A lightweight permission-based blockchain for IoT environments

Cooper, Andrew; Zaidi, Syed Ali Raza; Shakir, Muhammad Zeeshan; Ahmadi, Hamed

Published in:
Proceedings of the 2020 International Conference on UK-China Emerging Technologies (UCET)

Accepted/In press: 27/07/2020

Document Version
Peer reviewed version

A Lightweight Permission-Based Blockchain for IoT Environments

1st Andrew Cooper
School of Electronic and Electrical Engineering
University of Leeds
United Kingdom
scac@leeds.ac.uk

2nd Syed Ali Raza Zaidi
School of Electronic and Electrical Engineering
University of Leeds
United Kingdom
s.a.zaidi@leeds.ac.uk

3rd Muhammad Zeeshan Shakir
School of Computing
University of the West of Scotland
United Kingdom
muhammad.shakir@uws.ac.uk

3rd Hamed Ahmadi
Department of Electronic Engineering
University of York
United Kingdom
hamed.ahmadi@york.ac.uk

***Abstract*—The Internet of Things (IoT) is a rapidly evolving field. As it continues to create new applications in life, industry, agriculture and infrastructure, new challenges surface with them. The application of IoT networks range from medical devices to industrial robots. Such application dynamics present several new networking challenges. IoT networks are inherently heterogamous in terms of device capabilities, data volume, velocity and variety. This paper explores the possibility and effectiveness of combining IoT with the original concepts of blockchain technology in maintaining a verifiable ledger of polymorphic transactions to enable secure communication between devices. We present a practical implementation of a flexible blockchain system as a lightweight attachment to IoT devices. This system embeds a policy manager which defines and regulates the permissions required for authorised transactions; only a single device is permitted to amend this policy. We then demonstrate this system inside a reference topology by simulating activity between multiple homogeneous IoT devices communicating over the blockchain. We include devices of varying capabilities to demonstrate how those with more resources can support the chain whilst weaker devices consume from it.**

***Index Terms*—blockchain, IoT, edge, lightweight, permission**

I. Introduction

Security is a paramount consideration in any networking environment, exposing low-power devices on the open internet is an inherent and widely researched vulnerability in IoT given their generally weaker nature. If blockchain technology was to power an IoT network, devices within can confidently rely on it for data exchange in every transaction.

A. Short Background on Blockchains

Blockchains were introduced as a way of maintaining a distributed database of all transactions grouped in blocks. When signed, these blocks are published to the network and become publicly traceable. Each block has a cryptographic hash which ensures authenticity by featuring in neighbouring blocks, forming the 'chain'. In cryptocurrencies, the block hashes serve as guarantors of transaction ownership and authenticity as the block contents are part of the input toward computing the hash. The reference to the hash of the predecessor block forms the ancestral chain. To successfully tamper with an already-decided block, an adversary must re-compute all successor blocks whilst continuing the present chain with new transactions, which becomes more difficult as the audit trail expands. This fraudulent chain must then be submitted to, and accepted by, the majority of peers among a decentralised network, which is unlikely without the adversary controlling the majority of the network. Obtaining this proof of work (PoW), a piece of data traditionally expensive to produce yet simple to verify, is known as mining. With traditional blockchains, the PoW is computed by finding a nonce which, when paired with properties and containing transactions of the block, create a signature in a format stipulated by the blockchain.

However, a traditional blockchain implementation would be unsuitable in the context of IoT, as working inside tight resource constraints is a core characteristic of IoT development. Embedded devices are often low in processing power and storage capacity, and are therefore unlikely to cope with sustaining the blockchain as well as delivering their original purpose unimpaired. This will require designing a lightened blockchain with its mechanics tailored to suit an IoT environment.

B. Related Work

A review of previous studies addressing the security challenges found in IoT applications has shown a shared concern for privacy and traceability. A recent study by Hewlett Packard [1] found 70% of IoT devices to be vulnerable to attack, citing privacy protection and weak authorisation as significant contributing factors. The principles of blockchain technology can address these. The work of Zheng et al. [2] discusses the sensitive nature and volume of data IoT devices produce ushers a need for protection and governance should they malfunction or become compromised. Adireddy, Gottapu, and Aravamudhan note in [3] that the multiplicity of IoT devices

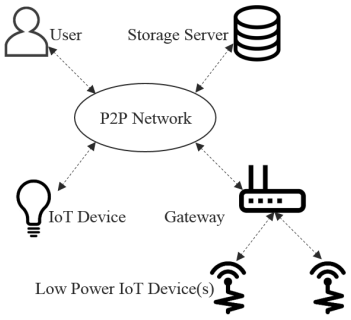

Fig. 1. High level overview of participants, including a cluster head (bottom right).

