ICDS Spring 2025

Cybersecurity

Introduction to Cryptography and Blockchain

Logistics

- **About Quiz 2 grade (6%)**
  - Recitation 003 & 005: Tuesday (April 29) 3-5 PM @730 (Zhaonan)
  - Recitation 004 & 006: Wednesday (April 30) 3:45-5 PM @744 (Sven & Jennifer)

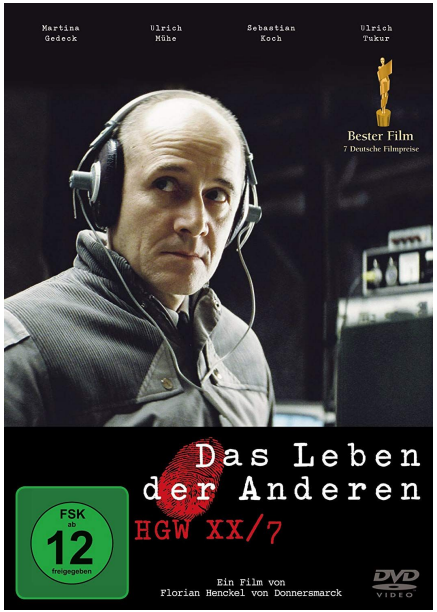
- **About Final Project (15%)**
  - **Group project! 2-3 students per team**
    - Registration done by April 28
  - **Start early!**
    - Read guideline; Divide and conquer!
    - Sample start code to be released after UP3 due
    - Submission (code & video) deadline: **May 17!**

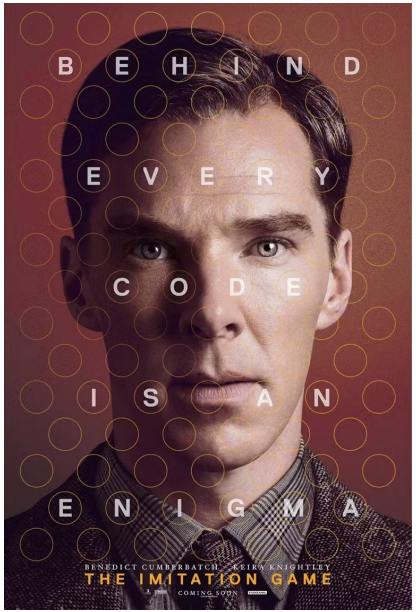
Agenda

- Symmetric encryption
  - Caesar cipher and the Enigma machine
- Asymmetric encryption (Public key cryptography)
  - Diffie-Hellman key exchange protocol
  - RSA
- Block chain
- Appendix: Bitcoin

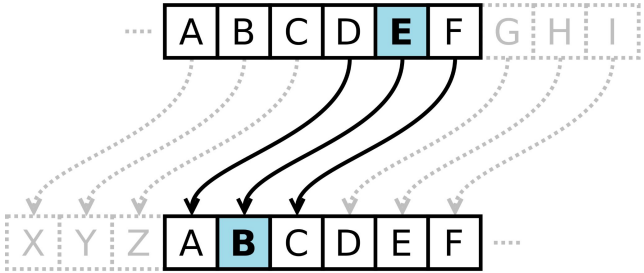
Cryptography?

- Definition: the study of techniques for secure communication (hiding/coding info.) in the presence of adversarial behavior



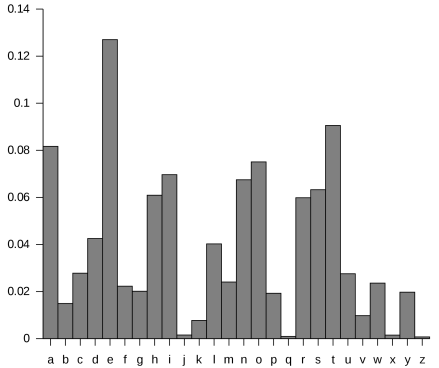


Caesar cipher (shift cipher; substitution cipher)



- Each letter in the plaintext is replaced by a letter with a fixed number of positions down the alphabet.

- It is easy to break a Caesar cipher.
  - Brute-force attack: We can try out all shifts and look for readable text.
  - Statistical attack: We can compare the frequency of the letters used with the frequency of the language itself to find the map between them.

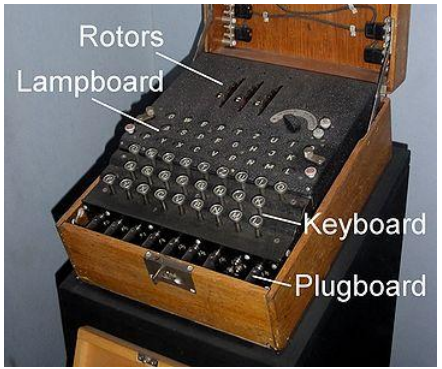


E and T are most frequent in English.

The Enigma Machine

- Polyalphabetic substitution cipher: Use a different alphabet for every character in the message to defend statistical attacks.

The Enigma Machine



- Three rotors scramble the 26 alphabet.
- One person inputs the plaintext in the keyboard, and the bulbs on the lampboard are lighten at each key press, which shows the ciphertext. (Entering ciphertext transforms it back into readable plaintext.)
- The rotor changes the electrical connections between the keys and the bulbs with each keypress.
- 3 rotors have 26^3 combinations and each machine has a unique connection to 26 keys (So, **>150 trillion patterns** in total); no way to break it by statistical attack!

Symmetric cryptography

- It uses the **same** key to both encrypt and decrypt the original message.

- Both parties need a shared symmetric key to set up a secure channel.
  - In Caesar cipher, both sides need to know the shift first.
  - Each Enigma machine has a key-book.

The key-distribution problem

- In symmetric cryptography, the two parties need to know the key before the communication.
- One needs to distribute the key to the other. ⇒ But if one does not meet the other party beforehand, how can they secure the key distribution?

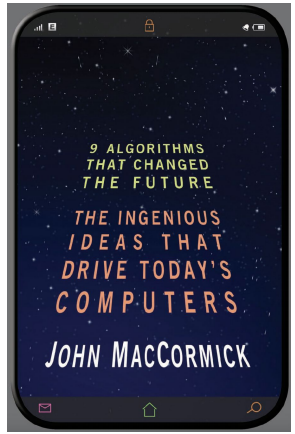
- In the era of the Internet, we often need to communicate with strangers, (e.g., sending your password to a bank server you never logged in before.) ⇒ Symmetric encryption is not a good choice for internet communication due to the key-distribution problem. (We need asymmetric cryptography.)

Asymmetric cryptography

Also known as public key cryptography,

- Using pairs of keys
  - public keys: being disseminated widely (anyone can have it)
  - private keys: only known to the owner
  - The data encrypted with a public key can only be decrypted by its corresponding private key, and vice versa.

