BLOCKCHAIN TECHNOLOGY FOR IOT ACCESS CONTROL

Vashi Chaudhary
2018A7PS0243H

INTERNET OF THINGS(IoT)

IoT is a new model that contains all wireless networks, actuators and mobile networks where each component has a unique address and they communicate amongst each other using radio-Frequency identification devices (RFID)

INTERNET OF THINGS(IoT)

The Internet of Things (IoT) as a concept is fascinating and exciting, but one of the major challenging aspects of IoT is having a secure ecosystem encompassing all building blocks of IoT-architecture.

Applications of IOT

- Transportation and logistics domain - Logistics, Assisted driving, Mobile ticketing, Monitoring environmental parameters, Augmented mapsv
- Healthcare domain - tracking, identification, authentication, data collection, sensing
- Smart environment domain - comfortable homes/offices, industrial plants, smart museums and gyms
- Personal and social domain - social networking, historical queries, losses thefts

OSI MODEL : Open System Interconnection

7. **Application Layer** - Applications that work at Layer 7 are the ones that users interact with directly. Eg Chrome.

6. **Presentation Layer** - it represents the preparation or translation of application format to network format. Ex Encryption/Decryption

5. **Session Layer** - Functions at this layer involve setup, coordination and termination between the applications at each end of the session

4. **Transport Layer** - The Transport Layer deals with the coordination of the data transfer between end systems and hosts

3. **Network Layer** - this layer is responsible for packet forwarding, including routing through different routers

2. **Data Link Layer** - The Data Link Layer provides node-to-node data transfer (between two directly connected nodes), and also handles error correction from the physical layer

1. **Physical Layer** - which represents the electrical and physical representation of the system

TCP/IP Model

- The **TCP/IP model** is a concise version of the OSI model
- This model has just 4 layers

TCP/IP model	OSI model
Application Layer	Application + Presentation + Session
Transport Layer	Transport
Inter-network Layer	Network
Network Access Layer	Data link + Physical

Application Layer

- Combination of application, presentation and session layer in OSI model.
- Provides interface to the user
- Presents data in user readable format
- Data encapsulation, encryption/ decryption

Transport Layer

- Also called Host-to-Host layer.
- Acts as an entity between network and application layer.
- Provides error free delivery of data
- Shields the applications from complexity of data.

Two protocols -

- TCP - Connection oriented. Reliable, slower and costly
- UDP - Connectionless. Unreliable but faster

Internet layer

Allows host to inject packets into any network.

Two protocols -

- IP - Proper delivery of packets to the destination
- ICMP - error reporting protocol
- ARP - Address of a known hardware host using a known IP address

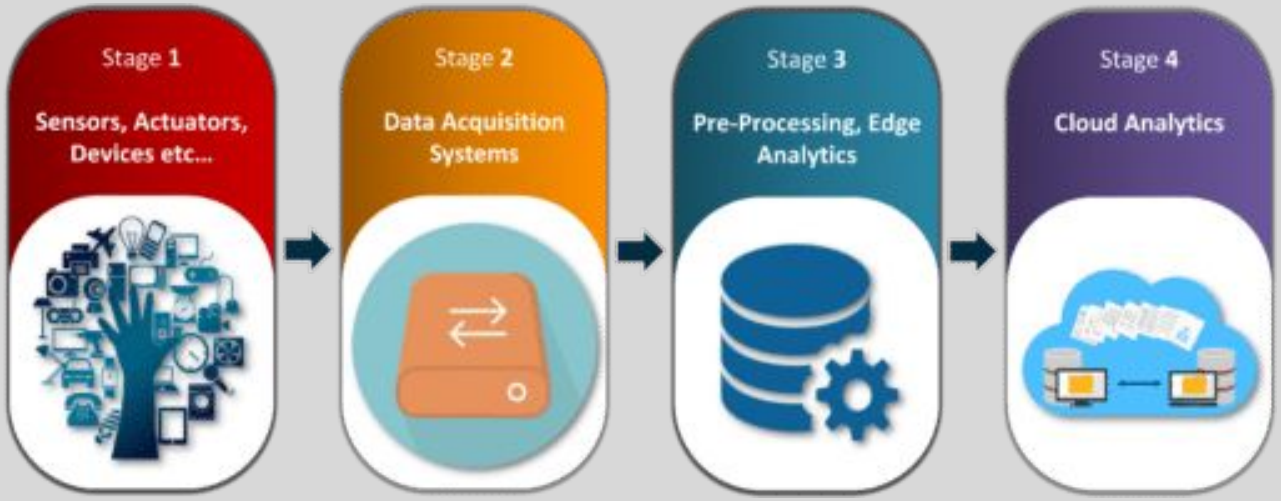
Network access layer

Combination of data link and physical layer.

