Introduction to Cryptography 04/03/2018

Lecture 19: Altcoins-1

Instructor: Vipul Goyal *Scribe: Raghav Behl*

1 Limitations of Bitcoin

Bitcoin has several limitations, some of which were spoken about in the last lecture. Before looking at Bitcoin's limitations, it's important to discuss the basic properties of Bitcoin which cause a lot of them.

1. Bitcoin has a block size limit of 1MB, i.e. it can house a maximum of 1Mb of transaction information per block.
2. Bitcoin has a rate limit of 10 minutes, i.e. a maximum of one block is mined (read: added to the chain) every 10 minutes.

1.0.1 Scalability

These properties limit the number of transactions that the Bitcoin network can verify. Right now, the network can achieve a maximum of 7 transactions per second. We discuss this issue in more depth later in the lecture.

1.0.2 Slow transactions

Due to the network having a rate limit on the number of blocks added to the chain, the transactions take some time to complete. Moreover, due to fork resolution, a lot of wallets take precautions and wait for 2 or more confirmations (read: blocks to be mined) before registering the transaction.

1.0.3 Wastage of electricity

Bitcoin's network uses a Proof-of-Work algorithm, which has the basic properties outlined above. Miners solve complicated puzzles to verify transactions and earn Bitcoin as reward. This consumes a lot of electricity as the larger the Bitcoin miner network gets, the more the difficulty increases to mine Bitcoin. Consequentially, the network consumes more power than the country of Columbia at present.

1.0.4 Usability (Can be better)

Bitcoin is completely decentralized and while this is great for network security, it presents a lot of usability issues. In the Fiat Money system, a bank takes care of any issues arising with one's bank account, making the system very usable. No such structure exists in the Bitcoin world, one is responsible for one's own wallet and consequentially one's secret key. No reversals exist as a feature of the network.

1.0.5 Anonymity

The Bitcoin network makes all transactions between public keys visible. A user on the network participates in a transaction by using a public key, a cryptographic pseudonym. One can track users in a number of ways, from tracking payments to monitoring one's IP address related to transactions. Moreover, alt-coins like Zcash exist which employ Zero-knowledge proofs to make transactions completely anonymous.

1.0.6 Government Intervention and Regulations

Governments remain a big threat to the future of the Bitcoin network as they have various tools at their disposal to halt the Bitcoin network. For eg. Due to a reservoir of funds at their disposal, governments can potentially launch a 51% attack against the Bitcoin network. Governments control the monetary policy of their respective country, and crytpocurrencies like Bitcoin are a direct threat to that. So, the adoption and regulation of Bitcoin in most countries is still up in the air.

1.0.7 Quantum Computers

Quantum computers pose a real threat to all Proof-of-Work blockchains like Ethereum and Bitcoin as they are extremely powerful and can be used to launch a 51% attack against blockchain networks. The threat is a while away though and a solution is being thought through.

2 Mining Pools

Due to the extremely high hash rate supporting the Bitcoin network it is extremely difficult to reap rewards for small-scale miners. Consequentially, many miners pool their hash rate towards a mining pool in the search for greater rewards.

Mining pools employ an algorithm known as partial proof of work by which they put small miners to work on less computationally intensive problems. In the case of Bitcoin, it is to calculate a nonce with (k-k') zeroes. They can then calculate the work done by smaller miners and distribute rewards accordingly.

Problem with mining pools Pools can become very powerful and potentially launch a 51% attack. As of now, the mining pools have done a good job of policing themselves. At a point in 2014, a mining pool called Ghash.io accounted for 42% of the mining power of the Bitcoin network. They quickly moved to scale down, to reassure users, investors and miners.

3 Existing Proposals to solve blockchain Scalability

First, we need to ask why not increase the block size? Why not decrease the time it takes to mine a new block?

If we increase the block size to include more transactions, block propagation through the network will take more time, increasing the number of forks in the network. Hence, increasing the cost for new miners to enter the network. This will in turn lead to centralization of mining in the hands of the wealthy, increasing the probability of an attack.

