Abstract:
A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and re-join the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Key words: digital signature, proof-of-work, peer-to-peer network.

I. INTRODUCTION

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust-based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating dispute. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party. What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

II. WHAT IS BITCOIN?

Bitcoin is digital currency released as an opensource software in 2009[2]. It is a decentralized cryptocurrency produced by all the participating nodes in the system at a defined rate. The chain of Bitcoins created over a period and are linked to each other called Blockchain. It can be used to search any past transaction happened over the network between Bitcoin addresses. When a new block of transactions is created, it gets added to the Blockchain [2]. The new transaction records are continuously added to Bitcoin's public ledger and this process is called Bitcoin mining. Secure Hash Algorithm 256 (SHA-256) [2] which is a cryptographic hash function is used by Bitcoin. We can determine the integrity of a given data by comparing the execution output of SHA-256 algorithm called “hash” with an already known and expected hash value.[2] A hash algorithm converts a large volume amount of data into a fixed-length hash. And same data will always produce same hash but any slight modification in data will completely change the hash.

III. WHAT IS BLOCKCHAIN?

Blockchain is a transaction database that contains information about all transactions that have been performed in the past and that work on the Bitcoin protocol. It creates a digital ledger for transactions and allows all network participants to edit the ledger in a secure manner that is shared across a distributed network of computers. To make any changes to the current block of data, all nodes in the network algorithms evaluate, verify, and match transaction information with the Blockchain record. If the majority of the contract agrees in favour of the transaction, it will be approved and a new block is added to the existing chain. Blockchain metadata is stored in the Google LevelDB by Bitcoin Core client.[2] We can visualize Blockchain as a vertical stack with blocks on top of each other and the bottom block acting as the basis for the stack. The individual blocks are connected to each other and indicate the previous block in the chain. Individual blocks are defined by fragmentation created with the SHA-256encrypted hash algorithm on top of the cluster. A block will have one parent but can have a multiple child, each of which refers to the same main block and therefore contains the same hash in the previous block hash field. Each block has a hash of the original block in its own head, and the fragmentation sequence connecting an individual block with its main block creates a
large chain that indicates the first block called the formation block.[2]

IV. WHAT IS GENESIS BLOCK?

The first block #0 created in 2009 is referred to as Genesis block in Blockchain.[2] It is common ancestral parent of all new blocks created and if traversed backward in time we will reach the genesis of blocks in the end. Genesis block contains a text message "The Times 03 / Jan / 2009 Chancellor on brink of second bailout for banks"[2]. The message was embedded in the first block by Bitcoin’s creator Satoshi Nakamoto. It offers proof of date when Genesis block was created which refers to the headline of British newspaper The Times. The donation for Genesis block was made By John Wnuk and Jayden McAbee on June 9 2016 which contains the first bitcoin wallet.[2]

V. HOW BITCOIN WORKS?

Bitcoin uses an encryption algorithm for the digital signature algorithm (ECDSA) to ensure that only legitimate owners have access to the funds. When Bitcoin is sent, it creates a transaction message and attaches the ECDSA public key to the new owner. Each Bitcoin is associated with the ECDSA public key of its current owner. A new transaction is broadcast over the Bitcoin network to inform everyone that the new owner of these currencies is the owner of the new key. Bitcoin kiosks are Internet-connected machines that allow depositing cash for paper receipts or by transferring money to a public key on Blockchain. When bitcoin is sent, it attaches the public key to the new owner and expects it with the private sender key. The sender’s signature on the message confirms that the message is original and that transaction history is kept by everyone so that it can be easily verified. It uses the asymmetric encryption algorithm for public keys and the concept of public and private keys for data encryption and decryption. If the message is encrypted using a public key (P_\text{k}), then a private or secret key (S_\text{k}) is necessary for decryption. Public keys can be shared with anyone, but private keys must be kept confidential. Participants can make public key pairs (P_\text{k} and secret keys S_\text{k}). Bitcoin does not need a third party because it distributes ledgers called Blockchain. Users who are interested in devoting CPU power to run special programs are called Bitcoin miners, and in the end, they form a network to defend the Blockchain. In the process of mining Bitcoin, users create new Bitcoin coins and transactions flow through the network. Bitcoin reduces the supply algorithm which means that the bonus for mining Bitcoin blocks is cut in half after every 210,000 blocks. The block making rate is adjusted every 2016 blocks to ensure making about 6 blocks per hour. The amount of Bitcoin created for each block is set to decrease geometrically and will decrease by 50% of every 210,000 pieces, almost 4 years.[2]

