Performance Analysis and Application of Mobile Blockchain

Kongrath Suankaewmanee[1], Dinh Thai Hoang[1], Dusit Niyato[1], Suttinee Sawadsitang[1], Ping Wang[1], and Zhu Han[2]
[1] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2] Department of Electrical and Computer Engineering, University of Houston, USA

***Abstract*—Mobile security has become more and more important due to the boom of mobile commerce (m-commerce). However, the development of m-commerce is facing many challenges regarding data security problems. Recently, blockchain has been introduced as an effective security solution deployed successfully in many applications in practice, such as, Bitcoin, cloud computing, and Internet-of-Things. In this paper, we introduce a new m-commerce application using blockchain technology, namely, MobiChain, to secure transactions in the m-commerce. Especially, in the MobiChain application, the mining processes can be executed efficiently on mobile devices using our proposed Android core module. Through real experiments, we evaluate the performance of the proposed model and show that blockchain will be an efficient security solution for future m-commerce.**

***Index Terms*—Blockchain, mobile commerce, Android application, mobile security.**

I. Introduction

The Technavio's market research analyst predicted that the global mobile commerce (m-commerce) market will grow at a compound annual growth rate of more than 27% by 2020 [8]. M-commerce is gaining acceptance globally due to its variety of advantages, such as, offering multiple payment options to consumers, streamlining the procurement processes, and proving better price choices due to stiff market competition. However, the key concern about security for m-commerce needs to be addressed for the continued growth of m-commerce.

Recently, blockchain has been introduced as an effective technology for data security problems. Furthermore, it has been implemented successfully in many applications, e.g., Bitcoin Wallet [1], Ethereum [11], and Internet-of-Things (IoT) [3]. Generally, blockchain is a distributed database that is replicated and shared among members of a network [2]. With blockchain, when a transaction is created, it will be verified parallelly and transparently by some nodes in the network through mining processes. After that, transactions are grouped into blocks, and the links between blocks and their content are protected by cryptography and cannot be forged. Once entered into a blockchain, transactions cannot be erased. Thus, a blockchain contains an accurate, time-stamped and verifiable record of every transaction, and hence the network does not need a central authority. As a result, the blockchain technology is popularly used in systems requiring high security and transparency, such as, Bitcoin [9] and smart digital contract Ethereum [11].

In blockchain technology, the mining process plays a crucial role in verifying and adding transaction records to the public ledger, i.e., the blockchain. In a mining process, the miner, i.e., the node taking the responsibility for mining a transaction, is required to verify the transaction and solve the proof-of-work problem in order to find a new hash for the new incoming block to store the verified transaction. This process is complicated and usually executed on powerful devices with high computational capacities and energy supply, e.g., servers and computers. Therefore, blockchain technology has not been adopted widely in m-commerce. However, the breakthrough of mobile technologies together with the improvement of hashing algorithms recently allow the mining processes to be able to implement on mobile devices efficiently, thereby opening new opportunities for the development of blockchain in m-commerce.

In this paper, we first introduce a new m-commerce application using blockchain technology, namely, MobiChain, to secure transactions. In MobiChain, we develop a Mobile Blockchain Application Programming Interface (API) which allows the mining processes to be performed on mobile devices effectively. We then conduct real experiments to evaluate the performance of the proposed module in terms of computation time, energy consumption, and memory utilization. Through experimental results, we show that blockchain technology will be an efficient security application for future m-commerce.

II. Related Work

The success of Bitcoin has motivated researchers to investigate and adopt the blockchain technology to many areas, such as, smart contract [11], finance [9], human resource management [6], supply chain [7], and IoT [3][10]. For example, in [3], the authors proposed a lightweight installation of the blockchain technology for a smart IoT house. In each house, multiple IoT devices (e.g., smart phones, personal desktops, and sensors) are connected to the same network. Under the

978-15386-3652-7/18/$31.00 ©2018 IEEE

proposed model, only authorized users can access and control devices in the house, and messages received by the authorized users are protected and unable to be modified by any malicious users.

The authors in [4] introduced a new decentralized personal data management platform that allows users to control their own data, thereby preventing the data breach from a third-party. The platform combines blockchain and off-blockchain storage to construct a personal data management framework. As such, the blockchain will recognize the users as the owners of their personal data, and thus the users are not required to trust any third-party. The authors in [5] then extended [4] by improving the encryption algorithm and introducing a new method, called, Proof-of-Credibility Score (PCS), for mining processes. Different from [4] where the trust score of the node is accumulated on how much good actions a node took, the proposed PCS method utilizes the connection between nodes to calculate the credibility score. The numerical results then showed that with the proposed hybrid blockchain of credibility score, the system security can be improved.

