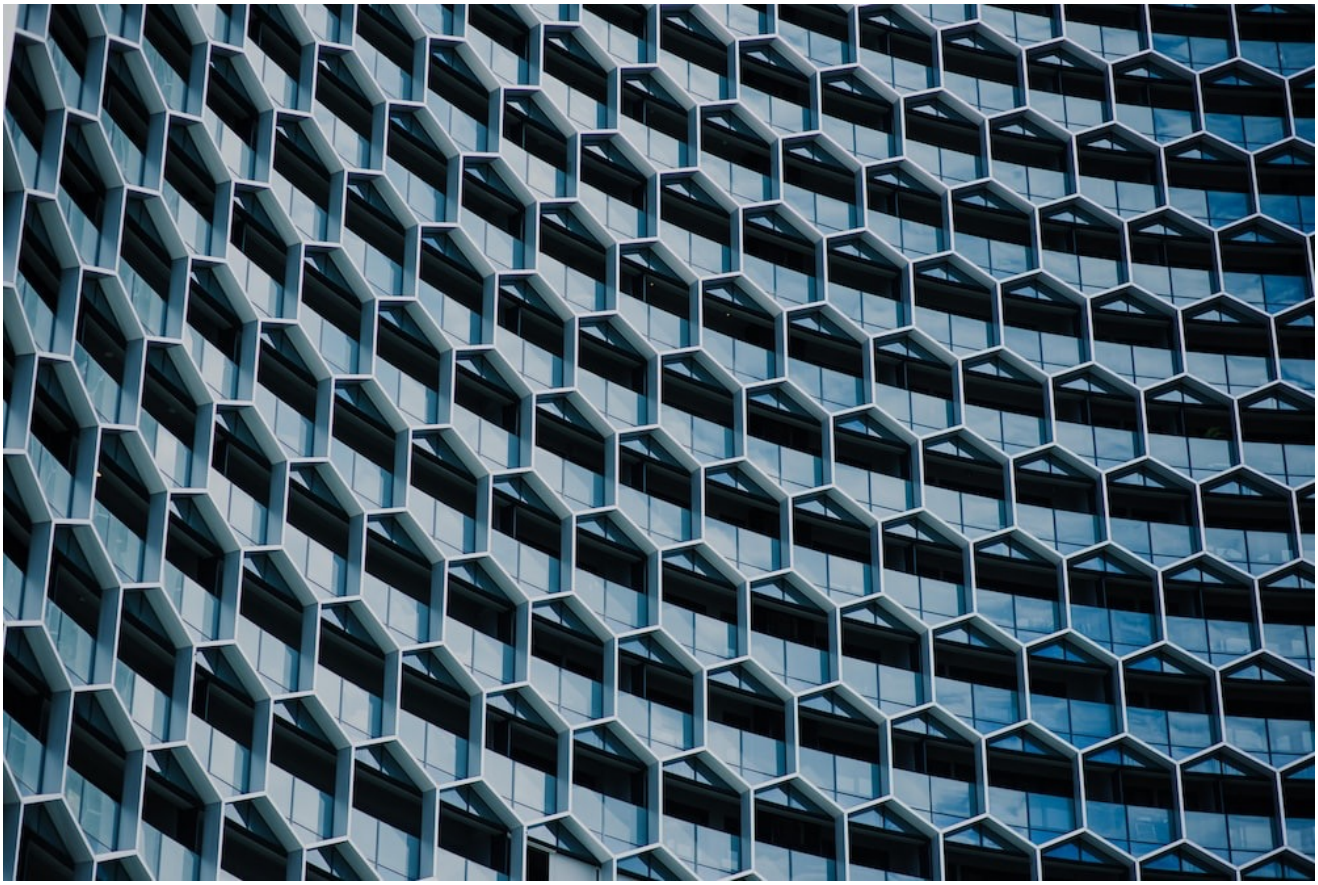
Alpha Financial
Technology Research



Do fake bitcoins exist or is it possible to adulterate Blockchain?

Research paper on the blockchain consensus network and its weaknesses.

Alpha Financial
Technology Research

Blockchain Vulnerabilities

Blockchain technology has been touted as a revolutionary innovation that can bring about significant improvements in various sectors of the economy, including finance, healthcare, logistics, and more. Despite its numerous benefits, blockchain technology is not immune to weaknesses and vulnerabilities that can be exploited by bad actors to compromise the system's integrity and security.