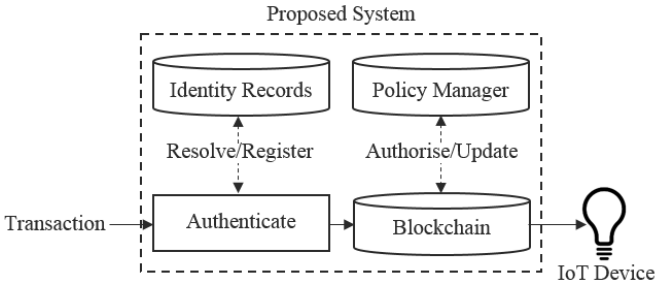

Fig. 2. Pathway of the request process on an IoT device.

on a network along with their reduced capabilities adds to their vulnerable nature and offer a solution based on public key infrastructure.

Others have proposed integrating blockchain concepts with IoT, such as Zhang et al. [4], who found the strong interconnection of IoT networks leaves them vulnerable to adversaries deploying malicious systems into the network and are easily compromised, demonstrating the need for access control. They propose a system leveraging smart contracts to regulate access control, with blockchain supporting a distributed ledger of the contracts. Danzi et al. [5] explore the feasibility of integrating blockchain distributed ledger technologies (DLTs) within a wireless network, finding DLT an effective guarantor of information but at significantly higher network traffic. These tudies are only relevant to this article and by no means this is exhaustive of the list of the related work on DLT. Readers interested in DLT are directed to [6] and references therein.

II. Methodology

This section presents the design of the proposed system and includes the changes made to traditional blockchain technology to accommodate for IoT devices and the nature of their environment. The system is designed as a lightweight, software-based attachment connected over a standard P2P network.

Although a public chain, only authorised participants, who must exist on the policy, may receive from and submit to the chain. Extremely low power devices unable to participate in the chain may rely on direction from a cluster gateway outfitted with the system described in this paper, as illustrated in the bottom right of Fig. 1, wherein an array of low-power sensors are under the management of a more robust gateway capable of participating in the blockchain. When another device requests a reading from a subordinate device with appropriate authorisation, the cluster head receives this request and probes accordingly. The cluster head then propagates the reading on to the network from its subordinate device.

There is the possibility a weaker device under a cluster head can become compromised, feeding malicious data unbeknownst to the cluster head. A node must therefore be able to suspect a breach and verify the integrity of another. For the procedure, we recommend inspection of existing patterns such as challenge-response attestation between a challenger and a prover. The former requests a memory checksum be computed ad hoc by the latter, using a nonce to prevent precomputed or masqueraded attacks, similar to the PoW found in blockchains. This technique is adaptable to an IoT application, as seen in the work of Krauß, Stumpf, and Eckert in [7], who extend the process to include neighbours in peer policing through behaviour analysis, which would be processed by a more capable node (in this application, the cluster head).

Unlike traditional implementations, this proposed blockchain stipulates no additional requirements from miners in hash computation or PoW. There is no expected composition for hashes, such as involving nonce tokens as with cryptocurrency blockchains. Members of the block and its transactions within are given as input to a SHA-2 function with a 256 bit digest. There is no economic incentive offered to miners for throttling the block production rate as this would only apply undue exertion on participants. However, with no transaction fees, devices cannot mark the importance of their transaction as with cryptocurrency blockchains, where miners prioritise confirmation of those offering a higher processing fee [8].

A. Device Integration

All devices rely on the same proposed system for requests as illustrated in Fig. 2, which serves as an intermediary between the network and the device itself. Devices connect to the network and receive their commands from the blockchain in an event-based fashion through callbacks. Authorised device transactions are relayed through an event for the device to process. Fig. 2 illustrates this flow through the direction of arrows and system boundary lines, showing the process from a received request passing through the local blockchain system and, if authorised, on to the device itself. Administrative transactions, however, are executed directly within the blockchain.

When a participant device is booted, it first attempts to load the cached blockchain from disk into memory if it exists, verify its integrity and connect to peers to synchronise and reconcile any changes (e.g., policy amendments) from the latest chain which took place during since going offline.

B. Storage

Space complexity is a significant constraint of blockchain, particularly when considering an IoT application. Devices with

limited disk space are automatically ineligible as network participants; the linear sequencing of blockchain [9] will require some onboard storage volume - often scarce or absent entirely in IoT devices. How a participant manages its memory and storage does not affect the mechanics of the overall network. For example, we can reduce the burden on weaker devices by serialising the blocks into files of a suitable data-interchange format, named by their sealing hash for ease of access, and maintaining an in-memory, ordered list of the block hashes instead.