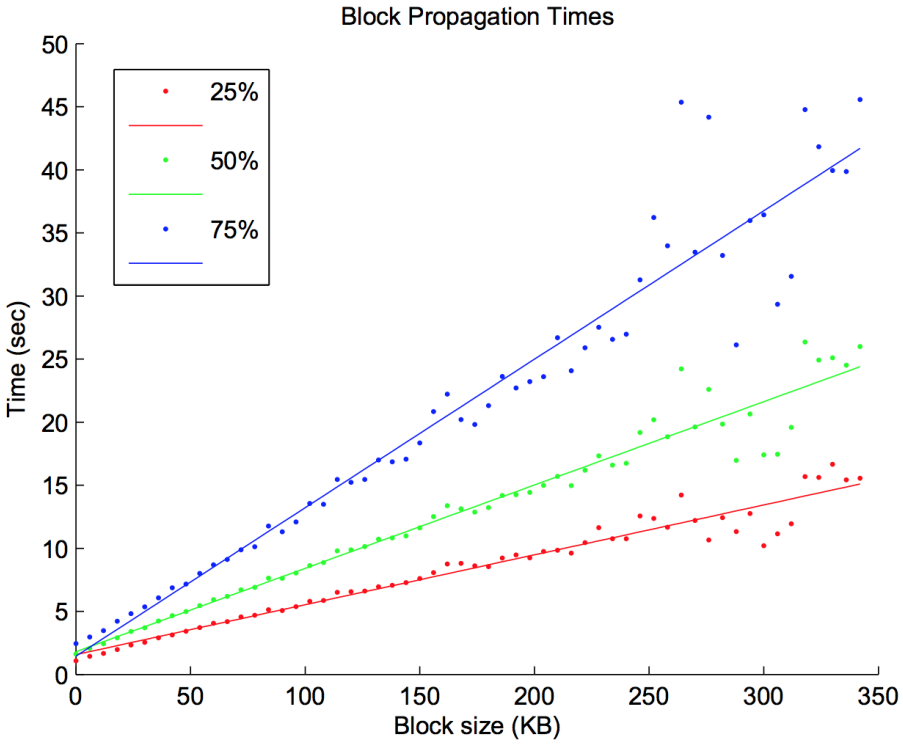

Figure 1: The relation between the block size and the time it took to reach 25% (red), 50% (green), and 75% (blue) of monitored nodes, based on data provided by Decker and Wattenhofer. [1]

If we decrease the time after which a new block is mined, effectively increasing the rate of information being added to the blockchain linearly, we will increase the number of forks in the chain. This will make it possible for a malicious miner to attack the network with less than 50% hash power.

3.0.1 GHOST (Greedy Heaviest Observed Subtree)

The GHOST protocol is an innovation first introduced by Yonatan Sompolinsky and Aviv Zohar in December 2013 to solve the scalability issue plaguing a lot of the proof of work based blockchains. The Ethereum blockchain implements a simplified version of GHOST which only goes down seven levels. Another implementation of this protocol, called Casper, is at present being proposed for the Ethereum blockchain by Vlad Zamfir.

Sompolinsky an Zohar researched the effects of increasing the transaction rate in the Bitcoin protocol by increasing the block size (figure 1) to hold more transactions and by increasing the number of block creation events. They found that both of these methods increase conflict between blocks and lead to the network being more susceptible to attacks like double-spending.[1]

GHOST selects the heaviest subtree rooted at the fork in case of a conflict. This is a departure from the longest-chain rule which is used by Bitcoin right now. The advantage of GHOST over the longest-chain rule is that no matter what the block size or block creation rate of the network is, the security of the network is not compromised.

Figure 2 is a scenario where GHOST shows it's security over the longest-chain rule in a forked chain. The attacker's chain (6 blocks) here is longer than the main chain of the network ending in 5B and would assume the place of the main chain under the longest-chain rule. But under GHOST, the shortest, but honest chain wins as it has more block weight.

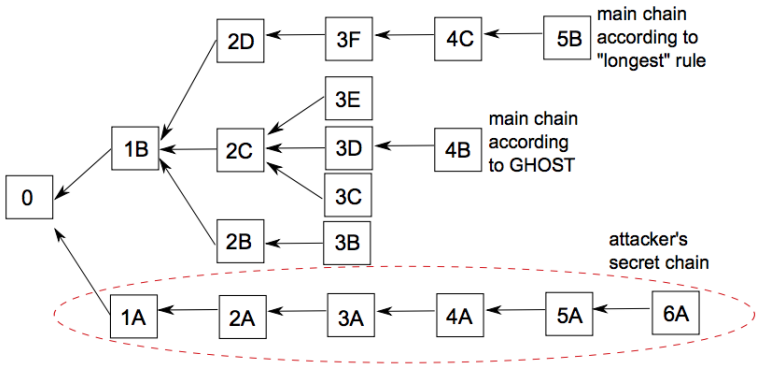

Figure 2: A scenario showing benefits of GHOST over the longest-chain rule. Here the attacker's chain would become the main chain under the longest-chain rule, but not under GHOST. [1]

A simplified algorithm is as follows:

Assuming the genesis block is the first block mined. Then, given the i^{th} block, pick the $i+1^{st}$ block as follows:

- Observe all of $B_i s$ children (subtrees)
- Choose the heaviest subtree i.e. one with most number of blocks
- The root of this subtree is B_{i+1}

Properties of GHOST

The Convergence of History

Let every t_b be the time every block B was either adopted or rejected by all the nodes. Then $Pr(t_b < \infty) = 1$. So, every block created is either fully adopted or fully abandoned.