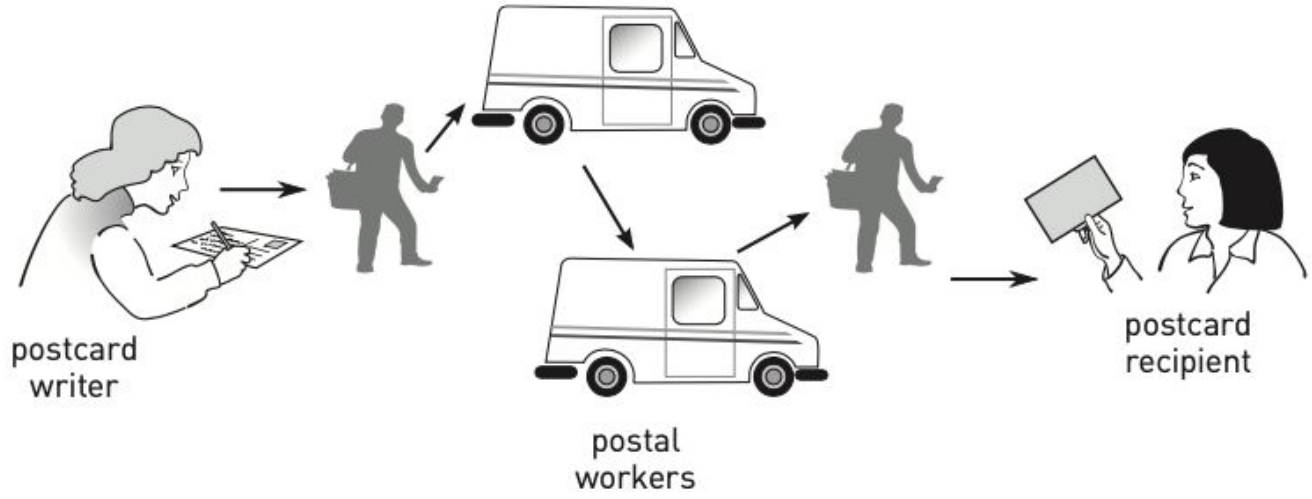
- Approaches
  - Diffie-Hellman key exchange protocol *(Chapter 4 in “9 algorithms”)*
  - RSA *(Chapter 9 in “9 algorithms”)*



Diffie-Hellman Key Exchange Protocol

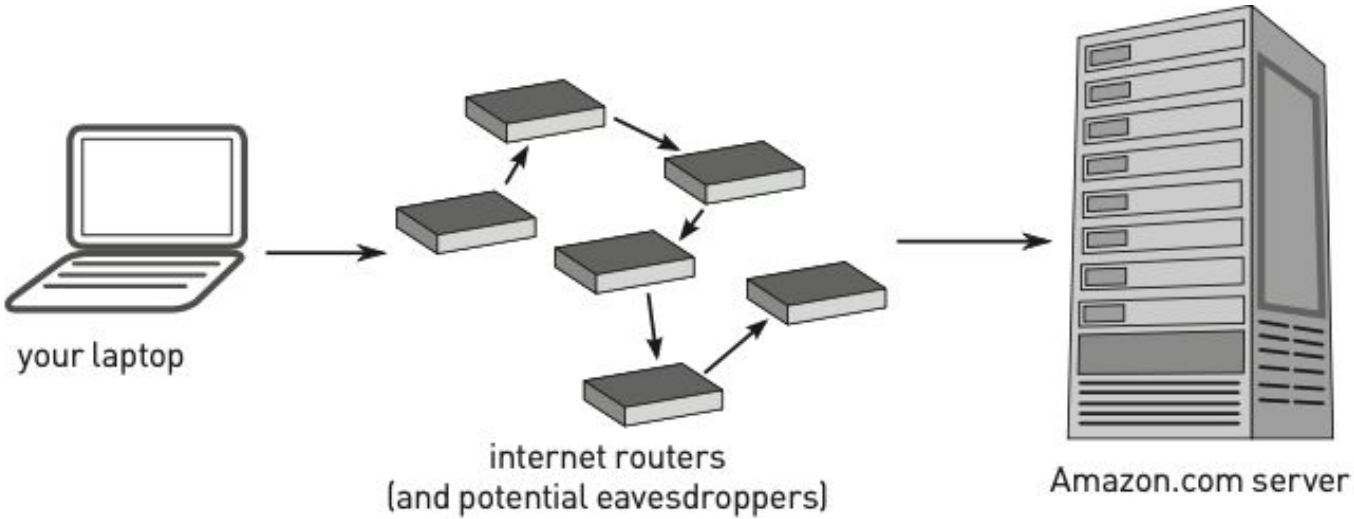
- It creates a shared secret between two parties of the communication, so that they can establish a secure communication channel over an untrusted network.
  - Developed by Whitfield Diffie and Martin Hellman in 1976;
  - One of the earliest practical implementations of public-key cryptography

Sending message on a postcard



It’s obvious that sending a postcard through the mail system will not keep the contents of the postcard secret.

Similar to sending a postcard



A credit card number sent from your laptop to Amazon.com can easily be snooped by an eavesdropper if it is not properly encrypted.

Can we send secrets on a postcard?

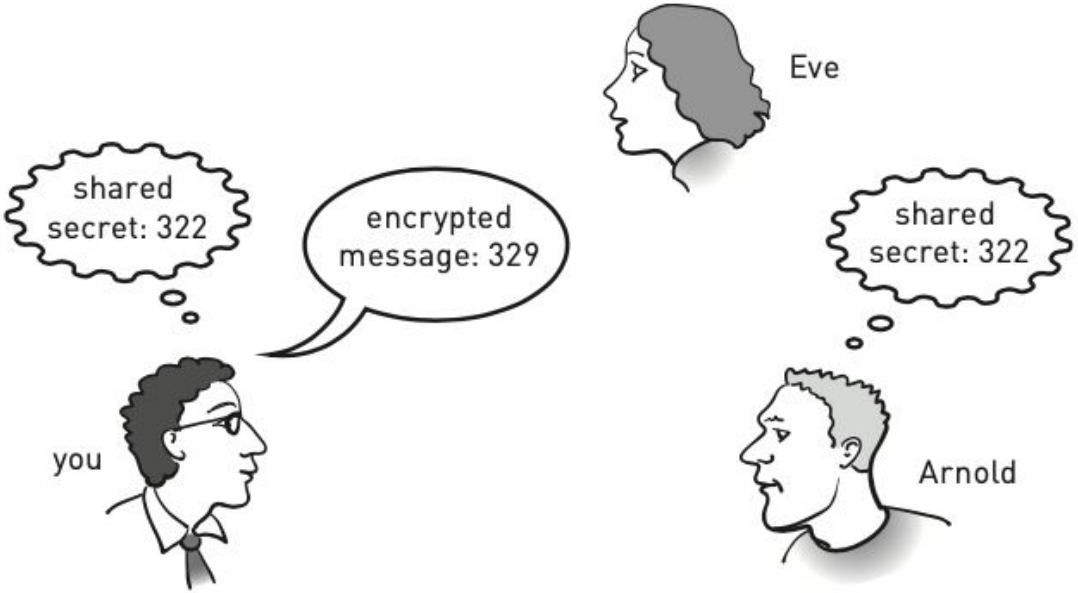
Yes! We can encrypt messages with a shared secret.

- Assume you and your friend Arnold shared a secret, e.g., the number of the house (322 on Pleasant street) you lived in when you were young.

