Self-adaptation made easy with Blockchains

Peter E. Sedgewick
University of Kent, UK
ps448@kent.ac.uk

Rogério de Lemos
University of Kent, UK
r.delemos@kent.ac.uk

ABSTRACT

This position paper describes how blockchains facilitate the implementation of distributed self-adaptive systems. We demonstrate how the master/slave decentralised control pattern for self-adaptive systems, integrated with a permissioned blockchain, can protect nodes of a network against attacks by continuously adapting the membership of an access control list. Whenever malicious behaviour is detected, consensus on an updated access control list is reached, and that node is removed from the network. Using a smart home, as an example, we demonstrate that a permissioned blockchain is able to maintain a consistent view of a network of Internet of Things (IoT) devices in the presence of malicious nodes.

CCS CONCEPTS

• **Software and its engineering → Software design engineering**; • **Security and privacy** → *Access control*;

KEYWORDS

self-adapive systems, blockchains, IoT, MultiChain, access control

ACM Reference Format:
Peter E. Sedgewick and Rogério de Lemos. 2018. Self-adaptation made easy with Blockchains. In *Proceedings of SEAMS '18 (SEAMS 2018).* ACM, New York, NY, USA, Article 4, 2 pages. https://doi.org/10.1145/3194133.3194150

1 INTRODUCTION

In this paper, we investigate how the master/slave pattern, identified as one of the patterns for decentralised control in self-adaptive system [**?**] can be implemented within a framework of a blockchain. This will be presented in the context of a private network of interconnected Internet of Things (IoT) devices. The proof-of-concept implementation uses an open source permissioned blockchain (MultiChain [1]) for maintaining an immutable record of transactions in order to handle malicious behaviour.

Blockchains provide one common virtual trusted ledger, which is replicated, produced collaboratively, and validated in a distributed fashion. Validated information is incorporated into the blockchain, after a consensus protocol ensures that the nodes agree on a unique order of the transactions. Taking as a reference the master/slave pattern, a blockchain is able to fulfil the activities of the Monitor and Execute stages of the MAPE-K loop that should be deployed on the nodes of private network, while the activities of the Analysis and Plan stages should be performed by a centralised node. The Knowledge (Model) can either be kept by each node as an access control list (ACL), or inferred by analysing the transactions of a block.

There have been a couple of publications regarding IoT security and blockchains [**? ? ?**]. For example, it has been argued that blockchains can solve a great number of IoT's security and privacy issues, and a subsequent architecture has been proposed for supporting a blockchain-based smart home [**?**]. While in [**?**], the authors introduce a novel access control mechanism using Bitcoin's scripting abilities. However, the challenge of protecting a system against behavioural uncertainties of malicious nodes was not covered. The goal of this paper is to apply the lessons learned from Self-adaptive Authorisation Framework (SAAF) [**?**] into blockchains in order to protect a private network against malicious behaviours. This exercise has shown that security should be considered in the context of other quality attributes, in our case dependability, which is fundamental for achieving consensus.

2 BACKGROUND ON BLOCKCHAINS

Blockchains provide a verifiable history of transactions between nodes in a network, and require the signing of messages via public-key cryptographic techniques. By maintaining an immutable record of previous transactions and using their hashes as a required input for succeeding transactions, the system maintains a permanent and accountable ledger of transactions that are explicitly linked. In this way, no previous transaction can be altered by any party, as doing so would cause cryptographic hashes to change. In a blockchain, transactions traditionally enter a *mempool* (in-memory store) after submission, and await to be *mined*. Mining is the process whereby valid transactions are collected into a *block*, and committed to the chain. Upon mining, the latest block is considered to be the most up-to-date accepted version of transactions.

As multiple nodes can submit different and valid blocks at the same time, a consensus protocol is needed to decide between competing blocks. Numerous consensus protocols exist, such as, proof-of-work, proof-of-stake (PoS) and Mining Diversity.

Permissionless (or public) blockchains, such as Bitcoin [2] and Ethereum [3], permit any node running appropriate client software to connect and mine blocks. Permissioned (or consortium) blockchains differ from a public blockchain since it can allow or deny specific addresses to view transaction information and participate in the network. Access control mechanisms are an integral of the latter, and our work manipulates those mechanisms (i.e., ACL) depending on the perceived attacks.

