Design of A Blockchain-Based Lottery System for Smart Cities Applications

Da-Yin Liao, Straight & Up Intelligent Innovations Group Co. San Jose, CA, USA

XueHong Wang, Center of Lottery Study in China Peking University, Beijing, China.

2017 IEEE 3rd International Conference on Collaboration and Internet Computing

중앙대학교 소프트웨어학과
20150249 김용재

Contents

1. Background

2. Smart Contract and Hawk Model

3. Basic Lottery Operation

4. Fairness, Transparency and Privacy

5. Conclusion

Background

- The outcome of Lottery is determined solely by chance, **with no skill** involved.
- Existing lottery games **rely on a centralized black box of services and operations**, which causes distrust on players or citizen in many aspects.

Question about existing Lottery System

- Is the Ticket / Lottery real?
- Is the drawing random and secure?
- Is the winning ticket added after the drawing?
- Is the jackpot winner real?
- Is the distribution of funds fair and real?
- Why are so many corruptions, frauds and scandals of the lottery authority reported from time to time?

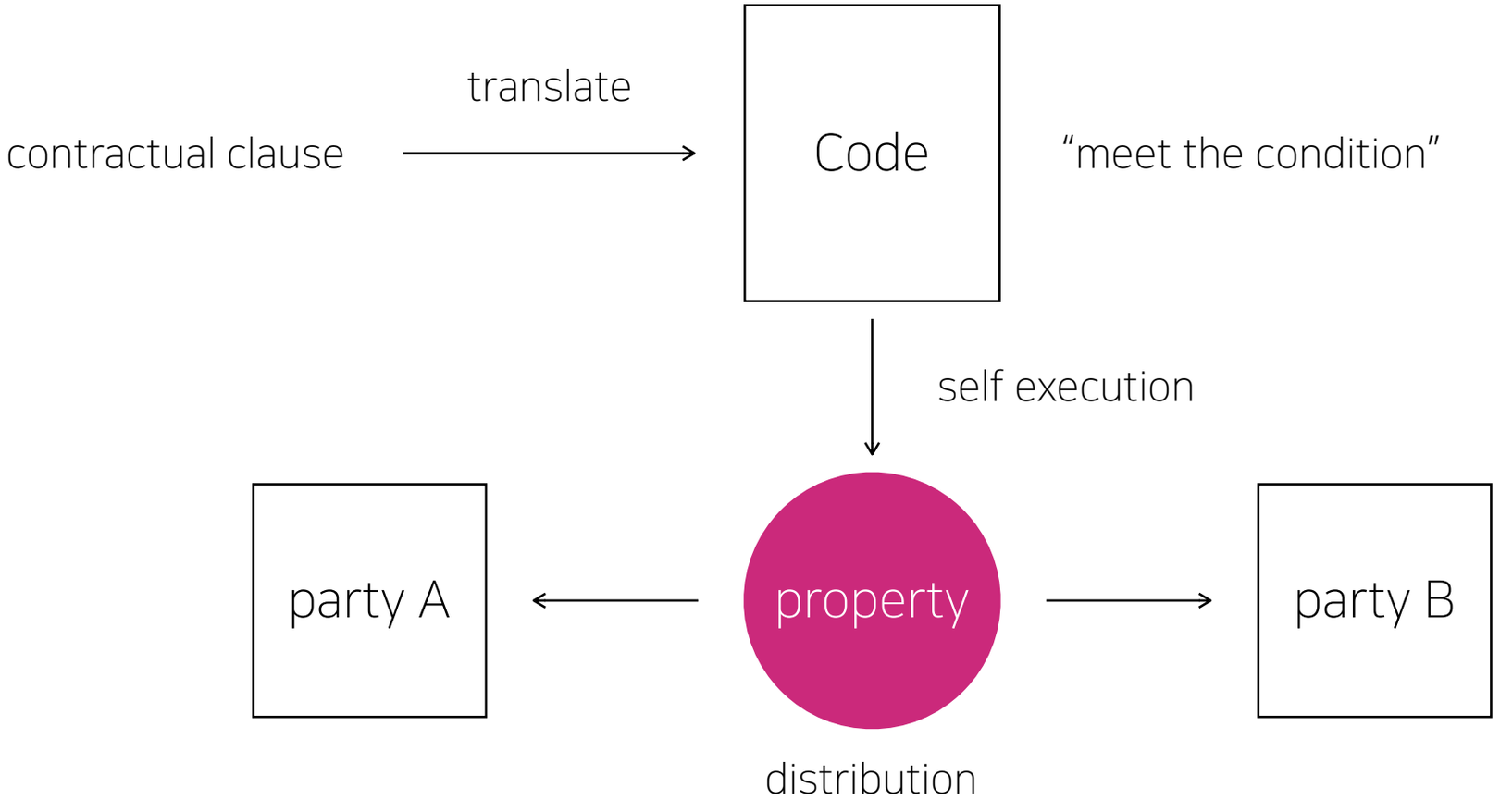
> The emerging **blockchain technology shows a glimpse of solutions** to fairness and transparency issues faced by lottery industries.

Blockchain in FairLotto

- Blockchains introduce new ways of decentralization and delegation of services with autonomous interacting pieces of code, also referred to as **smart contracts.**
- Every interaction with all business processes is strongly cryptographically authenticated.
- We can ensure evaluating payment, ticketing, and payouts in distribution environments and guarantee that operations will be properly enforced by all interesting entities.

