Contents

1 A general introduction to digital currency

For all digital currency solutions, the double spending problem must be solved. That is, they must deal with the potential flaw of allowing an amount of currency to be spent more than once. In more traditional online transaction systems, banks act as a central trusted authority. They can screen transactions, deciding which are valid and invalid. Additionally, if double spending is inadvertently allowed, they can simply charge the user a fee with the full authority of law. This allows for a simpler solution to the double spending problem [1]. Unfortunately, it also creates a single point of failure in terms of both availability and integrity. In cryptocurrency solutions, this problem holds greater weight as they tend to be distributed by nature. Solving this problem using a peer to peer network was a big part of Satoshi Nakamoto's Bitcoin whitepaper which created the idea of a blockchain. In this paper he proposed a currency in the form of a chain of digital signatures. These signatures contained information about transactions, and are grouped together into contiguous units called blocks. These blocks are chained together, forming the blockchain which contains all transaction data since the currencies genesis. A user's account is determined by an address, and the user's balance is solely determined by the net sum of all transactions which have ever occurred on that address. This blockchain would be shared across

the network. However, in order to solve the double spending problem, there has to be a way for the network to determine the true blockchain. It would still be possible for different versions of the blockchain to exist in the network, and some of these chains would disagree about the state of a particular user's account. This is where Nakamoto's consensus algorithm of Proof-of-Work comes in, which will be described in the following section along with a more detailed look at the exact mechanisms surrounding [2]. Since then many cryptocurrencies have spawned with many different configurations and consensus algorithms.

2 Types of Blockchain

Since Nakamoto's proposal, many different implementations of blockchains have spawned, with many different configurations and consensus algorithms. His proposal was of a public, permissionless blockchain, and most but not all of the blockchains since then have followed suit. Generally, blockchains can be classified as public or private based the ability to read and submit transactions to the blockchain. In a private blockchain, only certain users would be allowed to participate. Additionally, blockchains can be classified as permissioned or permissionless. Permissionless blockchains allow any user or machine to participate in the verification of transactions into blocks, which grow the blockchain. Included in this is the process of coming to a consensus about the true state of the blockchain. Permissioned blockchains have preselected authorities which may verify transactions and decide upon the state of the blockchain. Generally all permissionless blockchains are public, as a user that could not submit transactions would have no incentive to participate in the verification and consensus process. Concurrently, permissioned blockchains are generally private or semi-private. From this point forward, these 4 classifications will be simplified as such to public and private [3].

2.1 Blockchain Permission

2.1.1 Public Blockchain

As discussed before, public blockchains tend to require going to greater lengths to solve the double spending problem. There is the potential for malicious users to be involved, and consensus algorithms must be robust enough to handle this. Generally, public blockchains use proof of work or proof of stake consensus algorithms. These algorithms will be discussed in greater detail later, but generally they give a user influence in the consensus

process proportional to some resource held by that user. Rather than place trust in a central authority, users place trust in the blockchains algorithm which usually comes with concrete, mathematical guarantees. These algorithms usually have clear avenues of manipulation, (e.g. in the form of a 51% attack where users holding over half of the resources involved in consensus collude in order to manipulate the system), but they require such a prohibitively large amount of resources or occur with such an infinitesimally low probability that the the blockchain may treated as secure. As a corollary, users also gain the certainty of knowing that there is no central authority that could at some point manipulate or change parts of the system without users' consents. A final advantage of public blockchains is the network effect gained by being a open. A public blockchain may more easily gain users which inherently gives the blockchain greater value. A currency that has no users has little value [4]. So far, public blockchains have mainly been discussed from the perspective of integrity. Generally, public blockchains offer little in terms of confidentiality as transactions are public. There are, however, specific public blockchains geared towards privacy such as Monero which aims to obsfuscates the sending address, receiving address, and amount transferred by transactions [5]. How this achieved will be discussed later. From the lens of accessibility, public blockchains tend to be much better than more traditional centralized currencies. Due to their distributed nature, public blockchains tend to remain resilient to both target denial of service attacks as well as localized network outages.