Responsible for -

- Error control
- Flow control
- Access control
- Bit conversion

IoT Architecture



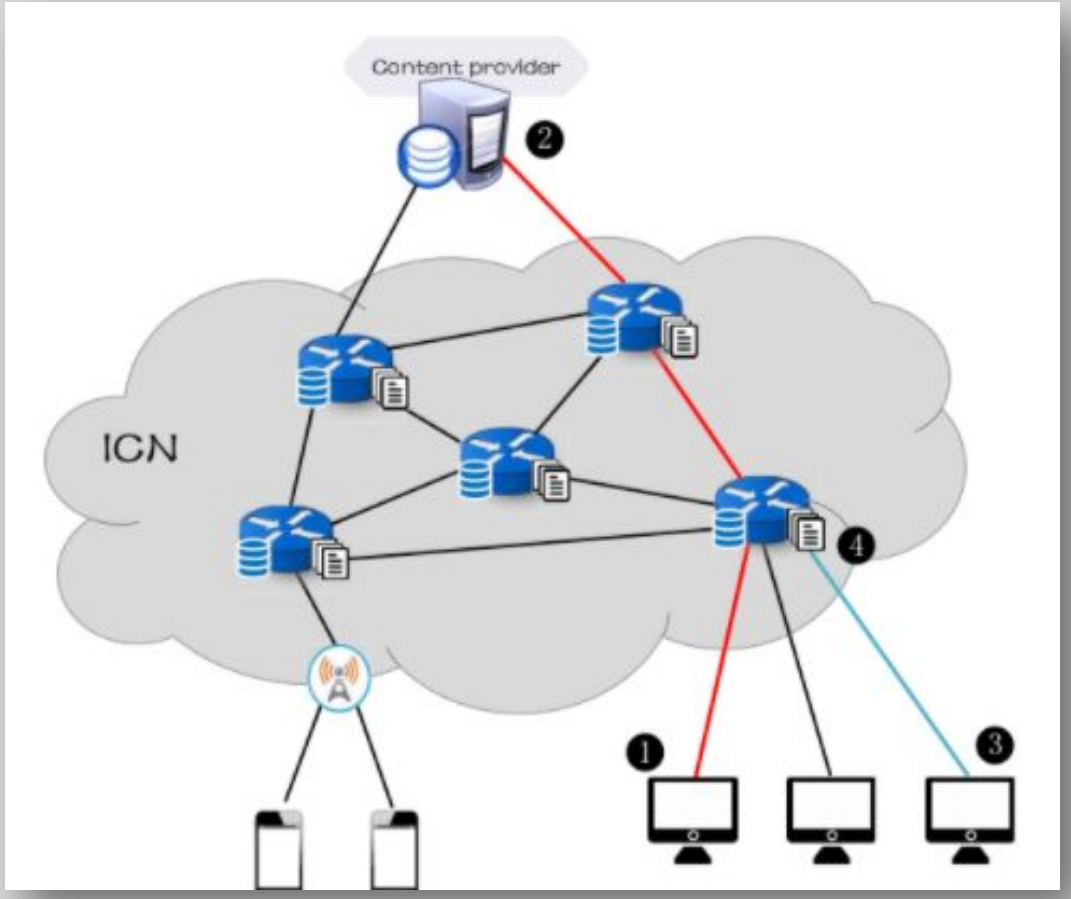
STAGE 1: Sensing and actuating stage covers and adjusts everything needed in the physical world to gain the necessary insights for further analysis.

STAGE 2: The vital importance of this stage is to process the enormous amount of information collected on the previous stage and squeeze it to the optimal size for further analysis

STAGE 3: The prepared data is transferred to the IT world. In particular, edge IT systems perform enhanced analytics and pre-processing here.

STAGE 4: It enables in-depth processing, along with a follow-up revision for feedback

