Peer to Peer Networks: Hidden Transactions in Blockchains

Aniket Sharma
Roll No. 170111

April 2021

1 Abstract

Blockchains are a secure method to keep track of transactions between the nodes in a Peer to peer network. This technology is widely used in the cryptocurrency sector, most popularly is Bitcoin. Though it is good for secure transactions, but also one's privacy should be maintained. As blockchain is regarded as the most secure technology but does not allow much privacy, so we shall take up the topic of introducing hidden transactions (or private transactions) in blockchains.

We shall first see the working and security principles of blockchain and then we shall figure out which factors in blockchains act as a constraint for a hidden transaction. In the rest of the part I shall include my take on how the hidden transactions can be incorporated without altering the main framework of blockchain which ensures security.

Blockchain first was introduced in Bitcoin in 2009 by Satoshi Nakamoto. So throughout the report we shall see the examples or demonstration in terms of bitcoin's functioning.

2 Key Principles of Blockchains

Blockchain is a decentralized way of securing the information about all the past transactions. Blockchain is built upon a peer to peer network which helps the nodes to constantly monitor the information and figure out a false transaction(s). In this section we shall look at the functioning that enables the blockchain to do so.

When a node in a P2P network makes a transaction to another node in the network, then the sender of the bitcoin must send a message to the network containing the information about the (sender, receiver, amount). Further to make sure that it is the genuine sender that sends the message, the sender has

to send the message with a digital signature on it using the sender's node's RSA generated private key. This signed message broadcasted to the network constitutes a transaction.

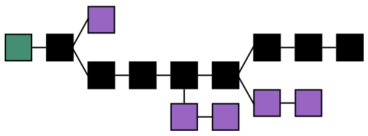

Figure 1: Visualization of a blockchain.

The term Block refers to a set of transactions which are encapsulated and secured in one phase of the blockchain and have a 'proof of work' entity attached to it. A block contains the following information:

- The hash of the previous block
- The list of transactions made
- Proof of work

The blocks are arranged in a sequence with every new block appended in the end of the sequence, it is possible for new blocks to create a new branch(fork) in the tree, but we shall see how it is taken care of, in next section. The hash of the previous block is done through a cryptographic hash function, SHA256.

The proof-of-work is a number or a sting that must be attached along with the block. For a block to be valid, its hash must satisfy a certain accepted criteria, for example in bitcoin the hash of a block must have n-many number of 0's in its hash's starting n-bits (the n changes over time). As can be observed, the only way for a cryptographic hash to meet a given condition is through hit and trial. So here to create a block that satisfies the given criteria one need to do hit and trial to get a valid proof-of-work.

When a node calculates the proof-of-work then the block is completed. This node then sends the blocks to all the other nodes in the network, and following which all the nodes update their blockchains. As proof-of-work is a purely hit and trial methord so it takes some time for a block to form (called block time), so it can be said that each block has some computational power, resources and time associated with it.

Another and the most crucial principle of blockchain is that all the nodes tend to give credibility and authenticity to the list of blocks which most computation time associated with it. Hence it will always prefer the longest chain and discard the other forks in the tree as they will be faulty or fake chains.

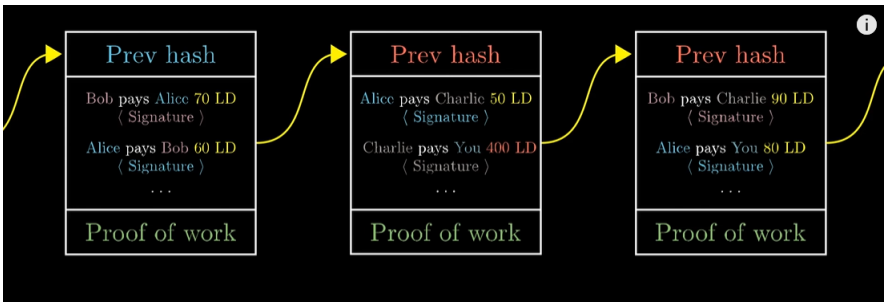

Figure 2: Links and content of the blocks.