2.1.2 Private Blockchain

It is possible to further divide private blockchains into fully private blockchains and consortium blockchains. In a fully private blockchain, consensus is carried out by a single organization. These may be useful for blockchains used internally by a company for things like data storage and internal auditing. Consortium blockchains involve a pre-selected set of participants in the consensus process. This is useful in situations where several companies have a common aim and wish share common data and split computational cost. Since private blockchains involve a much smaller and often colluding/cooperating participants, it is much easier to change the consensus algorithm, offering private blockchains greater flexibility. This also opens up the possibility for the blockchain to manipulate transactions, for instance reclaiming currency transferred as the result of a hack or exploit. As another advantage, since there are fewer node, private blockchains may reach a consensus much more quickly than public blockchains. In addition, pri-

vate blockchains do not need to worry about a 51% attack as validators are known to be cooperative. Private blockchains may also more naturally offer confidentiality given that read permissions are restricted. Since private blockchains are most often carried out by private organizations, they may afford the resources to ensure that nodes are well-connected and may afford quick repairs in the event of network outages [4]. Despite this, private blockchains remain more vulnerable to denial of service attacks.

2.2 Blockchain Consensus Algorithms

2.2.1 Proof of Work

Proof of work is the solution that Nakamoto proposed for the first blockchain. It treats computational power as a resource which a user may use to gain influence in the consensus process. In proof of work, transactions are shared across the network between nodes, which group them into blocks. These blocks must then be hashed in such a way that they are linked to the last block in the existing agreed upon blockchain. This hash is computationally expensive, and must be tried repeatedly with different values of a nonce, a part of the hash input. Once a nonce is found which satisfies a certain requirement (in this case a certain number of repeating zeroes at the beginning of the hash) the block may be added to the chain. As compensation for performing this computation, the user which found the hash is compensated a certain amount of currency in this block. However, not every existing blockchain is the true blockchain. Participants should only honor the longest blockchain. The records existing in other blockchains are irrelevant. As a result, those participating in consensus are incentivized to only attempt to add blocks to the longest blockchain. Otherwise, there computation will not reward them, as adding blocks to a shorter blockchain would not result in any actual gain. However, it is possible that two users could find hashes to add a block to the longest chain at nearly the same time. This is not a problem though, as it is unlikely that new hashes for both of these chains will be found a approximately the same time again, so eventually one chain will outpace the other. In order to ensure that these probabilities fall within acceptable ranges, the hash must be difficult enough that blocks are not added too frequently resulting in these types of race conditions. On the other hand, the hash can not be too difficult or transactions will take a long time to process. As a result, the difficulty of the hash changes. In Nakamoto's proposal this is done by altering the number of zeroes required at the beginning of a hash. It should be noted, that the level of trust in a transaction is not static. When

a block is first found containing a transaction, it is the most likely that another blockchain could outpace that chain, to become the agreed upon chain. As more blocks are linked onto an existing block, more trust can be placed in those transactions. In the event, of an attack attempting to outpace the accepted blockchain in order to modify past blocks, Nakamoto computed the following probability that the attack succeeds given they have a certain share of the computational power in the network. Note, that even 10% share for bitcoin today would be a tremendous amount of computational power [2].

2.2.2 Proof of Stake

Proof of stake algorithms are quite similar. The main difference comes from how new blocks are created. In proof of stake the algorithm randomly selects a participant the right to create a block. The participant can then create this block by linking it to some previous block of their choice. That block will also reward the chosen participant a certain amount of currency for participating in the consensus process. The chance that a participant is selected is proportional to the amount of currency that user holds. Like in proof of work, only the longest blockchain is honored, so participants are incentivized to link to the longest exist blockchain so that they may receive their reward. Proof of stake algorithms hold numerous advantages of proof of work. From a security perspective, proof of stake offers much less risk to integrity than proof of work. Both algorithms' primary threat comes in the form of 51% attacks. In proof of stake 51% attacks are much less viable for two reasons:

1. As a user attempts to gain greater share of the currency, the price of the currency will be driven up by demand. This makes acquiring a majority nearly impossible

2. Even if a user were to gain a majority of the currency, they would then be heavily incentivized not to maliciously manipulate the blockchain. This is because they have an incredibly large stake in the value of that currency, and manipulating the blockchain would surely cause the price of the currency and the value of their stake to drop dramatically.

In addition, to the added security benefits proof of stake can be seen as a much more environmentally friendly (and as a result a more economical) approach as proof of work results in large amounts of unnecessary electrical consumption as the result of the computation involved in computing hashes [6].

