A High-Performance Hybrid Blockchain System for Traceable IoT Applications

Xu Wang[1,2], Ping Yu[1,3], Guangsheng Yu[1,2], Xuan Zha[1,4], Wei Ni[5], Ren Ping Liu[1,2], and Y. Jay Guo[1]

[1] Global Big Data Technologies Centre, University of Technology Sydney, Australia
{Xu.Wang-1,Guangsheng.Yu,RenPing.Liu,Jay.Guo}@uts.edu.au
{Ping.Yu-2,Xuan.Zha}@student.uts.edu.au
[2] Food Agility CRC Ltd, 81 Broadway, Ultimo, NSW, Australia, 2007
[3] State Key Laboratory of Networking and Switch Technology, Beijing University of Posts and Telecommunications, China
[4] China Academy of Information and Communications Technology, Beijing, China
[5] Data61, CSIRO, Australia
Wei.Ni@data61.csiro.au

Abstract. Blockchain, as an immutable distributed ledger, can be the key to realize secure and trustworthy IoT applications. However, existing blockchains can hardly achieve high-performance and high-security for large-scale IoT applications simultaneously. In this paper, we propose a hyper blockchain architecture combining the security of public blockchains with the efficiency of private blockchains. An IoT anchoring smart contract is proposed to anchor private IoT blockchains into a public blockchain. An IoT device management smart contract is also designed to trace sensory data. A comprehensive analysis reveals that the proposed hybrid blockchain system can achieve the performance of private blockchains and resist tampering.

Keywords: Blockchain · Internet of Things · Smart Contract.

1 Introduction

The Internet-of-Things (IoT) technology has been digitizing the physical world with ubiquitous smart devices [1]. The IoT data in centralized IoT systems are exposed to various threats, e.g., natural disasters and cyber-attacks. Meanwhile, the centralized IoT system cannot guarantee data transparency and therefore, goes against IoT data sharing among multiple parties. By storing data in blocks with their hash values chained in a peer-to-peer network, the blockchain technology is able to provide immutable, transparent and trustworthy ledger services for IoT applications [1]. The blockchain technology has been employed in IoT as the root of trust, e.g., access control [2] and data securing [3].

A significant challenge of applying the blockchain technology into IoT applications is the trade-off between performance and security [4]. To be specific,

blockchains can be divided into private (consortium) blockchains, e.g., Hyperledger Fabric [5], and public blockchains, e.g., Ethereum [6]. Running in small-scale networks, private blockchains are able to provide high capacity and low latency services but can only tolerate limited failures or attacks. For example, Fabric can achieve 3,500 transactions per second with a latency of less than one second [5]. Public blockchains, on the contrary, can tolerate large-scale failures but can only provide low capacity and long delay. For example, Bitcoin can only accept seven transactions per second and needs a confirmation time of 10 minutes [1].

In this paper, we propose a hybrid blockchain system combining the advantages of public blockchains and private blockchains. The proposed hybrid blockchain system is able to achieve the capacity of thousands of transactions per second with a latency of seconds based on the private blockchain technology. The hybrid blockchain system can achieve tamper-proof of existing blocks and stop a limited number of victim nodes from creating forged blocks and fake anchoring proof. An IoT device registration and revocation smart contract is developed to ensure the traceability of IoT sensory data and prevent forged IoT data. We carry out a comprehensive analysis of the performance and security of the hybrid blockchain system. Our analysis indicates that the hybrid blockchain system can achieve the performance of private blockchains and resist tampering even all the IoT blockchain miners are compromised.

The rest of the paper is organized as follows. In Section 2, the related works are surveyed, followed by the proposed hybrid blockchain system in Section 3. A comprehensive analysis of the hybrid blockchain system is carried out in Section 4, followed by the conclusions in Section 5.

2 Related Work

As pieces of computer code running on top of blockchain, smart contract translates various assets, such as IoT devices and digital assets, into virtual identities in blockchain, and enables them to interact with other assets [7]. Smart contract technology plays an important role in providing authentication rules, managing and securing IoT data [8] in an automatic manner.

