Blockchain

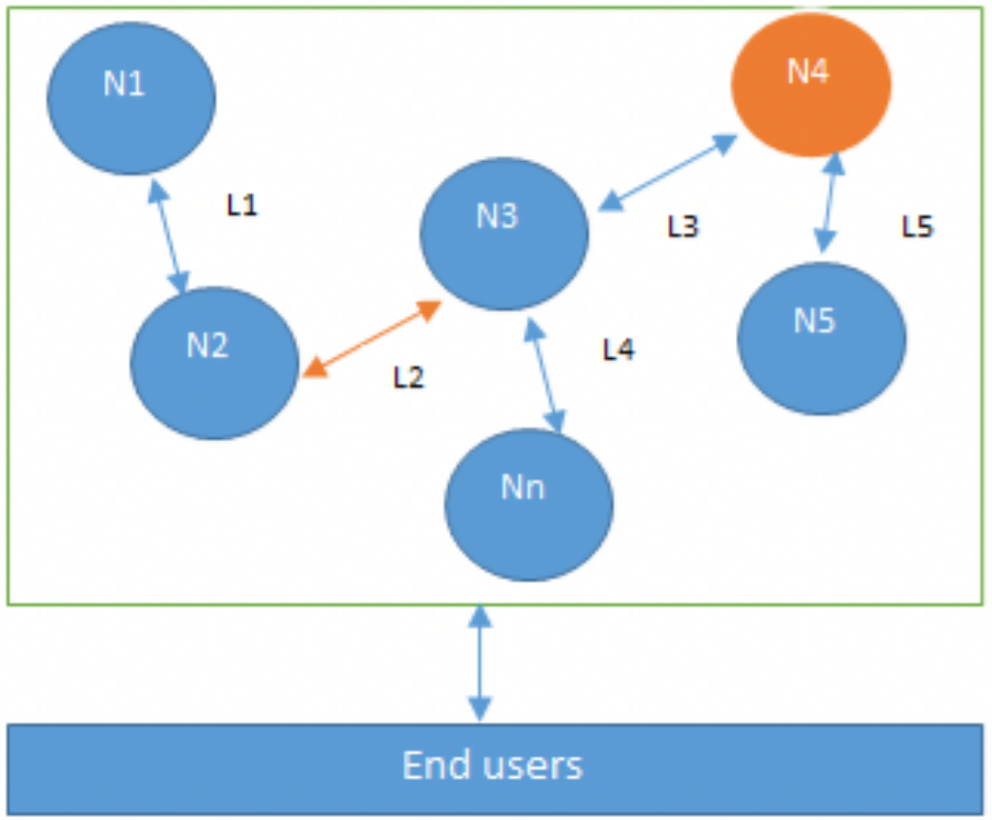
-a brief overview

Pilkyu Jung

SOOMPAY KOREA

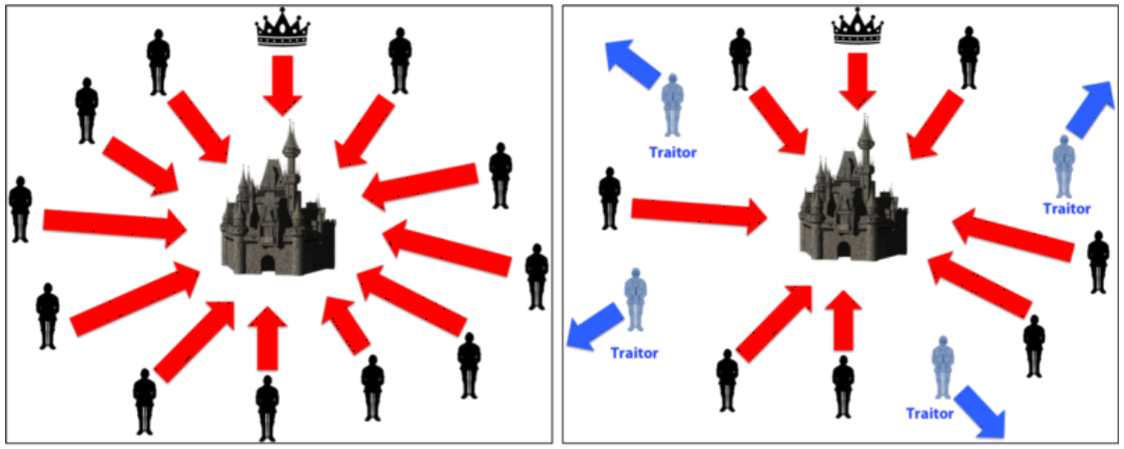
Index

Intro: Distributed systems



- A computing paradigm. Two or more nodes work with each other in a coordinated fashion in order to achieve a common outcome. End users see it as a single logical platform.

Intro: Byzantine Generals problem



Coordinated Attack Leading to Victory Uncoordinated Attack Leading to Defeat

- As an analogy with distributed systems, generals can be considered as nodes, traitors can be considered Byzantine (malicious) nodes, and the messenger can be thought of as a channel of communication between the generals.

- This problem was solved in 1999 by Castro and Liskov who presented the Practical Byzantine Fault Tolerance (PBFT) algorithm. Later on in 2009, the first practical implementation was made with the invention of bitcoin where the Proof of Work (PoW) algorithm was developed as a mechanism to achieve consensus.