3 How is blockchain secure?

If one needs to temper with the data then there are two possible scenarios which can be attempted to create a fraud. We shall see them as well as see how the blockchain tackles each of them.

1. Tempering with the data. If one attempts to tamper with the data stored, ie say change a information in a transaction stored in one of the blocks, then notice that when the data is changed then the hash value of the block will also change and due to the cryptographic hashing, will no longer make the block valid. To make the block satisfy the criteria of the hash, the person who has tempered must generate the new proof-of-work that shall satisfy the criteria on the hash. Now, notice that in the block we also incorporate the hash of the previous blocks. So here the successor of the tempered block should change the hash of the tempered block, therefore it will also require a new proof-of-work. This logic will repeat for all successive nodes and hence the proof-of-work for each node after the tempered node will have to be calculated again. As we have already showed that calculating proof-of-work will require a lot of resource and time. So with limited resource one can't create the longest forking chain of blockchain and hence the blockchain will never consider it's fork as a valid list of transactions.

2. Adding a new block with faulty transaction list. If one attempts to add a faulty block to the current blockchain, then he/she must also be able to built upon its successor blocks to keep it's corresponding fork as the longest fork in the tree and hence be considered the authentic one. But since with limited resources one can't keep that pace with competition from other nodes with much more combined resources, hence it becomes impossible to execute such fraud.

4 Drawbacks of using Blockchains

The following are the disadvantages of using the blockchain in a P2P network, and may constraint P2P networks to use blockchain technology.

1. Requires a lot of computational power and resources.
2. All transactions are publicly available.
3. Complicated implementation and algorithms for finding the proof-of-work
4. Does not provide instant transaction authentication in the blockchain tree due to block time.

5 Returning to the problem of Hidden Transactions

As we saw that Blockchain operates on peer to peer network, so its framework is decentralised, ie, each node in the network maintains its own copy of 'blocks' which represents all the past transactions. Although it is important and acts as an advantage to the blockchain by maintaining its security, but still by making all the transactions publicly available it makes a compromise with ones privacy. As a person/node's all transactions are publicly visible to everyone, his/her privacy may be disturbed. We shall try to tackle this problem in the below discussion.

6 Potential Idea for Enabling Hidden Transactions

This section we will discuss the foundation of the key problems that are encountered while trying to figure out a mechanism for hidden transactions, which will be followed by my ideas on solving those problems and implementing hidden transactions in blockchains. One of the main security aspects of the blockchain was that it did not rely on any central entity, but instead each node keeps a copy of the block chains, thus to compromise the security one needs to take control of atleast half of the nodes in the network. As we are willing to implement the hidden transaction's with the same blockchain-level security, so it is best to not have a central entity to be an intermediately. While now there are firms that may facilitate the transactions while keeping the identities anonymous, but for my study, I tried to find the alternative Peer to Peer technique to do the same.

In the usual blockchain, whenever a transaction is made, then all the nodes involved in the transaction (eg sender, receiver etc.) publish the transaction details signed by the nodes to inform the minor nodes about the transaction. The minors use the nodes public key to decrypt and confirm that the transaction is verified by the node.

If we want to keep the identities of the nodes anonymous, then the following need to be done

1. The nodes involved in the process must not be the nodes that sends the publish request for the transaction to the minors.
2. The transaction details can no longer be digitally signed using the node's private key. As otherwise the signer is bound to reveal his/her identity so that the minor can use the corresponding public key of the node to decrypt the transaction details to process the transaction (eg how much bitcoins were transferred etc.)
3. In the Transaction details that the sender/receiver prepares, they must not mention the NodeIDs of the nodes involved in the transaction.

We need to address the above three problems with keeping the ground rules explained in the previous section intact to get the hidden transactions on blockchain possible while maintaining the same level of security. The following subsections discusses each of the problems successively, elaborating the problem and discussing proposed solution.