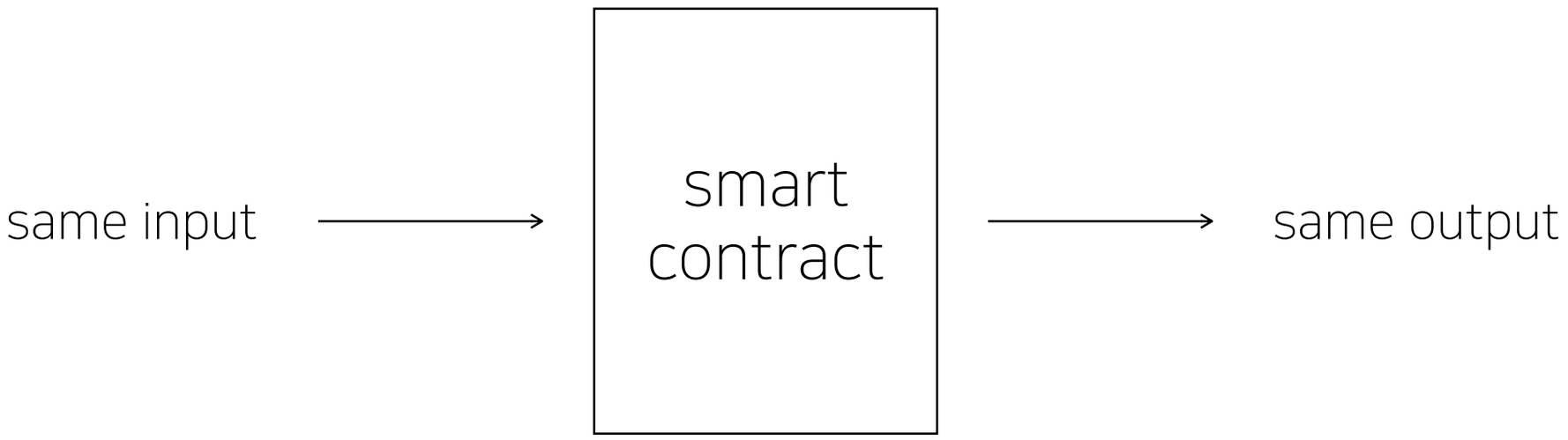
Smart Contract

- It translate contractual clauses into code and embeds them into property that can self-enforce them.
- It can minimize **the need for trusted intermediaries between transacting parties** and the **occurrence of malicious or accidental exception.**



Smart Contract

- Every node in a smart contract-enabled blockchain is running a virtual machine (VM) and the blockchain network acts as a distributed VM.
- A smart contract is deterministic; the same input will always produce the same output.



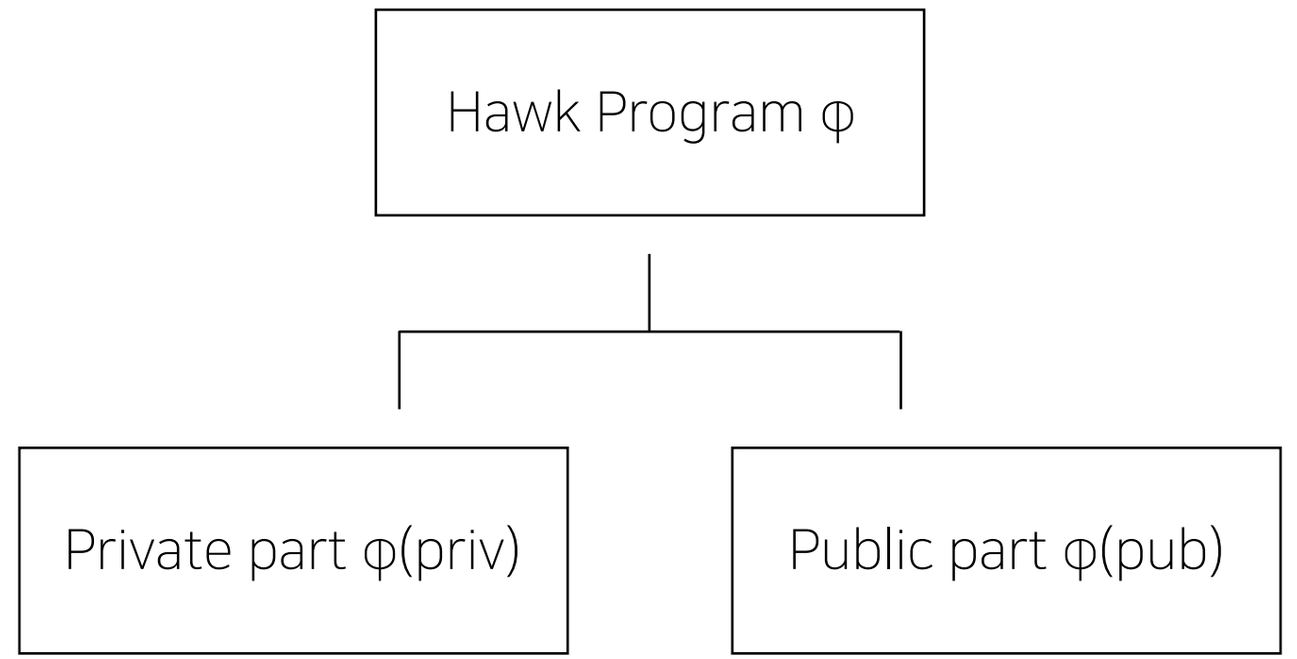
Hawk Model

- As **lack of privacy** is a major hindrance toward the broad adoption of decentralized smart contracts.
- Hawk is decentralized smart contract system that **stores no financial transactions in the clear on the blockchain.**
- It retains transactional privacy in the public.

