BLOCKCHAIN

CUHK Open Innovation Lab

DOWNLOAD

- https://nodejs.org/en/
- https://github.com/ethereum/mist/releases
- https://github.com/ethereum/web3.js/blob/develop/dist/web3.js

BLOCKCHAIN

BLOCK – FIRST BLOCK

- Genesis Block (block 0 / block 1)
- Hardcoded
- Does not reference a previous block

BLOCK – FIRST BLOCK

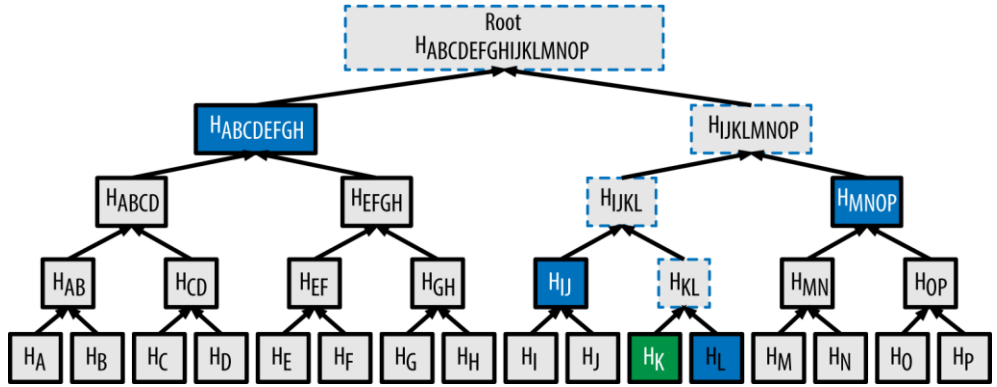
```
{
    "config": {
        "chainId": 15,
        "homesteadBlock": 0,
        "eip155Block": 0,
        "eip158Block": 0
    },
    "difficulty": "200000000",
    "gasLimit": "2100000",
    "alloc": {
        "7df9a875a174b3bc565e6424a0050ebc1b2d1d82": { "balance": "300000" },
        "f41c74c9ae680c1aa78f42e5647a62f353b7bdde": { "balance": "400000" }
    }
}
```

BLOCK

```
class Block {
    constructor(index, previousHash, timestamp, data, hash) {
        this.index = index;
        this.previousHash = previousHash.toString();
        this.timestamp = timestamp;
        this.data = data;
        this.hash = hash.toString();
    }
}
```

BLOCKS

- Block header
  - Timestamp, difficulty, …
  - Merkle root of transactions in current block
  - Hash of pervious block
- Transactions

