# Goal of this lecture note

- Transactions
- Time-Stamp Server
- Estonian Blockchain
- Proof-of-Work

# 1 Transactions

- Bitcoin Transactions
  - Bitcoin is a chain of signatures.

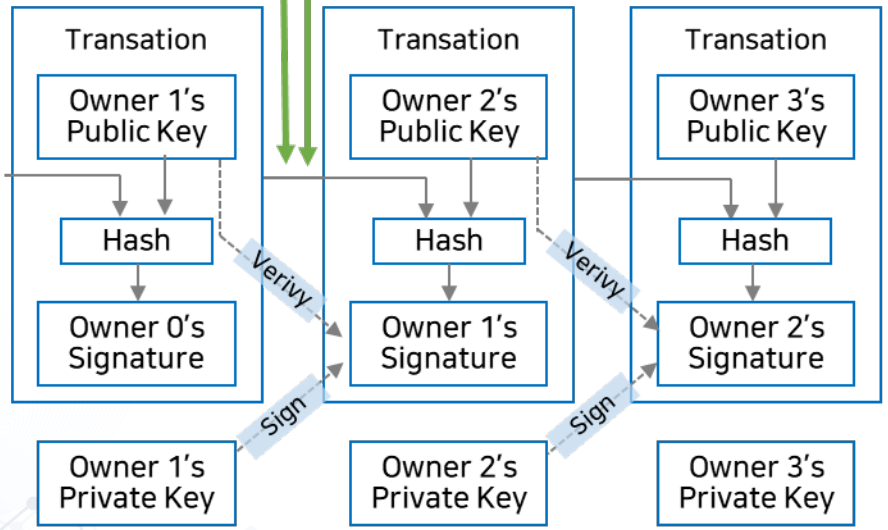
> "We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership."

  - In the sequel, quoted sentences in box are from Satoshi.

# 1 Transactions

- Bitcoin Transactions
  - Bitcoin is a chain of signatures.

2nd Tx box
1. The first TX shows that O1 owns the coin
2. O1 can transfer it to anybody, say O2
3. O1 writes TX 2
4. O1 asks Owner2, the new owner, for his public key
5. O1 hashes the received public key and TX1, and calculates the hash value




6. To show his ownership status, O1 signs the hash value and leaves the signature in TX2
7. Now, anybody can verify O1's signature with O1's pubic key written in TX 1

Once TX 2 recorded and published, anybody can easily see TX 2 and knows that O1 has transferred his coin ownership to O2

# 1 Transactions

- Double Spending Problem

"The problem of course is the payee can't verify that one of owners did not double-spend the coin.
A common solution is to introduce a trusted central authority, or mint that checks every transaction for double spending.
After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent.
The problem with the solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank."

# 1 Transactions

- Double Spending Problem

> "We need a way for the payee to know the previous owners did not sign earlier transactions.
> For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend.
> The only way to confirm the absence of a transaction is to be aware of all transactions.
> In the mint based model, the mint was aware of all transactions and decided which arrived first."

# 1 Transactions

- Double Spending Problem

> "To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received.
> The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received."

[1] W.Dai, "b-money", http://weidai.com/bmoney.txt, 1988.

# 2 Timestamp Server

- Timestamp Server
  - The solution Bitcoin propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5].

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements, "In 20th Symposium on Information Theory in the Benelux, May 19993
[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document, " In Journal of Cryptology, vol 3, no 2, pages 00-111, 1991.
[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," in Sequences II : Methods in Communication, Security and Computer Science, pages 329-334, 1993.
[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In Proceedings of the 4th ACM Conference on Computer and Communications Security, pages 28-35, April 1997.

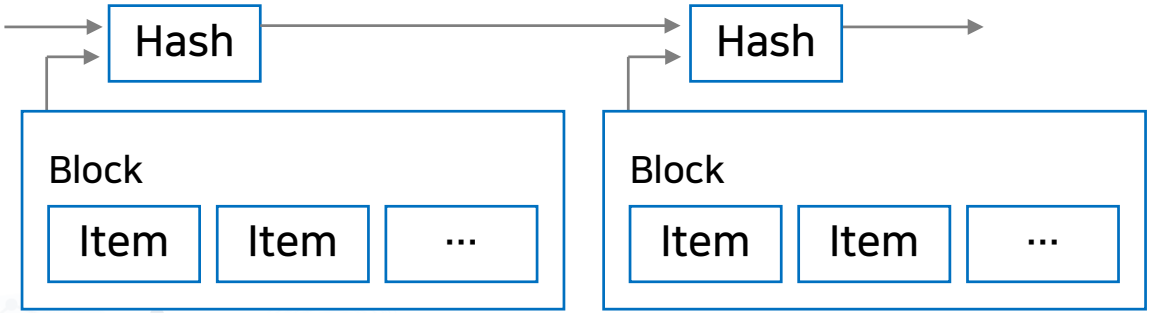
## 2 Timestamp Server

- Timestamp Server

> "The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash.
> Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it."