Blockchain has been introduced into IoT applications for access control [2], data securing [3], etc. A smart contract was developed in [2]. The smart contract allowed managers to define access control policy for IoT data. In [3], a secured IoT data storage system was presented over a blockchain. Specifically, the data was stored in Distributed Hash Tables (DHTs) while the pointer to the DHT storage address was secured in a blockchain. The blockchain also took charge of the access control of the IoT data.

The combination of public blockchains and private blockchains can achieve the security of the public blockchain with the efficiency and confidentiality of the private blockchain [9,10,11]. In [9], the private blockchain was introduced to verify the accuracy of IoT data and broadcast transactions, while the public blockchain was used to verify consistency and store data. A hybrid blockchain

system combining a public blockchain and private blockchains was developed in [10]. The public blockchain was introduced as external monitors to store public information, e.g., geolocation information from trucks. The private blockchains were used for sensitive information, e.g., shipment information.

By utilizing the immutability of public blockchains, data can be anchored to a public blockchain to create a timestamp proof [12]. The anchoring technology was further extended to IoT applications [13,2,14,3]. For example, IoT devices were clustered into groups where the header nodes signed the IoT data centrally and attached the data to the public blockchain directly [14].

3 Hybrid Blockchain System

3.1 Overview

The proposed hybrid blockchain system consists of a public blockchain and multiple IoT blockchains, as shown in Fig. 1. The hybrid blockchain system provides a trusted and reliable IoT data storage service that only the sensory data from registered IoT devices can be recorded and verified.

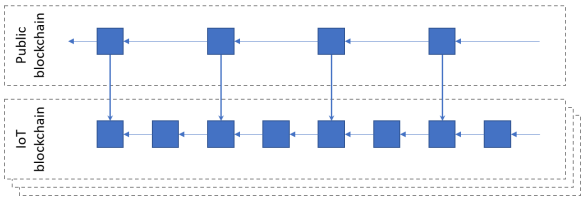

Fig. 1. The proposed hybrid blockchain system.

The public blockchain is considered to be trustworthy and used as the root of trust for IoT blockchains. This is a practical assumption because popular Proof-of-Work (PoW) blockchains can resist attacks with less 25% of the total computing power [15]. The public blockchain takes in charge of low-frequent functions, e.g., IoT blockchain initialization and IoT blockchain anchoring, by running smart contracts. As a public popular blockchain platform, Ethereum can be a good choice for the public blockchain, which has been widely adopted in researches [6,16].

IoT blockchains can be private blockchains and run at high speeds for high-capacity and low-latency IoT services. IoT blockchains adopt consortium consensus protocols, e.g., Practical Byzantine Fault-Tolerant (PBFT) [17]. Each IoT blockchain, consisting of miners and IoT devices, can be set up and maintained by a consortium. The IoT devices collect and send sensory data to the IoT blockchain in the form of transactions. The miners are requested to be registered

in the public blockchain and managed by participators of the IoT blockchain. The miners validate IoT transactions and generate blocks in IoT blockchain.

All the nodes, including IoT blockchain miners and IoT devices, have their unique public/private keys and can be identified by their corresponding public keys (or the equivalent blockchain addresses). The private keys, with the assumption of absolute security, are the foundation of blockchain security and prevent identity forging attacks [16]. Note that, miners of an IoT blockchain use the same pairs of public/private keys across the public blockchain and IoT blockchain.

The IoT blockchain blocks are anchored to the public blockchain periodically with a predefined interval of t to improve data integrity. This is achieved by calculating the hash values of the anchoring blocks in the IoT blockchain and then registering the hash values on the public blockchain. In order to stop attackers from forging and/or stopping the blockchain anchoring, a vote-based consensus protocol is proposed over smart contracts.