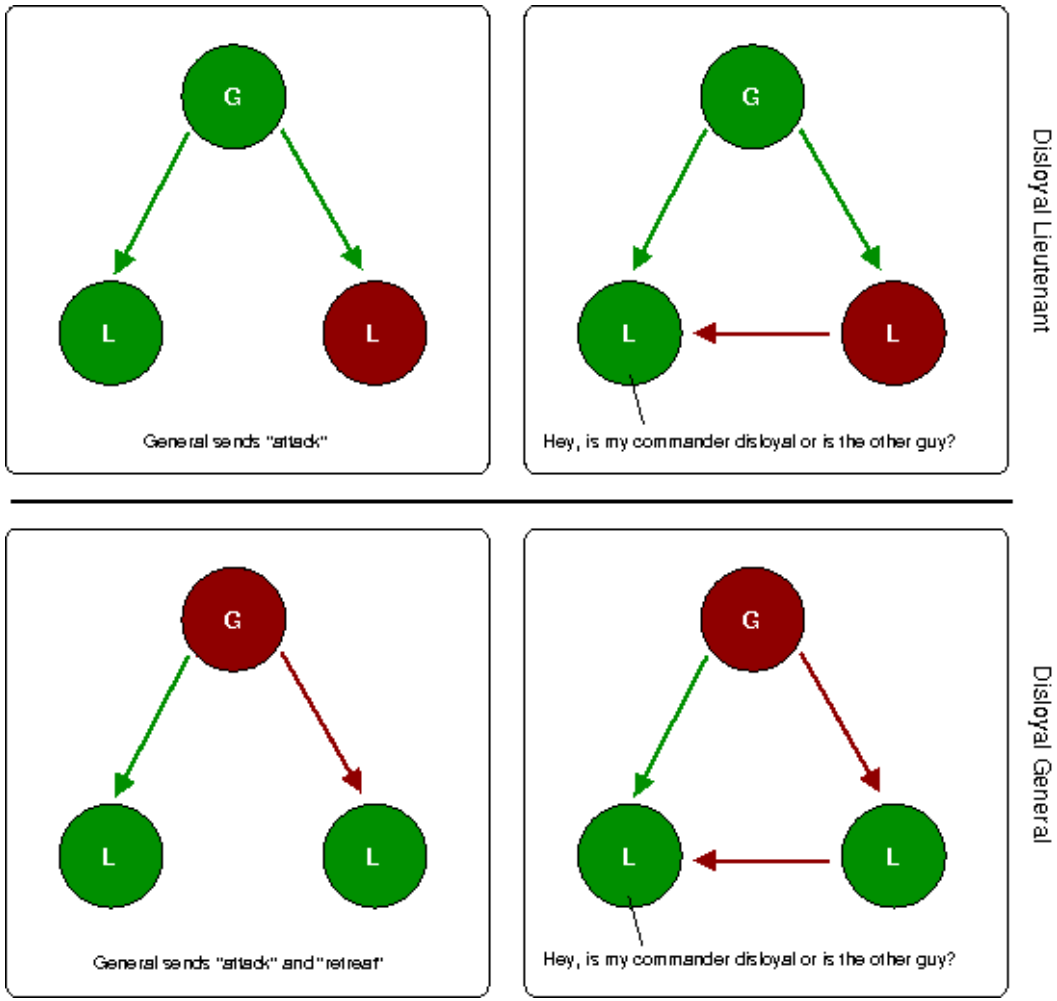


HOW PEERS ARE FOUND

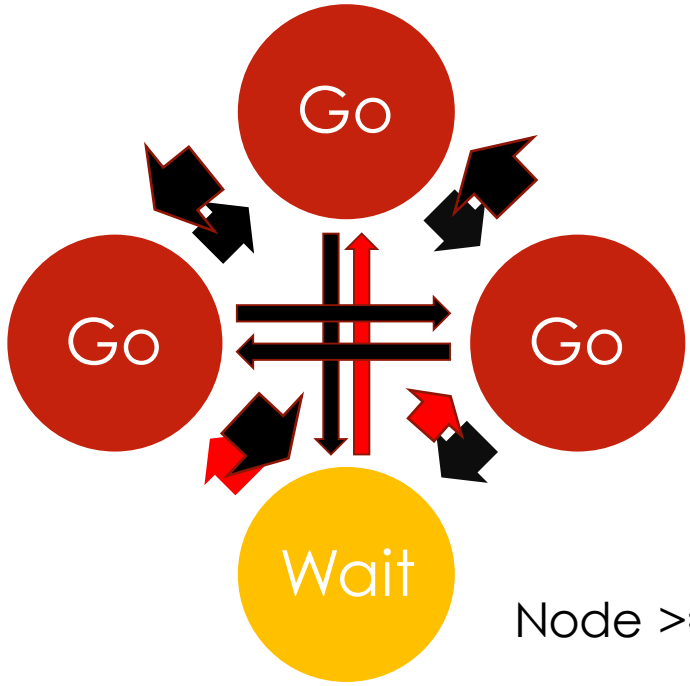
- 1. continuously attempts to connect to other nodes on the network until it has peers

- 2. Static nodes
  eg:
  "enode://f4642fa65af50cfdea8fa7414a5def7bb7991478b768e296f5e4a54e8b995de102e0ceae2e826f293c481b5325f89be6d207b003382e18a8ecba66fbaf6416c0@33.4.2.1:30303“