Intro: Consensus

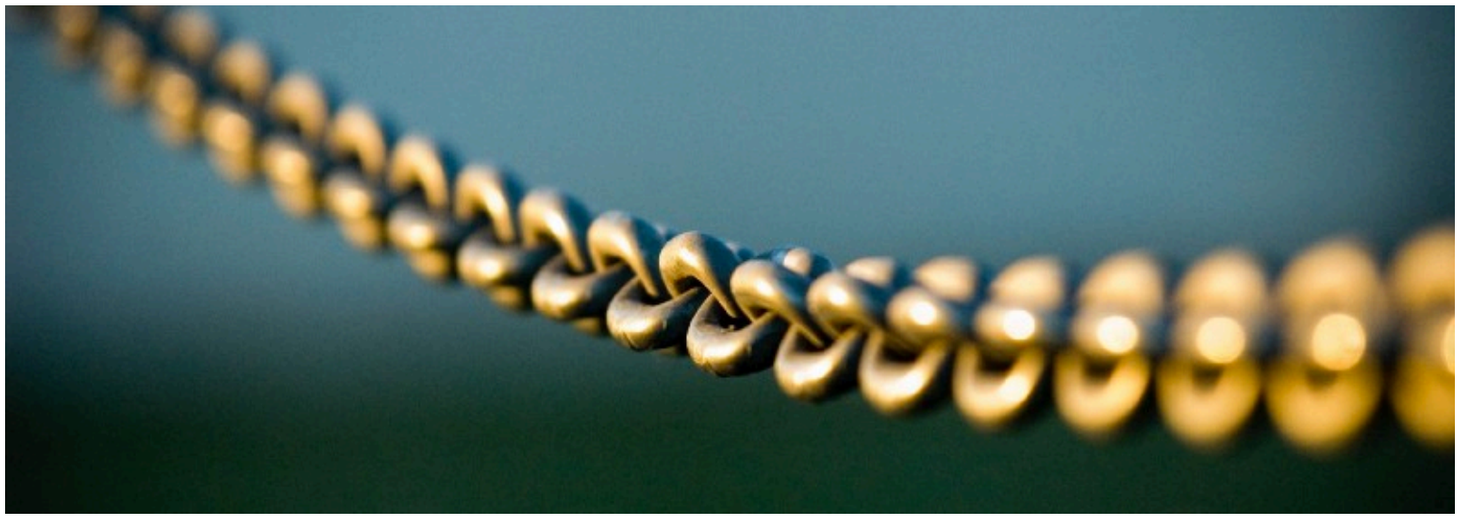
- Consensus is a process of agreement between distrusting nodes on a final state of data. It is easy to reach an agreement between two nodes (for example in client-server systems) but when multiple nodes are participating in a distributed system and they need to agree on a single value it becomes very difficult to achieve consensus. This concept of achieving consensus between multiple nodes is known as distributed consensus.

Intro: Consensus mechanisms

- There are various requirements which must be met in order to provide the desired results in a consensus mechanism. The following are their requirements with brief descriptions:

1) **Agreement:** All honest nodes decide on the same value.
2) **Termination:** All honest nodes terminate execution of the consensus process and eventually reach a decision.
3) **Validity:** The value agreed upon by all honest nodes must be the same as the initial value proposed by at least one honest node.
4) **Fault tolerant:** The consensus algorithm should be able to run in the presence of faulty or malicious nodes (Byzantine nodes).
5) **Integrity:** This is a requirement where by no node makes the decision more than once. The nodes make decisions only once in a single consensus cycle.

Intro: So, What is Blockchain?



- Blockchain at its core is a peer-to-peer distributed ledger that is cryptographically secure, append-only, immutable (extremely hard to change), and updatable only via consensus or agreement among peers.

- From a business point of view a blockchain can be defined as a platform whereby peers can exchange values using transactions without the need for a central trusted arbitrator.

SOOMPAY KOREA

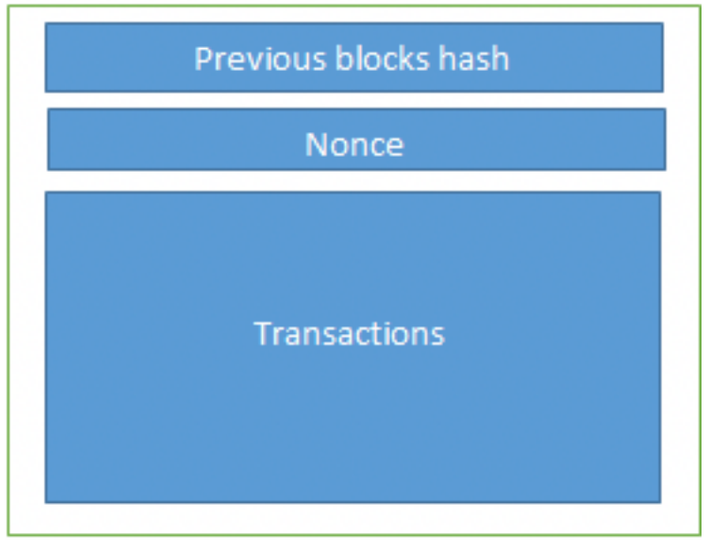
Intro: Various technical definitions of blockchains

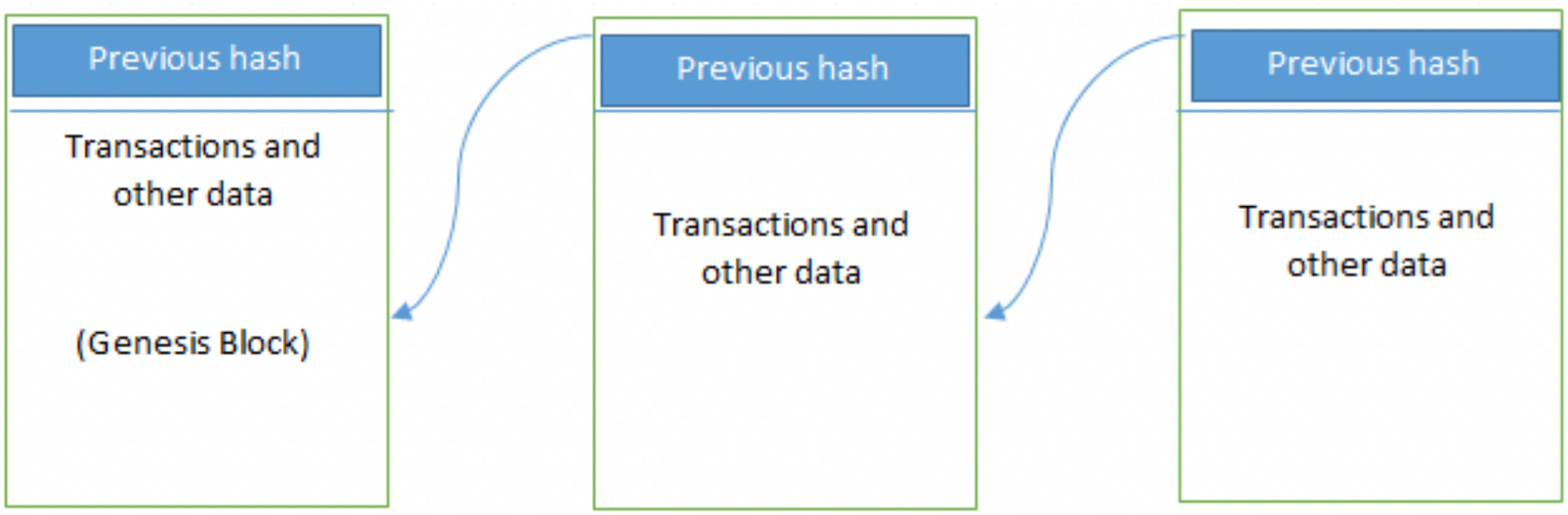
- Blockchain is a decentralized consensus mechanism. In a blockchain, all peers eventually come to an agreement regarding the state of a transaction.