Then, you can encrypt the message like this

- “Hey Arnold, remember the number of my family’s house on Pleasant street? Well, if you take that house number and add on the number I’m thinking of right now, you get 329. ”
- Arnold can calculate the number by 329-322. But others cannot as they don’t know the shared secret.

Using the shared secret



The trick: the message 7 is encrypted by adding it to the shared secret, 322. Eve cannot although she knows the whole conversation between you and Arnold.

A problem

- In the previous solution, we assume you and Arnold are friend and share a secret.
- In the real world, we sometimes have to communicate with people who are strangers to us (over the Internet).

Can we set up a shared secret via the public network without others knowing it?

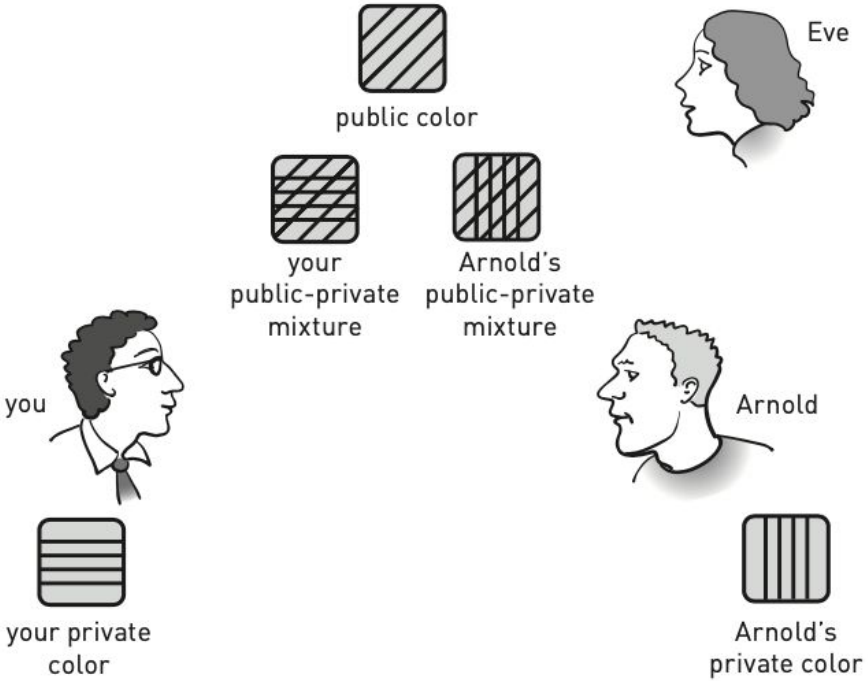
- Yes; by using a public key

Making a public key

Diffie-Hellman key exchange (paint-mixing trick)

- Suppose you, Arnold, and Eve are in a room.
- There are many colors available.
- Each of you has a huge collection of various pots of paint which are clearly labeled with its color.
- Everyone can mix the paint secretly without the others seeing.
    - Once the paints are mixed, they cannot be separated.
- You can share the mixtures of paint, placing a batch of them at a corner of the room for others to pick up

Making a public key



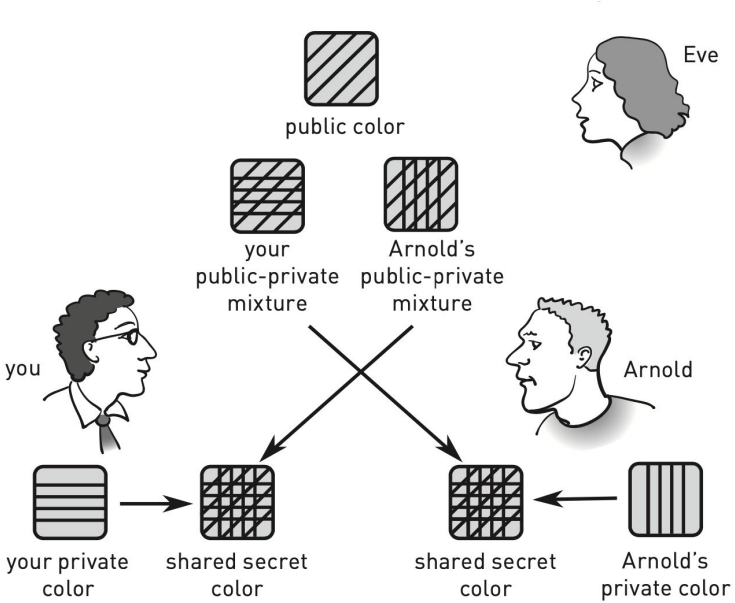
Now, you are going to create a share secrete with Arnold.

1. You and Arnold each choose a “private color”.
2. One of you announces the ingredients of a new different color which is called the “public color”
3. You and Arnold create a mixture by combining the public color with your private color, which is called “public-private mixture”.
4. You and Arnold pick up the “public-private mixtures” of each other and mix them with your own “private color”.

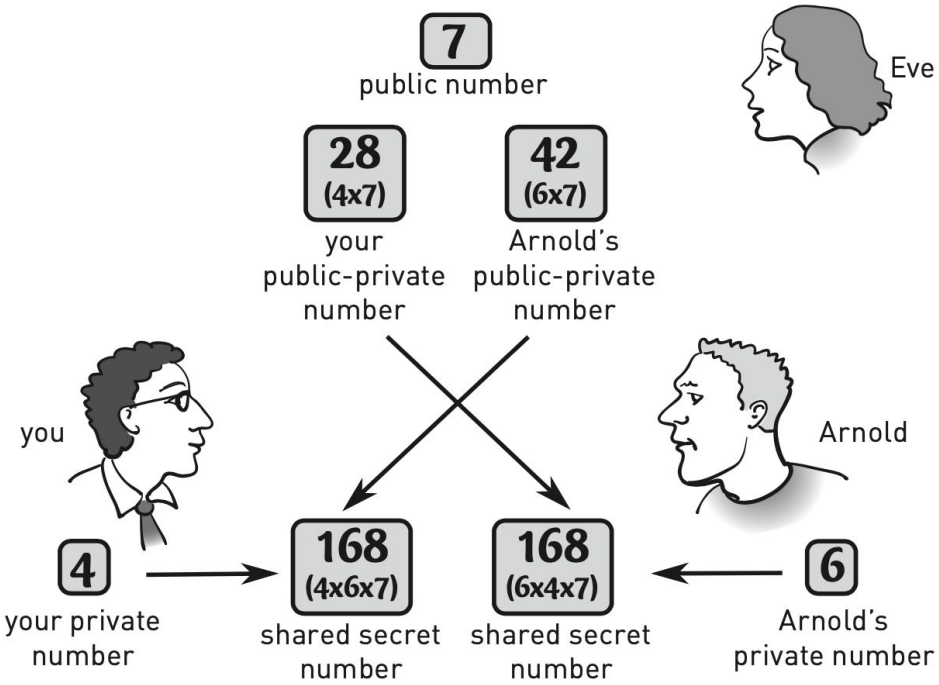
Making a public key

Now, you and Arnold have identical mixtures! But Eve won't.