BYZANTINE GENERALS' PROBLEM



BYZANTINE FAULT TOLERANCE

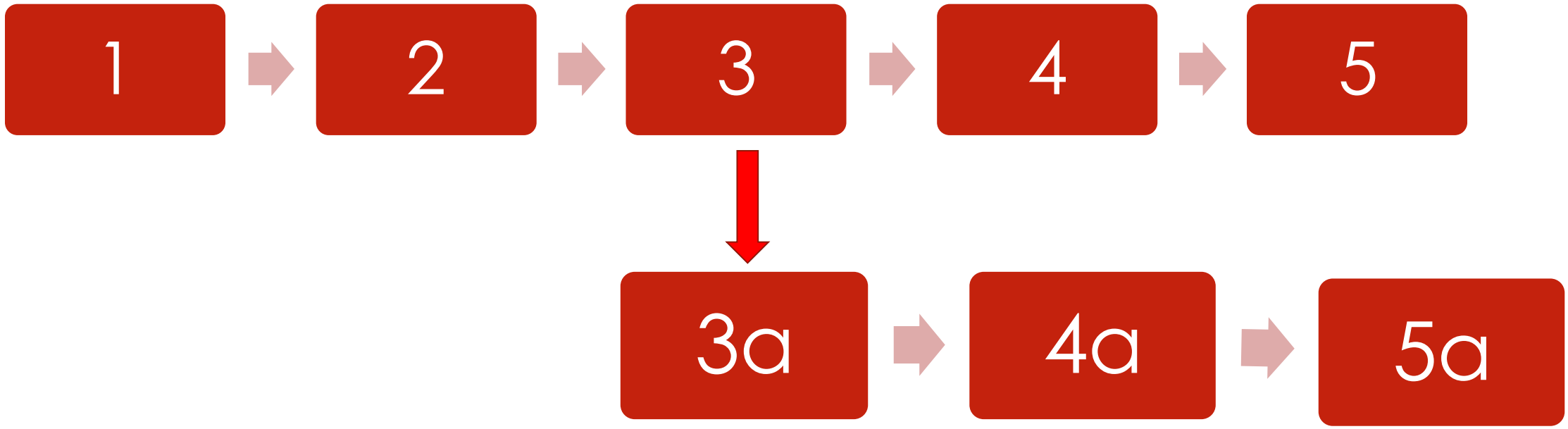


Node >= 3m+1

51% ATTACK

- After a block is generated, it requires validation from other peers
- If malicious nodes has 51% Power within the network, it can reject blocks generated by others

HARD FORK



SOFT FORK

DATA STORAGE PROBLEM

- Problem: if f(x) represent the block size for block number x. f’(x) is an increasing function , lim(x->inf) f(x) = infinity.

- Solution 1: Mining Pool
- Solution 2: Reduce data on each block
- Solution 3: Semi-Distributed Network

Each file and all of the **blocks within it** are given a **unique fingerprint** called a **cryptographic hash**.

IPFS **removes duplications** across the network and tracks **version history** for every file.

Each **network node** stores only content it is interested in, and some indexing information that helps figure out who is storing what.

When **looking up files**, you're asking the network to find nodes storing the content behind a unique hash.

IPFS

APPLICATION: DECENTRALIZED DNS

- A good DNS should be **global**, **human-meaningful** and **decentralized**

- **Certificate Authorities** can be wrong

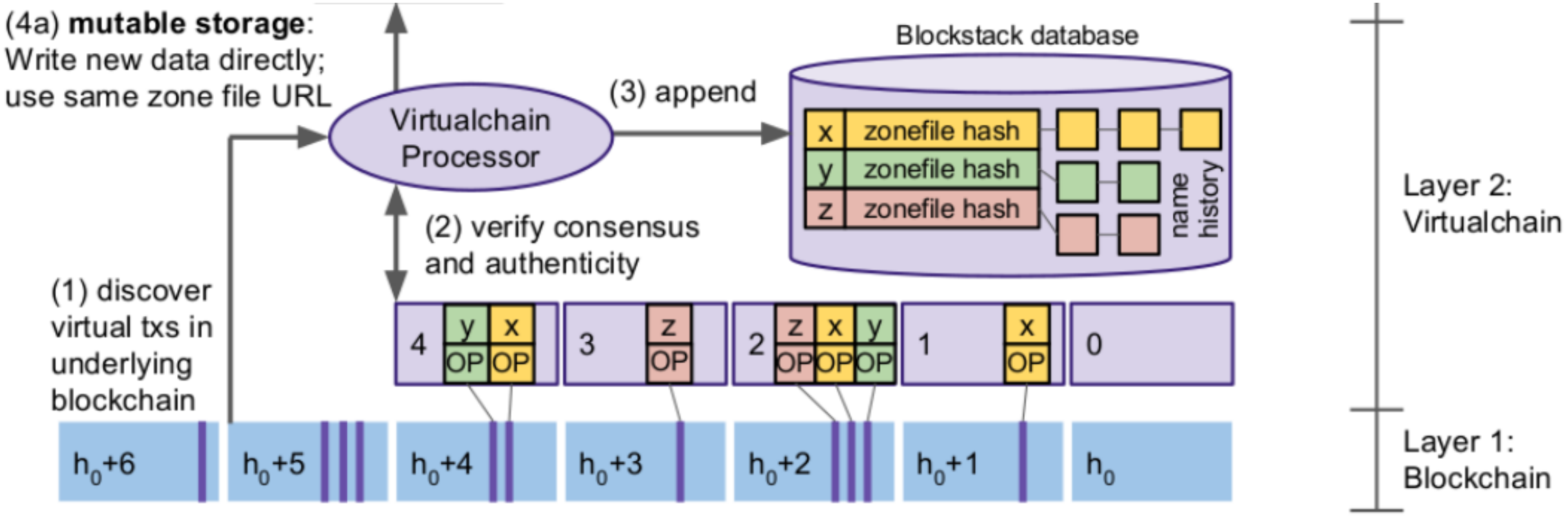
There have been notable screw-ups by CA's that didn't require getting their systems cracked. In 2016, the CA WoSign issued a certificate for github.com to someone who only proved that they had a user account on GitHub [14]. Also in 2016, the CA Comodo issued a certificate to the wrong person because the CA used OCR image-scanning software to read a picture of the website owner's email address, and the OCR algorithm misread the email address [15].