VI. PROPOSED SYSTEMS

Bitcoin can be used anonymously to make transactions between account holders, anywhere, anytime throughout the world, making it attractive to criminals and terrorist organizations. They can use Bitcoin to buy or sell illegal goods such as drugs or weapons. Most countries do not explicitly define the legitimacy of Bitcoin, preferring a wait and see approach. The decentralized and anonymous nature of bitcoin has challenged many governments about how to allow the use of law while preventing criminal transactions.[3] The system that we propose will be designed so that transactions can be tracked again when malicious actions are detected. For each addition of a transaction to the blockchain, a significant amount of time is wasted even though the transaction may look very similar to what actually happened. This can be regarded as one of the defects that can be somewhat overcome by our model with the ability to create groups of trusted parties, through which the private key can be shared so that transactions can be completed quickly. Bitcoin Wallet is a collection of private keys but can also refer to client software used to manage those keys and make transactions on the Bitcoin network. The original Bitcoin client stores private key information in a file called wallet.dat by following what is called the “bitkeys format”. Our system will suggest a method where private keys in the wallet are shared among group members and ensure an additional layer of protection for users to stay away from any dangerous or suspicious actions.

VII. DOUBLE SPENDING

We define electronic currency as a series of digital signatures.[1] Each currency owner is transferred to the next currency by digitally signing the hash of the previous transaction and the public key of the next owner and adding it to the end of the currency. The payee can check the signature to verify the ownership chain. The problem, of course, is that the recipient cannot verify that one of the owners has not duplicated the currency. The general solution is to provide a reliable central authority, or mint, that verifies each transaction to double expenses.[1] After each transaction, coins must be returned to mint to issue new coins, and it is not believed that coins issued directly from mint will only be spent twice. The problem with this solution is that the fate of the entire money system depends on the company that manages the mint, with every transaction you have to go through, just like a bank. We need a method for the recipient of the payment to know that the previous owner has not signed the previous transaction. For our purposes, the first transaction is attractive, so we don't care about the next double-spending effort. The only way to confirm that there are no transactions, is to know all transactions. In the mint-based model, mint is aware of all transactions and decisions that first arrive. To achieve this without a trusted party, the transaction must be announced publicly [8], and we need a system for participants to agree on a date for the order of receipt. The recipient needs to show that at the time of each transaction, most contracts agree that it is the first receipt.

VIII. TIMESTAMP

Our solution begins with a time stamp server. The timestamp server operates by taking a hash of a block of items to be temporarily stamped and widely published, such as in a newspaper or Usenet publication [5]. The timestamp proves that the data must be present at the time, obviously, in order to reach fragmentation. Each timestamp includes the previous timestamp in fragmentation, forming a chain, with each additional timestamp reinforcing the one before it[1].

IX. PROOF-OF-WORK

To implement a peer-to-peer timestamp server, we will need to use a working system directory, similar to Adam Back’s
Hashcash [7]. Proof of business includes a hash value scan at hash, as with the SHA-256, hash starts with a zero-bit number. The required mean work is exponential in the number of required zero bits and can be verified by implementing one hash. For our timestamp grid, we perform proof of work by increasing the nuance of the block until a value is found that gives the required zero bit-bit hash. Once the CPU effort is made to make it meet proof of work, the block cannot be changed without rework. Since subsequent blocks are bound by chains after them, work to change the mass will include returning all blocks after. Proof of action is basically one CPU and one vote. The majority decision is represented by the longest chain, which has the most evidence of the invested work effort. If the majority of CPU power is controlled by honest nodes, the honest chain will grow faster and outperform any of them. To modify a previous block, the attacker would have to re-prove proof of action for the block and all of the blocks after it, then catch up with the honest nodes and bypass them. [1]The likelihood of catching an attacker is greatly reduced with the addition of subsequent blocks. To compensate for the increased hardware speed and the varying interest in contract operation over time, the difficulty of proving work is determined by a moving average that targets the average number of blocks per hour. If created very quickly, the difficulty increases.