Manager

- The manager is a special party in facilitating the execution of Hawk.
- In Hawk, the manager can see users's inputs and is trusted not to disclose users' private data.
- The manager is not a trusted third party as those in conventional transactions.
- The manager cannot affect the correct execution of the Hawk program.
- In the context of lottery games, the role of the manager is like the lottery provider, while the users are the lottery players.

Program



The private part takes

input data (selected numbers for lottery)

currency units (BTC)

It **computes to settle the payout distribution** amongst the parties.

In public part, **no private data and money is referred, manipulated, and used.**

Basic Lottery Operation

- By using the proposed core module, all mining processes of the blockchain technology can be executed on Android devices.
- The function of the core module are to
  - create blocks
  - verify the correctness through mining processes
  - link the verified blocks to the chain.
- The core module requires an application to work as a front-end.
- In fact, the core module can be used in various applications, for example, **file sharing, smart contracts,** and **credit member systems.**

Database Function

- The main functions of chain are to **record the private data in the local device** and to **broadcast this chain to all connected devices in the network.**
- The chain includes three different data structures.
    1) account
    2) transaction
    3) block
- In MobiChain, The data structure is followed by the JSON format.
- The private key is stored locally at the mobile node, while the public key is broadcast to other nodes.

Database Function

```
{
 "type": "account",
 "username": /*String of username*/,
 "private_key": /*String of private key's account*/,
 "public_key": /*String of public key's account*/
 "create_date": /*Date time of creating*/
}
```

Account data structure

```
{
 "id": /*Result string after hashing everything inside
     transaction excluding signature */,
 "signature": /*String of the combination between
     transaction and private key's sender*/,
 "timestamp": /*Time of creating*/,
 "transaction": {
  "data": {
   "payload": /*Any string in JSON format*/,
   "uuid": /*String of the unique identification number*/
  },
  "owner": [/*String of public key's sender*/, /*String
     of public key's destination*/]
 }
}
```

Transaction data structure

```
{
 "id": /*Result string after hashing block_number,
     tx_hash, previous_block, and nonce*/,
 "block_number": /*Integer of the current block number*/,
 "votes":[
    {
     "node_pubkey": /*String of public key's miner*/,
     "signature": /*Result string after vote is signed by
      using private key's miner*/,
     "vote": {
      "is_block_valid": /*Boolean that present the block
     valid status*/,
      "previous_block": /*String ID of the previous block
     */,
      "timestamp": /*Time of block creating*/,
      "voting_for_block": /*Same with the ID*/
     }
    }
   ],
 "version": "1",
 "tx_hash": /*Result string after hashing  all
     transactions in the block*/,
 "block": {
    "transactions": [/*list of transactions*/],
    "voters": [/*list public key's of voters*/]
 },
 "nonce": /*Integer of the hashing time. Note that
     hashing is done iteratively until the conditions are
      met*/
}
```

Block data structure

Database Function

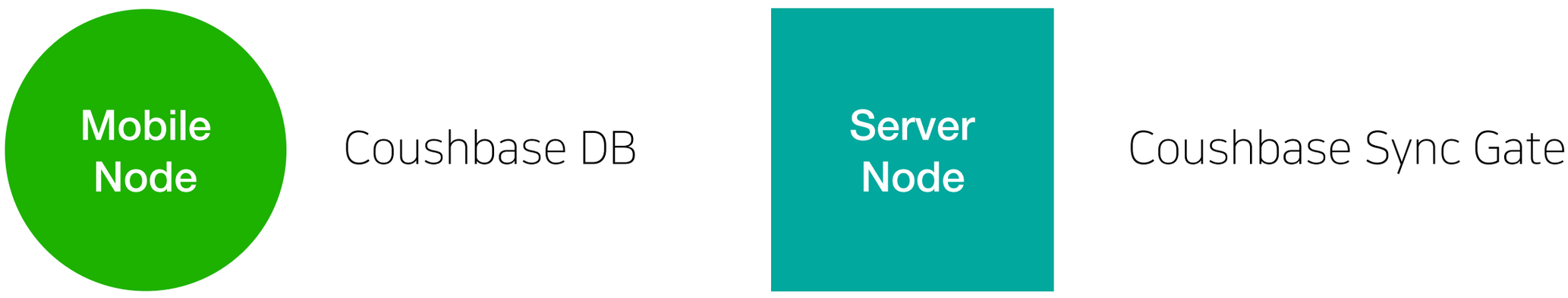
- In the MobiChain system, the blockchain is stored in a database. The database is implemented on both mobile devices and servers.

Mobile Node

- Coushbase Lite database instead of SQLite **(NoSQL)**
- It is **suitable for a real-time system.**