- Eve cannot obtain the shared secret color because there is not way of "unmix" the mixed paint.
- If she adds two public-private mixtures, she still don't know what the shared color is as there is an extra public color in it.
    - If she mix any two mixtures she has, there are too many public color.



Replacing colors by numbers



- The public number is 7. You and Arnold multiply your private numbers with it and share the "mixtures".
- Then, you multiply Arnold's public-private number with your private number, you can obtain the secret number, same to Arnold's.
- Now you can encode your message with the secret number.

How to make the mixture irreversible?

Multiplication is not good for mixing operation because it can be reversed by division. In real life, we use **modular exponentiation** to perform the mixing of numbers.

- Modular exponentiation:

  Given base b, exponent e, and modulus m, the modular exponentiation c is c = b^e mod m, i.e., c is the remainder of b^e / m.

e.g., given b = 5, e = 3, and m = 13, c = (5^3 mod 13) = 8.

The unmixing operation of modular exponentiation is called discrete logarithm which cannot be calculated efficiently by any known methods.

The algorithm

1. You and Arnold each choose a private number
2. You and Arnold publicly agree on two public numbers: one is the modulus m, another is the base b.
3. You and Arnold each create a public-private number (PPN) by the following
   - $PPN = base^{Private\ number} \bmod m$
4. You and Arnold each take the other's PPN and mix it in with your own private number by
   - $Shared\ secret = other\ person's\ PPN^{Private\ number} \bmod m$

Proof of the effectiveness

Let the public numbers be m (modulus) and b (base); let p and q be two private numbers. We have two public-private numbers p_n and q_n.

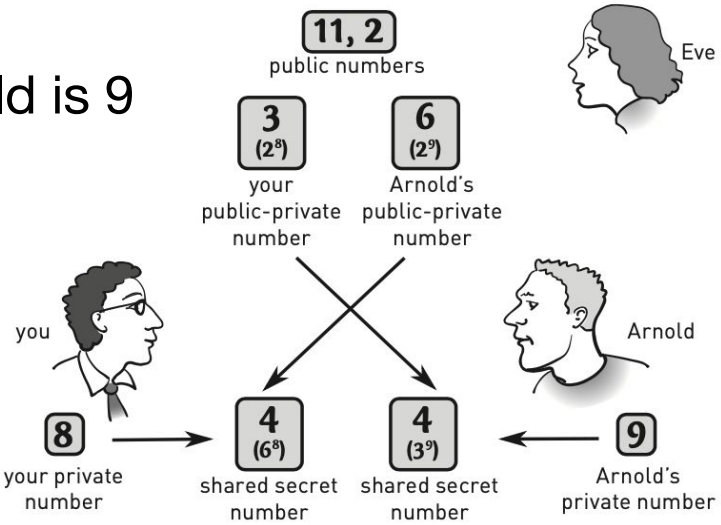
Since $(km + c)^n = (km)^n + (km)^{(n-1)}c + \ldots + c^n$ (where km is the part that divisible by m, c is the remainder), we have $(km + c)^n \bmod m = c^n \bmod m$. So, it holds the following,

- $(b^p)^q \bmod m = (km + p_n)^q \bmod m = p_n^q \bmod m$
- $(b^q)^p \bmod m = (hm + q_n)^p \bmod m = q_n^p \bmod m$

Therefore, the algorithm generates a shared secrete.

Example:

- Public numbers (11, 2), i.e., m = 11, b = 2
- Your private number is 8 and that of Arnold is 9
- Calculating the public-private numbers:
  - Yours: $3 = 2^8 \bmod 11$
  - Arnold's: $6 = 2^9 \bmod 11$
- Calculating the shared secret number:
  - On your side: $6^8 \bmod 11 = 4$
  - On Arnold's side: $3^9 \bmod 11 = 4$



Now, you can encode the massage with the secret number. (using symmetric encryption)

Diffie-Hellman protocol in practice

The Diffie-Hellman key exchange protocol helps ensure the confidentiality of the data transmitted over the HTTPS connection.

- When you go to a secure website that starts with “https:”, your computer and the web server is communicating with create a share secret using Diffie-Hellman protocol.

Task: secure the chat using Diffie-Hellman protocol

- In our chat system, all messages will be transferred by the server.
- To prevent being eavesdropped, we can encrypt the messages between all parties.
- You can build a cipher between every two clients in the chat group, using Diffie-Hellman protocol.

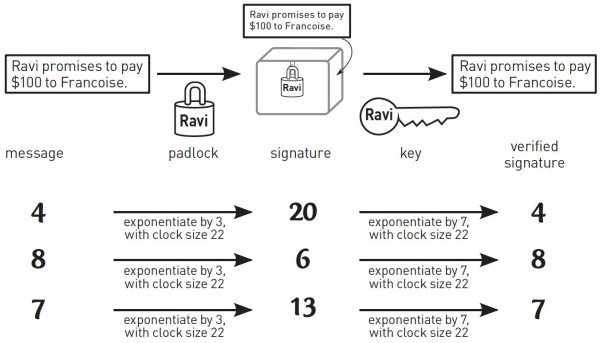
- How will the “search” still work if your encrypt the messages?

RSA

- Named after its inventors: Ronald **R**ivest, Adi **S**hamir, and Leonard **A**dleman

- <u>Encrypting messages using public keys and decrypting it using private keys</u>
  - If a message has been encrypted with the public key, it can only be decrypted by the private key paired to it.
  - If a message has been encrypted with the private key, it can be decrypted by the public key paired to it.

- Widely used in daily applications
  - e.g., secure messaging, digital signatures



Locking and unlocking messages using exponentiation.

The RSA algorithm

1. Selecting two primes, p and q, and obtaining the modulus n = p x q;
2. Generating lcm (i.e., the lowest common multiple) of (p -1), and (q -1)
3. Selecting a number e from [1, lcm] as a public key; taking n as another public key;
4. Generating the private key d by d = 1/e mod lcm;
5. Encoding message m by $c = m^e \bmod n$
6. Decoding message m by $m = c^e \bmod n$

Proof skipped. In case you are interested, refer to:
https://www.cse.cuhk.edu.hk/~taoyf/course/bmeg3120/notes/rsa-proof.pdf

Secure messaging

To use RSA in secure messaging,

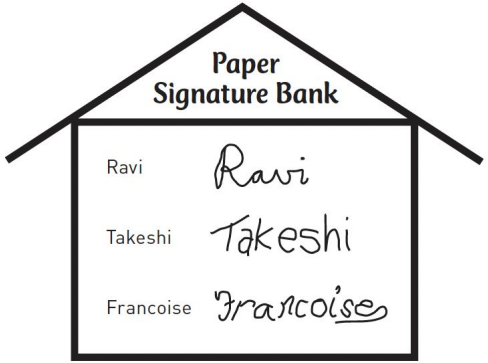
- one should generate the private key and public key,
- then, he/she shares the public keys to others;
- the others will use the public keys to encode the messages send to the person;
- he/she will decode the message by his/her private key

Task: secure our chat system using RSA

- Each party in the chat group should share his/her public key to the others
- When he/she sends messages to the others, he/she needs to encrypt the message using public keys of others
- You can build a encryption-decryption system for this task.

