Notes on Cryptocurrency and Blockchain Protocols

Mahdi Zamani
Visa Research, Palo Alto, CA
mzamani@visa.com

Proof-of-Work. Introduced by Dwork and Noar [DN93], a proof-of-work is a proof showing that some moderately hard work has been done by the prover. Such proofs should be difficult to produce, but should be easily verifiable by the verifier. For example, guessing the correct sequence of a combination lock is a proof-of-work, because it is hard to find the correct combination but once produced, can be easily verified; just enter the combination and see if the lock opens.

A proof-of-work is often requested by a server from a client to protect against denial-of-service attacks by clients. The common scenario is that the client who is requesting a service from a server is challenged by the server to solve a moderately hard puzzle. The client then solves the puzzle and sends the proof-of-work to the server. The server verifies the proof and if accepted, grants the service to the client.

Definition. Let d be a positive number and c, x be bit strings. A function $f(d, c, x) \rightarrow \{0, 1\}$ is called a *proof-of-work function* if given d, c, and x, the function can be easily computed, but given d and c, finding x such that $f(d, c, x) = 1$ is computationally difficult but feasible. We refer to d as *difficulty*, to c as *challenge*, and to x as *nonce*. Given d and c, any x such that $f(d, c, x) = 1$ is called a *proof-of-work*.

In bitcoin, for example, the proof-of-work function returns true if and only if SHA-256 cryptographic hash function returns a bit string starting with d zeros in response to the input $x|c$.

The Flat Model [GKL15]. To model the hashing power of parties in a blockchain protocol, it is usually easier to assume all parties have the same quota of hashing power per round. The *non-flat model* where parties have different hashing powers can be easily captured by clustering the flat-model parties into larger virtual entities that are comprised by more than one flat-model party. For example, mining pools in Bitcoin can be thought of such aggregations of flat-model parties.

Common-Prefix Property [GKL15]. Let n be the total hashing power in the network and t be the total hashing power of the adversary. If $\frac{t}{n-t}$ is suitably bounded below 1, the blockchains maintained by honest nodes will have a large common prefix. More specifically, if an honest party removes k blocks from the end of its local chain, then the probability that the resulting chain will not be a prefix of another honest party's chain drops exponentially in the security parameter.

Let μ denotes the fraction of honest nodes, ρ denote the fraction of adversarial power, and p denote the hardness of the proof of work. Garay *et al.* [GKL15] show that if $\mu > \lambda\rho$ for some $\lambda > 1$ such that $\lambda^2 - p\lambda + 1 \geq 0$, then the blockchain maintained by the honest nodes will possess a large common prefix.

Chain-Quality Property [GKL15]. For any coalition of nodes (following any mining strategy), the fraction of blocks in the blockchain discovered by these nodes is exactly proportional to their collective hashing power. It is shown that the ratio of blocks in the chain of any honest node that are contributed by malicious nodes is bounded by $\frac{t}{n-t}$.

Chain Growth [KP16]. The number of blocks that are added to the blockchain during any given number of rounds.

Selfish Mining Attack [ES14]. A pool of parties keeps its discovered blocks private to intentionally fork the chain. When the public chain approaches the pool's private branch in length, the selfish miners reveal blocks from their private chain to the public in order to waste computational resources of honest miners. It is shown that the honest miners waste proportionally more than selfish miners. Therefore, the selfish pool's rewards exceed its share of the network's mining power, incentivizing rational miners to join the selfish mining pool.

Proof-of-Stake (PoS) [KRDO16]. The idea is to pick a node as the leader with a probability that is proportional to its stake in the system (i.e., the number of coins the node owns). Similar to a PoW-based consensus, the chosen leader gets to broadcast a block to the network. This process is called *minting* (rather than mining as in PoW-based consensus). PoS needs a secure coin flipping protocol to select an unpredictable leader randomly with respect to the distribution of stakes.

Bribery Attack. The attacker bribes some miners to not publish their blocks to the network so that the attacker can build an alternative chain secretly with their blocks and based on a block prior to the one containing a transaction T. After T gains the necessary number of confirmations (e.g., six), and the attacker's chain becomes longer than the honest chain, the attacker publishes its chain. His chain will be accepted as the new valid chain, and as a result, T is reversed after being confirmed. Most PoS-based consensus protocols including [KRDO16, Mic16] are vulnerable to bribery attacks while such attacks are much harder in PoW-based consensus, because in the latter, bribed miners have to solve new PoWs again and within a short amount of time to reverse an already-confirmed transaction.