# 2 Timestamp Server

- Timestamp Server



## 2 Timestamp Server

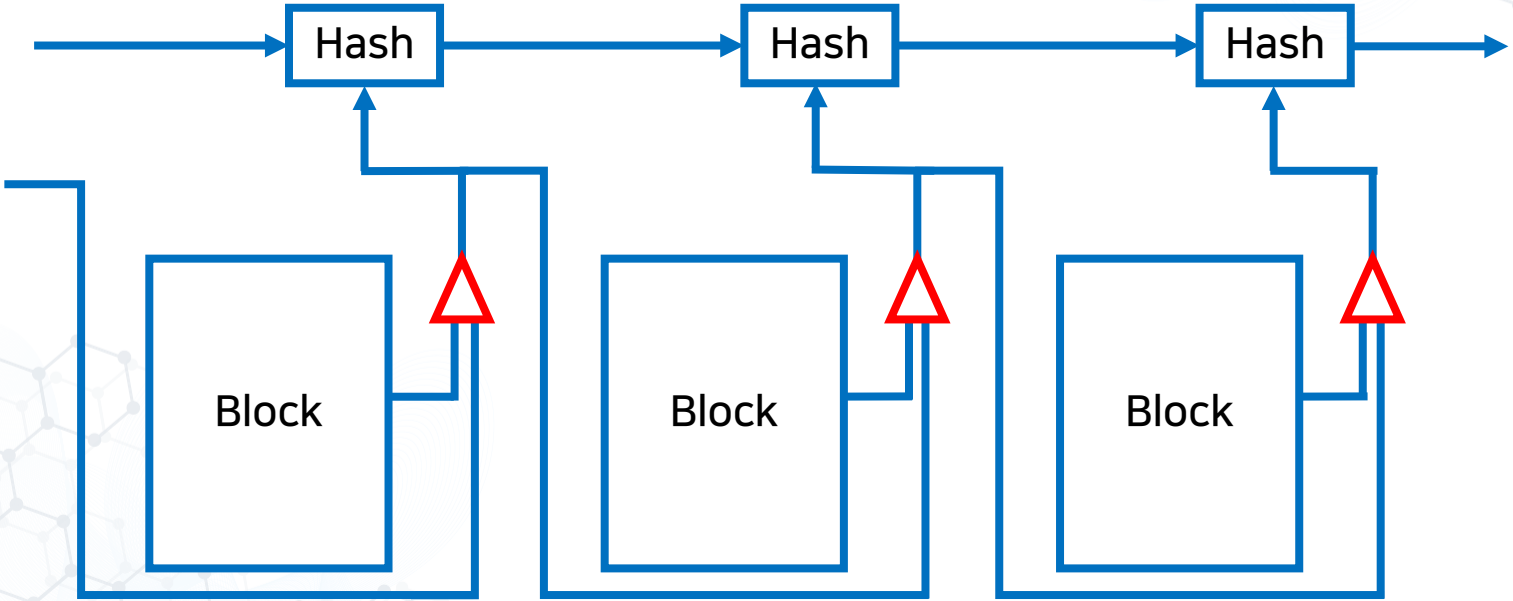
- Timestamp Server
  - If a timestamp server indicates the existence of hash value at a certain time point, then a legitimate ledger can indeed be made?
    - If hash values only are published while no block contents are published, there will be no issue of scalability, and privacy can be kept since no one other than the parties involved in the transactions can see the content of transactions!
    - But how can one verify for coin ownership and double spending transactions?

## 2 Timestamp Server

- Timestamp Server
  - The problem is to decide who should run the timestamp server?
  - If a government runs it, it becomes a private Blockchain (while social terms it is a public chain)!
  - What possible problems are there if it is run by government?