Devices on the network are not required to mine or archive the chain themselves if their processing power is limited; they may still observe and submit transactions but are unable to author blocks as they do not possess a full local copy of the chain required to verify continuity; devices which do are known as full nodes [10]. A full node with expansive storage can serve as dedicated archivist of the chain's ancestry. It is recommended to maintain at least one sibling copy of the chain as measures of both backup and protection, should a node become compromised and attempt to tamper with the chain.

III. Implementation

This section presents a novel software-based prototype of the proposed system, demonstrating how the characteristics of such a system would translate into software. The source code of this project is made available under a public repository[1]. Derivatives, improvements, and pull requests are welcomed.

A. Formulation

The blockchain offers two constructors for initialisation. The first accepts a parameter for an administrator profile identifier, used when formulating the blockchain as the first device on a new network. The second accepts a block vector, which is submitted to the local blockchain to verify and synchronise, used when joining an existing network. It is during this stage the policy is reconstructed and the former constructor is called, declaring the administrator recorded in the genesis transaction. As the synchronisation process can increase disk and memory activity, it is only once—during device boot. Any capable device may implement this protocol, but only the administrator may introduce it to the network by amending the policy. Participants of the blockchain may include appliances themselves, dedicated miners, archivist servers or a management device of a low-powered device cluster. The policy is maintained in-memory for faster lookup. Amendments to the policy are made through a distinct transaction type by the administrator carrying only the changes for the relevant device.

B. Transactions

Devices receive instruction as output from the blockchain, propagated via an appropriate callback. This design relieves the device from the repetitive task of evaluating a transaction's authenticity and determining how to act. All rejected transactions are propagated to an optional separate callback, which more powerful devices may register to monitor for spikes in rejected transactions, possibly indicating a compromised device on the network, and alert the administrator. The callback registrar does not require a corresponding entry for each type of transaction and event, it is left entirely to the device to register interest or share callback methods.

[1]Repository available at: https://github.com/aco/iot-blockchain.

Algorithm 1 Authorising, storing, and executing a transaction.

procedure APPENDTRANSACTION(*transaction)
  $transactionType \leftarrow$ **typeid** of $transaction$
  if $transaction$.authorise(Policy) **then**
    if $transactionType$ administrative **then**
      execute $transaction$ passing Policy Manager
    else
      execute $transaction$ passing **nullptr**
  else
    invoke $rejectionCallback$ passing $transaction$
  append $transaction$ to getLeadBlock().$transactions$
  if $callbackRegistrar$ includes $transactionType$ **then**
    invoke $callback$ passing $transaction$
  else
    invoke $genericCallback$ passing $transaction$
  if $transaction.expedited$ **then**
    seal $leadBlock$
  else
    invoke $transactionBroadcastResponder$

A device transaction is authorised when there is a corresponding policy entry for the device specified by the transaction, and when the entry contains sufficient permission for the transaction author, depending on the action (read or write). The policy considers an absent entry for either key (resource or profile) as unauthorised. Authorised transactions are offered the policy header or propagated to the device for execution. Transactions with absent, banned or unknown identities are rejected and do not pass to the device (unless it specifically registers a rejected transaction callback).

All transactions in this system derive from a base Transaction class. Each subclass of transaction implements the virtual methods from the base to account for their additional properties and individual purpose, in keeping with the Liskov substitution principle. Transactions inherit an execute method, allowing the blockchain to delegate execution to the subclass to the same benefit above. As a preventive measure, only administrative transactions recognised as valid by the blockchain are offered a pointer to the policy manager. This process is denoted in Algorithm 1.

IV. Case Study

To demonstrate this blockchain system, we describe in this section a case study simulating a network of devices in the context of a typical smart home, running inside a single application to mimic a P2P network. It should be noted there is no explicit requirement for devices to follow the same standard in other use cases of this system.

- **Device:** The base for all participating devices. It includes a callback for transactions and blocks, a unique identifier, and a live value. Standard devices can author and broadcast transactions on the network, but cannot mine or publish blocks. These devices should register a callback for at least device transactions.
- **ParticipantDevice:** This form of device is assumed to be more powerful than the base and is capable of mining and publishing blocks on the network. It registers individual callbacks for all forms of transaction. It is also able to broadcast an entire block to the network.
- **ClusterDevice:** The cluster device subclasses from the participant device and is assumed to have at least equal computing power to its superclass. This device directs a subset of individual devices (which do not maintain their own copies of the blockchain) and registers an override for (authorised) inter-device transactions.