As the blockchain application keeps growing, the scalability is a big challenge. The authors in [19] proposed a BigchainDB system using NoSQL database format to address this problem. The blockchain pipelining is adopted to allow the system to be scalable when adding blockchain-like characteristics to the distributed database. In m-commerce, transactions should be mined at mobile nodes to support direct peer-to-peer data exchange and sharing. This is especially important for the case when mobile nodes have no (or limited access to) Internet connection. Currently, there are some Android applications which was developed to perform mining processes on mobile devices for Bitcoin application, such as, Easy Miner [12] and LTC and Scrypt Miner PRO [13]. However, they are still demo versions and have not yet completed. More importantly, the platform to support general blockchain operations is still missing because Bitcoin applications are used for digital currency applications only. This is the motivation to introduce the MobiChain in this paper.

III. System Model and Assumptions

The proposed MobiChain system is illustrated in Fig. 1, which consists of multiple mobile devices and multiple server nodes. Each mobile node has a backlog or queue to store pending tasks. The mobile node is connected to a server node either via the Internet or a local direct connection. All server nodes are installed *Sync Gateway* [18] to broadcast data, and they are always connected to other server nodes via the Internet. In this case, we consider a message passing scenario, e.g., *Mobile Node A* sends a message to *Mobile Node C*. This message can be a data transfer request or a radio resource sharing request, which needs to be verified. This message is referred to as a transaction. We assume that all the mobile nodes are installed the MobiChain application with functions shown in Section IV, and they have already registered user accounts. These accounts are to provide public and private keys which are unique to the users. For notational convenience, let *M(X)* represent *Mobile Node X* ($\forall X \in \{A, B, C, D, E\}$), and *S(Y)* represent *Server Node Y* ($\forall Y \in \{F, G, H\}$).

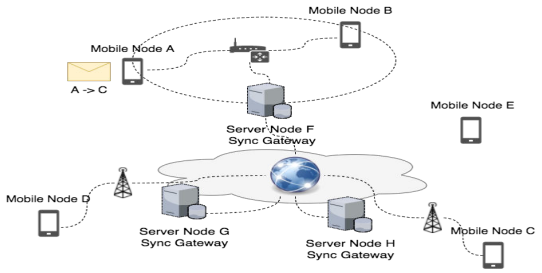

Figure 1: The MobiChain system model.

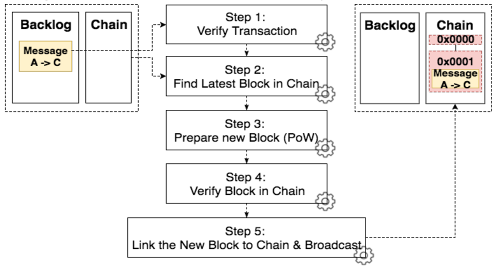

Figure 2: The mining process in the MobiChain system

A. Message Passing Process

We consider a message passing process from *M(A)* to *M(C)*. This process includes the following steps:

- **Step 1:** The sender, i.e., *M(A)*, creates a transaction. In general, a transaction consists of three parts, i.e., text message, sender's information, and destination's information. After that, the sender puts the transaction into the sender's backlog.
- **Step 2:** The sender uploads the transaction to its connected server, i.e., *S(F)*.
- **Step 3:** The Sync Gateway, i.e., *S(F)*, broadcasts the transaction to every node in the network. In Fig. 1, *M(B)*, *M(C)*, and *M(D)* have the same transaction in their backlogs as they are connected to the network. Note that *M(E)* does not have the same transaction because it is currently not connected to the network.

B. Mining Process

After receiving the transaction from *M(A)*, we assume that nodes *M(B)* and *M(C)* are dedicated or they want to mine this transaction, e.g., they will receive some rewards to mine this transaction. These nodes will then start the mining process. The mining process includes five steps as illustrated in Fig. 2. The detail is as follows:

- **Step 1:** A miner, i.e., *M(B)* or *M(C)*, queries the transaction from its backlog, and verifies this transaction by

checking whether the transaction is modified or not, and whether the transaction exists in the blockchain or not. If the transaction is not modified and does not exist in the blockchain, the miner performs Step 2. Otherwise, the mining process is terminated and the miner reports the problem to all other nodes in the network.
- **Step 2:** The miner finds the latest block in the chain, and stores the identification number (ID) of the latest block for the future use. The latest block may be referred to as the previous block.
- **Step 3:** This step is referred to as Proof-of-Work (PoW) process. In the PoW process, the miner will create a new block by hashing the information iteratively. The information includes (i) the IDs of the previous block and the created block, (ii) the block number (sequence number) of the created block, (iii) the verified transaction, (iv) the date/time stamp, and (v) the signature of the miner. We use `SHA3_256` algorithm [14] for the hashing process.
- **Step 4:** This step is to verify all information, e.g., hash ID and transaction information, in the chain. After creating the block ID, the miner verifies all the existing blocks in the chain. This process guarantees that the information of all blocks in the chain has not been changed.
- **Step 5:** The miner needs to check whether the created block has been added to the chain by other mobile nodes or not. If not, the miner will add the created block to the chain.

Note that in Fig. 2, the chain is empty. Hence, the identification of the previous block is zero, and the transaction that contains the message from *M(A)* to *M(C)* is pushed to a new block. The new block is then linked to the previous block in the chain.

IV. Implementation

The proposed core module is a Mobile Blockchain Application Programming Interface (API) for the Android environment. By using the proposed core module, all mining processes of the blockchain technology can be executed on Android devices. The functions of the core module are to create blocks, verify the correctness through mining processes, and link the verified blocks to the chain. Therefore, the core module requires an application to work as a front-end. In the previous section, the core module is implemented as a message passing application. In fact, the core module can be used in various applications, for example, file sharing, smart contracts, and credit member systems. In this section, three functions of the core module are discussed in detail.

A. Database Function

The main functions of the chain, i.e., the database, 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, i.e., account, transaction, and block structures, as presented in Fig. 3, Fig. 4, and Fig. 5, respectively. Note that the data structure is followed by the JavaScript Object Notation (JSON) format, and the explanation of each element is given in the comments. Moreover, a user has a set of public and private keys. The private key is stored locally at the mobile node, while the public key is broadcast to other mobile nodes as shown in the account data structure in Fig. 3. In the MobiChain system, the blockchain is stored in a database. The database is implemented on both mobile devices and servers. On the mobile devices, we use Coushbase Lite [17] instead of the SQLite database. On the server nodes, Coushbase Sync Gateway [18] is implemented to receive and broadcast data to the devices.

```
{
”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*/
}
```

Figure 3: Account data structure followed by JSON format.

```
{
”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*/]
}
}
```

Figure 4: Transaction data structure followed by JSON format.

```
{
”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*/
}
```

Figure 5: Block data structure followed by JSON format.

B. Main Function

The Main Function is executed when one of the three following scenarios happens.

- When a mobile node sends a new message, the Main Function creates a transaction and assigns the transaction to the sender’s backlog.
- When a mobile node receives a transaction, the Main Function assigns the new transaction to the receiver’s backlog.
- 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 as discussed in Section III-B.

In short, the Main Function will be used to create transactions and blocks, assign transactions into the backlog, and verify transactions in the backlog and blocks in the chain.

C. *Cryptography Function*

The cryptography function can be separated into three parts, i.e., cryptography-hashes, key-signature, and encode-decode. In the cryptography-hashes, the SHA3-256 algorithm can be used [14]. When a user registers for an account in the MobiChain system, a pair of private and public keys is generated by ED25519 public key signature system [15], which will be used in signing and verifying a signature. For encode and decode processes, the Base58 schemes can be used [16].

V. Performance Evaluation

All experiments presented in this section were performed on a Samsung Galaxy Tab S2 8.0 (T715), i.e., a mobile node, and a workstation with Intel Xeon CPU E5-1630, i.e., a Sync Gateway. The total energy consumption on the mobile device was measured by VideoOptimizer program [20]. The Mobilechain system and the core module were implemented by the Android Studio and Software Development Kits (SDK) tools. If the nonce is not specific in an experiment, the nonce is set to be zero. The nonce value is set in Fig. 5. In this section, we analyze memory utilization according to the number of blocks in the chain and the mining process performance. As shown in Fig. 2, the mining process has five steps. Most of the mining process computation is on Step 3 (chain verification) and Step 4 (Proof-of-Work), while those of Step 1, Step 2 and Step 5 are negligible. Thus, only the chain verification and the Proof-of-Work processes are analyzed in this section.

A. *Memory Utilization*

Memory utilization of the proposed MobiChain is shown in Fig. 7 under three different sizes of blocks, i.e., one transaction per block, three transactions per block, and six transactions per block. Here, the content of each transaction is fixed at 20 characters. In Fig. 7, if we increase the number of transactions in each block, the memory utilization can be reduced remarkably.