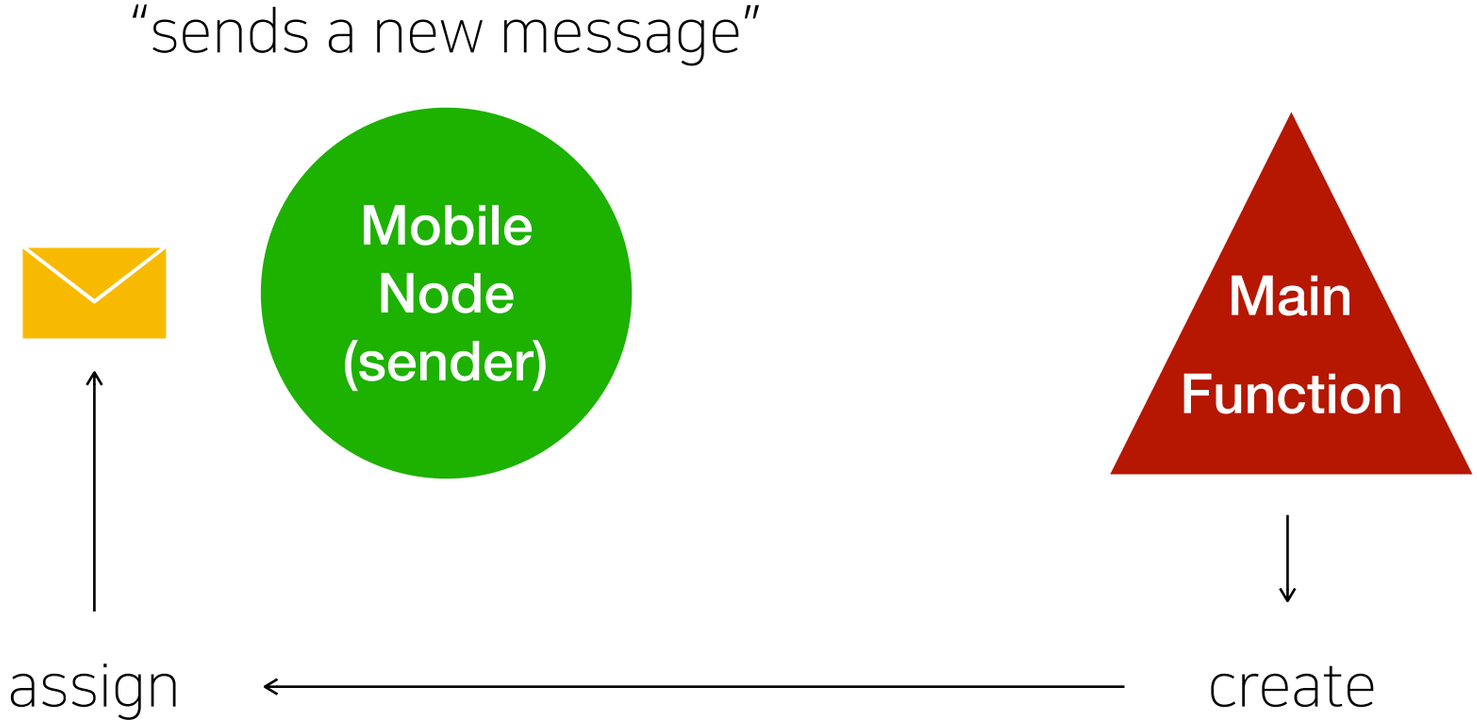
Server Node

- Coushbase Sync Gate is implemented to receive and broadcast data to the devices.



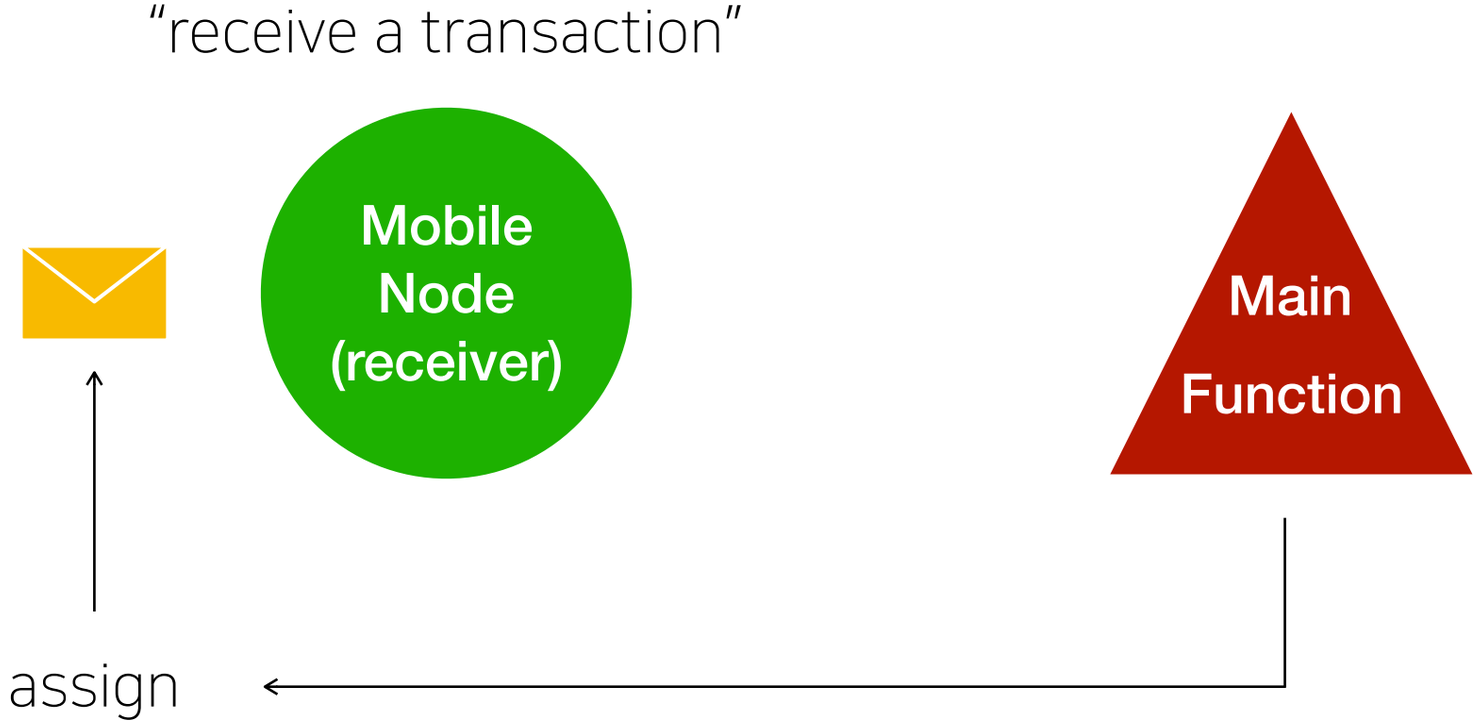
Main Function

1. When a mobile node sends a new message, the Main Function creates a transaction and assigns the transaction to the **sender's backlog**.