Paper signatures

- handwritten signatures on paper
- copies are stored at some trusted institution, e.g., banks.
- verified by comparing with the copies



How we use signatures? → e.g., One day you received a paper statement:

“I promise to give A to everyone in this classroom.

Signed, Xianbin Gu ” -- Is this true?

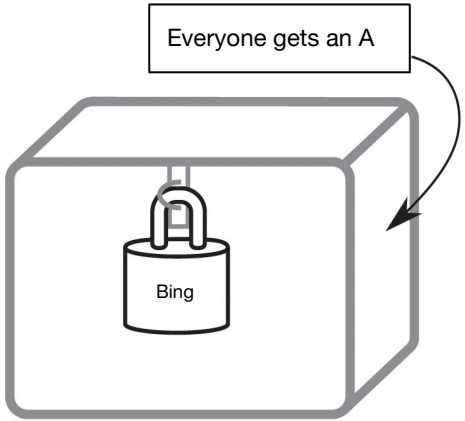
- If the sign is verified as genuine, then it is!

Digital signatures

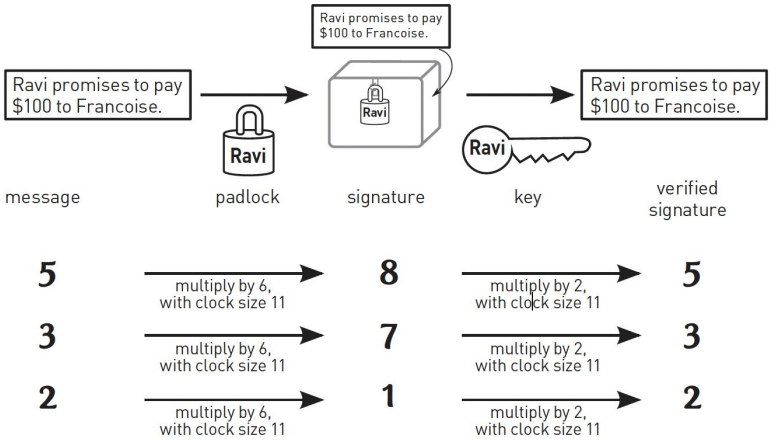
- Digital signatures can’t be in the same form of paper signatures because they can be forged easily. (e.g., simply by the “copy and paste”)

The idea: signing with a padlock

- Make a padlock that can **only** be locked by its owner. (using his private key to encrypt it)
- The owner makes a claim (under the witness of others) and locks it into the box.
- People can ask the owner to open the box and verify the claim. (sharing a public key)

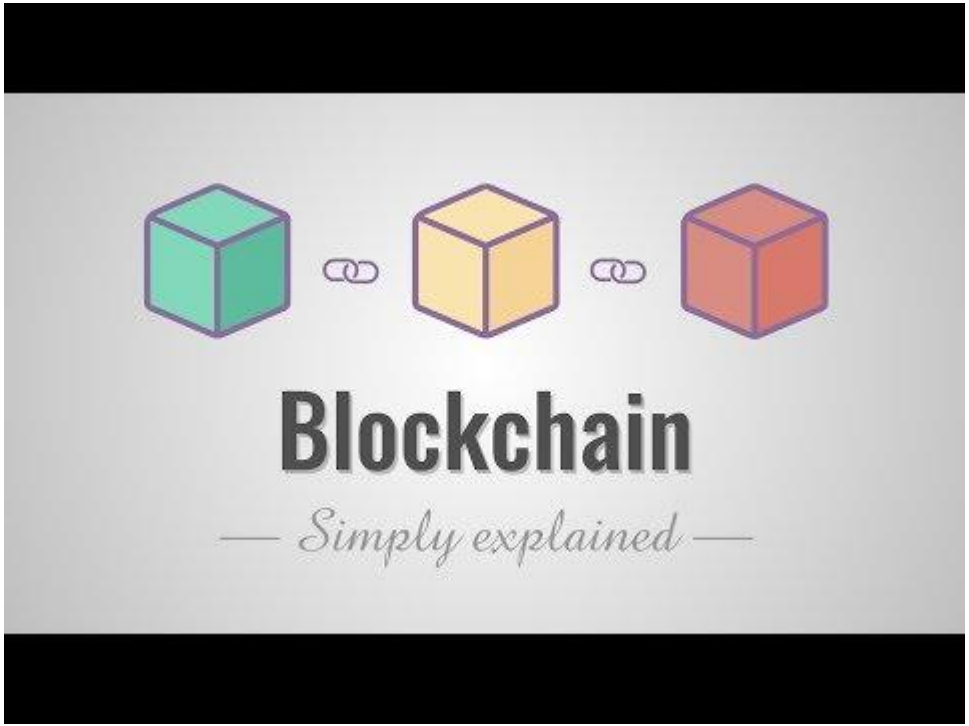


Lock a numeric message using a padlock



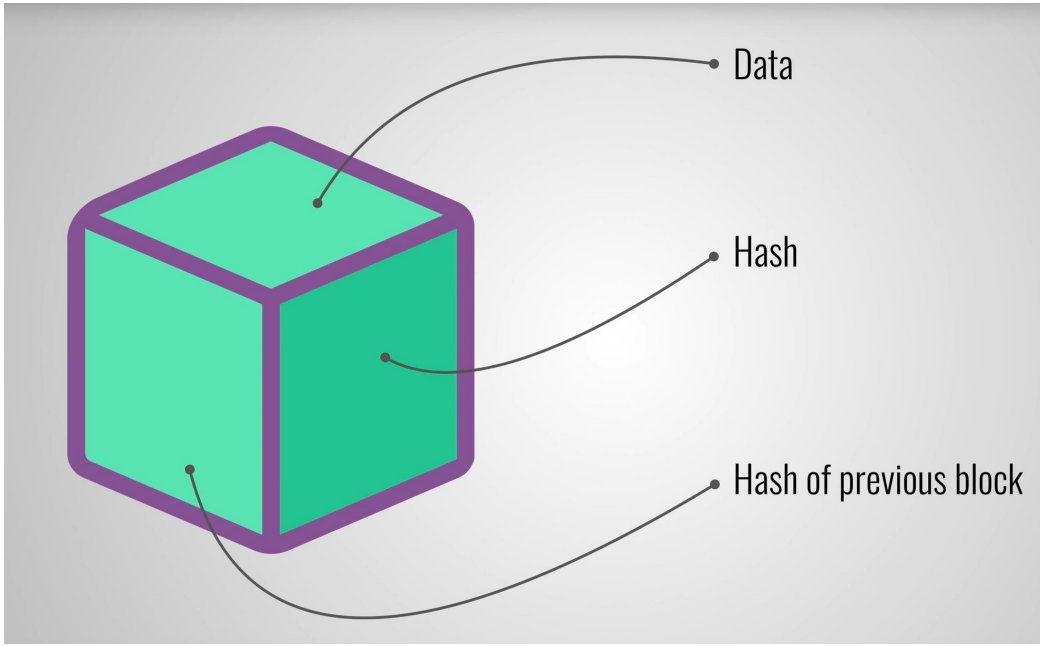
- Share the public key to others
- Make a claim (let everyone know what you have said)
- Convert it into some numbers (encrypt the claim with your private key; it is like you sign a signature)
- People can decrypt the encrypted claim using your public key, so they can ensure the claim is signed by you.