System Architecture

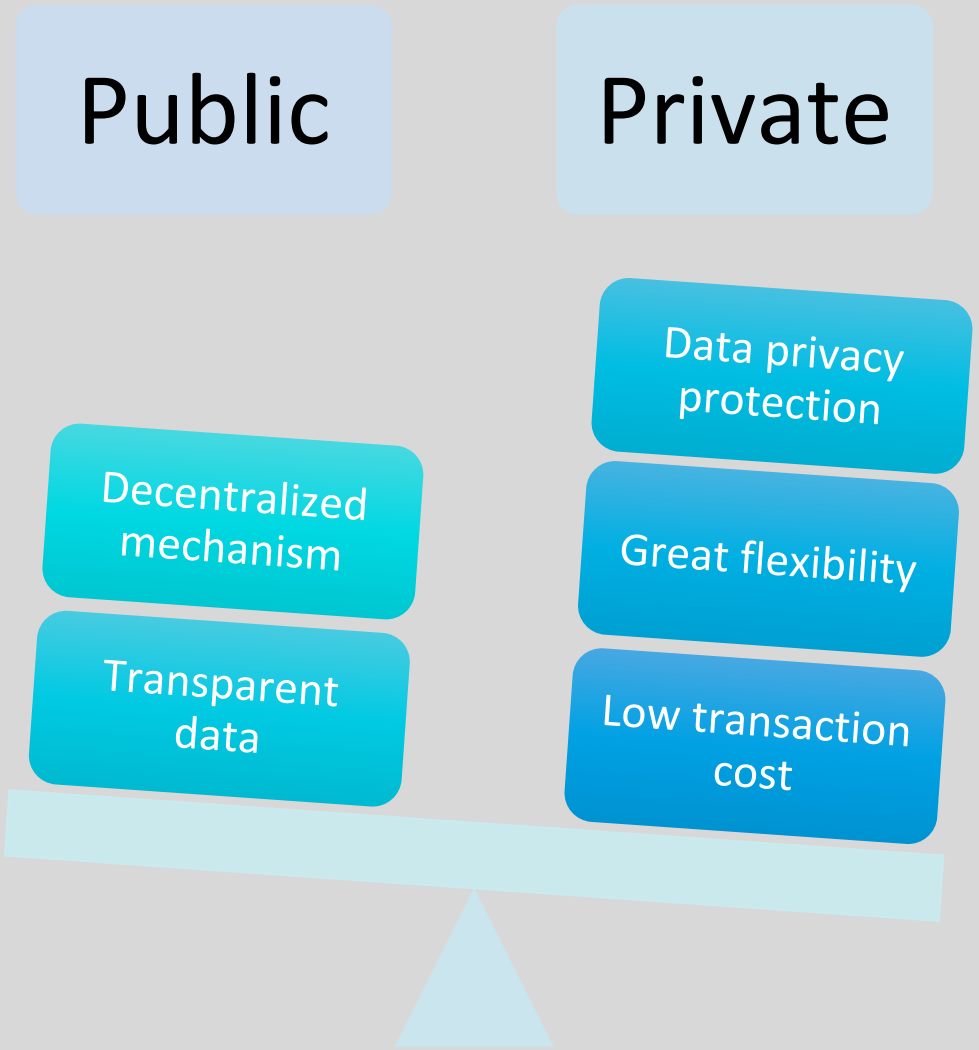


- Content Providers – Content providers are responsible for publishing the data file
- Internet Service Provider – This component caches data and forwards it to users
- Users – Data consumers who have different authority to access data

CURRENT IoT ARCHITECTURE

- Current IoT devices depend on server/ Client system(centralized system) and with the amount of data being handled , chances of cyber attack is huge.
- All devices are identified, authenticated and connected through cloud servers that support huge processing and storage capacities. Connections between devices have to exclusively go through the internet, even if they happen to be a few feet apart.
- In IoT access control systems, each lock, access controller, card reader and other associated devices are provided with unique IP addresses with which they communicate among themselves. These devices are connected through wireless networks to their mobile/software application.
- We can consider the example of vehicle control in an industry chain where identifying connected devices (vehicles, products, etc.) permits their localization and tracking. Obviously, getting this type of information instantly can improve the global system functioning and efficiency by immediate intervention when needed. Identification affects many aspects of the global IoT system, including conception, architecture, access rules, etc.

BLOCKCHAIN



1. Blockchain is a database that maintains a continuously growing set of data records. It is distributed in nature, meaning that there is no master computer holding the entire chain. Rather, the participating nodes have a copy of the chain. It's also ever-growing — data records are only added to the chain.
2. When someone wants to add a transaction to the chain, all the participants in the network will validate it. They do this by applying an algorithm to the transaction to verify its validity. What exactly is understood by "valid" is defined by the Blockchain system and can differ between systems. Then it is up to a majority of the participants to agree that the transaction is valid.

A Blockchain consists of two types of elements:

- Transactions are the actions created by the participants in the system.

- Blocks record these transactions and make sure they are in the correct sequence and have not been tampered with.

BLOCKCHAIN-WORKING

1. A blockchain is decentralized, so there is no single authority that can approve the transactions or set specific rules to have transactions accepted. That means there's a huge amount of trust involved since all the participants in the network have to reach a consensus to accept transactions.
2. Most importantly, it's secure. The database can only be extended and previous records cannot be changed (at least, there's a very high cost if someone wants to alter previous records).
3. A set of approved transactions is then bundled in a block, which gets sent to all the nodes in the network. They, in turn, validate the new block. Each successive block contains a hash, which is a unique fingerprint, of the previous block.
4. Blockchain technology is the missing link to settle privacy and reliability concerns in the Internet of Things. Blockchain technology could perhaps be the silver bullet needed by the IoT industry.
5. It can be used in tracking billions of connected devices, enabling the processing of transactions and coordination between devices; this allows for significant savings for IoT industry manufacturers.

