Goal of this lecture note

- Network
- Blockchain Scalability
- Block Header
- Consensus
- Payment and Change
- Privacy

1 Network

- Network
  - The steps to run the network are as follows

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

1 Network

- Network

> "Nodes always consider the longest chain to be the correct one and will keep working on extending it.
> If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first."

1 Network

- Network

> “In that case, they work on the first one they received, but save the other branch in case it becomes longer.
> The tie will be broken when the next proof-of work is found and one branch becomes longer, the nodes that were working on the other branch will then switch to the longer one.”

1 Network

- Network

> “New transaction broadcasts do not necessarily need to reach all nodes.
> As long as they reach many nodes, they will get into a block before long.
> Block broadcasts are also tolerant of dropped messages.
> If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.”

1 Network

- Network
  - Are there any guarantee for transactions to be included into blocks?
  - With a large incentive(tx fee), a tx can be put on high priority.
  - But if the production rate of txs is higher than the service rate, then there must be some transactions not to end up in the blockchain.

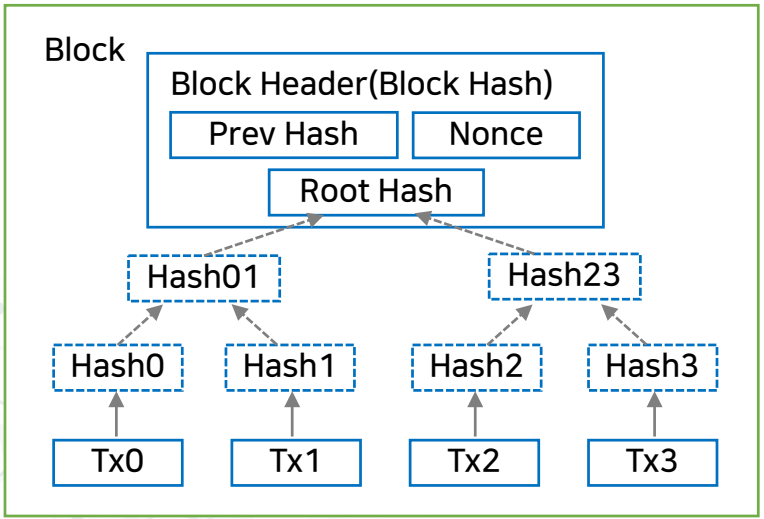
2 Blockchain Scalability

- Reclaiming Disk Space

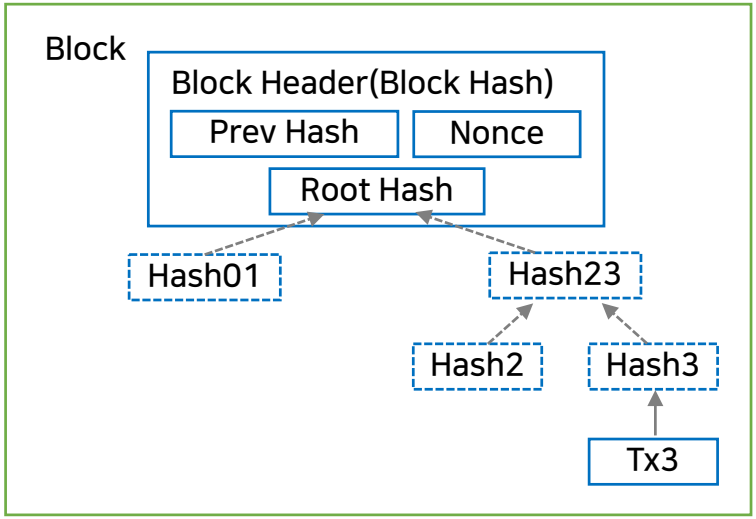
"Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree[7][2][5], with only the root included in the block's hash.
Old blocks can then be compacted by stubbing off branches of the tree.
The interior hashes do not need to be stored."

2 Blockchain Scalability

- Blockchain Scalability
  - Reclaiming Disk Space



Transaction Hashed in a Merkle Tree



After Pruning Tx0-2 from Block

2 Blockchain Scalability

- 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 is the block headers must be kept in memory."

2 Blockchain Scalability

- Blockchain Scalability
  - Use Merkle tree and save disk space.
  - Save the blockhash in the header.
  - Those tree branches recording past transactions are erased but the hash values are kept.
  - 80 byte Blockheader

(1) Prev hash
: 256 bit = 2^8 = $2^{5*(2\text{^}3)}$ = 2^5 Bytes = 32 Bytes
(2) Roothash = 32 Bytes
(3) Nonce = 4 Bytes = 32 bit
(4) Time
(5) Difficulty
(6) version

3 Block Header

- 80 Byte Block Header

Bytes	Name	Data Type	Description
4	version	int32_t	Indicates which set of block validation rules to follow.
32	previous block header hash	char[32]	A SHA256(SHA256()) hash in internal byte order of the previous block's header. This ensures no previous block can be changed without also changing this block's header.
32	merkle roothash	char[32]	A SHA256(SHA256()) hash in internal byte order. The merkle root is derived from the hashes of all transactions included in this block, ensuring that none of those transactions can be modified without modifying the header.