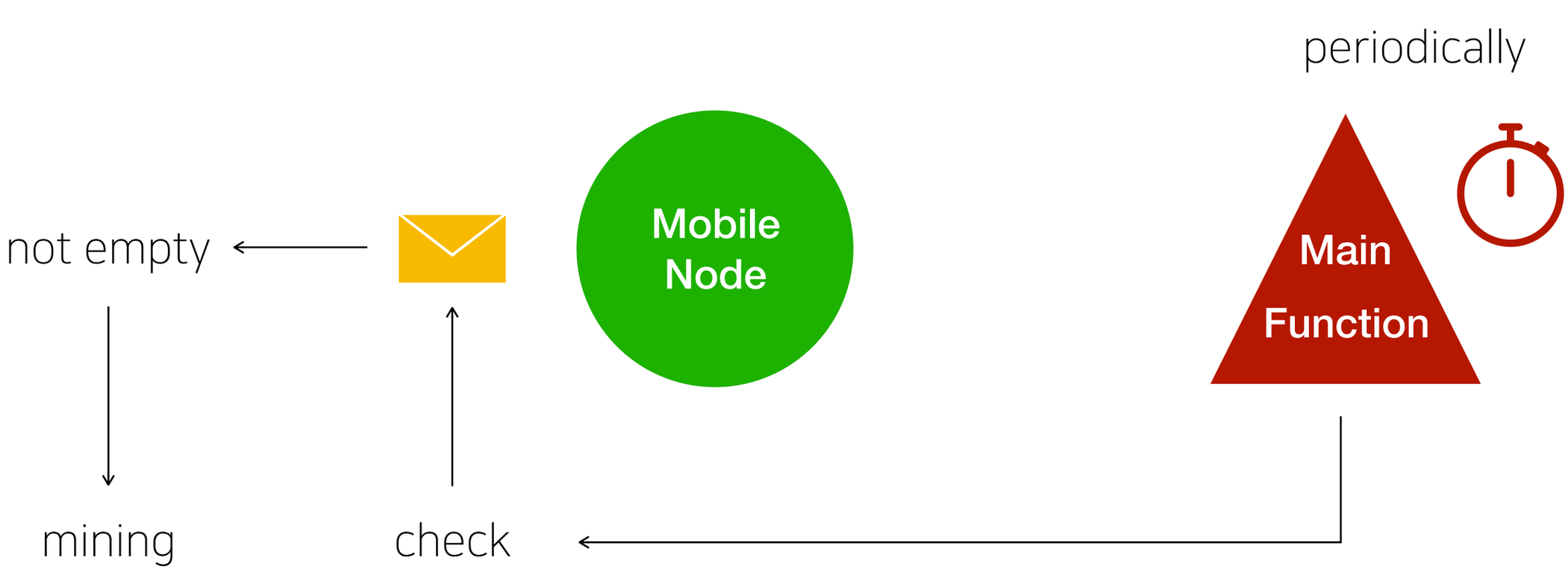
Main Function

2. When a mobile node receives a transaction, the Main Function assigns the new transaction to the **receiver's backlog.**



Main Function

3. The Main Function is executed periodically to **check whether a mobile node's backlog is empty or not.** If the backlog is **not empty**, the main function will **perform the mining process.**



Cryptography Function

- The cryptography function can be separated into three parts.
  - **cryptography-hashes** : SHA3-256 algorithm
  - **key-signature** : ED25519 public key signature system
  - **encode-decode** : Base58 schemes

Fairness, Transparency and Privacy

Experiment environment

- Mobile device (Mobile Node) : Samsung Galaxy Tab S2 8.0 (T715)
- Server (Sync Gateway) : Workstation with Intel Xeon CPU E5-1630
- Total energy consumption on the mobile device was measured by VideoOptimizer program
- If the nonce is not specific in an experiment, the nonce is set to be zero.

Memory Utilization

- The content of each transaction is fixed at 20 characters.

Test Case

1) 1 transaction per block
2) 3 transaction per block
3) 6 transaction per block

Test Result

- If we **increase the number of transactions in each block**, **the memory utilization can be reduced** remarkably.
- If we store 3 or 6 transaction in one block, the memory utilization can be reduced by 33% or 55%, respectively.