- Blockchain is a distributed shared ledger. Blockchain can be considered a shared ledger of transactions. The transaction are ordered and grouped into blocks.

- Blockchain is a data structure; it is basically a linked list that uses hash pointers instead of normal pointers. Hash pointers are used to point to the previous block.

Intro: Generic structure of a blockchain





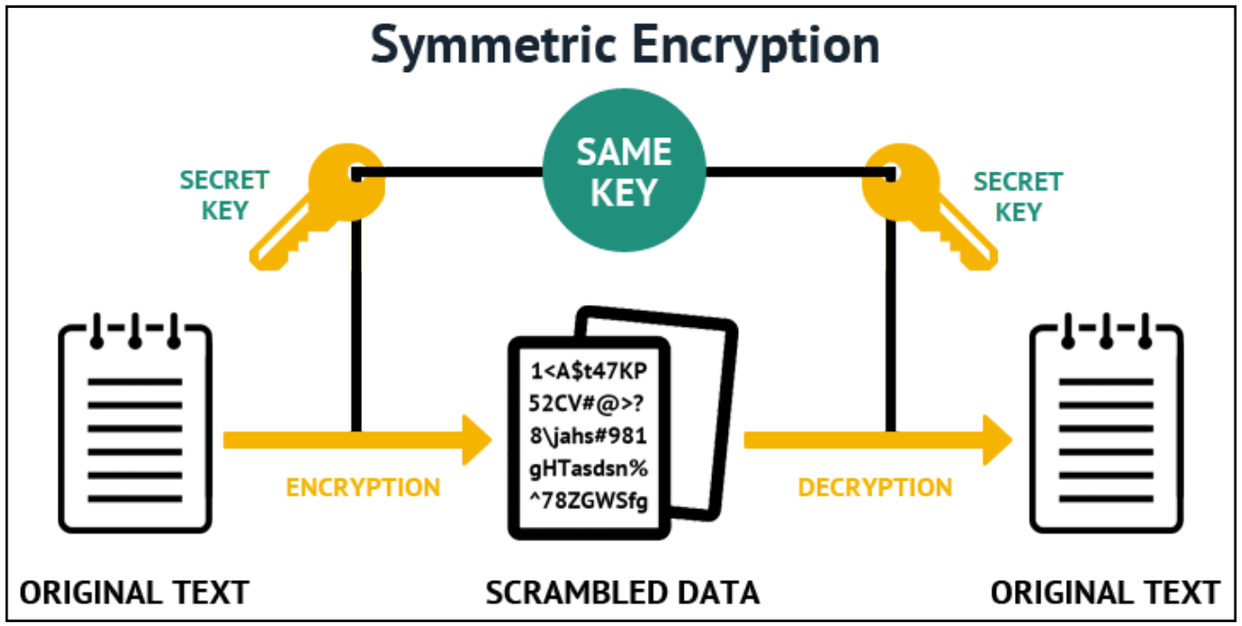
SOOMPAY KOREA

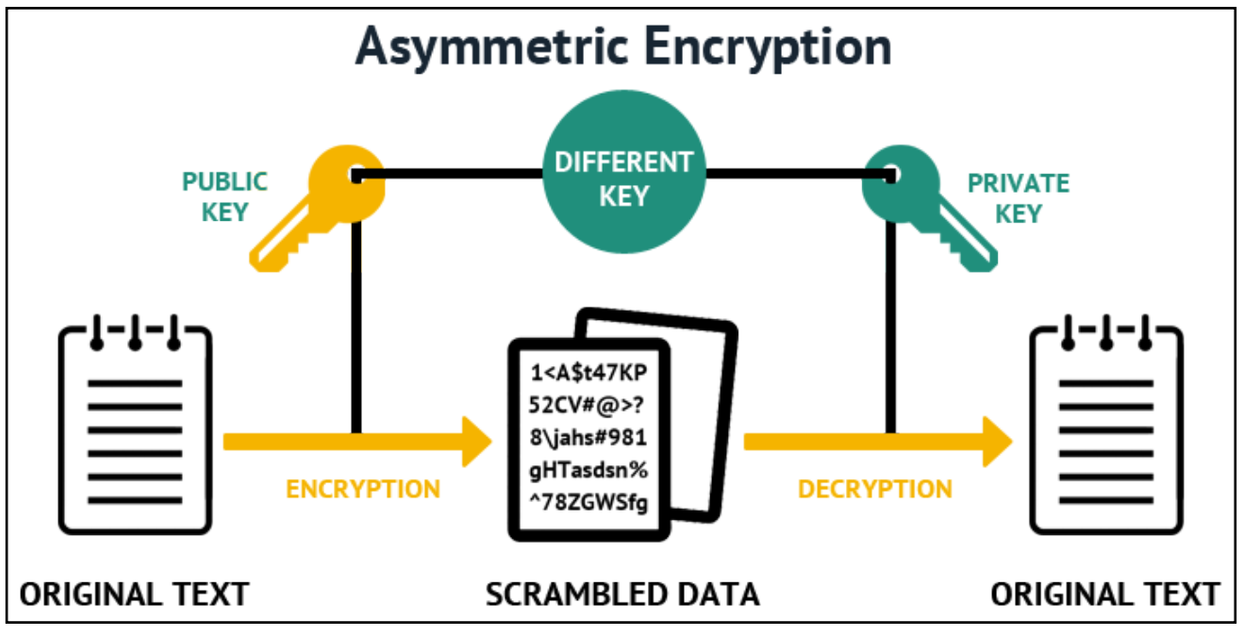
Intro: Cryptographic hash function

- It is like a blackbox that takes a string as input (“*SoomPayKorea*”) and returns a fixed size arbitrary string (“*b519254fd37fd4e07efe7bec6fe4202f198da389142a81e894e4bb41431d830a*”). To be usable cryptographically it must have certain properties:

1) Same input always returns same output
2) It’s quick to compute
3) You can’t reverse engineer the “*b519254fd37fd4e07efe7bec6fe4202f198da389142a81e894e4bb41431d830a*” that comes from “*SoomPayKorea*” without brute force (trial and error)
4) A small change in the input will change the output a lot
5) It’s unfeasible that two inputs generate the same output

Intro: Symmetric vs Asymmetric





Intro: Nodes

- A node in a blockchain network performs various functions depending on the role it takes. A node can propose and validate transactions and perform mining to facilitate consensus and secure the blockchain. This is done by following a consensus protocol. (Most commonly this is PoW.) Nodes can also perform other functions such as simple payment verification (lightweight nodes), validators, and many others functions depending on the type of the blockchain used and the role assigned to the node.

