Security in Blockchain

Yusuf AKSOY

Abstract—**Blockchain applications are popular, and they can be a solution to the problems in different areas. There are different challenges exist in the blockchain such as creating identities, providing anonymity, proof-of-work, digital signatures, and building attack tolerant systems. This paper examines the fundamental concepts of blockchain to maintain security and privacy with Bitcoin as a use case.**

Keywords—blockchain, security, privacy, Bitcoin

I. What Is Blockchain And Its Applications

Blockchain is simply transforming a digital property considering safety and security and everyone informs what is changed [4].
It can be considered as a public secure ledger contains transaction history and secured by cryptographic techniques. What distinguishes blockchain from widely used databases over the years is that it has the tamper-evident and decentralized characteristics. In blockchain, data is publicly available for the users and any user can verify the correctness of the data. Also, it is decentralized therefore there is no need to trust third-party authorities to maintain blockchain. There are many applications [6] for blockchain, here are the most commons:

A. Financial Applications

Financial applications can use to keep records of financial transactions without the need for third-party authority. Different cryptocurrencies exist and one of the most successful applications of the blockchain is Bitcoin. Although cryptocurrencies and blockchain concepts are not new, after publishing “Bitcoin: A Peer-to-Peer Electronic Cash System” paper in 2008 by Satoshi Nakamoto, cryptocurrency concept has become popular and attracted a lot of attention. It uses an electronic payment system based on cryptographic algorithms instead of third-party authorities in the commerce [1]. Therefore, cryptocurrencies can be considered as decentralized. With this technology, merchant and customer can directly trade based on cryptocurrency.

B. Smart Contracts

Smart contracts are publicly available and verifiable codes that committed to the blockchain ledger. There is no need to rely on a lawyer when using a smart contract because it provides an agreement between users. Smart contracts can be triggered by an event and code executes [7]. They have certain conditions or configurations, and users can involve the system while protecting their anonymity. For example, contribution to a venture and distribution of profits is possible by using smart contracts. In Ethereum smart contract codes execute on the Ethereum Virtual Machine (EVM). Codes can be written by using different languages such as Solidity or Vyper [10].

C. Internet of Things(IoT)

The nature of immutability can be adapted to IoT devices to communicate securely and detect malicious actions [6].

Cryptographic techniques have a key role in the cryptocurrency concept because they are the basis of mining, tamper-evident transactions, verification of data records in the blocks and data integrity.

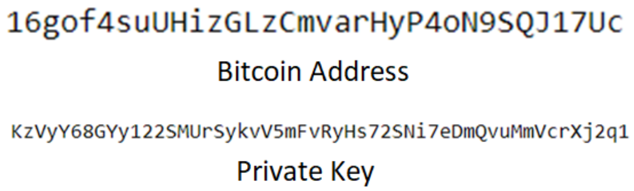
II. Identites

Most of the cryptocurrencies use asymmetric cryptography to identify users. A user has Bitcoin address which can be used to represent public key and a private key which grants user’s ownership for a given address. The first private key needs to be generated and then the public key can be derived by using it. Retrieving private key from the public key is not possible [2].

Public keys provide anonymity to users and they look like random numbers. A user can have many different public keys, therefore there can be many Bitcoin addresses of an individual user. There is no centralized identity manager, hence identity generation is not restricted [8].

Bitcoin addresses can be shared by other users in the network, while private keys should only be known by the owner of the address. Both Bitcoin addresses and private keys can be used for encryption and decryption processes. **Figure 1** shows an example of a generated Bitcoin address-private key set for Bitcoin. In Bitcoin, private keys are 256 bits and public keys are also 256 bits. Bitcoin addresses are shorter versions of the public keys and they are 160 bits.

In Bitcoin the public key can be generated by using the Elliptic Curve Digital Signature Algorithm (ECDSA). Specifically, “secp256k1” standard elliptic curve is using for Bitcoin. 128-bit security is provided for Bitcoin [8][5].



16gof4suUHizGLzCmvarHyP4oN9SQJ17Uc

Bitcoin Address

KzVyY68GYy122SMUrSykvV5mFvRyHs72SNi7eDmQvuMmVcrXj2q1

Private Key

Figure 1: Example of a Bitcoin address-private key set

III. Hash Fuctions

Hash functions can take any size of data as an input and convert them to fixed-size output data. The output will always be the same size without depending on the input data size.

Also, hash functions have an avalanche effect, it means even a small change causes a completely different result. For example, **Figure 2** shows SHA256 hashing of two given words, even though they have a difference of letter, their results after the hashing are completely different.