One of the most significant weaknesses of blockchain technology is its susceptibility to 51% attacks. In a blockchain network, transactions are verified and added to the ledger by a network of nodes or miners who solve complex mathematical problems to create new blocks. When a single miner or group of miners controls more than 50% of the network's computing power, they can potentially control the network and rewrite transaction history. This could allow them to spend the same coins multiple times, effectively double-spending, or even reverse transactions that have already been validated.

While the probability of a 51% attack is very low, there are other vulnerabilities that are being exploited by malicious hackers and causing great concern in the industry.

We have developed software capable of exploiting these vulnerabilities in the Bitcoin and Ethereum network, allowing to adulterate blockchain blocks and generate fake transactions and in this document, we are going to talk about both theoretical and practical vulnerabilities, how they are exploited, and how to prevent them.

Alpha Financial
Technology Research

How is it possible to adulterate the blockchain network or generate fake bitcoin?

Before diving into how the attack works, here are some pre-context about proof-of-work protocols to have a better understanding.

1.Not everyone participating in the network, or using the network to perform a transaction is required to have a full copy of the digital ledger. As of March 2023, the size of the Bitcoin network is 470 GB. Imagine that if it was the case, you will need to have a smartphone with enough storage space just to make a transaction. That is rather impractical. Participants can interact with the network as a lightweight client, without having a full copy of the ledger.

2.Propagation of information (ie. publishing of a new block) across the network is not an instant affair. Information is relayed from nodes to nodes and the propagation of information from one node to the rest of the network depends on the degree of connectedness of one node to the network. There can be significant delays for an information to be propagated throughout the network, which is one of the inefficiencies that attackers can leverage on.

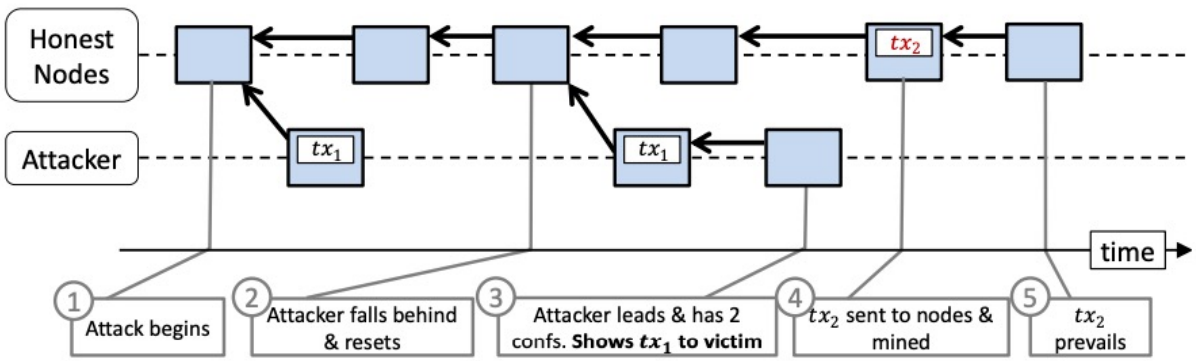
3.It is possible whereby two or more competing blocks are published onto the network, resulting in a fork event. In such an event of dispute, the longest chain will be determined as valid and all the other shorter chains will be invalidated.

4.For a miner to mine a new block, it has to find the latest block that was published on the network to work on and miners will mine on the first published block they find on the network. Note, as mentioned in point 3, it is possible to have two different miners connected to different parts of the network to assume that the blocks that they find are the latest block. Both of them can be mining on different chains.

5.Miners can choose to withhold publishing the fact that they have successfully mined a new block and choose to continue mining on their own block without the network knowing, resulting in a fork. This way the network and the miner will be working on separate chain of blocks. And when the miner choose to publish, the longer chain will be considered as valid.

How is it possible to adulterate the blockchain network or generate fake bitcoin?

We shall term the chain worked on by honest miners as the honest chain and the chain done by the attacker as the secret chain. With the above mentioned in mind, a generalised attack is as follows:



The attacker finds the latest block on the honest chain to mine on. At this point, it starts working on a secret chain with the first block containing the transaction, tx1, she wishes to revert later. If the attacker realises that her secret chain is shorter than the honest chain, she will abandon the operation and find the next latest block on the honest chain to restart the operation.

When the attacker successfully built a chain longer than the honest chain, she will publish her secret chain to the victim's node. At this point, the victim's node will believe that the attacker's chain is valid as it is the longest chain and will honour the transaction.