Intro: Full Nodes

- A full client, or “full node,” is a client that stores the entire history of bitcoin transactions (every transaction by every user, ever), manages users’ wallets, and can initiate transactions directly on the bitcoin network. A full node handles all aspects of the protocol and can independently validate the entire blockchain and any transaction. A full-node client consumes substantial computer resources (e.g., more than 125 GB of disk, 2 GB of RAM) but offers complete autonomy and independent transaction verification.

Intro: Lightweight Nodes

- A lightweight client, also known as a simple-payment-verification (SPV) client, connects to bitcoin full nodes for access to the bitcoin transaction information, but stores the user wallet locally and independently creates, validates, and transmits transactions. Lightweight clients interact directly with the bitcoin network, without an intermediary.

Intro: Third-party API Nodes

- A third-party API client is one that interacts with bitcoin through a third-party system of application programming interfaces (APIs), rather than by connecting to the bitcoin network directly. The wallet may be stored by the user or by third-party servers, but all transactions go through a third party.

Intro: Addresses

- Addresses are unique identifiers that are used in a transaction on the blockchain to denote senders and recipients. An address is usually a public key or derived from a public key.

- While addresses can be reused by the same user, addresses themselves are unique. In practice, however, a single user may not use the same address again and generate a new one for each transaction. This newly generated address will be unique.

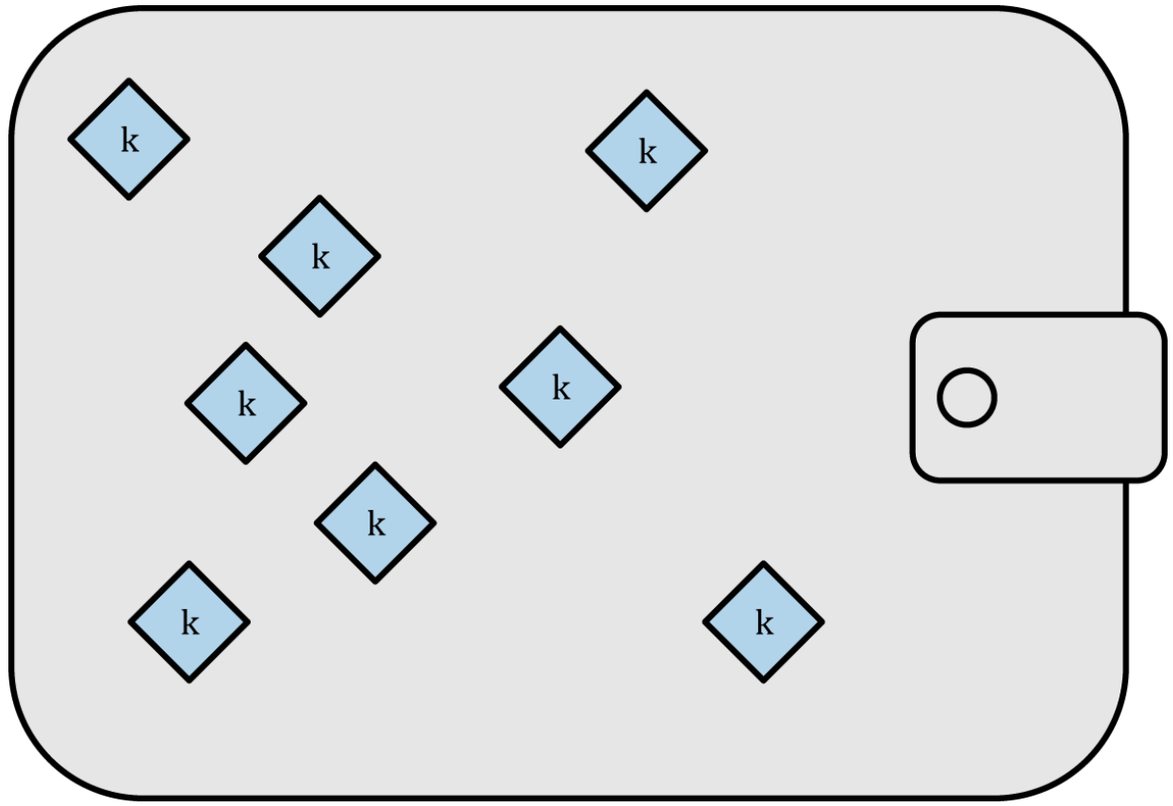
- Users generate a new address for each transaction in order to avoid linking transactions to the common owner, thus avoiding identification.

Intro: Wallet

- A wallet is an application that serves as the primary user interface. The wallet controls access to a user's money, managing keys and addresses, tracking the balance, and creating and signing transactions.

- A common misconception about bitcoin is that bitcoin wallets contain bitcoin. In fact, the wallet contains only keys. The "coins" are recorded in the blockchain on the bitcoin network. Users control the coins on the network by signing transactions with the keys in their wallets. In a sense, a bitcoin wallet is a keychain.