Security and Privacy Requirements

The blockchain-based access control solution has to consider a set of security and functional properties, defined as follows:

- Authenticated access control — the proposed scheme has to ensure an efficient access control to outsourced data, where requesting entities are authenticated.
- Management efficiency — the proposed scheme should offer efficient management processes.
- Privacy through pseudonymity and unlinkability measures — entities' privacy is preserved thanks to the pseudonymity supported by the blockchain and the inability to directly link some data to an entity, or an access session to a requesting user identity. Privacy is strengthened with untraceability in case of one-time blockchain accounts, with one account used per system interaction.
- Auditability — each data owner should have a transparent view over how data are collected, accessed and processed.

MODEL 1 For Blockchain controlled IoT Access

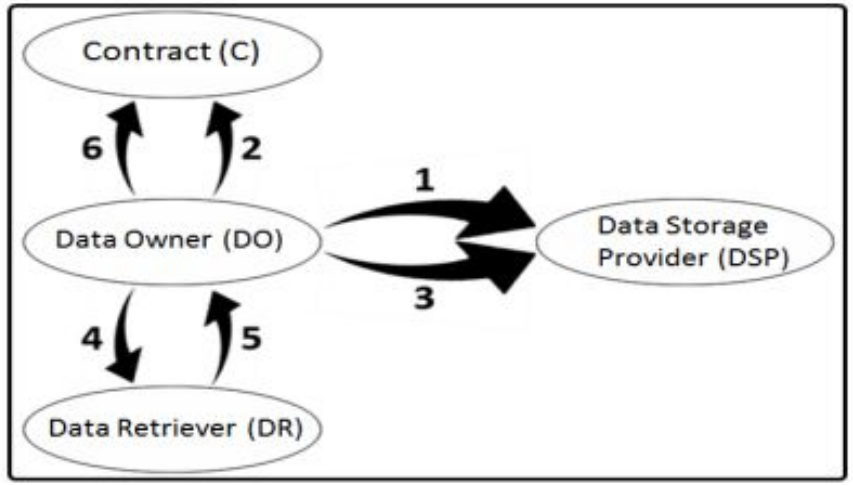


Figure 1: Whitelist Creation Process

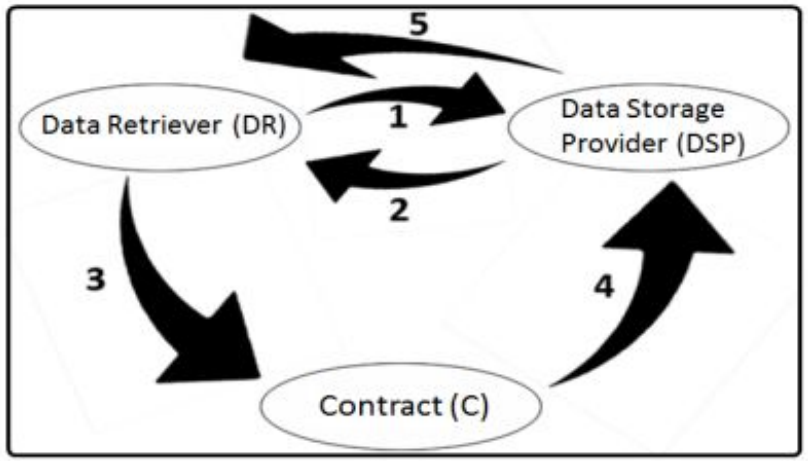
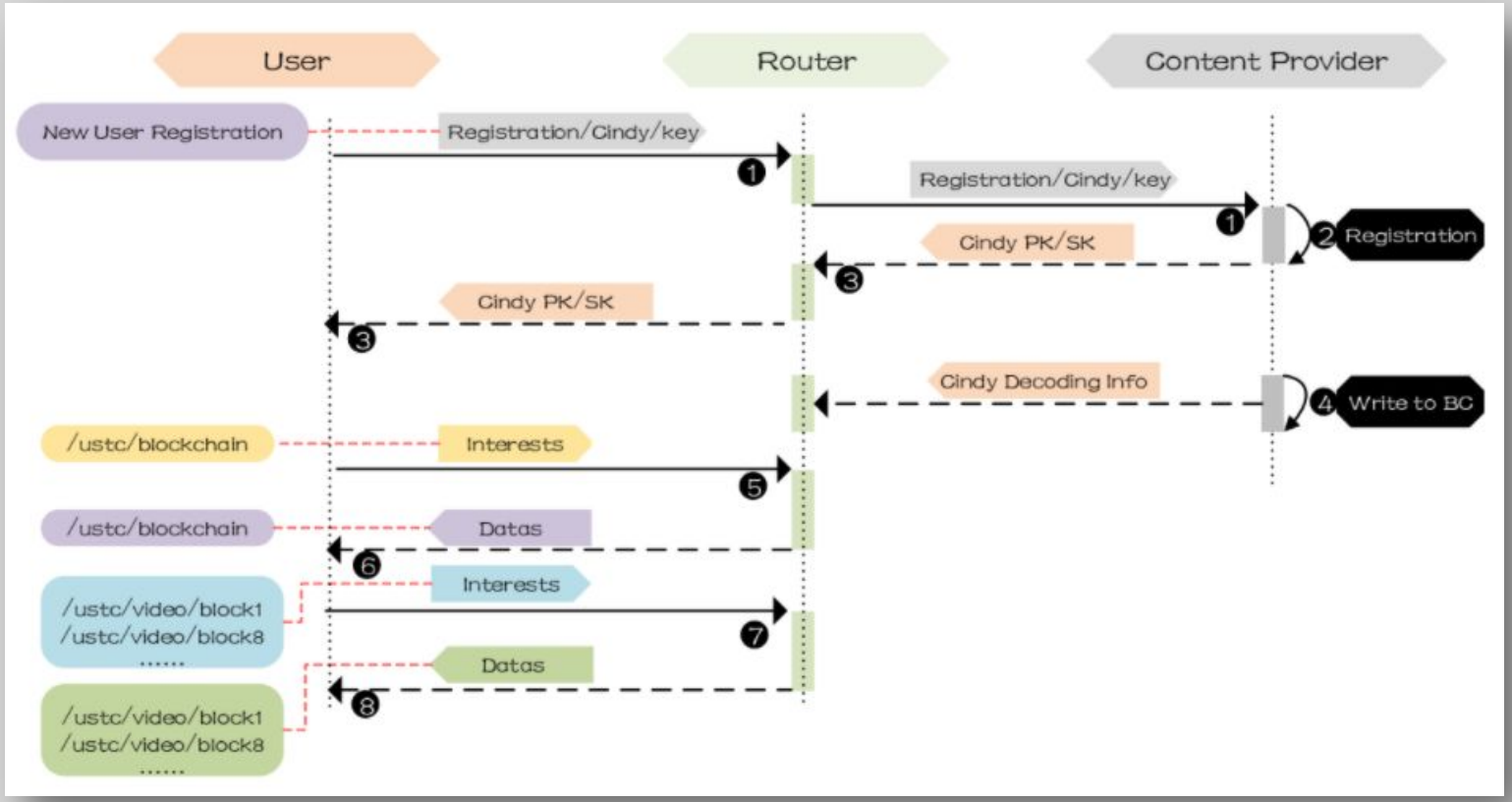