$$Memory\ \ Utilization\ =\ c_b\ +\ c_tT\ +\ c_dD$$

T = number of transactions in one block

D = number of digits of ***block_number***

Memory Utilization

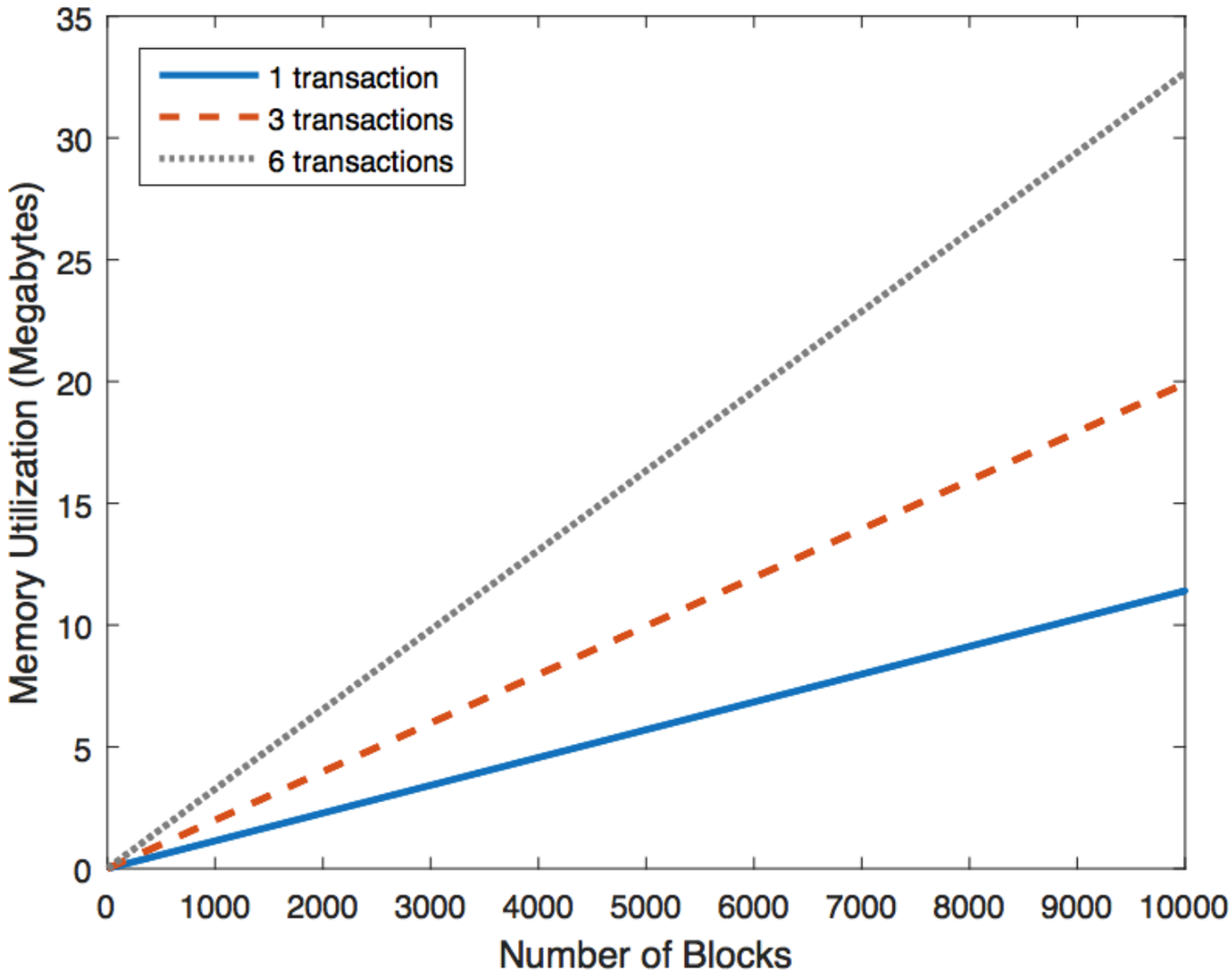


Figure 6: The memory utilization when the number of blocks increases.