The attacker then submits a conflicting transaction, tx2, whereby the source of funds is the same as tx1, to the honest network. As the honest chain is not aware of the attacker's chain, the honest chain will grow long enough (it is safe to assume that the honest network has a significantly higher combined hashrate power as compared to the victim's node) for tx2 to be accepted by the network and termed as valid.

This will result in tx1 being invalidated. The attacker has successfully receive a promised value from the victim without sending anything in exchange.

How is this attack executed?

This attack is executable when a dishonest miner, who has control over 2 full nodes network, connect one of these (node A) directly to the service of an exchange. Then the second full node (node B) interconnects it with other nodes that are well positioned within the blockchain network. In order to know which nodes to connect to, the miner must monitor the instant in which the nodes transmit the transactions, and how they then propagate them to the other nodes in the network.

Thus, you will be able to know which nodes are the first to transmit the operations and you will be able to connect with the objective service and with well positioned nodes. After establishing the necessary connections, the miner generates a valid block privately. At that moment, create a pair of transactions, which will have different values, being a high value transaction and a low value transaction.

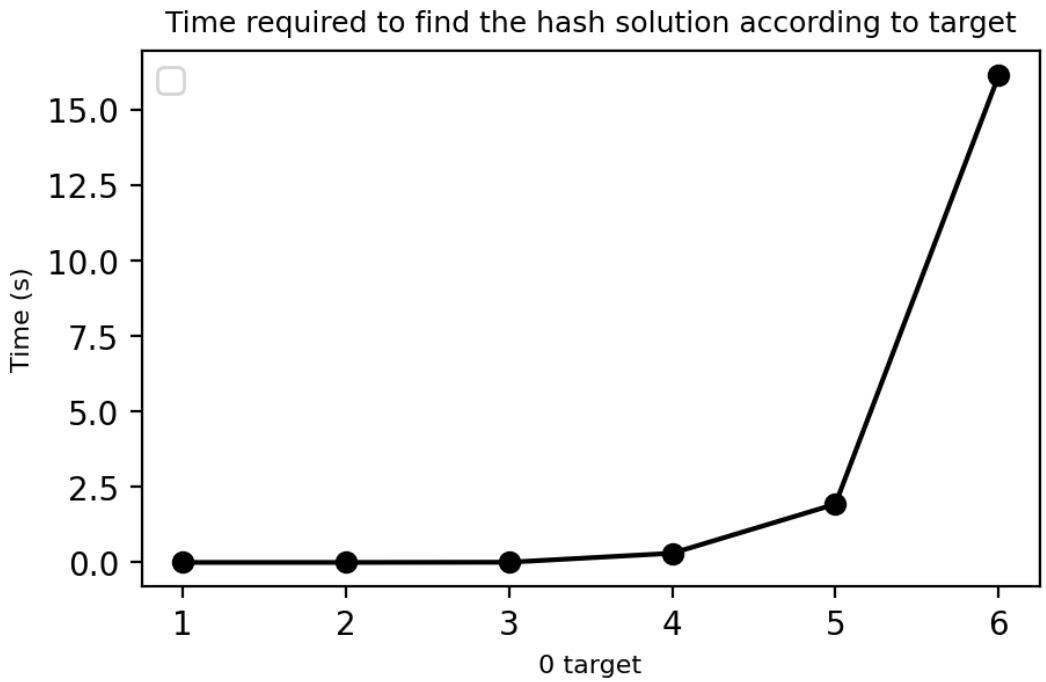
For example, the first transaction may be 25 BTC or more, and the second transaction may be as low as 0.1 BTC. Subsequently, the miner keeps the mined block on hold and assigns node A the high value transaction, that is, the 25 BTC transaction. This will be the transaction that will be directed to make a deposit within the exchange service.
When the miner detects a block advertisement on the network, it immediately transmits the block pre-terminated by it, directly to the exchange service, along with the recently generated block on the network. This in the hope that the rest of the nodes will consider their block as valid and assume them as part of the main chain. Thus, this block will be confirmed, and therefore, the 25 BTC transaction that is included in it is validated.

Once the exchange service confirms the 25 BTC transaction, the attacker makes a withdrawal from the exchange for the same number of cryptocurrencies that they deposited in the previous transaction (25 BTC). The attacker then sends the second transaction created, the 0.1 BTC transaction, to the network from node B. In order to create a fork that causes the network to reject and invalidate the first transaction. If this fork survives, the first transaction with the 25 BTC deposit will be invalidated, but the withdrawal will have been made. Therefore, the attacker will have succeeded and the exchange will lose 25 BTC.