3.2 Steps to Run an IoT Blockchain

IoT Blockchain Registration. Let m denote the number of IoT blockchain miners. An IoT blockchain is registered on the public blockchain first before starting the IoT blockchain. This is realized by a smart contract on the public blockchain. The smart contract stores the IoT blockchain ID, hash of the genesis block, and a list of IoT blockchain miners.

The IoT blockchain registration needs to be confirmed by all the miners of the IoT blockchain. In other words, the IoT registration smart contract needs m different confirmations and cannot tolerate any failure.

IoT Devices Registration and Revocation. IoT devices need to be registered on the IoT blockchain before sending sensory data. The devices are identified by their blockchain addresses/public keys. The public keys of IoT devices are registered by the IoT blockchain miners for the verification of IoT transactions. The IoT revocation also needs to be confirmed by IoT blockchain miners. The private keys of IoT devices can only be seen by the IoT devices, which suppresses forged transactions.

Suppose that the IoT device registration/revocation can tolerate τ_m failures and needs at least $(m - \tau_m)$ confirmations. The threshold τ_m can be defined by the IoT consortium to trade off between security and failure tolerance. The registration and revocation of IoT devices can be realized by an IoT management smart contract on the IoT blockchain. The smart contract storing IoT blockchain miners and registered IoT devices can only be called by IoT blockchain miners.

IoT Transaction Generation. The IoT devices generate transactions by embedding sensory data in the data field of the transactions, sign the transactions with their private keys and send the signed transactions to IoT blockchain miners.

IoT Blockchain Growing. The IoT blockchain miners collect IoT transactions and then verify the senders by checking whether the senders are registered in the IoT management smart contract. If the transactions are from registered

IoT devices, the transactions are kept for further mining. Otherwise, the transactions will be dropped by the miners.

The IoT blockchain miners then run consensus protocols to mine transactions into blocks. The miners also anchor the IoT blockchain into the public blockchain by sending the hash values of check blocks with a predefined interval to the smart contract in the public blockchain. The hash of the check block in the IoT blockchain must be consistent with the registered block in the public blockchain. The verification process can be adopted as a part of the block consensus protocol.

Suppose that the IoT blockchain anchoring can tolerate τ_r malicious miners and needs at least $(m - \tau_r)$ different confirmations. The threshold τ_r is defined by IoT blockchain miners to trade off between integrity and failure tolerance.

IoT Sensory Data Verification. The proposed hybrid blockchain system realizes a chain of trust where the trust is passed from the public blockchain to IoT transactions through the IoT blockchain anchoring contract, IoT blockchain miners, and the IoT device management contract.

To verify an IoT transaction, the examiner first verifies the latest check block and the IoT blockchain miners with the help of the IoT anchoring smart contract on the public blockchain. The examiner then checks the identity of the sender of the IoT transaction with the IoT device management contract on the IoT blockchain at the height of the IoT transaction. The examiner lastly checks the signature of the IoT transaction. The IoT transaction is trustworthy if all the above steps are verified successfully.

3.3 Smart Contracts

We propose an IoT blockchain anchoring contract on the public blockchain to connect the public blockchain and IoT blockchain and an IoT device management contract on the IoT blockchain for IoT device registration and revocation.

IoT Blockchain Anchoring Contract. This contract provides blockchain registration and anchoring service for a single IoT blockchain. An IoT blockchain can be registered with its IoT blockchain ID, the hash value of the genesis block and miners. The contract needs to be deployed on the public blockchain by one of the miners. The IoT blockchain ID, the hash value of the genesis block and miner list can be hard-coded into the smart contract. In this way, the other IoT blockchain miners only need to give approval to the contract rather than pass all the data to the contract for registration.

Functions in the IoT blockchain anchoring contract can be given as follows.