6.1 Problem 1: Transaction details publisher

If the nodes involved in the transaction publishes the details then they can be traced down by finding the sender of the transaction details. To maintain their anonymity, a third party node that is not involved in the transaction must be chosen and it should publish the details of the transaction. There are two options, first, we can have some predefined nodes that can be used as intermediately nodes for publishing the transaction details, or second option is that the publishing node can be randomly chosen for each transaction. As told earlier that my approach is to find a reliable peer to peer technique for implementing hidden transactions. As the former approach is central based, so we shall be going ahead with the later method, which is to chose the publishing node randomly.

Let us take the example of bitcoin. Let's say there is a transaction between two nodes S and R, where node S is the sender and node R is the receiver of cryptocurrency in the process. Now, in my proposed method, nodes S and R establishes a communication and mutually generate a shared random number Ran. The random publisher node then shall be

$$Y = (S + R + Ran)\%N$$

Here N is the number of nodes in the DHT network. Both sender(S) and receiver(R) shall now send the transaction details to node Y, which will substantiate the transaction and send a request for its publishing to the minors.

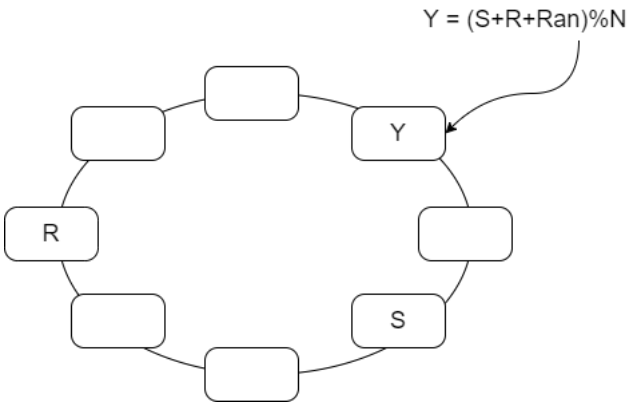

Figure 3: Publisher node Y in the DHT

6.2 Problem 2: Alternative of Signatures

As explained earlier, if the transaction details are signed for verification, then one must also reveal its identity (NodeID), so that the content can be decrypted with the corresponding nodes public key. Thus the node is no longer anonymous.

A potential way I developed to solve this problem is as follows:

1. The receiver node(R) and the sender node(S) of the bitcoin example establishes communication and generates a pair of asymmetric keys.

2. The receiver node, takes the transaction details and encrypts it using it's side of the asymmetric key. It then also generates a TransactionID by passing the transaction details through SHA256 cryptographic encoder. Finally it combines the encrypted transaction details, with the TransactionID and sends the combination to Node Y.

3. On the sender nodes side, as it also has a copy of the transaction details, hence it can also produce the same TransactionID as generated by receiver, by passing the transaction details through SHA256. Node S takes its side of the asymmetric key and along with the TransactionID sends it to Node Y.

4. Node Y shall receive the packets from both S and R. Trough the same TransactionID attached with the messages, it can deduce that the two messages are complimentary to each other. It shall the use the asymmetric key sent by Node S and use it to decrypt the transaction details sent by node R. If it is decrypted successfully, then this will ensure that the verification from both the sender and receiver is complete. Now node Y shall send a request for publishing the transaction details.