Alpha Financial
Technology Research

Proof of Work

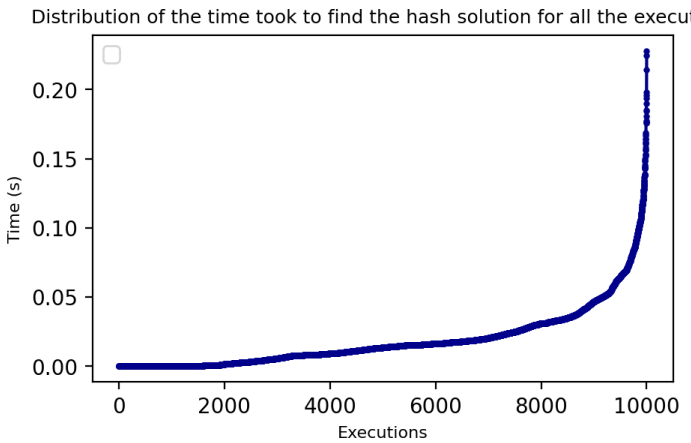
We are going to conduct a proof of work mining where we want to see how the average time needed to find the solution evolves with the difficulty adjustment. For that we will be running code which from this challenge 5JskLFx82fGh7eFP3c12XXX will add a generated random nonce and then hash the whole new variable until we the target of x 0's at the beginning of the new hash is obtained.



One can easily notice an exponential trend between the number of 0's required before the hash and the time needed to achieve it, which tends to increase exponentially.

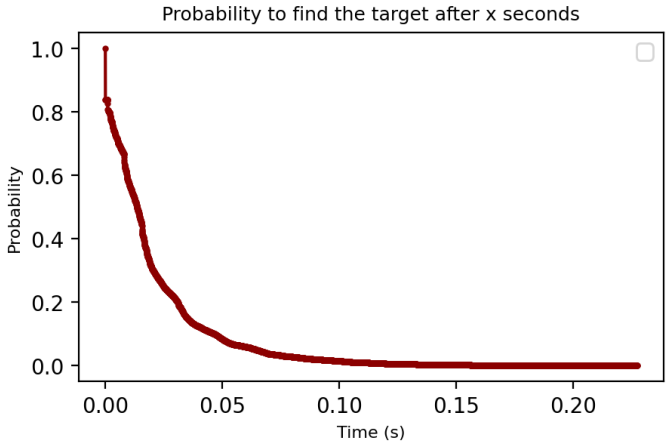
Exponential law

Let's repeat the same task 10 000 times but this time with a target of 3 only, here is a distribution of the time it took for each execution.



Obviously at a certain point it becomes more unusual that the search time exceeds a certain threshold.

From this we can then deduce the probability of meeting the target after a certain time. The computations are available in the code, the given results reflect in an obvious way an exponential law, according to the research paper the lambda parameter has a value of 1/600.



One plus two

The attack starts when the attacker mines a block (block A) and keep it secret, from that moment he will secretly try to mine two more blocks but keep in mind that if he wants to broadcast them all he needs to push more blocks than honest blocks (B) mined.

We define q the probability that the attacker finds a block, q is then strongly correlated to the hash power of this miner within the network. Thus the probability p=q-1 refers to the other shares of the network, the honest miners.

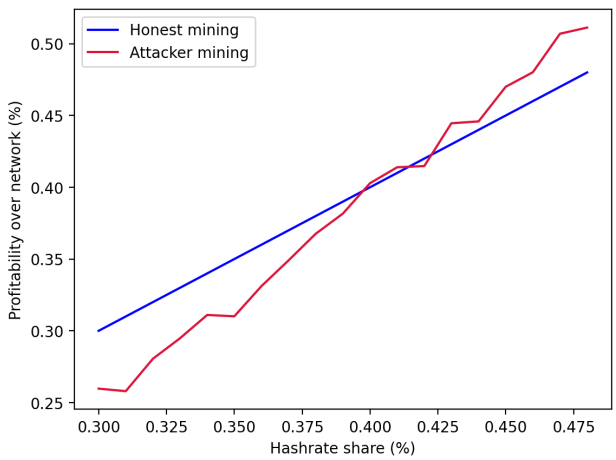
In the following table M is the probability that such a block sequence occurs, R and H are the coefficients respectively associated with the attacker and the honest miner for the expectation computation.