출처:: https://bitcoin.org/en/developer-reference#block-headers

3 Block Header

- 80 Byte Block Header

Bytes	Name	Data Type	Description
4	time	uint32_t	The block time is a Unix epoch time when the miner started hashing the header (according to the miner). Must be strictly greater than the median time of the previous 11 blocks. Full nodes will not accept blocks with headers more than two hours in the future according to their clock.
4	nBits	uint32_t	An encoded version of the target threshold this block's header hash must be less than or equal to. See the nBits format described below.
4	nonce	uint32_t	An arbitrary number miners change to modify the header hash in order to produce a hash less than or equal to the target threshold. If all 32-bit values are tested, the time can be updated or the coinbase transaction can be changed and the merkle root updated.

출처:: https://bitcoin.org/en/developer-reference#block-headers

4 Consensus

- It is a way to resolve a conflict.
- Longest chain is trusted
  - Simplified Payment Verification

> "It is possible to verify payments running a full network node.
> A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in."

4 Consensus

- Longest chain is trusted
  - Simplified Payment Verification

> "He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it."

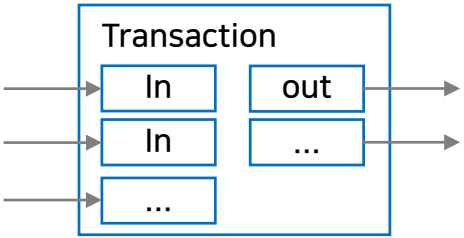
5 Payment and Change

- Payment and changes
  - Combining and Splitting Value

> "Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender."

5 Payment and Change

- Payment and changes
  - Combining and Splitting Value



How to get the change?

6 Privacy

- Privacy, by Anonymous Pub Key
  - Privacy

> "The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes the method, but privacy can still be maintained by breaking the flow of information in another place by keeping public keys anonymous."

6 Privacy

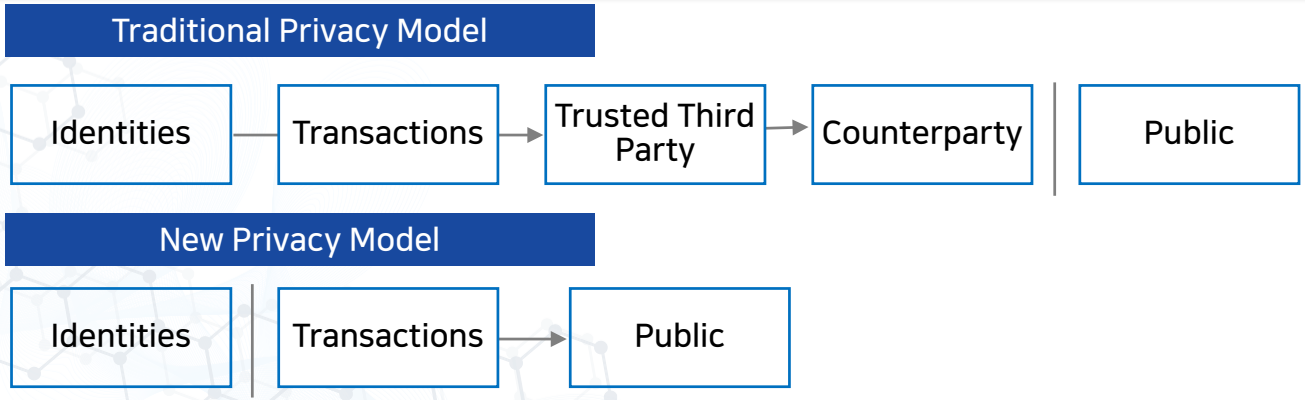
- Privacy, by Anonymous Pub Key
  - Privacy

> "The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone."

6 Privacy

- Privacy, by Anonymous Pub Key
  - Privacy

> "This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the 'tape', is made public, but without telling who the parties were."



6 Privacy

- Privacy, by Anonymous Pub Key
  - Privacy

> "As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner.
> Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner.
> The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner."

6 Privacy

- Privacy, by Anonymous Pub Key
  - Blockchain is published.
  - Privacy is maintained
    by keeping public key anonymous!
  - Additional privacy
    by using new public key per transaction!