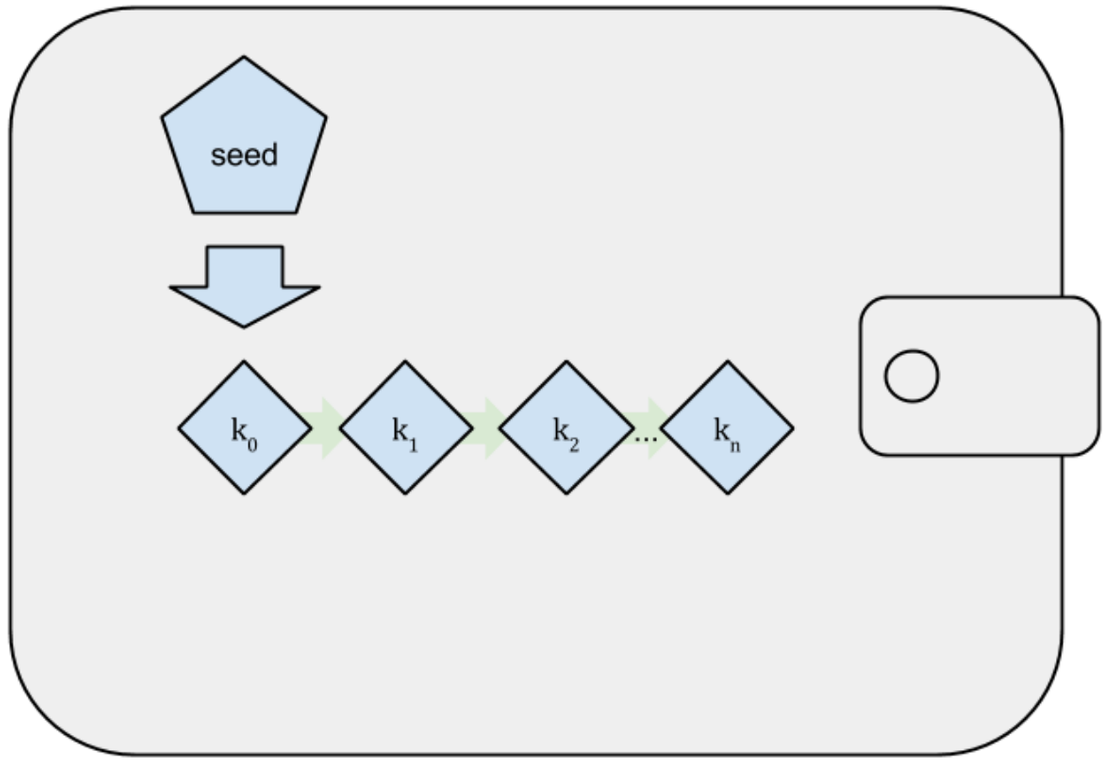
Intro: Nondeterministic Wallet



- A nondeterministic wallet, where each key is independently generated from a random number. The keys are not related to each other. This type of wallet is also known as a JBOK wallet from the phrase “Just a Bunch Of Keys.”

SOOMPAY KOREA

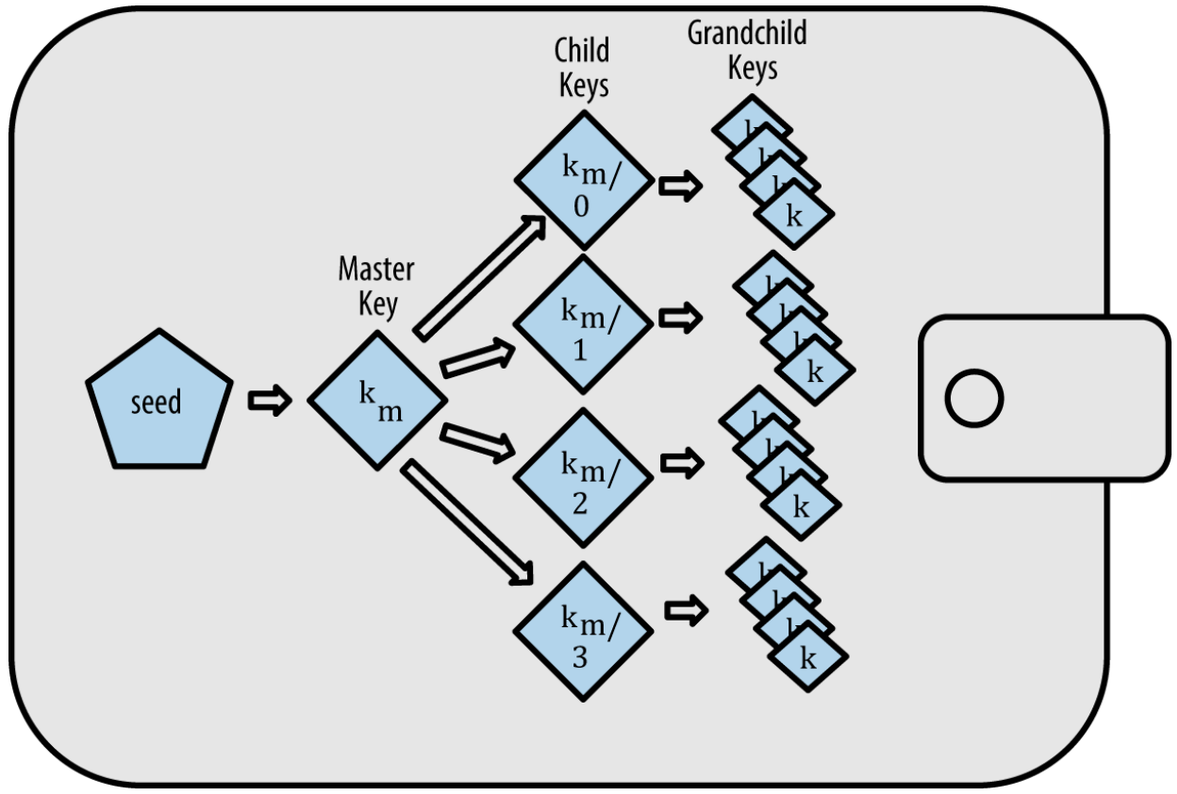
Intro: Deterministic(Seeded) Wallet



- A deterministic wallet, where all the keys are derived from a single master key, known as the seed. All the keys in this type of wallet are related to each other and can be generated again if one has the original seed. Deterministic wallets are initialized from a seed. To make these easier to use, seeds are encoded as English words, also known as mnemonic code words.

SOOMPAY KOREA

Intro: HD Wallets (BIP-32/BIP-44)



- The most advanced form of deterministic wallets is the HD wallet defined by the BIP-32 standard. HD wallets contain keys derived in a tree structure, such that a parent key can derive a sequence of children keys, each of which can derive a sequence of grandchildren keys, and so on, to an infinite depth.

Intro: Wallet

- Desktop Wallet

  A desktop wallet was the first type of bitcoin wallet created as a reference implementation and many users run desktop wallets for the features, autonomy, and control they offer.

- Mobile Wallet

  A mobile wallet is the most common type of bitcoin wallet. Running on smartphone operating systems such as Apple iOS and Android, these wallets are often a great choice for new users. Many are designed for simplicity and ease-of-use, but there are also fully featured mobile wallets for power users.

Intro: Wallet

- Web wallet

  Web wallets are accessed through a web browser and store the user's wallet on a server owned by a third party. This is similar to webmail in that it relies entirely on a third-party server. Some of these services operate using client-side code running in the user's browser, which keeps control of the bitcoin keys in the hands of the user.