Figure 2: Access to Resources Process

Data owner collects the data from DSP and DR and implements a smart contract.

DR and DSP use the contract to implement access control

MODEL 2:



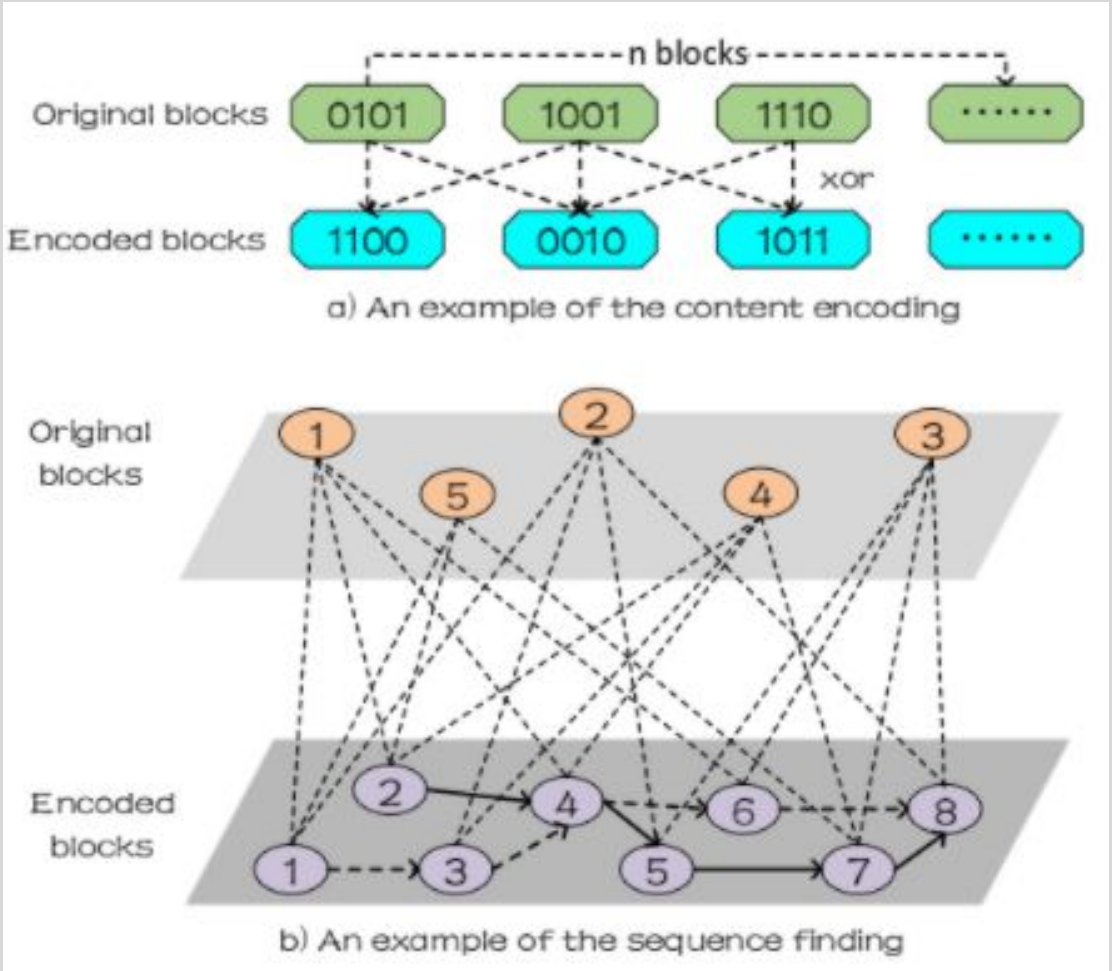
Step 1: The users should register to the CP firstly by sending his username and password through safe channel ❶. The CP verifies user's username and password and generates the public/private key pairs ❷. Then the network returns the public/private key pairs to user ❸.