2.2.3 Byzantine Fault Tolerance

Byzantine fault tolerance is characteristic which denoted an algorithms ability to remain resilient in the face of such difficulties as malicious actors, unstable connections, and hardware failures. It is represented as a fractional amount of a system that may fault without comprising the integrity of the system. The name comes from a classic paper in computer science call the The Byzantine Generals Problems. This paper outlined an abstraction for building robust computer systems. In this abstraction several Byzantine generals surround an enemy city. The generals must decide if they should attack or retreat; however, the generals are only able to communicate via messengers. As part of the problem, the messengers may be intercepted or otherwise fail to deliver their message, and some of the generals may be traitorous and purposely send incorrect or conflicting messages. It would be ruinous for some generals to attack while other retreat, so it is essential that they attempt to follow the same plan despite these difficulties [7]. While this description does not capture the full complexity of the problem, it should be clear how this problem maps very directly to consensus algorithms in a blockchain. Since this papers publication nearly 40 years ago, much research has gone into analyzing the problem and proposing solutions. Some blockchains such as Tendermint, have created consensus algorithms by adapting solutions to the Byzantine Generals Problem [8]. In addition, there exists a variation of proof of stake based upon Byzantine fault tolerance in which consensus can be achieved after a single block. Aside from assisting in algorithm design, Byzantine fault tolerance provides another method of analyzing existing consensus algorithms. Such analyzes have been performed on existing proof of stake and proof of work algorithms. One analysis put Bitcoin, the original proof of work blockchain, at ~49.5% fault tolerance under typical network conditions, and Ethereum, the most popular proof of stake blockchain, at ~46% [6][9].

3 Additional Considerations

In addition, to these configuration many other exists each with their own use case and security profile. Even with a single classification, many different implementations can alter use case and security profile. For instance, Monero, another public proof of work blockchain, was created to emphasize confidentiality. In Monero public addresses are distinct from those that take in transactions. When a transaction occurs special one-addresses are created and linked with a users public address. The currency is then trans-

ferred between these special addresses to become usable by the recipient and unusable by the sender. However, beyond this additional layer of a public address mapping to multiple other address, there is further obsfucation in the form of transaction mixing. This is where a during a transaction, funds are shuffled between addresses in such a way that the net result of the actual underlying transaction is not clear from a single physical transaction. As a result of this obsfucation, a degree of accessibility is lost as the transactions shown on the blockchain are essentially meaningless. A user may only know about transactions they participated in using a secret key linked with their public address [10].

4 Conclusion

Blockchains are an interesting new technology with a lot of potential for changing how we look at currencies as well as security and distributed computing. The consensus algorithms used by blockchains are constantly evolving, and the next generation of blockchains may look quite different from those discusses here. Even in the current ecosystem, blockchains are very diverse and scattered. In fact, while we mainly considered blockchains which aim to implement a digital currency, many other use cases for blockchains exist. Some have proposed their use for things like unique personal identification and database access control [11][12]. These sorts of applications will surely have diverse security profiles and interesting algorithmic solutions. For these reasons, blockchains are worthy of attention as new opportunities and capabilities for their use are explored.

5 References

[1] U. Chohan, "The Double Spending Problem and Cryptocurrencies," University of New South Wales, December 19, 2017.

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Bitcoin.org, 2008.

[3] S. Aras, V. Kulkarni, "Blockchain and Its Applications - A Detailed Survey," International Journal of Computer Applications, December 2017.

[4] V. Buterin, "On Public and Private Blockchains," Ethereum Blog, August 7, 2015.

[5] "Introduction," The Monero Project, 2018.

[6] "Proof of Stake FAQ," Ethereum, 2018.

[7] L. Lamport, R. Shostak, M. Pease, "The Byzantine Generals Problem," SRI International, July 1982.

[8] "Consensus Overview," Tendermint, 2018.

[9] A. Miller, J, LaViola Jr. "Anonymous Byzantine Consensus from Moderately-Hard Puzzles: A Model for Bitcoin," University of Central Florida.

[10] "A Low-Level Explanation of the Mechanics of Monero vs Bitcoin in Plain English," Monero. 2017.

[11] H. Orman, "Blockchain: the Emperors new PKI?", IEEE Internet Computing, April 24, 2018.

[12] O. Nathan, A. Pentland, G. Zyskind, "Decentralizing Privacy: Using Blockchain to Protect Personal Data," IEEE Security and Privacy Workshop, July 2015.