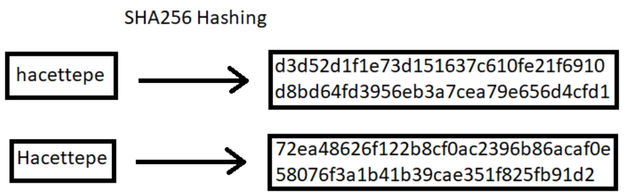

SHA256 Hashing
hacettepe
d3d52d1f1e73d151637c610fe21f6910
d8bd64fd3956eb3a7cea79e656d4cfd1
Hacettepe
72ea48626f122b8cf0ac2396b86acaf0e
58076f3a1b41b39cae351f825fb91d2

Figure 2: Hashing of two words with using SHA256 hashing algorithm

Avalanche effect in hash functions makes difficult to tamper data by an attacker. An attacker may try to change input data without changing hash, but hash functions are collision resistant algorithms. It can be considered computationally infeasible to attack and find a collision in this way because its possibility is too small [8].

Because of the properties of hash function, they are appropriate to provide security in blockchain applications. Blockchain word comes from blocks are constructed to build a chain manner. It is like a linked-list data structure, each block points another block. Each block header has a hash pointer that points the previous block, and it is possible to traverse through the blocks by using these hash pointers [1][8]. **Figure 3** shows traversing the blocks by using the previous hash pointer in block header.

The beginning of the block is called “genesis block” and it is possible to reach it from the most recent block by traversing in the chain. This property avoids changing the hash pointer because an attacker should change the whole blockchain. It is not possible because users can verify the integrity of the blockchain [5].

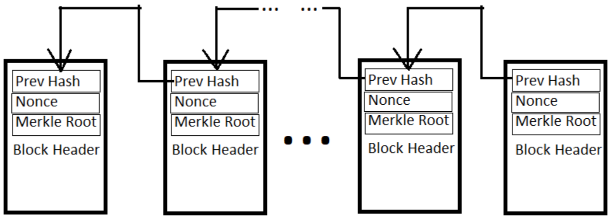


Figure 3: Using hash pointers enables to traverse the blocks and avoids tampering a block.

There is another important concept that blockchain using is called Merkle Tree. It is a binary tree in which nodes are linked together with hash pointers. With using Merkle Root, we can traverse down to nodes. Security of the Merkle Tree relies on the same concept with traversing blocks by using hash pointers. If an attacker tampers a node at the bottom of the tree, then the parent node’s hash will not match with it anymore. Therefore, if a node’s hash is changed this propagates to the parent nodes and at the end hash of the root changes. Any attempt to tamper data blocks can be detected in this way [8].

Merkle Tree is using to prove a node’s membership to the corresponding tree. Using the Merkle Tree concept brings logarithmic complexity to reach a node and ignoring unrelated leaves of the tree [8]. This decreases the verification time of the data.

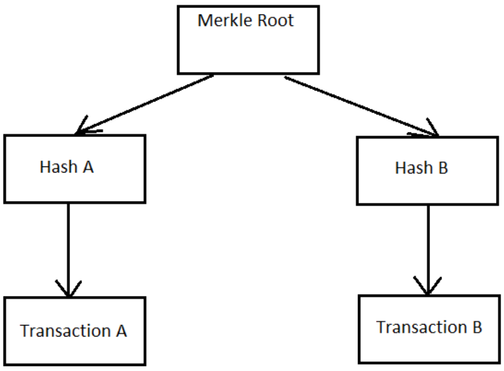


Figure 4: Transactions can be verified by using Merkle Tree concept in Bitcoin [1].

Merkle root is included in the block’s header as **Figure 3** shows. **Figure 4** shows transactions and they can be verified by using the Merkle Tree concept.

IV. Digital Signature

To use digital signature concept in blockchain, a user should be able to verify the signed data, so that data in the block can be considered valid. A signature should only be used by the owner of the signature, which means it shouldn’t be generated by anyone else.

Another consideration is that a signature should be used only for what it was signed for. It shouldn’t be used to sign other data. The data can be signed using a private key and verified by the public key. The digital signature algorithm in Bitcoin is ECDSA and it is collision-resistant algorithm [5].

V. Decentralization

Decentralization is an important concept in Bitcoin. Because all transactions should be agreed by users instead of a centralized authority. Decentralization is distributed in Bitcoin, users broadcast their transactions to the network publicly and agree on single order history of the transactions. Simply all nodes receive transactions and verify the transaction with using concept proof-of-work. Whenever a node finds proof-of-work, it broadcast to the network. If the proof-of-work is correct nodes maintain chain of blocks by creating a new block on top of it [1]. Each node can verify these processes.

To maintain the Bitcoin, there should be a consensus between users, it means the majority of the users should be agreed. The system itself should be resistant to any malicious activities because there is no central authority to maintain the system.