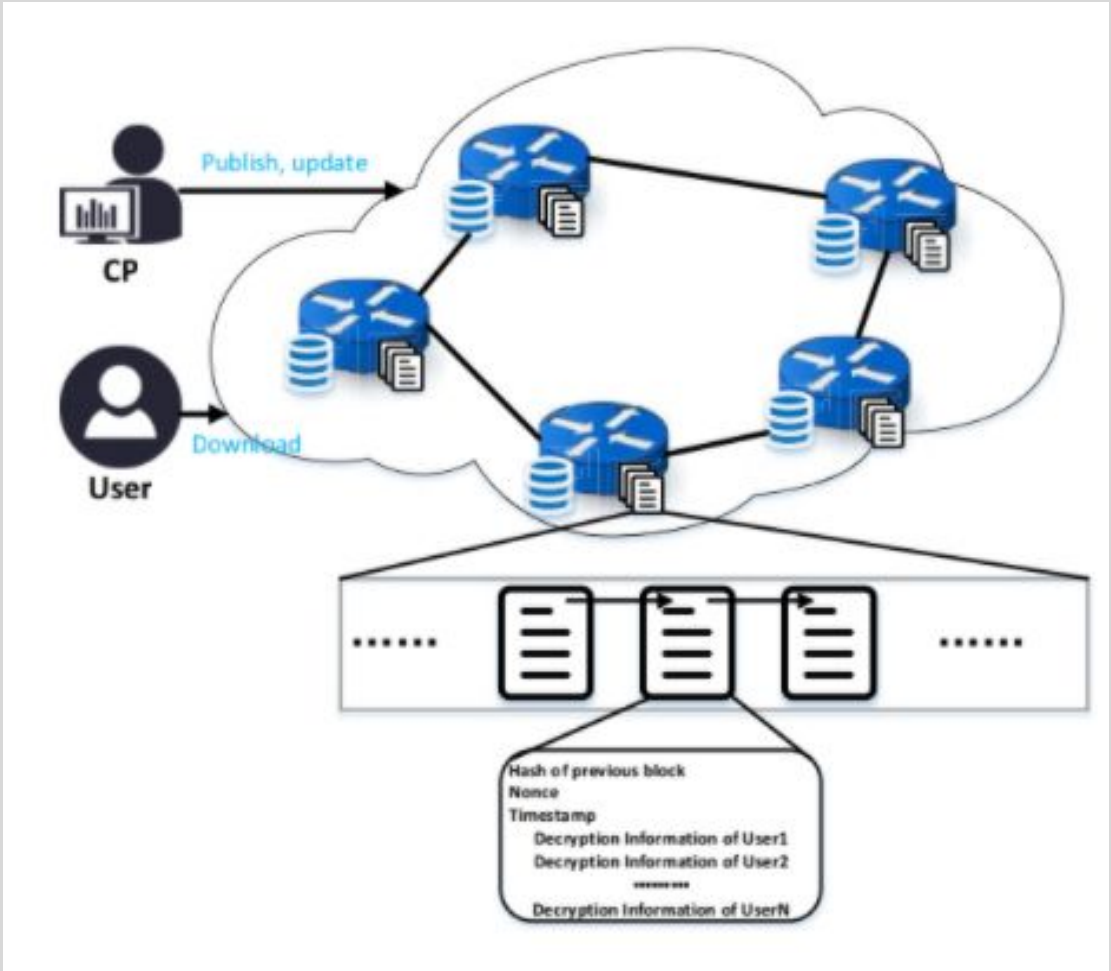
Step 2: The CP generates the data decryption information, which would be encrypted by the user's public key, for this user. The generated information is formed into blocks and appended to the end of the blockchain. Then the CP releases new block to the whole network ❹.