B. *Proof-of-Work Process*

In this experiment, the hash process is executed iteratively until the first three digits of the hash value equal zero. At each iteration, the hash value is combined with nonce, which will be increased by one in each iteration. The execution time of Proof-of-Work process is random due to the hash condition. The number of hash iterations until the condition is met is found to be random.

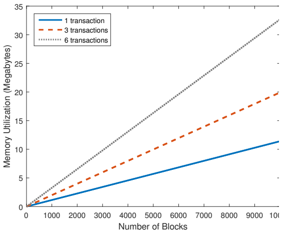

Figure 6: The memory utilization when the number of blocks increases.

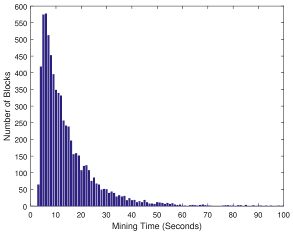

Figure 7: The distribution of mining time v.s. number of blocks.

In our experiment, we create 7,156 blocks and use the mobile device to mine these blocks. In this case, it took 3.5 days to execute the Proof-of-Work processes for all 7,156 blocks. The histogram of these blocks is shown in Fig. 6, which can be expressed as a gamma long tail distribution [21]. The experiment is filtered to show only 0 to 100 seconds. According to Fig. 6, 88.06% of blocks need to use 3 to 30 seconds to perform the Proof-of-Work process, while only 4.79% perform longer than 100 seconds. At the peak points, 23.23% of the total blocks use 5 to 7 seconds. In our experiments, 803 hashing iterations are executed per second, and thus the peak points use around $4,015$ to $5,621$ hashing iterations before meeting the condition.

C. *Chain Verification Process*

The execution time and energy consumption of the chain verification process are presented in Fig. 8 and Fig. 9, respectively. The execution time and energy consumption are measured from the beginning of the chain verification process until the end of this process. For multiple threads, the measurement is from the beginning until the last thread completes. Two types of experiments were conducted for both one-thread and four-thread scenarios. Each of them is performed with one, three, and six transactions per block. Each block contains 20 random characters. As expected, as the number of blocks in the chain increases, the execution time and energy consumption increase accordingly.

One important finding is that the chain verification process is executed faster and consumes less energy when transactions are grouped together in a block. In particular, as shown in Fig. 8, when there are 3 or 6 transactions grouped in one block, the total execution time per block can be reduced approximately 30% or 40%, respectively. When the execution time is reduced, the energy consumption will be decreased accordingly. However, in practice, having more number of transactions in a block can cause more delay if the transactions are generated randomly. Thus, the tradeoff for the number of transactions per block is worth

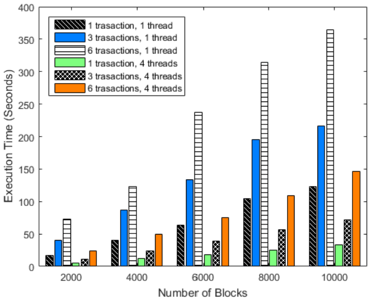

Figure 8: Execution time of the chain verification process.

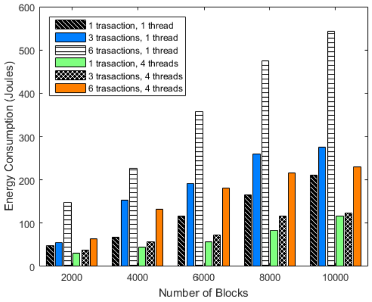

Figure 9: Energy consumption of the chain verification process

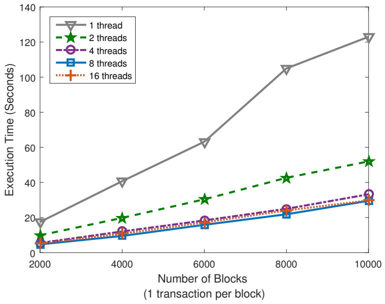

Figure 10: The execution time of the chain verification process by using multiple threads.

investigating.

In Fig. 10, we show the execution time of one transaction per block with different threads. Clearly, as the number of blocks increases, the execution time rises. However, interestingly, when we increase the number of threads, the execution time is not always reduced. Specifically, when we increase the number of threads from 1 to 2 or from 2 to 4, the execution time is reduced approximately twice. However, if we keep increasing the number of threads, the execution time reduces insignificantly. Here, the best execution time is achieved when the number of threads is 8 because the Android device support 8 processing cores and each core has one thread.