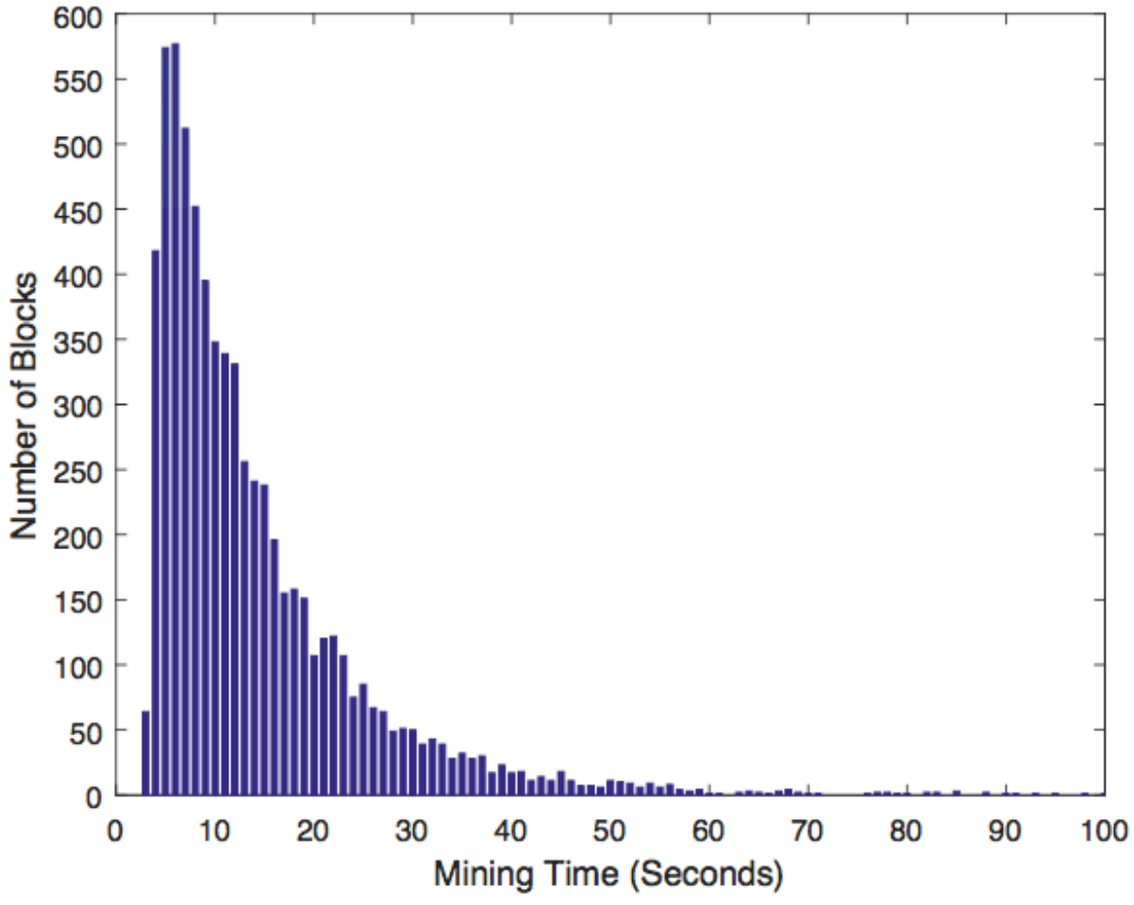
Proof-of-Work Process

- The hash process is executed iteratively until **the first three digits of the hash value equal zero.**

Test Condition

- 7,156 blocks were used for this test.
- These blocks were mined using mobile device.

Proof-of-Work Process



Test Result

- The test result is filtered to show only 0 to 100 seconds.
- **88.06% of blocks** need to **use 3 to 30 secs** to perform the PoW process.
- Only 4.79% of blocks perform longer than 100 secs.
- At the peak points, 23.23% of total blocks use 5 to 7 secs.
- 803 hashing iterations are executed per second.
- Peak points use around 4,015 to 5,621 hashing iterations before meeting the condition.

Chain Verification Process

- The execution time and energy consumption are measured from the beginning of the chain verification processes **until the end of this process.**
- **For multiple threads**, the measurement is from the beginning **until the last thread completes.**