Blockchain



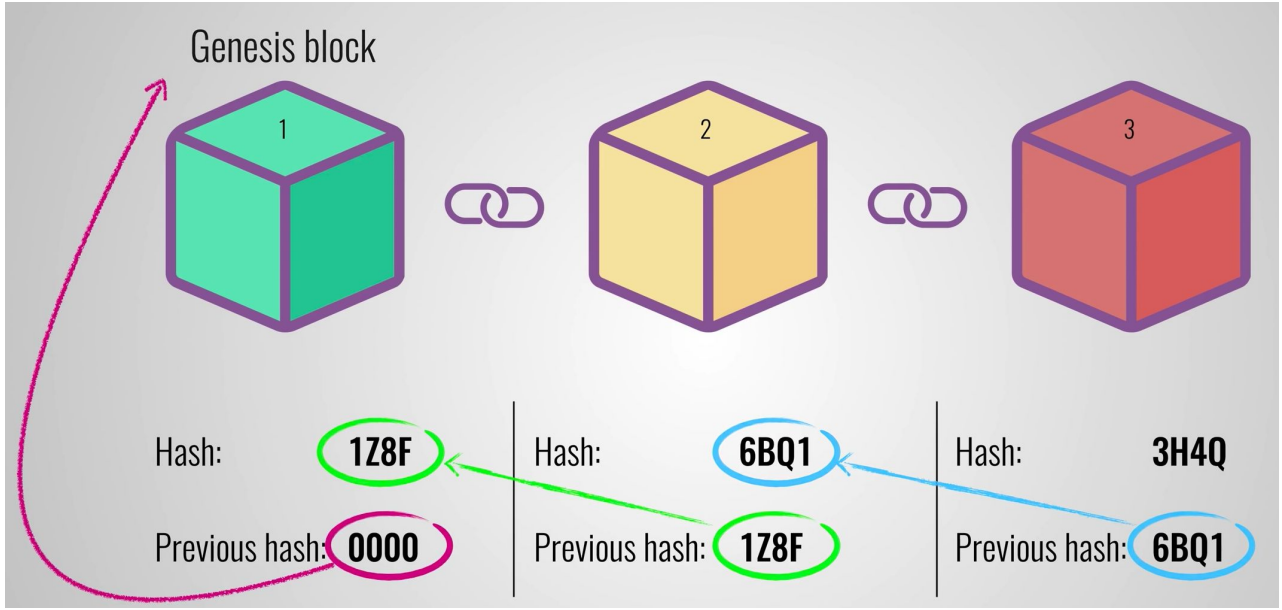
- Originally proposed for time stamping digital documents
- A “distributed ledger”

Each block contains three parts



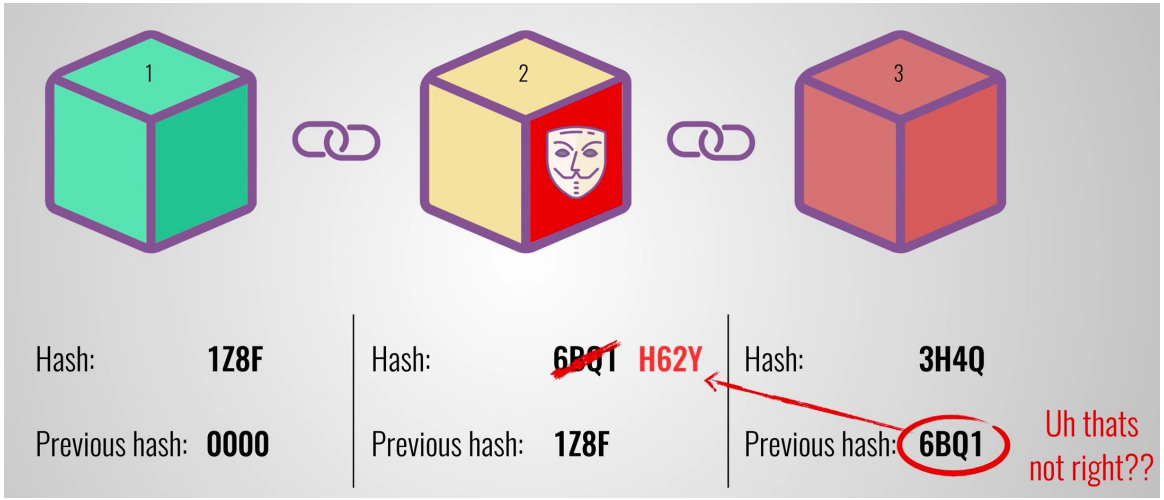
- Data: information carried by the block
- Hash: a peculiar label of the block, a “fingerprint”
- Hash of previous block: the fingerprint of the linked block in the front.