**IX. NETWORK**

New transactions are broadcast on all contracts. Each node groups new transactions in a cluster. Each knot creates a difficult clue to its mass. When a node finds evidence of action, it broadcasts the block to all nodes. The contract does not accept the ban unless all transactions in it are valid and have not been actually spent. The contract expresses its acceptance of the block by working to create the next block in the chain, using acceptable mass segmentation as a previous segmentation. The necklace always considers the longest chain to be the right chain and will continue to work on extending it. If two nodes broadcast different versions of the next block at one time, you may receive some nodes one or the other first. In this case, they work on the first branch that they received, but they save the other branch in the case where it becomes longer. The tie will be broken when the next proof of work is found and one branch becomes taller; The nodes that were working in the other branch will then be converted to the longest branch. New transaction broadcasts don't necessarily need to reach everyone. As long as it reaches many nodes, it will enter a cluster long before. Broadcasts tolerate dropped messages. If the node does not receive a block, it will request it when it receives the next block and realizes that it has missed.[9]

**X. INCENTIVE**

According to the agreement, the first transaction in a block is a special transaction that starts with a new currency owned by the block maker. This added incentive to the contract to support the network, and provided a way to distribute coins that were originally circulated, because there was no central authority to issue them. The constant addition of new coins is similar to gold miners who spend resources to add gold to circulation. In our case, CPU and electricity time is spent. Incentives can also be funded with transaction costs. If the value of the transaction result is less than the value of the input, the difference is the transaction fee added to the value of the block incentive containing the transaction. After a predetermined number of coins is circulated, incentives can be fully transferred to transaction costs and completely free of inflation. If a greedy attacker can accumulate CPU power over all honest nodes, he must choose between using it to deceive people by stealing payments, or using it to make new coins. He must find it more profitable than playing by the rules, as the rules that benefit him with new coins more than anyone else put together, rather than damaging the system and the validity of his wealth[9].

**XI. DISK SPACE**

Once the last transaction in a currency is buried under sufficient blocks, the spent transactions can be disposed of before disk space is made available. To facilitate this without breaking the fragmentation of the cluster, transactions are fragmented in the Merkle Tree[5][6], with only the root included in the fragmentation of the cluster. The blocks can then be compressed by removing tree branches. Retail not required. A block header without transactions will be about 80 bytes. Assuming blocks are created every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems usually sold with 2 GB of RAM starting in 2008, and Moore's Law predicts current growth of 1.2 GB per year, storage shouldn't pose a problem even if block heads should be kept in memory[9].

**XII. SIMPLIFIED PAYMENT VERIFICATION**

It is possible to verify payment without running a full network node. A user only needs to save a copy of the block header from the oldest proof work chain, which can be obtained by asking the network node until he is sure it has the longest chain, and getting the Merkle branch that links the transaction to the timestamped block. In. He cannot check the transaction for himself, but by linking it somewhere in the chain, he can see that the network node has received it, and a block is added after further confirming the network has received it.

**XIII. PRIVACY**

The traditional banking model creates a level of privacy by restricting access to information for interested parties and a trusted third party.[4] All transactions must not be publicly announced, but privacy remains unaffected. It is maintained by breaking the flow of information elsewhere, by keeping the public keys secret. The public can see that someone is sending money to another person, but without information linking the transaction to anyone. This is similar to the level of information published by exchanges, where the time and size of individual deals, the "tape", are announced, but without knowing who the parties are.[4]

**XIV. CONCLUSION**

We have recommended an electronic transaction system without relying on trust. We started with the usual digital signature coin framework, which provides strong, but incomplete ownership control without a way to prevent double disbursement. To overcome this problem, we recommend that peer-to-peer networks use proof-of-work to record the general history of transactions that are mathematically impractical for an attacker to change if an honest node controls most of the CPU’s power. This network is strong in its unstructured simplicity. The node works at once with a little coordination. It does not need to be determined, because the message is not
routed to a specific location and only needs to be delivered based on best effort. The node can go and return to the network as desired, and accept the chain of evidence as a clue about what happened during his departure. They strongly choose their CPU, declaring their acceptance of a valid block by trying to expand it and rejecting the invalid block by refusing to work on it. Any necessary rules and incentives can be applied to this compliance mechanism.

XV. REFERENCES

[1]. https://masteringbitcoin.neocities.org/


[3].https://www.investopedia.com/bitcoin


[9]. Bitcoin:A Peer-to-Peer Electronic Cash system”- Bitcoin.org, Satoshi Nakamoto