Block sequence	R	H	M	Description
B	0	1	p	Honest miner mined first, the attack cannot be performed.
AAA	3	3	q^3	1 + 2 successfull.
AAB	2	2	p*(q^2)	Attacker got 2 blocks and is ahead of only one, he pushes this two before being caught up.
ABA	2	2	p*(q^2)	Attacker got 2 blocks and is ahead of only one, he pushes this two before being caught up.
ABB	0	2	(p^2)*q	Attacker lost his advance and the first block secretly mined cannot be pushed anymore.

Therefore the expected values are :
•E[R] = 3(q^3) + 4p(q^2)
•E[H] = p + 3(q^3) + 4p(q^2) + 2(p^2)q

And the return should be E[R]/E[H]. The return level is heavily reliant on the probability p, i. e., the hashrate share of the attacker.

By running a simulation you can see graphically at what level this strategy is profitable, you can display the theoretical result next to it.

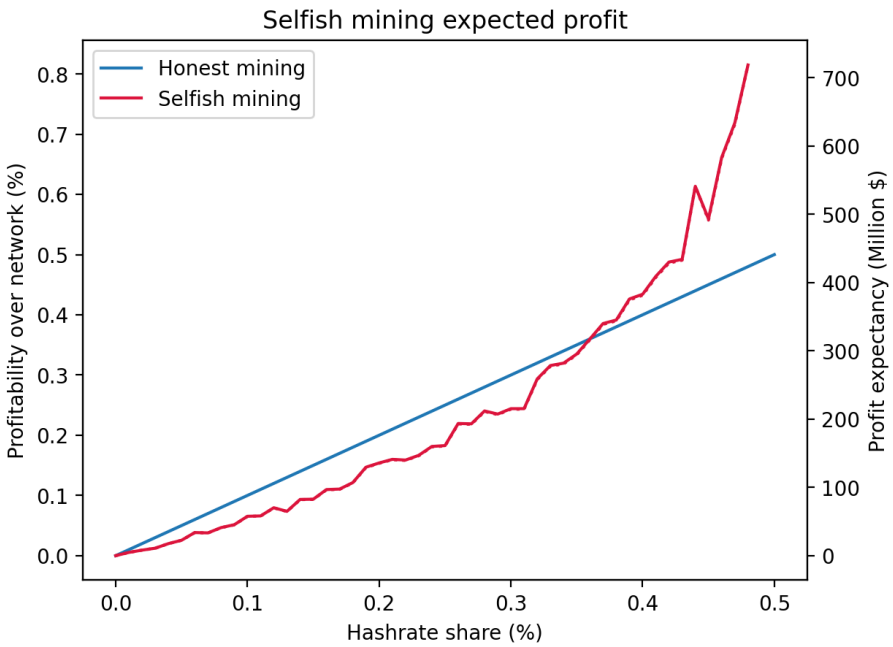


Selfish mining

The selfish miner continues to mine the next blocks but doesn't broadcast it maintaining his lead, therefore, there would be another fork in his hands. When the network is about to catch up with the selfish miner, it is at that moment when he releases his secretly mined blocks into the blockchain.
The rest of the network will consequently adopts this block solutions since the chain and proof of work is longer and more difficult.

Additionally the selfish miner get to claim all of his block rewards. The network connectivity is something important to take into account in this strategy. Indeed, if the selfish miner ever get catched up in the number of blocks he will need to broadcast his fork blocks right away and hope he will get chosen by the network, for that he will need to have a better connectivity than the miner who published the last honest block.

The parameters inputs for this simulation are the number of cycle attack and the connectivity with the network (%). The bitcoin price and the block rewards will affect the profit made by the selfish miner.



Double spending

Here is how the double spending attack works :

1. Start of the attack cycle.
2. The attacker mines honestly on top of the official blockchain *k* blocks with a transaction that returns the payment funds to an address he controls.
3. When he succeeds in pre-mining *k* blocks lading the honest miners, he keeps his fork secret, sends the purchasing transaction to the vendor, and keeps up mining on his secret fork.
4. If the delay with the official blockchain gets larger than *A* then the attacker gives-up and the double spend attack fails.
5. If the secret fork of the attacker gets longer than the official blockchain that has added *Z* confrmations to the vendor transaction, then the attacker releases his fork and the double spend is successful.
6. End of the attack cycle.If successful, the attacker has won all the published blocks rewards plus his double spend, otherwise 0.

Our simulation will only display the probability of success of such an attack not the gains made from it. Therefore the parameters inputs for this simulation are the number of attempts, the number Z of blocks confirmation and A the difference of blocks before giving up the attack.