Intro: Wallet

- Hardware wallet

  Hardware wallets are devices that operate a secure self-contained bitcoin wallet on special-purpose hardware. They are operated via USB with a desktop web browser or via near-field-communication (NFC) on a mobile device.

- Paper wallet

  The keys controlling bitcoin can also be printed for long-term storage. Paper wallets offer a low-tech but highly secure means of storing bitcoin long term. Offline storage is also often referred to as cold storage.

Intro: Then, Who made Blockchain?

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

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, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of

- In 2008 the paper was written on the topic of peer-to-peer electronic cash and introduced the term chain of blocks. This term over the years has now evolved into the word blockchain.

Intro: Bitcoin consists of

- A decentralized peer-to-peer network (the bitcoin protocol)
- A public transaction ledger (the blockchain)
- A set of rules for independent transaction validation and currency issuance (consensus rules)
- A mechanism for reaching global decentralized consensus on the valid blockchain (Proof-of-Work algorithm)

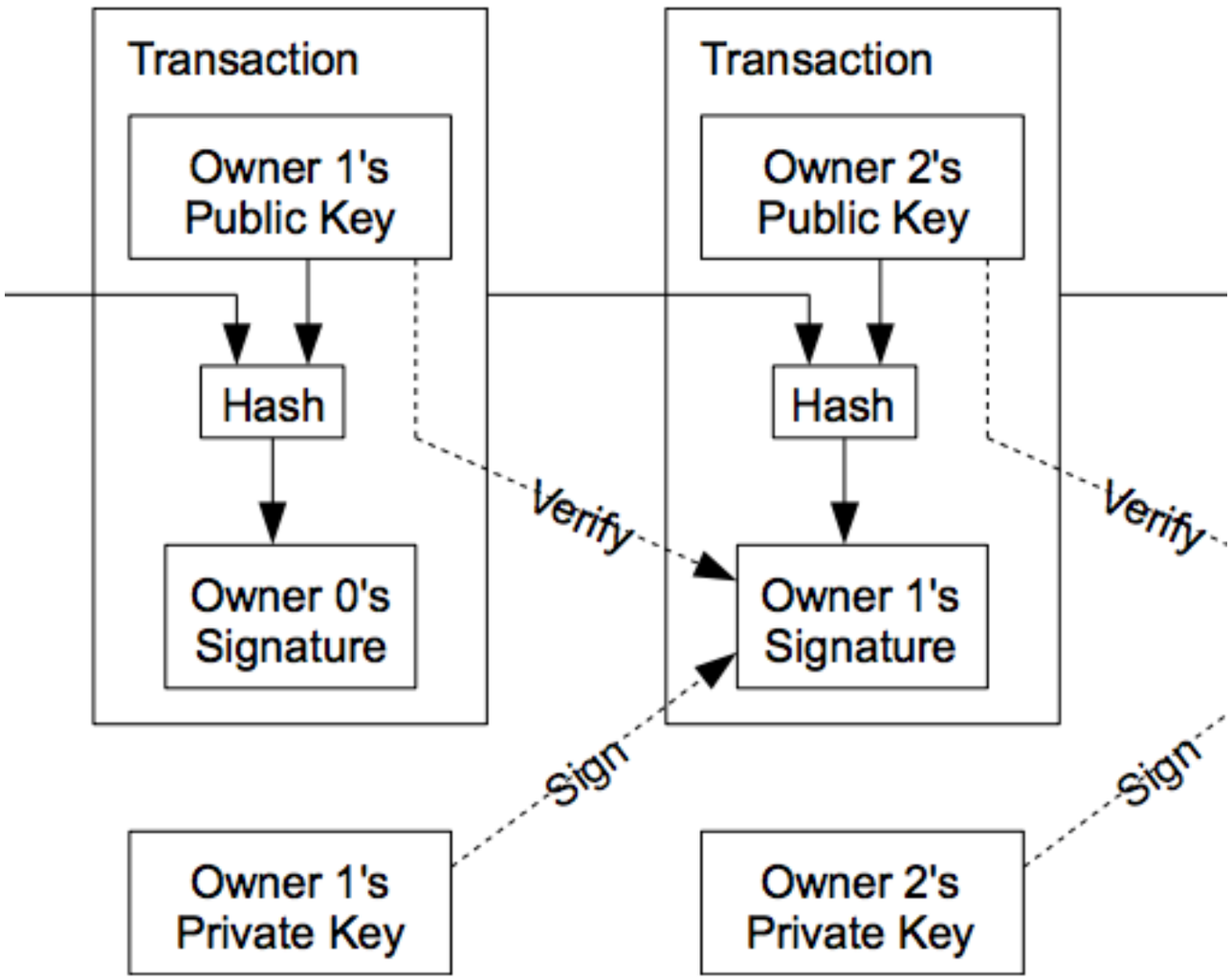
Transactions

- Transactions can be as simple as just sending some bitcoins to a bitcoin address.

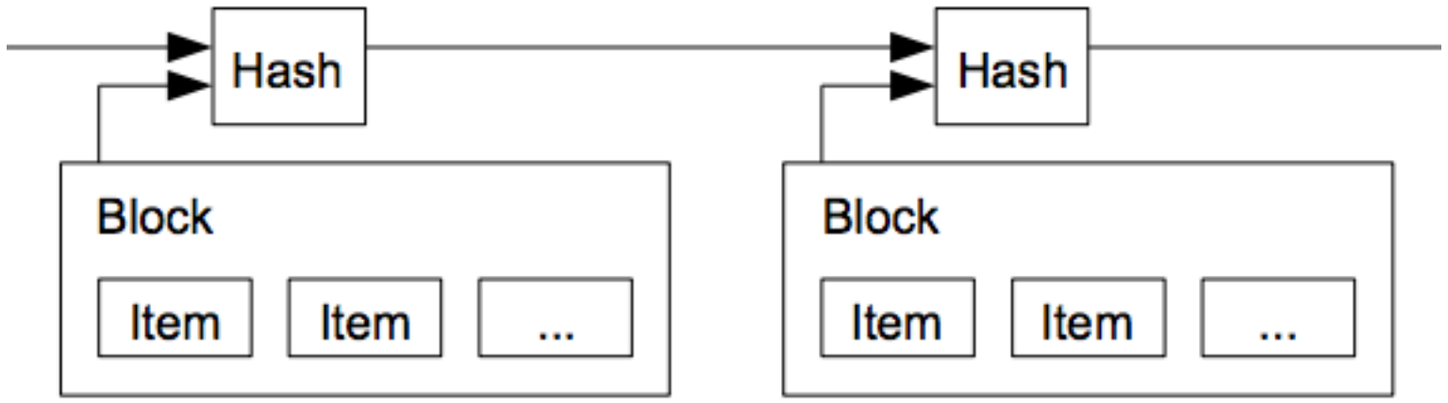
- Each transaction is composed of at least one input and output. Inputs can be thought of as coins being spent that have been created in a previous transaction and outputs as coins being created.