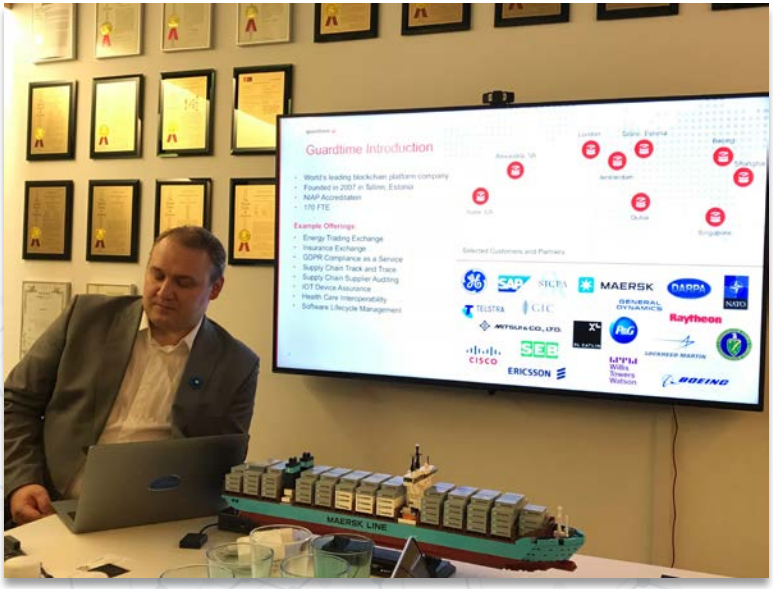
## 2 Timestamp Server

- Using an off-chain timestamp server



# 3 Estonian Blockchain

- Estonian Blockchain




전명산, '비트코인 이전에 에스토니아에 블록체인이 있었다.' 공공분야 블록체인 현장 르포_ #7: 에스토니아 정부와 가드타임의 도전. 2018.12.06 (https://www.coindeskkorea.com/33836/)

# 3 Estonian Blockchain

- Guardtime publishes its hashes regularly.


MARKETS, 2008. 09. 18. / https://guardtime.com/blog/6-reasons-why-encryption-isnt-working

# 3 Estonian Blockchain

- Estonian Blockchain is KSI.
  - "A blockchain is a distributed public record of events; an append-only record of events where each new event is cryptographically linked to the previous. New entries are created using a distributed consensus protocol.
  - This blockchain overcomes two major weaknesses of traditional blockchains, making it usable at industrial scale:" (Guardtime)

출처: guardtime federal, https://www.guardtime-federal.com/ksi/

# 3 Estonian Blockchain

## Scalability

- One of the most significant challenges with traditional blockchain approaches is scalability – they scale at O(n) complexity i.e. they grow linearly with the number of transactions.
- In contrast the E. blockchain scales at O(t) complexity – it grows linearly with time and independently from the number of transactions.

출처: guardtime federal, https://www.guardtime-federal.com/ksi/

# 3 Estonian Blockchain

## Settlement time

- In contrast to the widely distributed crypto-currency approach, the number of participants in KSI blockchain distributed consensus protocol is limited.
- By limiting the number of participants it becomes possible to achieve consensus synchronously, eliminating the need for Proof of Work and ensuring settlement can occur within one second.

출처: guardtime federal, https://www.guardtime-federal.com/ksi/

# 3 Estonian Blockchain

## Data Privacy

- KSI does not ingest any customer data; data never leaves the customer premises.
- Instead the system is based on one-way cryptographic hash functions that result in hash values uniquely representing the data, but are irreversible such that one cannot start with the hash value and reconstruct the data – data privacy is guaranteed at all times.

출처: guardtime federal, https://www.guardtime-federal.com/ksi/

# 4 Proof-of-Work

- Proof-of-Work

> "To implement a distributed timestamp server on a peer-to peer basis, Bitcoin uses a proof-of work system similar to Adam Back's Hashcash[6], rather than newspaper or Usenet posts.
> The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits."

[6] A. Back, "Hashcash – a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002.

# 4 Proof-of-Work

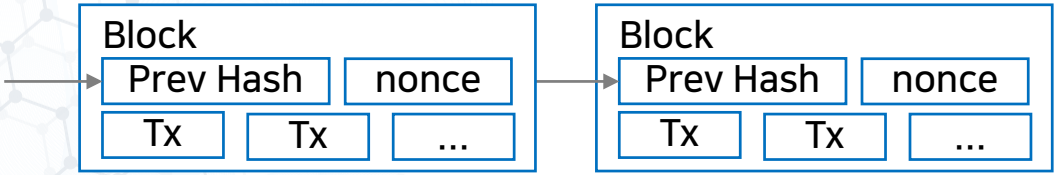
- Proof-of-Work

> "The average work required is exponential in the number of zero bits required and can be verified by executing a single hash."
> "For the timestamp network, Bitcoin implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits."

# 4 Proof-of-Work

- Proof-of-Work

> "Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it."



# 4 Proof-of-Work

- Proof-of-Work

> “The proof-of work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote.”

# 4 Proof-of-Work

- Proof-of-Work

> "The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains."

# 4 Proof-of-Work

- Proof-of-Work

> “To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes.”
> “We will show that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.”

# 4 Proof-of-Work

- Proof-of-Work

> “To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of work difficulty is determined by a moving average targeting an average number of blocks per hour.
> If they’re generated too fast, the difficulty increases.”

# 4 Proof-of-Work

- Proof-of-Work
  - Aim to make a timestamp Server in a P2P network

Why?

- Not to rely on any central authority.
- Central authority such as banks and states.
- Within a nation, the state government may run the timestamp server.
- But for trades overseas, P2P across different nations is needed.

# 4 Proof-of-Work

- Proof-of-Work
  - Solution?
    - Distributed timestamp P2P server network.
    - Distributed, thus, it is difficult to maintain the integrity of data.
    - To keep the integrity of data, PoW system is proposed!