Resilience to 50% attacks

Each node in the network creates new blocks at the rate of $c * \lambda$ (a Poisson process), where c is the computational fraction of the node in the network and λ is the rate of the Poisson process of block creation. *Assume an attacker with a block creation rate where it's $0 < c \leq 1$. The probability that a block B will be off the main chain at time(B) + t, where time(B) is the time the block was added to the main chain, tends to 0 as t goes to ∞.*

Problems with GHOST

GHOST does not solve all issues. In some cases miners and mining pools can exploit the Bitcoin protocol to gain better rewards than their fair share.

1. Miners better connected to the network gain rewards larger than their fair share.
2. Using the selfish mining strategy, small miners can deliberately add more forks to the network by keeping their blocks created private. This can enable them to create a private chain with a significant lead over the honest public chain. Eyal and Sirer describe this selfish attack in detail. [3]

3.0.2 Inclusive Block Chain Protocols

Motivation

The problem with present proof of work blockchains like Bitcoin is that the network security is compromised with high throughput. It's a trade off between security and throughput. We need a protocol where the protocol security imposes no bottleneck to the throughput of the network. The network will be secure under all throughput parameters and the limitations of the network will be the infrastructure supporting the network. This will bring more/faster transactions on the main chain.

Idea: SPECTRE: Serialization of Proof-of-Work Events, Confirming Transactions via Recursive Elections.

Spectre was proposed by Yoad Lewenberg, Yonatan Sompolinsky, and Aviv Zohar in 2016 to solve the throughput vs security trade off. They describe a DAG (Directed Acyclic Graph) based permissionless, distributed ledger that aims to maintain 50% resilience to attackers while enabling high throughput through the network.

Bitcoin has a hash function that considers only the previous block to form the next one, enabling the chain. Spectre allows several past blocks in its hash function which might have been created in parallel and be conflicting. The conflicts might include double-spending, where A tries to send money to B, and the same money to C, trying to bamboozle the network. However, this allows miners to create blocks concurrently and much more frequently as the nodes do not need to align their world view of the graph at the time of block creation.

Bitcoin's longest-chain rule can viewed as a voting mechanism. When conflicts in the chain occur, the nodes vote at the time of the next block's creation on which block they choose. This inhibits higher levels of throughput without disrupting the security of the network. SPECTRE employs a voting mechanism where the next block can safely vote on a conflicting previous block without making the miner align their world view at the time of the next block's creation. So, blocks are then voters and due to the design of SPECTRE, the majority vote becomes irreversible really quickly Using the majority vote one can extract all accepted (non-conflicted) transactions easily.

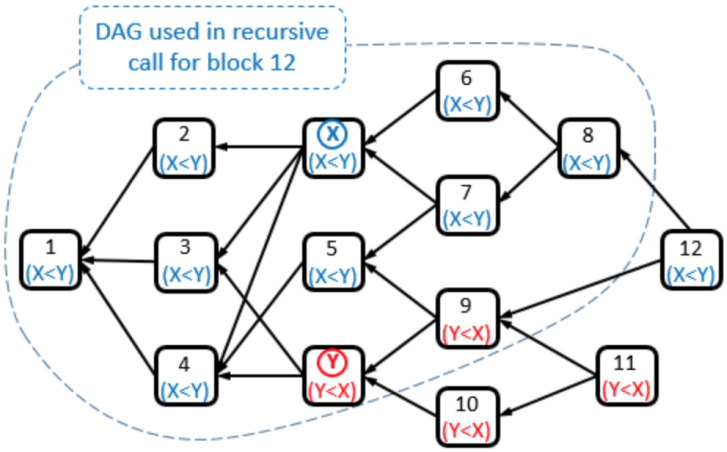

Figure 3: A voting procedure based on the order of blocks x and y. [2]

The voting process (Figure 3) to resolve conflicts is as follows:

- For a pair of blocks a and b in conflict, determine whether a defeats b (represented as $a < b$) or b defeats a (represented as $b < a$). Here $a < b$, or a defeats b, says that a is before in the order of precedence than b.
- Release a set of accepted transactions to the network. Accepted transactions in block a are ones which have defeated all conflicts and have all inputs accepted.

Appendix

Properties of SPECTRE [4]

Consistency Transactions are accepted if and only if the block wins the majority vote in a conflict and if all it's inputs are accepted.

Safety If a transaction is accepted by a node, then it's accepted by all nodes with a really high probability.

Weak Liveness A transaction without any conflicts, will be accepted by the main chain in the DAG really quickly.

References

[1] Yonatan Sompolinsky and Aviv Zohar, "Accelerating Bitcoins Transaction Processing Fast Money Grows on Trees, Not Chains." 2013.

[2] Yoad Lewenberg, Yonatan Sompolinsky and Aviv Zohar, "SPECTRE: Serialization of Proof-of-Work Events, Confirming Transactions via Recursive Elections." 2016.

[3] Ittay Eyal and Emin Gun Sirer "Majority is not Enough: Bitcoin Mining is Vulnerable." 2013.

[4] Aviv Zohar https://www.youtube.com/watch?v=5mEaBXl3BMM 2017