The blockchain



- Blocks are “linked” by the hashes

When a block is changed

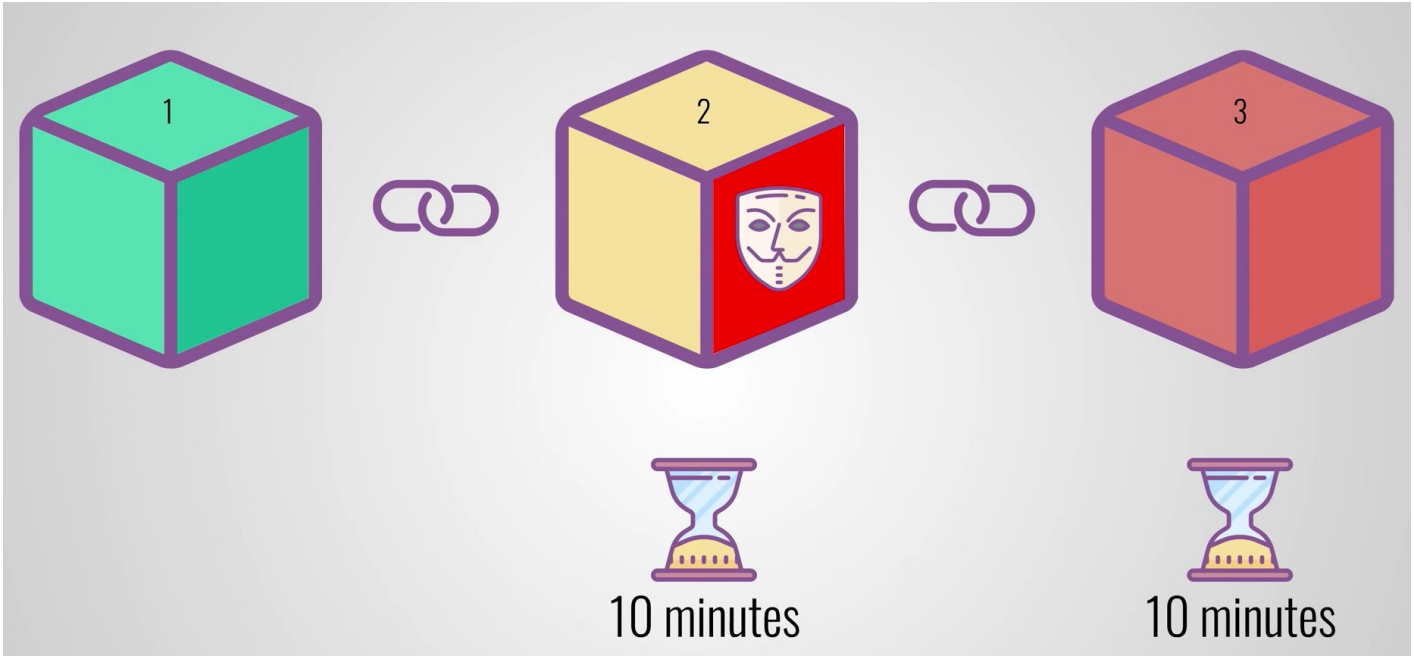


The hash is changed. To valid the change,

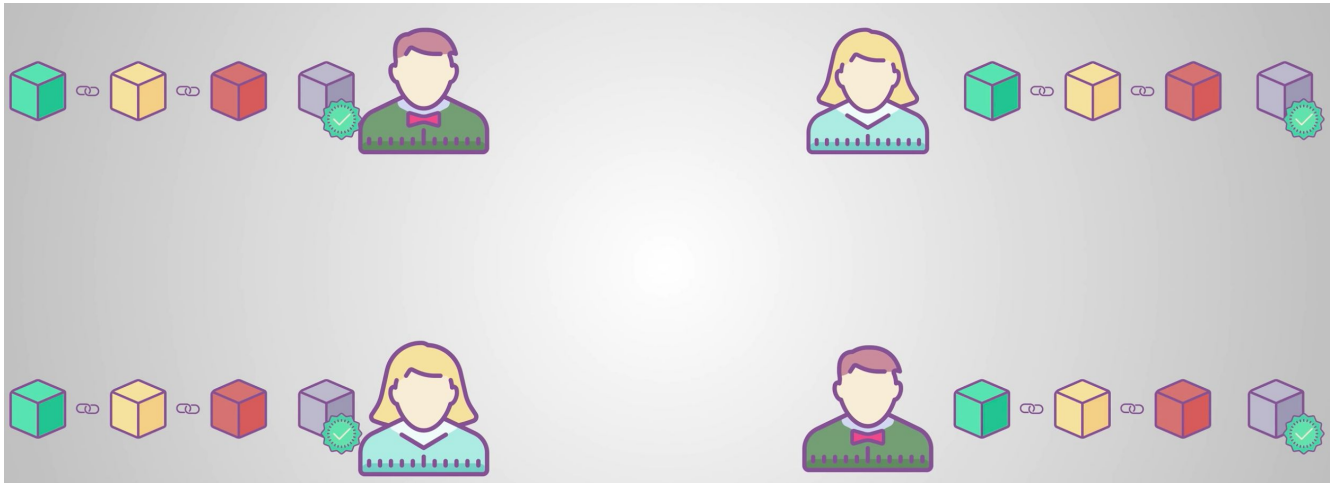
- The “previous hash” in the next block need to be changed, which results in the change of hash in that block, and so on.
- All blocks next to it should be changed.

Proof-of-work

- An algorithm to slow down the process of updating the hash of the next block.



P2P network: decentralizing



- Everyone in the network has a full copy of the blockchain
- New blocks are sent to everyone, being verified and added to the chain

It is almost impossible to tamper a block

If someone wants to successfully tamper a block, he has to

- Redo the proof-of-work for each block
- Take control of more than 50% of the P2P network.

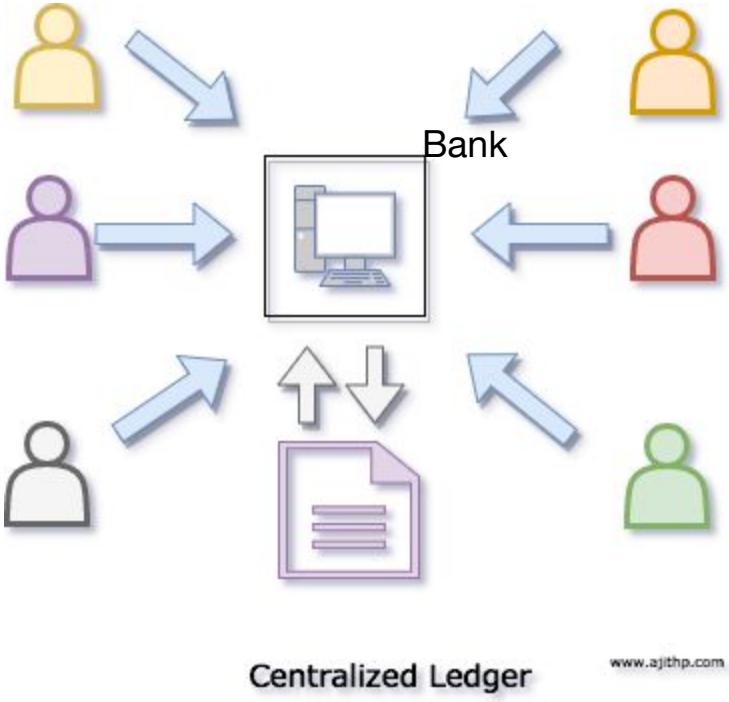
Currently, the size of blockchain of Bitcoin is about 160 GB, containing more than 500,000 blocks. Tampering one block is costly.

Appendix: Bitcoins

- A cryptocurrency
- Decentralized, peer-to-peer payment
- Technique foundation:
  - the Internet
  - Asymmetric encryption (RSA)
  - Block chain

Background

- Centralized transactions → needed a trusted third party (e.g., a bank)



A transfers $50 to **B**:

- A sends a request to his Bank for initiating the transfer
- Bank verifies the request, then subtract $50 from A's account and adds $50 to B's account if everything is ok.
- Bank updates its ledgers to reflect these changes.

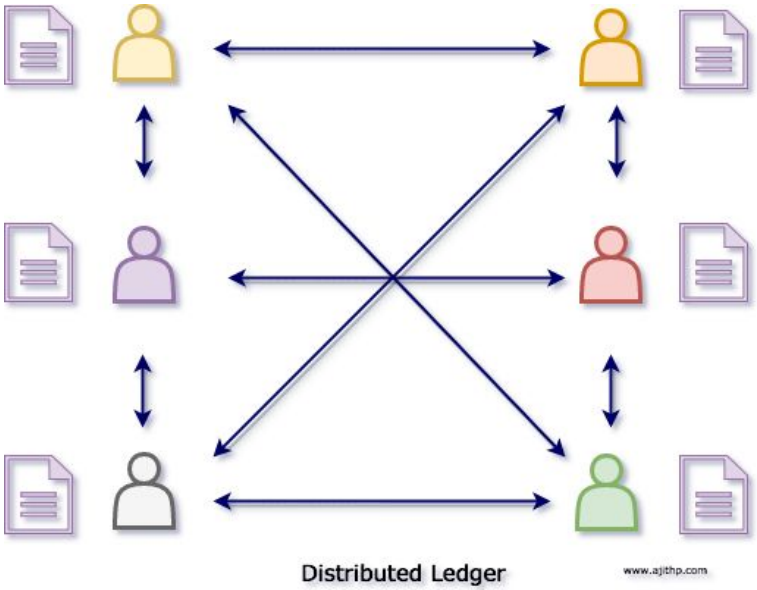
(a client-server model)

Background

- subprime mortgage crisis → banks are not trustworthy → 2008 financial crisis
- Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”
  - No one knows who he/she/them is

Decentralized transition

- Translation is done directed between two parties with a p2p network, no need for the trusted third party.