The device inheritance structure described above is solely for the demonstration of this system, devices may implement any portion(s) of this blockchain system. The only requirement is a callback for authorised device transactions. To simplify the network simulation, devices inherit from the Blockchain class to protect members from access by the network, in a similar fashion to the Policy Manager class hierarchy.

A. Synthetic Network

For the simulation, we construct a network of four devices based on the examples described in Section III, beginning with a standard device for the administrator. In reality, this may be a smartphone or tablet. Added to the network is a cluster head, responsible for two sensors detecting gas and flame. Finally, a standalone lightbulb is admitted to the network as a participant device.

When all devices have joined the network, an external helper method applies randomised permissions between random devices to ensure the policy will reject some transactions in the demonstration. All devices are permitted read access to the sensors within the cluster head of the network. Subordinates of the cluster are treated equally to peers, and each must have some entry in the policy header.

B. Performance

The practical implementation of this framework is written in C++17 for its efficiency and easier integration into testbed IoT hardware. The source was written under the C++ standard library with one external dependency, a header-only SHA-256 generator, used in hashing transactions and blocks. In simulation, using a 2015 MacBook Pro 2.2 GHz Intel Core i7 with 16 GB memory, the framework demonstrates high performance with an average processing rate of 45 unique, inter-device transactions per millisecond.

The majority of read operations on the framework run in $O(1)$ constant time, owing to extensive mapping of objects into sorted associative containers. As an example, the policy manager is implemented as a nested hash map for devices and their associated permissions for other devices on the network. Similarly, the majority of vector operations (such as appending transactions to a block) are run in amortized constant time. As with traditional blockchains, traversing the chain in-memory is only completed in $O(n)$ linear time as blocks are indexed numerically and not by hash. Indeed, $O(1)$ lookup is a possibility should a device opt to store blocks on disk, but the framework does not enforce such behaviour.

V. Conclusions

This paper proposes an approach to integrating both user and device actions and permission control in a single blockchain as a medium of communication. We begin with the core characteristics of blockchain technology, such PoW consensus, heaviest-chain favouring, transaction grouping and peer cooperation. Then, we refine for an IoT environment by eliminating economic incentives, such as rewarded mining, and introducing polymorphic transactions for communication. Not only can device access be regulated at a granular level, but its communications take place over the same protocol, leveraging the authentication already provided for in the permission system.

This system is generic enough to be extensible for other IoT-related use cases. In this prototype, the use of template functions and polymorphism requires little maintenance besides creating a new class of transaction. Devices may register a callback to the new transaction type in an identical fashion to existing types.

References

[1] Rawlinson, K. (2014) HP Study Reveals 70 Percent of Internet of Things Devices Vulnerable to Attack. HP Advisory.

[2] Zheng, Lirong, Hui Zhang, Weili Han, Xiaolin Zhou, Jing He, Zhi Zhang, Yun Gu, and Junyu Wang. "Technologies, applications, and governance in the Internet of Things." Internet of things-Global technological and societal trends. From smart environments and spaces to green ICT (2011).

[3] Adireddy, A., Gottapu, U. and Aravamudhan, A.P., 2016, October. Usercentric federation of access to Internet-of-Things (IoT) devices: a valet key for IoT devices. In 2016 International Conference on Circuits, Controls, Communications and Computing (I4C) (pp. 1-7). IEEE.

[4] Zhang, Y., Kasahara, S., Shen, Y., Jiang, X. and Wan, J., 2018. Smart contract-based access control for the internet of things. IEEE Internet of Things Journal.

[5] P. Danzi, A. E. Kalør, R. B. Sørensen, A. K. Hagelskjær, L. D. Nguyen, Č. Stefanović, Popovski and P. Popovski, "Communication Aspects of the Integration of Wireless IoT Devices with Distributed Ledger Technology.," arXiv, 2019.

[6] Sadek Ferdous, M., Jabed Morshed Chowdhury, M., Hoque, M.A. and Colman, A., 2020. Blockchain Consensus Algorithms: A Survey. arXiv, pp.arXiv-2001.

[7] Krauß, C., Stumpf, F. and Eckert, C., 2007, July. Detecting node compromise in hybrid wireless sensor networks using attestation techniques. In European Workshop on Security in Ad-hoc and Sensor Networks (pp. 203-217). Springer, Berlin, Heidelberg.

[8] Lo, S. and Wang, J.C., 2014. Bitcoin as money?.

[9] Babones, S., 2018. Smart 'blockchain battleships' are right around the corner.

[10] García-Bañuelos, L., Ponomarev, A., Dumas, M. and Weber, I., 2017, September. Optimized execution of business processes on blockchain. In International Conference on Business Process Management (pp. 130-146). Springer, Cham.