- If a transaction is minting new coins, then there is no input and therefore no signature is needed. If a transaction is to sends coins to some other user (a bitcoin address), then it needs to be signed by the sender with their private key and a reference is also required to the previous transaction in order to show the origin of the coins.

Transactions



Double spending & Timestamp Server



- Someone could add two transactions that consume from the same output unless we have a single chain which someone checks.

- A timestamp determines when an event occurred by using a sequence of characters. If we introduce this value in the hash generation of the transactions/blocks we make sure the data existed at that moment. There is no way to generate that hash with the same data for a different timestamp. That way no one can pretend something happened in a different order maliciously.

SOOMPAY KOREA

Proof of Work(POW)

- Transactions are broadcast to the entire network. At this point, they are not yet added to the chain. Miners (those that are going to perform the “work” to add the block to the chain) are going to perform hashing. They collect these transactions and put them in a block (as Merkle Root) together with the aforementioned timestamp, previous block hash, and some other relevant data like block height (what block # in the chain), and more. Having collected all this data in a block, they run it through the SHA256 hashing algorithm.

- What this basically does is it converts all that data into a string of characters that uniquely identifies that block and its data.

Proof of Work(POW)

- Change a tiny thing in the block's data and the entire hash changes (there is no known pattern for this but it is not random; change it back and you will get exactly the same hash).

- *SoomPayKorea -> b519254fd37fd4e07efe7bec6fe4202f198da389142a81e894e4bb41431d830a*

- *SoomPayKorea1 -> e5db60ee864f9be4517dde95a8634a0da45bcb0b4e9ab1e335e3321573ab9441*

Proof of Work(POW)

- The Bitcoin blockchain does not demand just a hash; it wants a hash that starts with (at the moment of writing) seventeen 0's. For example: *000000000000000006fb217d70740a895ce4966e2826325e31061bc433d7b186*

- How do miners get that hash? They need to add a number to the block's data which is called a 'nonce'. Nobody knows what number is needed to find the correct hash. The only way to find it is through trial-and-error: guessing.

- This process, guessing the right nonce, is what is referred to as 'mining' ⛏.

Proof of Work(POW)

- The miners with the largest CPU resources (most computational power) have the highest chance of being the first to find that correct nonce.

- As long as more than 51% of the CPU power is in the hands of honest nodes, it will be impossible for a malicious miner to consistently win the mining process and add false data to the chain. The longest chain is always the chain that is taken as the truthful chain.

- This process of adding a new block to the blockchain happens every 10 minutes or so. This is kept stable by the protocol adjusting the mining difficulty (# of starting 0's) accordingly as computational power grows over time.

Mining ⛏️

- Mining is a resource-intensive process by which new blocks are added to the blockchain. Blocks contain transactions that are validated via the mining process by mining nodes and are added to the blockchain. This process is resource-intensive in order to ensure that the required resources have been spent by miners in order for a block to be accepted. New coins are minted by the miners by spending the required computing resources. This also secures the system against frauds and double spending attacks while adding more virtual currency to the bitcoin ecosystem.

Mining ⛏️

- Miners are rewarded with new coins if and when they create new blocks and are paid transaction fees in return of including transactions in their blocks.

- Approximately 144 blocks, that is, 1,728 bitcoins are generated per day. The number of actual coins can vary per day; however, the number of blocks remains at 144 per day. Bitcoin supply is also limited and in 2140, almost 21 million bitcoins will be finally created and no new bitcoins can be created after that. Bitcoin miners, however, will still be able to profit from the ecosystem by charging transaction fees.

Incentive 💰

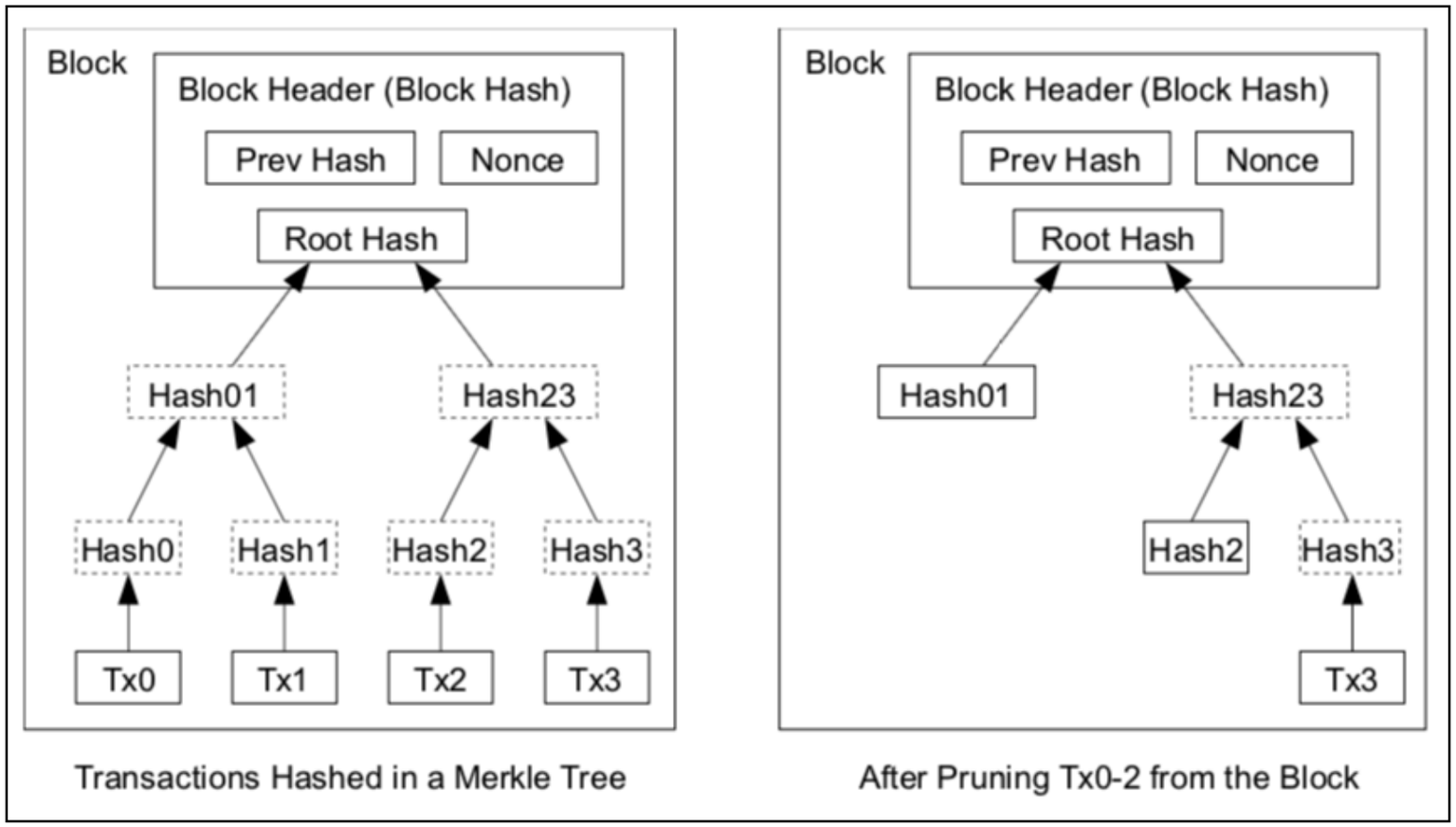
- Why would miners go through all that effort and pay a lot of money to obtain the computational power to mine?