Step 3: After waiting for some time, the user can get the latest blockchain from the ICN by sending an Interest ❺. The router caching the latest blockchain closest him sends the Data to him ❻.

Step 4: The user searches and obtains his own decryption information by his private key from the blockchain. From decryption information, he could get the original content successfully ❼ ❽ (the details would be introduced later).

XOR based Network coding and decoding





Combination of Blockchain and XOR based coding

Creating Block

CP constructs block containing decryption information and sequence. CP signs the block with his private key

Maintaining Blockchain

The whole blockchain is maintained by the CP, i.e., CP publishes the blockchain to the ICN and appends each new block to the end of the chain.

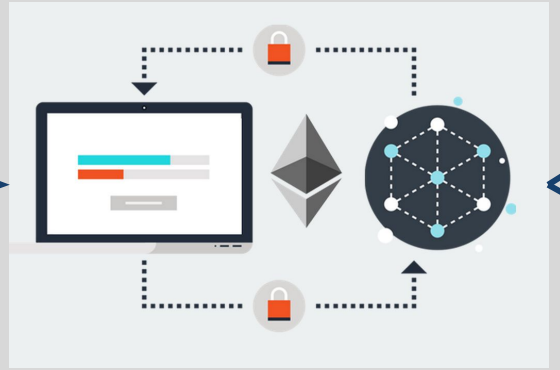
Using Blockchain

Users verify the block information with the CP's public key while downloading the whole blockchain from ICN.

IMPLEMENTATION

Proposed Model

Home automation system



Validation of sender nodes via smart contracts on ethereum blockchain

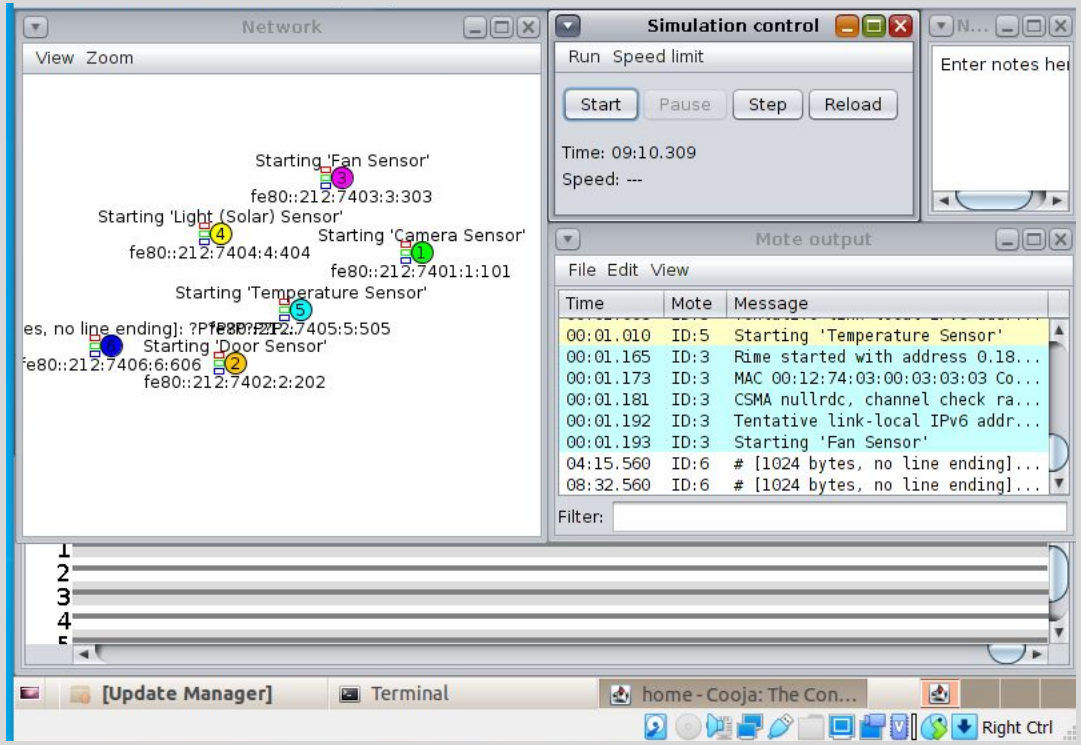
Malicious node detected

stop data transfer

everything safe

continue with the next transaction

COOJA - SIMULATION



11	5.09	202	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
12	4.143	101	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
13	4.218	505	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
14	4.715	303	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
15	5.38	404	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
16	6.701	202	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
17	6.767	101	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
18	7.846	505	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
19	7.321	303	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
20	9.056	404	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
21	9.525	202	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
22	9.577	101	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
23	9.663	505	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
24	10.133	303	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
25	11.96	404	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
26	12.417	202	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
27	12.481	101	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)
28	12.523	505	ff02::1a	ICMPv6	66	RPL Control (DODAG Information Solicitation)

Home Automation

SENDER

1. Solar Sensor
2. Temperature sensor
3. Fan sensor
4. Door sensor
5. Camera sensor

RECEIVER

1. Rpl-border-router

ETHEREUM

1. Ethereum is a decentralized open source blockchain featuring smart contract functionality
2. Ether is the native cryptocurrency token of the Ethereum platform.
3. A "smart contract" is simply a program that runs on the Ethereum blockchain
4. It's a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain.
5. they have a balance and they can send transactions over the network.