1. *IoT blockchain confirmation function:* This function manages the state of the IoT blockchain. The smart contract sets the state of the IoT blockchain as "pending" by default. The contract keeps a list of predefined IoT blockchain miners and their states. When all the miners have confirmed the IoT blockchain, the contract marks the IoT blockchain as "running".
2. *IoT block anchoring function:* This function implements the block anchoring service. The smart contract stores an object, including a block index, a block state, callers and the block hash values from the callers, for an anchored

block. After the smart contract has received $(m - \tau_r)$ confirmations from different miners with the same block index, block hash value, the block state is updated to be "registered".

IoT Device Management Contract. This smart contract manages IoT devices with their blockchain address. This contract is deployed in the genesis block of the IoT blockchain by one of the miners. The contract provides IoT device registration and revocation services by two functions. This smart contract only accepts function calls from IoT blockchain miners. We assume that these two functions are independent and can tolerate τ_m failures. In other words, both IoT registration and revocation need $(m - \tau_m)$ approvals from different IoT blockchain miners.

1. *IoT device registration function:* This function collects verification about IoT devices from IoT blockchain miners and then registers the IoT devices into the IoT blockchain. In the first time that an IoT device uploads IoT data to the IoT blockchain, it notifies its IoT blockchain address to all the IoT miners. The IoT miners call the IoT device registration function and pass the IoT blockchain address of the IoT device to the smart contract for the registration of the IoT device.
   If this function is called with a new IoT blockchain address, the contract creates a new IoT object containing the blockchain address, the default "unconfirmed" state, and the caller address. If this function is called with an existing IoT blockchain address, the contract first adds the caller into the corresponding IoT object. If the address is confirmed by $(m - \tau_m)$ different miners, the state of the IoT blockchain address is updated as "registered".
2. *IoT device revocation function:* This function collects IoT device revocation requests from IoT blockchain miners and updates the states of IoT devices. The IoT blockchain miners should reach consensus to revoke an IoT device. The IoT miners then call the IoT device revocation function and pass the IoT blockchain address of the IoT device to the smart contract.
   In the case that this function is called with an unknown IoT blockchain address, the smart contract just ignores the function call. In the case that this function is called with a registered IoT blockchain address for the first time, the smart contract adds a "pending" state and the caller address to the corresponding IoT object in the smart contract. In the case that the function is called to update a "pending" address, the smart contract adds the caller to the IoT object. If the revocation request for an IoT blockchain address is confirmed by $(m - \tau_m)$ different miners, the smart contract removes the IoT object of the IoT blockchain address.

4 System Analysis

4.1 Performance Analysis

The performance of IoT blockchain is upper bounded by the performance of its consensus protocol. Between block anchoring, the IoT blockchain can achieve the

performance of the consensus protocol. For example, the IoT blockchain can use the consensus protocol realized in the FastFabric blockchain with the capacity of up to 20,000 transactions per second [18].

When a block is anchored to the public blockchain, the IoT blockchain needs to temporarily synchronize with the public blockchain. Thus, the performance of the IoT blockchain is reduced around check blocks. However, long block anchoring intervals can have a negative impact on the IoT blockchain security, as will be analyzed in Section 4.2.

4.2 Security Analysis

We assume that the public blockchain is trustworthy and cannot be tampered. The IoT blockchain has been confirmed by all the IoT blockchain miners. Attackers can temporarily control v IoT blockchain miners at the same time. In this paper, we analyze the case that IoT device registration and block registration follow the same failure tolerance of the consensus protocol, i.e., $\tau = \tau_c = \tau_m = \tau_r$. Other cases can be similarly analyzed.

Stopping Services. We first analyze the attack that victim miners stop working. We can have the following conclusions. a) When $v \leq \tau$, the IoT blockchain can tolerate the failure and keep on growing and anchoring blocks into the public chain, and IoT devices can register in or revoke from the IoT blockchain; b) When $v > \tau$, the IoT blockchain cannot grow, and IoT device cannot register in or revoke from the IoT blockchain.