SECURITY VS FLEXIBILITY

- Facts
  - It is difficult to perform a permanent fork frequently.
  - Mining pool may cause 51% attack

- Question: How can we update our service on Blockchain?

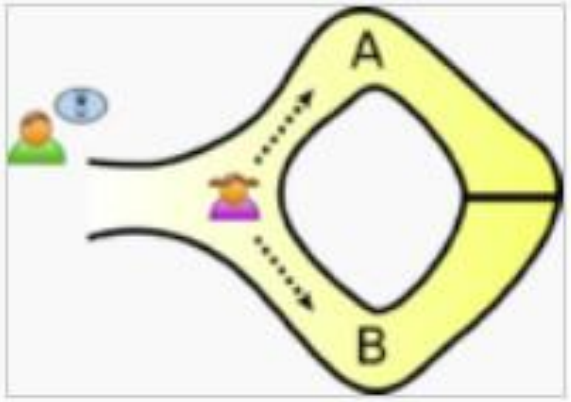
SOLUTION: VIRTUAL CHAIN



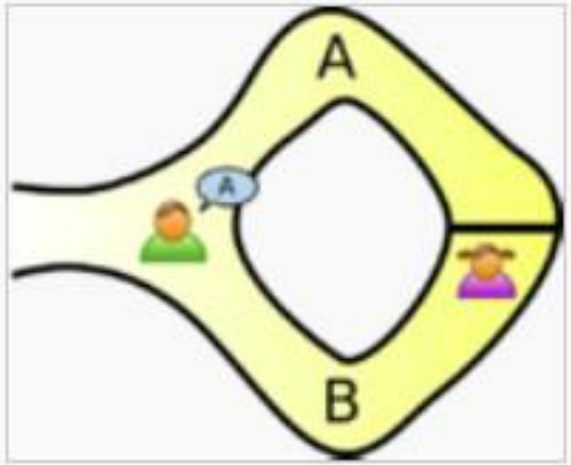
PRIVACY PROBLEM

- All data in Blockchain are public
- Is it possible to store personal information for identification process?

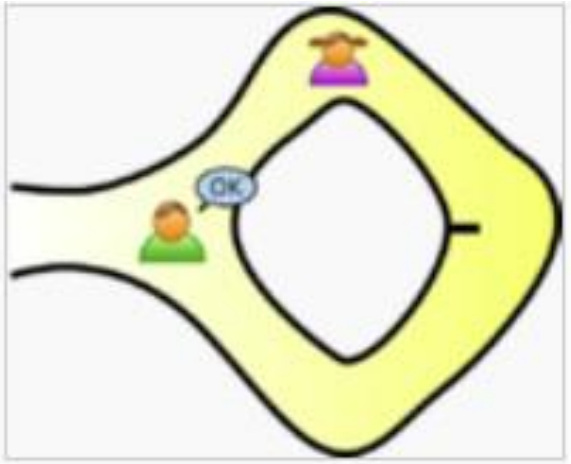
ZERO-KNOWLEDGE PROOF



Peggy randomly takes either path A or B, while Victor waits outside



Victor chooses an exit path



Peggy reliably appears at the exit Victor names

SMART CONTRACTS

- Simple program codes
- Language: Solidity
- Cannot edit after deployed
- X self-launch
- Keep simple!

DEMO

- Download Mist Wallet
- Installed NodeJS
- npm install –g ethereumjs-testrpc
- Web3.js

SITCON X HK 2017