[1] https://www.multichain.com/

[2] https://blockchain.info/

[3] https://www.ethereum.org/

SEAMS 2018, May 28–29, 2018, Gothenburg, Sweden,

ACM ISBN 978-1-4503-5715-9/18/05…$15.00
https://doi.org/10.1145/3194133.3194150

3 CASE STUDY: SMART HOME

3.1 Problem Statement

In a network of IoT devices, nodes are allowed to communicate with other nodes providing they know the addresses of their peer nodes. Since IoT devices often have vulnerabilities which can be leveraged by attackers for malicious purposes, it is important that nodes should not have the sole control of communication. By only exposing the blockchain address of a recipient node (a function of the private/public key of the node), nodes would communicate through a permissioned blockchain, ensuring that transactions stored in the *mempool* can later be analysed.

Let's consider the scenario of a household with several IoT devices. These devices are expected to communicate with each other to allow for a more integrated experience to the end user. For example, a movement sensor placed at the front door (D) may communicate with a kettle (K) that will start boiling when the owner enters the house each day around 5pm. In typical implementations, these devices may communicate using methods like HTTP, where D calls a REST endpoint exposed by K, for example. In this model there is no way for K to detect if D is acting out of character. An attacker may have been able to compromise the sensor on the door and maliciously begin boiling the kettle at arbitrary times. Considering the variety of IoT devices is growing in scope, the implications are potentially more serious.

In this example, we assume that each device has a certain behavioural profile, either set by an administrator or formed by the network of IoT devices based on their behaviour over time. K should identify D attempting to boil the kettle at unusual times or unusual frequency as suspicious and potentially *malicious*. As well as being intentionally compromised, some devices may begin to act out of character due to a fault in their software. Being able to monitor and analyse behaviour both increases security against attackers, and offers some protection against device malfunction. In this research, the support for access control will be a very simple list, where nodes have permissions for allowing or denying requests. In this context, a blockchain, associated with the closed household network of IoT devices, maintains a record of all transactions.

3.2 Experiment

We have implemented a permissioned blockchain using MultiChain because it allows to configure permissions at multiple levels of the multi chains, and to use federated mining (Mining Diversity) that can be more efficient.

A MultiChain was deployed in a network of IoT devices in which nodes had two distinct roles: *controller* and *device*. Controller nodes do not actively submit and receive transactions but rather monitor the network transactions, and administer the access control list (ACL), while acting upon malicious behaviour. The controller monitors newly mined blocks (through MultiChain's blockNotify event) in order to retrieve and analyse the block's transactions. If a node is perceived to act maliciously, the controller transmits a permission-update transaction by removing the offending node from the blockchain's ACL, preventing it from sending or receiving further messages. In our experiment, we modelled malicious behaviour on the basis of transaction volume within a specified time window (one minute).

In terms of the MAPE-K loop, the property of immutability inherent to a blockchain provides a distributed, safe way to execute the Monitor stage of a MAPE-K loop, whilst the decision reached through consensus allows the Execute stage to be performed by the collective efforts of a permissioned blockchain. In our experiment, in the Analysis stage the number of transactions per node were evaluated against a specified threshold, if a violation was detected the Plan would remove the node from the private network.

For our experiments, we have used 3 virtual machines running within a private network: Ubuntu 16.04, 1 CPU Core (Intel(R) Xeon(R) 2.40GHz), and 20GB HDD. Nodes communicate exclusively through the MultiChain rather than traditional methods like HTTP or TCP. Compared with a base level implementation using HTTP with no authentication, our MultiChain had a 750 fold decrease in throughput, largely due to the way we fetch transaction data through the use of the MultiChain JSON API. These are preliminary results, and significant improvements are expected if communication is done over TCP, for example.

One limitation of the proposed approach is that, permissionless blockchains consensus algorithms require computational resources that are impractical in the context of IoT. Another limitation specifically concerning Mining Diversity in MultiChain is that a malicious node may mine a block once in a while without any punishment. A promising solution for improving throughput is to use Byzantine fault tolerant consensus algorithms, like Hyperledger Fabric [4].

4 CONCLUSIONS

By implementing a simple proof of concept, we have shown in this paper that is feasible to incorporate self-adaptation into a blockchain in order to protect a network of IoT devices from malicious behaviour. We have demonstrated how properties of a permissioned blockchain can implement the Monitor and Execute stages of a MAPE-K loop. There is no inherent reason why the Analysis and Plan phases cannot be moved to functions of a blockchain. Novel systems for the addition and removal of nodes have been presented through the use of Smart Contracts , thus allowing the full MAPE-K loop to be executed through communal decisions and logic.

[4] https://hyperledger.org/projects/fabric