- Once the block is agreed upon, an extra transaction is added to the beginning of the block (often referred to as the ‘coinbase transaction’) which allocates newly created BTC to the winning miner’s wallet address, rewarding them for the work put in and providing a way to distribute coins into circulation. On top of that, each transaction in the block has a small transaction fee associated with it which also goes to the winning miner.

Reclaiming Disk Space

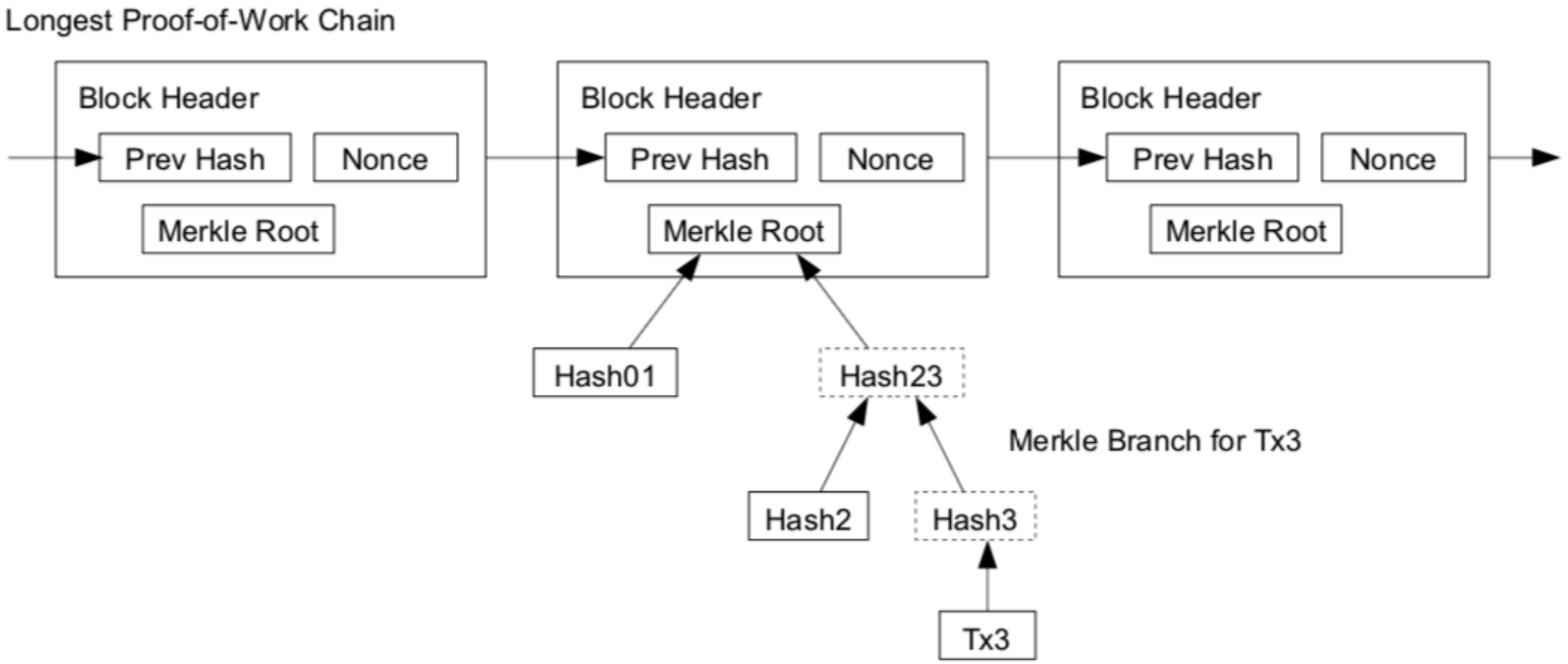
- Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree, with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.

Reclaiming Disk Space



- A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory.

Simplified Payment Verification



- In order to verify a payment, a user only needs to be able to link the transaction to a place in the chain by querying the longest chain of blocks and pulling the Merkle branch in which the transaction exists. If that user can do so, they can trust that the transaction has been valid given that the network has included it and further blocks have been build on it.

The end

- 감사합니다

References

- Satoshi Nakamoto, 2008, Bitcoin: A Peer-to-Peer Electronic Cash System
- Andreas Antonopoulos, 2017, Mastering Bitcoin (2nd Edition) – Programming the open blockchain
- Imran Bashir, 2017 , Mastering Blockchain (1st Edition)
- S Nakamoto, ,Bitcoin White Paper Made Simple - Blockchain Review
- https://medium.freecodecamp.org/satoshi-nakamotos-bitcoin-whitepaper-a-walk-through-3e9e1dee71ce
- https://medium.com/coinmonks/bitcoin-white-paper-explained-part-1-4-16cba783146a
- https://medium.com/coinmonks/bitcoin-white-paper-explained-part-2-4-d79fbc5e2adf
- https://medium.com/coinmonks/bitcoin-white-paper-explained-part-3-3-c06c1791a31b