SMART CONTACT

1. constructor
2. Contract variables
3. add function
4. getlist function
5. check function

```
pragma solidity ^0.4.0;

contract owner{

    address public owner;
    uint[] whitelist;

    constructor() public{
        owner=msg.sender;
    }
     function add(uint a) public
    {
        if(msg.sender==owner){
           whitelist.push(a);
        }
    }
    function getlist() returns( uint[]){
        return whitelist;
    }
    function check( uint a) public  returns(uint){
        uint i;
        uint flag=0;
        for(i=0; i<whitelist.length; i++){
            if(whitelist[i] == a){
                flag=1;
                break;
            }
        }

        if (flag==0){
            return 0;
        }
        else{
            return 1;
        }

    }
}
```

Deploying the smart contract

```
{
 "config":{
 "homesteadBlock":10
 },
 "nonce": "0x0000000000000042",
 "timestamp": "0x00",
 "parentHash": "0x0000000000000000000000000000000000
 "extraData": "0x00",
 "gasLimit": "0x8000000",
 "difficulty": "0x400",
 "mixhash": "0x0000000000000000000000000000000000000
 "coinbase": "0x3333333333333333333333333333333333
 "alloc": {}
}
```

```
"0x3656585c66e7cd28b988cdb80df3f222a766b1e82b19225907b057dc38164570"
> eth.getBalance(eth.accounts[0])
42000000000000000000
> admin.addPeer( "enode://4fbe7e94dc3a49b2d7334136982152491fdbae18b1a0472390605b8b6facc8433520575f42ccb19df5ea703422d1c7
true
> admin.peers
[{
    caps: ["eth/63", "eth/64", "eth/65"],
    enode: "enode://4fbe7e94dc3a49b2d7334136982152491fdbae18b1a0472390605b8b6facc8433520575f42ccb19df5ea703422d1c7d1c538
    id: "69fc2b8ceefca89424dbcccc5c2c081c6e94c27adfbcc35416bf47b3d1a04d2f",
    name: "Geth/v1.9.24-stable-cc05b050/windows-amd64/go1.15.5",
    network: {
      inbound: false,
      localAddress: "127.0.0.1:53096",
      remoteAddress: "127.0.0.1:30304",
      static: true,
      trusted: false
    },
    protocols: {
      eth: {
        difficulty: 1024,
        head: "0x6650a0ac6c5e805475e7ca48eae5df0e32a2147a154bb2222731c770ddb5c158",
        version: 65
      }
    }
}]
> personal.unlockAccount(eth.accounts[0],"123")
true
> eth.getBalance
function()
> eth.getBalance(eth.accounts[0])
42000000000000000000
> miner.start()
null
> eth.hashrate
0
> miner.setEtherbase(eth.coinbase)
true
> miner.start()
null
> eth.hashrate
28952
> miner.start()
null
> eth.hashrate
279363
> eth.getBalance(eth.accounts[0])
```

Deploying the smart contract

```
from typing import Dict, Any, Union
import json
import csv
from web3 import Web3, HTTPProvider

web3 = Web3(Web3.HTTPProvider(endpoint_uri='http://localhost:8545'))
web3.eth.defaultAccount= web3.eth.accounts[0]
web3.geth.personal.unlock_account(web3.eth.accounts[0], "123")
abi = json.loads('[{"constant":false,"inputs":[{"name":"a","type":"uint256"}],"name":"add",'
bytecode ="6080604052348015610010576000806101000a81548173ffffffffffffffffffffff
deploy = web3.eth.contract(abi=abi, bytecode=bytecode)
tx_hash = deploy.constructor().transact()
print(tx_hash)
tx_receipt = web3.eth.waitForTransactionReceipt(tx_hash,500,0.1)
print(tx_receipt)
```

```
greeter = web3.eth.contract(address=tx_receipt.contractAddress, abi=abi)
with open(r'C:\\Users\vashi\Downloads\homelog.csv','rt')as f:
    data = csv.reader(f)
    i=0

    for row in data:
        if i<5:

            tx_hash=greeter.functions.add(int(row[2])).transact()
            tx_receipt = web3.eth.waitForTransactionReceipt(tx_hash)
            print(tx_hash)


        i+=1
print(greeter.functions.getlist().call())
with open(r'C:\\Users\vashi\Downloads\homelog.csv','rt')as f:
    data = csv.reader(f)

    for row in data:
        print(row[2], greeter.functions.check(int(row[2])).call())
```

with open(r'C:\\Users\vashi\Dow... › for row in data › if i<5

THANK YOU