Coin vs Token. A coin is a unit of a digital currency that can be used like a fiat money (e.g., US Dollar) to transfer values and pay for any goods and services. In contrast, a token is a *ticket* that grants access to a particular service or resource. Usually, a token can only be used to buy a specific shared computing resource such as bandwidth, computation, and storage. Since tokens are backed by some external resources/services, they are usually less vulnerable to speculative investment problems that most digital coins suffer from. A person who has a token is free to sell it for any price that the market is willing to buy. Since the total number of tokens is usually limited, the cost of the token typically increases as less tokens become available unless the resource/service behind the token loses its value (e.g., is no longer seen useful).

Sidechain [ABW14]. A sidechain is a blockchain that independently grows in parallel to a main blockchain (called a *mainchain*). A sidechain is usually accompanied by a sidechaining mechanism that allows tokens from the main chain to be securely used within the sidechain and vice versa.

Orphaned Block. A valid block that is not part of the main chain. Such a block can occur naturally when two miners produce blocks at similar times or it can be caused by an attacker (with enough hashing power) attempting to reverse transactions. Those blocks that do not receive enough confirmations (e.g., at least six blocks on them) become orphaned.

Stale Block. A block which has already been mined so any solution to it will not be accepted as a new block by the network even if the solution is valid. The difference between an orphaned block and a stale

block is that an orphaned block is initially accepted by the majority of the network, but is later replaced by another block if a longer chain (that does not contain the orphaned blocks) is detected.

Uncle Block.

Nothing-at-Stake Attack. Some protocols that rely their decisions on how much stake each participant has in the system are vulnerable to a situation, where malicious participants transfer their stake into someone else's account and therefore have nothing to lose if the system's security breaks. This happens because such protocols build their security based on the idea of giving higher priorities to participants with higher stake in the system.

Mining Pool. When the difficulty for mining increased to the point where it could take a long time (like years) for individual miners to generate a block, bitcoin mining pools were created. With mining pools, miners can pool their resources together and share their computing power while splitting the mining reward equally according to the amount of shares they contributed to solving a block.

Transaction Processing Limit. As of 2016, Bitcoin's current block size limit is 1MB. To calculate the number of transactions per second (TPS) Bitcoin can process, we divide the block size limit (1MB) by the average size of transactions (250 bytes), divided by the average number of seconds between blocks (600 sec). This results in 6.6 TPS. In practice, bitcoin can handle 1-3 TPS as of September 2016 [Blo16].

Bitcoin Incentive. The miner who finds a block is rewarded with a new (i.e., never used before) bitcoin which is transferred to him via the first transaction of the block. The block reward creates an incentive for miners to support the network and to generate bitcoins.[1]
As of 2016, mining for bitcoins is more profitable than transaction fees. As more bitcoins are mined, the reward per bitcoin will diminish toward zero and the miners will profit more from processing transactions. The block reward started at 50 bitcoins in the *genesis block*, the first block ever created in 2009. This reward is divided by half every 210,000 blocks which take about 4 years to be mined on average.[2] As more bitcoins are mined, the block rewards will get smaller. As a result, transaction fees will become the only source of profit for miners.

References

[ABW14] Luke Dashjr Mark Friedenbach Gregory Maxwell Andrew Miller Andrew Poelstra Jorge Timon Adam Back, Matt Corallo and Pieter Wuille. Enabling blockchain innovations with pegged sidechains, 2014. https://www.blockstream.com/sidechains.pdf.

[Blo16] Blockchain charts: Transactions per block between April 2016 and September 2016, 2016. Available at https://blockchain.info/charts/n-transactions-per-block.

[1] Bitcoins are generated only from block rewards and transaction fees

[2] Since blocks are mined on average every 10 minutes, 144 blocks are mined per day on average. At 144 blocks per day, 210,000 blocks take on average four years to mine. In the first four years, 10,500,000 bitcoins were mined. Therefore, using a simple geometric series computation, the total circulation will be 21,000,000 bitcoins which will happen after about 95 years from 2009.

[DN93] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Advances in Cryptology — CRYPTO' 92: 12th Annual International Cryptology Conference Santa Barbara, California, USA August 16–20, 1992 Proceedings*, pages 139–147, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[ES14] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, pages 436–454, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[GKL15] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310. Springer, 2015.

[KP16] Aggelos Kiayias and Giorgos Panagiotakos. On trees, chains and fast transactions in the blockchain. Cryptology ePrint Archive, Report 2016/545, 2016. https://eprint.iacr.org/2016/545.

[KRDO16] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. Cryptology ePrint Archive, Report 2016/889, 2016. http://eprint.iacr.org/2016/889.

[Mic16] Silvio Micali. ALGORAND: the efficient and democratic ledger. *CoRR*, abs/1607.01341, 2016.