VI. Conclusion

In this paper, we have introduced MobiChain, a new m-commerce application using blockchain technology for data security. Different from most of current works, in our MobiChain system, mining processes can be performed on mobile devices through the proposed Android core module. This application provides useful functions for mobile transactions and promotes the development of m-commerce. Through experiment results, we have shown that blockchain technology is a practical solution for mobile devices to achieve security, efficiency, and scalability of data collection, processing, and sharing.

Acknowledgement

This work was supported in part by Singapore MOE Tier 1 under Grant RG33/16 and Grant RG18/13, in part by MOE Tier 2 under Grant MOE2014-T2-2-015 ARC4/15 and Grant MOE2013-T2-2-070 ARC16/14, NRF2015-NRF-ISF001-2277, US NSF CNS-1717454, CNS- 1731424, CNS-1702850, CNS-1646607, ECCS-1547201, CMMI- 1434789, CNS-1443917, and ECCS-1405121.

References

[1] T. Bamert, et al., "BlueWallet: The secure Bitcoin wallet," in *International Workshop on Security and Trust Management*, pp. 65-80, Wroclaw, Poland, Sept. 2014.

[2] I.-C. Lin and T.-C. Liao. "A survey of blockchain security issues and challenges," *International Journal of Network Security*, vol. 19, no.5, pp. 653-659, Sept. 2017.

[3] A. Dorri, et al., "Blockchain for IoT security and privacy: The case study of a smart home," in *IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 618-623, Hawaii, USA, Mar. 2017.

[4] G. Zyskind, O. Nathan, and A.S. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *IEEE Security and Privacy Workshops*, pp. 180-184, San Jose, USA, May 2015.

[5] D. Fu and L. Fang, "Blockchain-based trusted computing in social network," in *IEEE International Conference on Computer and Communications*, pp. 19-22, Chengdu, China, Oct. 2016.

[6] X. Wang, L. Feng, H. Zhang, C. Lyu, L. Wang, and Y. You, "Human resource information management model based on blockchain technology," in *IEEE Symposium on Service-Oriented System Engineering*, pp. 168-173, San Francisco, USA, Apr. 2017.

[7] H. M. Kim and M.Laskowski, "Towards an ontology-driven blockchain design for supply chain provenance," Open-Access Online Library, Aug. 2016.

[8] Global M-commerce Market 2016-2020. Technavio's Report.

[9] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system", White Paper, 2009. Available at: https://bitcoin.org/bitcoin.pdf

[10] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockChain for IoT," in *IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation*, pp. 173-178, Pittsburgh, USA, Apr. 2017.

[11] V. Buterin, "Ethereum: A next-generation cryptocurrency and decentralized application platform," Bitcoin Magazine, Jan. 2014.

[12] Easy Miner, Available at: https://play.google.com/store/apps/details?id=com.mr.app.ui&hl=en

[13] LTC and Scrypt Miner PRO, Available at: https://play.google.com/store/apps/details?id=com.miner.scrypt&hl=en

[14] S.J. Chang, R. Perlner, W. E. Burr, M. S. Turan, J. M. Kelsey, S. Paul, and L. E. Bassham, "Third-round report of the SHA-3 cryptographic hash algorithm competition," NIST Interagency Report 7896, Nov. 2012.

[15] Ed25519: high-speed high-security signatures. Available at: https://ed25519.cr.yp.to/.

[16] Base58Check encoding. Available at: .

[17] D. Ostrovsky, M. Haji, and Y. Rodenski, "Couchbase Lite on Android," in *Pro Couchbase Server*, pp. 283-292, Publisher: Apress, Berkeley, CA, 2014.

[18] D. Ostrovsky, M. Haji, and Y. Rodenski. "Synchronizing data with the Couchbase Sync Gateway," in *Pro Couchbase Server*, pp. 331-342, Publisher: Apress, Berkeley, CA, 2014.

[19] T. McConaghy, R. Marques, A. Mller, D. De Jonghe, T. McConaghy, T, G. McMullen, and A. Granzotto, "BigchainDB: a scalable blockchain database," White Paper, BigChainDB, 2016.

[20] VideoOptimizer. Available at: https://developer.att.com/video-optimizer/docs

[21] J. J. Buckley. *Long Tailed Distributions*. InFuzzy Probabilities and Fuzzy Sets for Web Planning 2004 (pp. 147-159). Springer Berlin Heidelberg.