Tampering Existing Transactions and Blocks. The attackers cannot tamper existing IoT transactions in the name of registered IoT devices. This is because the IoT transactions are signed with the private keys of IoT devices, and the private keys are kept by their owners. The tampered transactions, if any, cannot pass the signature verification. When $v < (m - \tau)$, attackers cannot tamper any existing block in the IoT blockchain. When $(m - \tau) \leq v \leq m$, attackers can tamper existing blocks after the last check block, e.g., forking the IoT chain and dropping transactions in existing blocks. The attackers cannot tamper any block before the last check block because the blocks are secured in the public blockchain by the anchoring.

IoT Devices Registration and Revocation. a) When $v < (m - \tau)$, attackers cannot register any IoT device nor revoke IoT devices. b) When $(m - \tau) \leq v \leq m$, attackers are able to register their own IoT devices and revoke any registered IoT devices.

Forging Transactions and Blocks. a) When $v < (m-\tau)$, attackers cannot create forged transactions nor blocks. b) When $(m - \tau) \leq v \leq m$, attackers can register their IoT devices and then create forged new transactions and blocks.

5 Conclusion

In this paper, we designed a hybrid blockchain system which anchors private IoT blockchains to a public blockchain. The proposed hybrid blockchain system is

able to provide high capacity and low latency services for IoT application while preserving data integrity with the help of public blockchains. An IoT device management smart contract on IoT blockchain was also proposed for IoT device registration and revocation to improve traceability.

6 Acknowledgement

This project was partially supported by funding from Food Agility CRC Ltd, funded under the Commonwealth Government CRC Program. The CRC Program supports industry-led collaborations between industry, researchers and the community.

References

1. Xu Wang, Xuan Zha, Wei Ni, et al. Survey on blockchain for Internet of Things. *Computer Communications*, 136:10 – 29, 2019.
2. Oscar Novo. Blockchain meets iot: an architecture for scalable access management in iot. *IEEE Internet of Things Journal*, 2018.
3. Ruinian Li, Tianyi Song, et al. Blockchain for large-scale internet of things data storage and protection. *IEEE Transactions on Services Computing*, 2018.
4. Xu Wang, Guangsheng Yu, et al. Capacity of blockchain based internet-of-things: Testbed and analysis. *Internet of Things*, 8:100109, 2019.
5. IBM. Behind the architecture of hyperledger fabric. https://www.ibm.com/blogs/research/2018/02/architecture-hyperledger-fabric/.
6. Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.
7. Konstantinos Christidis and Michael Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303, 2016.
8. Minhaj Ahmad Khan and Khaled Salah. Iot security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82:395–411, 2018.
9. L. Wu, K. Meng, et al. Democratic centralism: A hybrid blockchain architecture and its applications in energy internet. In *2017 IEEE International Conference on Energy Internet (ICEI)*, pages 176–181, April 2017.
10. Z. Li, H. Wu, et al. A hybrid blockchain ledger for supply chain visibility. In *International Symposium on Parallel and Distributed Computing*, June 2018.
11. J. Yu, D. Kozhaya, et al. Repucoin: Your reputation is your power. *IEEE Transactions on Computers*, 68(8):1225–1237, Aug 2019.
12. Chainpoint. Chainpoint. https://chainpoint.org.
13. Faizod. Faizod.anchoring. https://faizod.com.
14. Ali Dorri, Salil S Kanhere, et al. Towards an optimized blockchain for iot. In *Proc of 2nd Int. Conf. IoT Design and Implementation*, pages 173–178. ACM, 2017.
15. Christopher Natoli, Jiangshan Yu, et al. Deconstructing blockchains: A comprehensive survey on consensus, membership and structure, 2019.
16. X. Wang, X. Zha, et al. Attack and Defence of Ethereum Remote APIs. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, Dec 2018.
17. G. Yu et al. An optimized round-robin scheduling of speakers for peers-to-peers-based byzantine faulty tolerance. In *2018 IEEE GC Wkshps*, Dec 2018.
18. C. Gorenflo, S. Lee, et al. Fastfabric: Scaling hyperledger fabric to 20, 000 transactions per second. *CoRR*, abs/1901.00910, 2019.