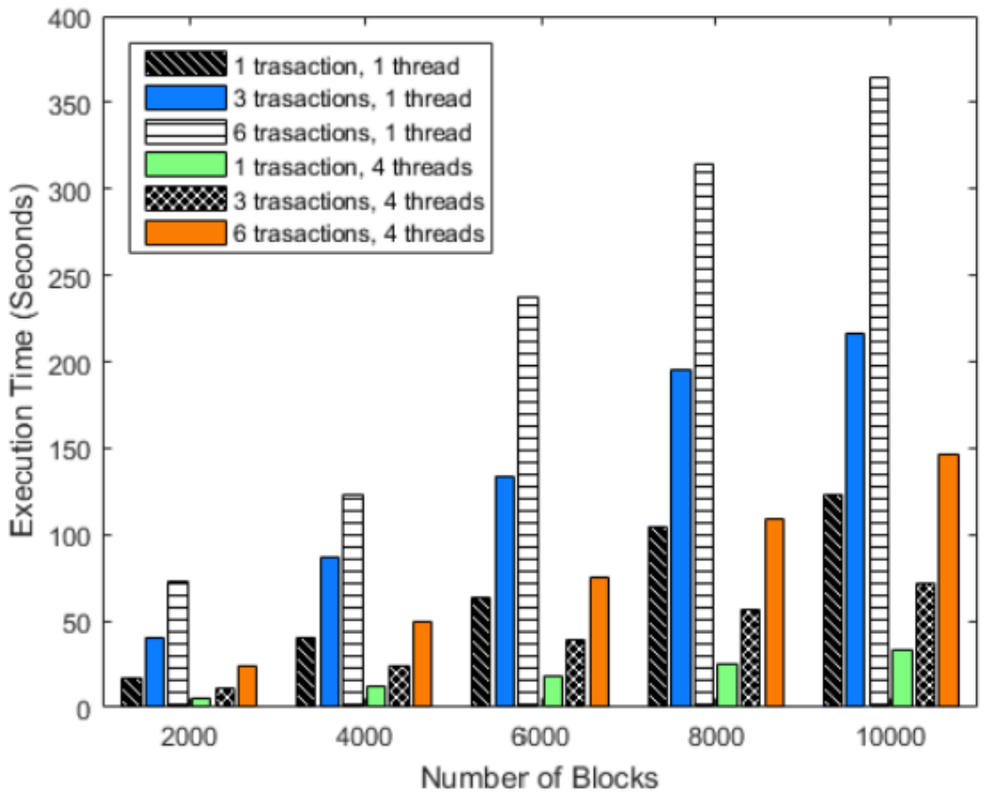
Test Case

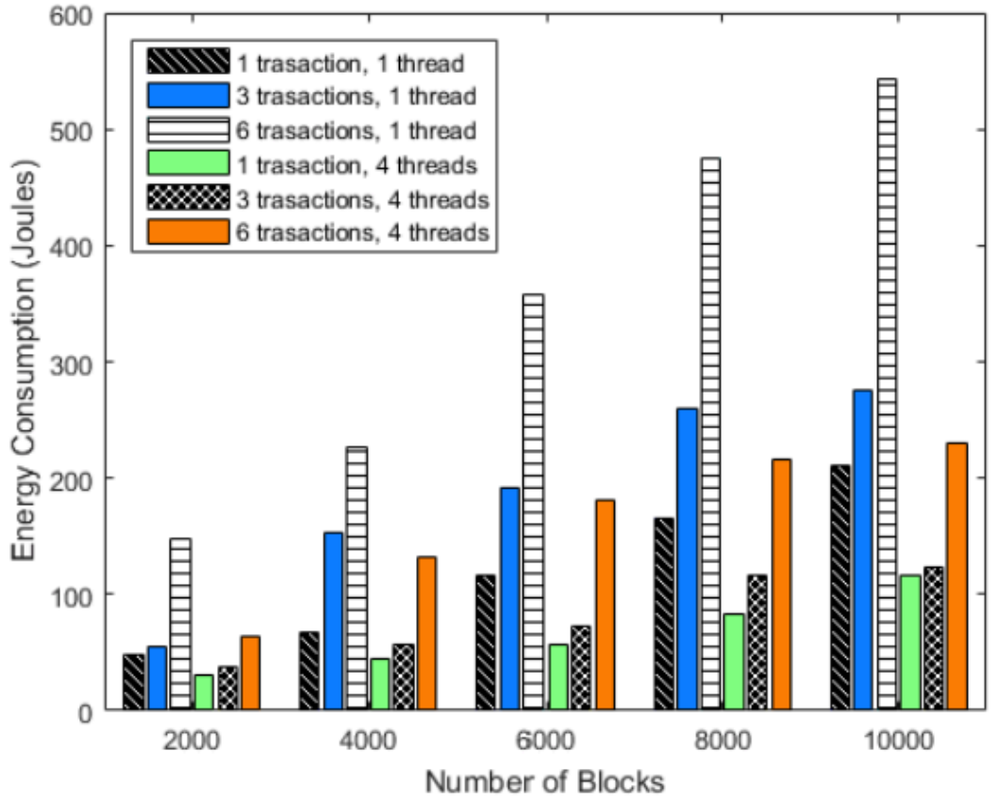
1) 1 thread
   - 1 transaction per block
   - 3 transaction per block
   - 6 transaction per block
2) 4 thread
   - same as above

Chain Verification Process

Test Result

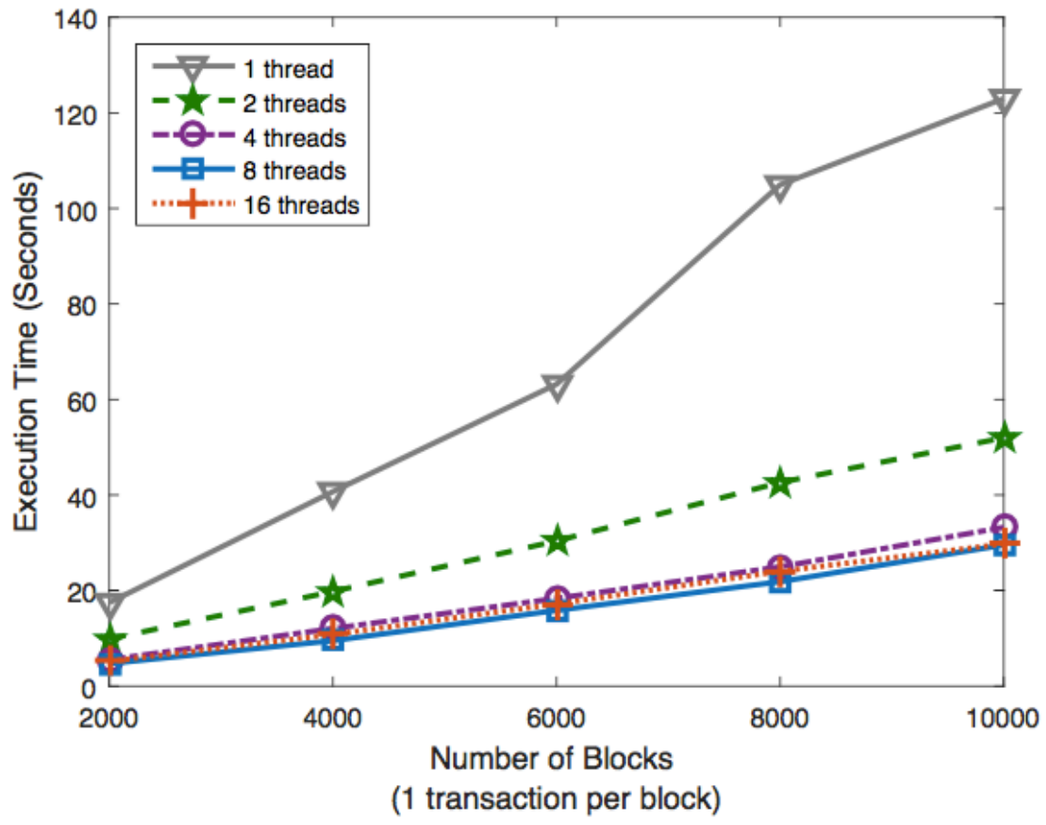
- As the **number of blocks in the chain increases, the execution time and energy consumption increase** accordingly.
- **Transaction are grouped together** in a block -> **faster / less energy.**
- In practice, having more number of transactions in a block can cause more delay if the transactions are generated randomly.





Chain Verification Process

- When we increase the number of threads, the execution time is not always reduced.
- If we keep increasing the number of threads, the execution time reduces insignificantly.
- Android device support 8 processing cores and each core has one thread.



Conclusion

- MobiChain, a new m-commerce application using blockchain technology for data security.
- It can perform mining process on mobile devices through.
- Experiments show that blockchain tech is a practical solution for
  - security
  - efficiency
  - scalability
  - processing…
- MobiChain system will be extended for **offline mining** and **propose data synchronization algorithms when mobile nodes are reconnected** to the network.