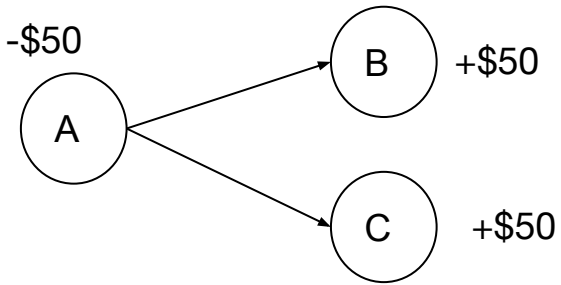
- Everyone has a ledger.

If A transfers $50 to B in decentralized model:

- A informs B about the transfer, updating his ledger, subtracting $50 from his account
- B receives A's message, updating her ledger, adding $50 into her account

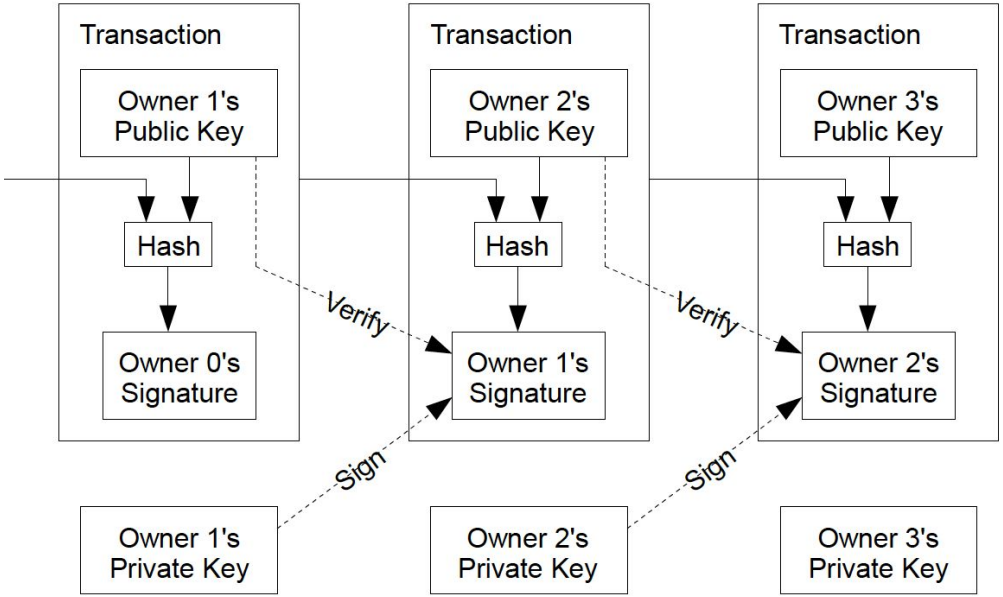
Issues of a P2P network: how to solve them?

- How to make transitions non-reversible? ⇒ block chain, proof-of-work
- Double-spending problem: one can perform two similar transitions simultaneously ⇒ “first come first serve”
  - Assume A has $50 in total. In a p2p network, A can pay B $50, and pay C $50 at the same time, because the money in A’s account changes only when the ledgers are changed. Such double-spending will end up in conflicts.



- Both transactions go into a pool of unconfirmed transactions, but only the first transaction gets confirmations (blocks containing transactions from preceding blocks and new transactions) and is verified by miners in the next block.

Transaction



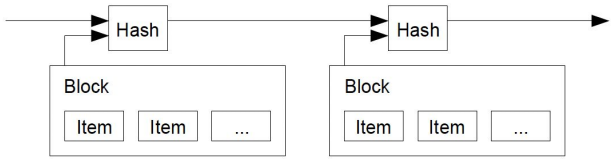
Transactions are signatures (using RSA)

- Hash (Previous transactions + payee's public key) → payer's signature (using RSA)
- Others can verify the transaction by the payer's public key
- The payee can get the payment by his/her private key (i.e., pay them to others)

E.g., When owner 2 gets the payment from owner 1, he needs to provide his public key to the hash. It will mix the public key and the previous transaction by hashing, then, owner 2 uses his private key to generate a signature for the output of the hash

Proof-of-work

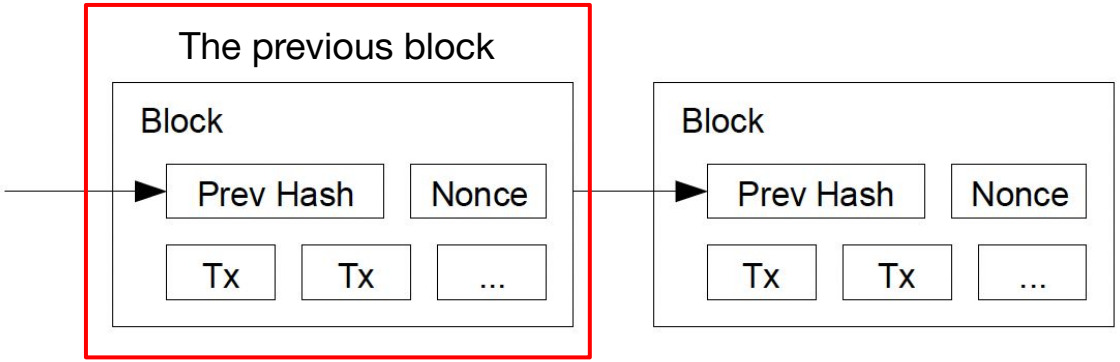
- A transaction should be added into a blockchain so to get confirmed



- But computers all around the world work on this simonuously, they may generate different blocks in the same time;
- Which block should be added to the chain? → Proof-of-work → so, every 10 minute only one block can be generated.

Proof-of-work

- Generate a new block: two steps



1. SHA256(previous block) + transactions in the new block + other information of the new block
2. Finding a random number; the first 72 digits of its SHA256 are all 0s. (the output of SHA256 contains 256 digits.)

Incentive

- Encouraging people making blocks to support the chain, distribute transactions
- Two parts: transaction fee + block reward
- A block reward refers to the number of bitcoins you get if you successfully mine a block (i.e., create a block and add it to the chain)
    - At inception, each bitcoin block reward was worth 50 BTC → The block reward is halved after the discovery of every 210,000 blocks, which takes around four years to complete. As of February 2019, one block reward was worth 12.5 BTC.
    - The number of BTC will reach 21,000,000 in 2040, no more block reward then.