VI. Most Common Attacks

There are different attacks on the blockchain, for example, 51 Percent, Double Spending, and Selfish Mining are the most common attack models.

In 51 Percent Attack, if most of the computing power (51 percent or more) in the network is malicious then control of the chain can be taken. This may lead to making possible other attacks such as double-spending [9]. Even if 51 Percent Attack has succeeded and attackers attempted to spend invalid coins, honest nodes would not accept the invalid transactions and there will be a fork. Fork means the chain is separated at that moment and users are not agreed on the chain. **Figure 5** shows an example of the fork after 51 Percent Attack. 51 Percent Attack causes to lose confidence over the system [8].

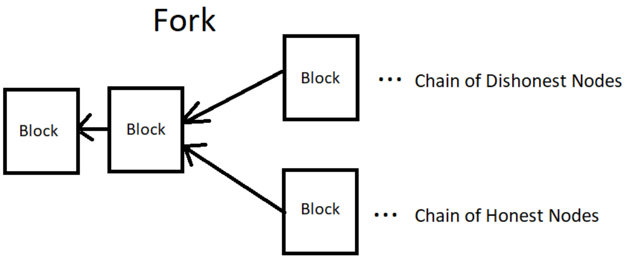


Figure 5: Fork in the chain after 51 Percent Attack

Double spending attack is spending the same coins twice. A person buys something from a merchant and makes a transaction to the merchant. But if the customer makes another transaction using the same coin causes double - spending. In the network, transactions not always included in the chain in order, because nodes tend to extend the longest valid branch. There is a possibility that only double-spending transaction attempts included in the longest chain. In this case, the customer's first transaction to the merchant becomes "orphan block" and double-spending succeeds. To avoid this merchant may wait for a certain time to ensure there is no double-spending attempt and receives coins. The probability of a successful double-spending attempt decreases after several block confirmations [8].

There are different techniques to prevent double-spending attack, for example, in a proposed method customer creates a deposit and be penalized by loss of deposit if there is a double-spending attack [9].

There is another type of attack called selfish mining. Selfish mining occurs when a miner founds two blocks and ahead of the chain by two blocks. If miner keeps secret these blocks and broadcast to the network all in once, the longest chain will be the selfish miner's block. This causes the waste of computing power of the network [8][9].

VII. Proof-Of-Work

Proof-of-work depends on a cryptographic puzzle that uses SHA-256. Bitcoin's proof-of-work proposes if the majority of the computing power is honest, then the growth of the blocks will be fast, and an attacker needs to modify all the work and surpass the honest nodes. This is type of work is difficult for the attacker [1].

When concatenating nonce, previous hash pointer and transactions and then taking the hash of them should satisfy a certain target value. A miner tries nonce values one by one and checks whether if the resulted target is satisfied with the target condition. Following inequality should be satisfied [8][9]:

H(nonce || previous hash || concatenation of transactions in the block) < target

The hash contains zero bits at the beginning and these bits determine the difficulty of the proof-of-work process. Difficulty changes according to varying interest of the nodes and hardware speed in the network. The difficulty increases if the proof-of-work problem becomes easy for nodes and vice versa [1].

References

[1] S. Nakomoto, Bitcoin: A Peer-to-Peer Electronic Cash System, Available: https://bitcoin.org/bitcoin.pdf *(access time 02.01.2020)*

[2] https://support.blockchain.com/hc/en-us/articles/360000951966-Public-and-private-keys *(access time 02.01.2020)*

[3] M. Conti, E. Sandeep Kumar, C. Lal, S. Ruj, A Survey on Security and Privacy Issues of Bitcoin, 2018

[4] https://www.coindesk.com/learn/blockchain-101/what-is-blockchain-technology *(access time 02.01.2020)*

[5] R. Zhang, R. Xue, L. Liu, Security and Privacy on Blockchain, ACM Computing Surveys, Vol. 1, No. 1, Article 1, January 2019 (Available: https://arxiv.org/pdf/1903.07602.pdf)

[6] Q. E. Abbas, J. Sung-Bong, A Survey of Blockchain and Its Applications, ICAIIC 2019

[7] https://blockgeeks.com/guides/smart-contracts/ (Access time 04.01.2020)

[8] A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, Bitcoin and Cryptocurrnecy Technologies (Pages: 40-42, 24-25, 34-35, 58-61, 64, 71, 72, 161), Draft, Feb 9, 2016

[9] T. T. Huynh, T. D. Nguyen, H. Tan, A Survey on Security and Privacy Issues of Blockchain Technology, 2019

[10] https://ethereum.org/developers/#getting-started *(access time 02.01.2020)*