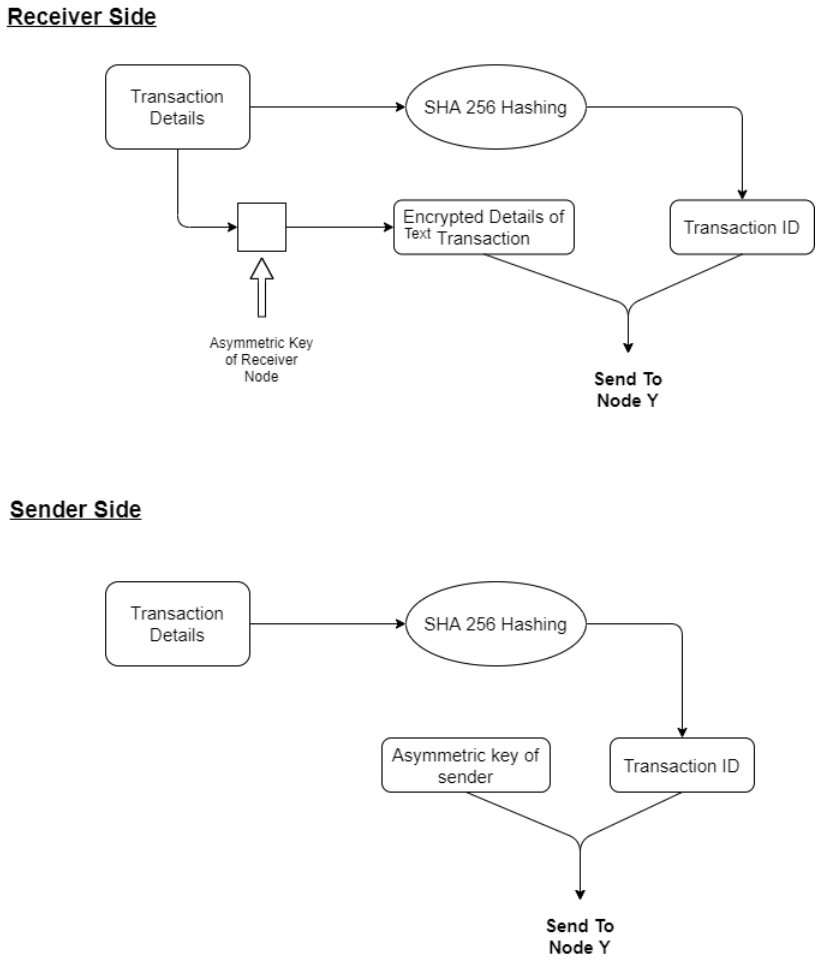

Figure 4: An overview of the process at the receiver and sender sides.

6.3 Problem 3: Hide NodeID's in transaction details

In transaction details, the nodeID of the sender and receiver can't be directly used else any node can read the blockchain and find the history of ones transactions. I though about creating an anonymous mapping of nodeID's to another set of hidden nodeIDs that shall correspond to a second DHT, which shall store each nodes concurrency balance. But this shall compromise the bitcoin level security, as then only the corresponding nodes which stores the copy of ones balance will have to be compromised to change the value of the balance.

7 Shortcomings and Future Steps

The above proposal has discussed the fundamentals of the task of creating a Peer to Peer hidden transaction mechanism in the blockchains. The proposed approach has also laid the ground rules and basis for future development of the technology, which shall be worked upon in the coming time. The inadequacies and drawbacks of the current proposal are listed below. These need to be worked upon in future while strictly respecting and maintaining the basics of blockchain and not compromising with it's security.

1. Node Y, has the responsibility to verify the transaction details from node S and node R by using the asymmetric key to decrypt the details sent from R. So this can also give the power to Node Y to send a faulty entry, and claim itself as the intermediately publisher for the entry.

2. The nodeIDs in the transaction details must be hidden and must therefore need to be mapped with an anonymous mapping as discussed in section 6.3. The proposed method in section 6.3 is inadequate because of it's low level security.

3. The messages sent by sender and receiver to node Y, have a chance to be strategically be tracked down in some DHT routings. For example chord with log routing table, a node with the nodeIDs of every node in the network, can use backtracking to narrow down the potential senders/receivers for which it was the intermediately node. As the nodeIDs are generated randomly, so there is a possibility that node Y can pinpoint the exact sender/receiver node.

8 A ray of hope

If we are able to enable hidden transactions then we will be successful in maintaining privacy in our transactions. This is a huge step, but can be used for unethical reasons also. Hence the technology should be handled carefully and the network should incorporate of only trusted people/nodes. If this is done right then it can be a boon for everyone, and if we don't prevent it from misuse then the same can turn into bane. So lets spread the word to safeguard this potential technology, and prevent its misuse.