Decentralised Blockchain Security Architecture for IoT and MQTT Brokers

Project Plan

Konrad Dryja

k.dryja.15@abdn.ac.uk

Department of Computing Science,
University of Aberdeen, Aberdeen AB24 3UE, UK

Introduction

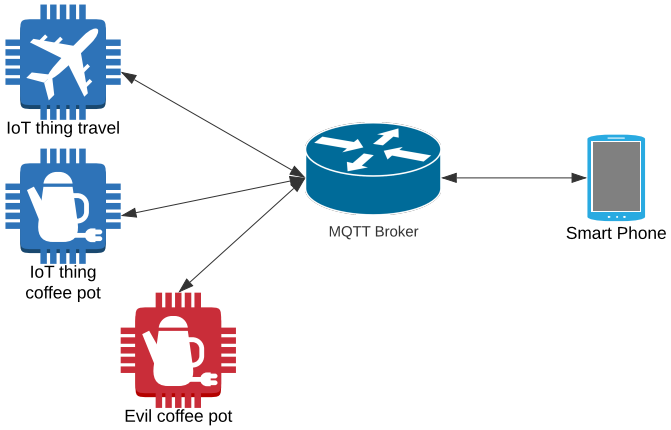
IoT is a growing field with presence getting larger from day to day. Equipment such as bus timetables, doorbells, thermostats or even pacemakers capable of sending alerts directly to emergency services is omnipresent. Sensitive data is very often handled (especially in the last example) and leak or misrepresentation can have catastrophic consequences. Additionally, those devices have limited computation power, which is sacrificed in favour of extended lifespan to avoid frequent maintenance and/or battery replacement. This makes it a very lucrative target for malicious actors. Researchers from University of Michigan performed a demo attack on a pacemaker, extracting personal information and changing the configuration.[1] This example and more clearly presents a need for solution capable of securing and closing off access to unwanted agents.

What is more, those devices frequently do not directly communicate on a peer-to-peer basis, but instead pass through intermediary resource handling the distribution of data. MQTT Brokers[1] are one of them, providing a Publisher/Subscriber architecture, allowing for information exchange without the constant need of interconnectivity between clients, instead of relying on created topics used as a relay. MQTT is also not a piece of software, but rather a standard describing the operation of such backend and leaves the implementation up to end-users. The documentation also clearly states that the security is not a part of the standard, but rather leaves this decision up to the implementors, offering several alternatives[2]. And while the transport of data can be assumed to remain secure (as MQTT operates over TCP/IP, thus encrypting the data using TLS), the access is often not. For example, an attack could be conducted, where malicious device communicates with the broker, impersonating another IoT hardware and finally compromise confidentiality, as shown on Figure 1.

Common implementation of MQTT brokers includes basic authentication based on a username and a secret password, offering limited to no logging capabilities. For example, one of the most popular brokers being Mosquitto[3], as of version 1.6.8 offers password

[1] https://mqtt.org/
[2] https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html#_Toc3901261
[3] https://mosquitto.org/man/mosquitto-conf-5.html

Figure 1: Attackers can impersonate IoT devices



based authentication with ACL capabilities restricting some credentials only to given topics. TrustLens project from University of Aberdeen presented solution of attaching semantic annotations to ongoing operation of a broker and in process allowing higher transparency of broker's state[2].

I would like to propose a universal, pluggable module following the description of MQTT standard (and thus potentially making it independent from the used MQTT implementation) handling Authentication, Authorization and Accountability of IoT devices and consumers attempting connection to the brokers. In order to make the platform more resilient and independent from single points of failure (often being the central ACL repository), I will deploy it on a blockchain - which by convention supports recording of transactions (by recording them on the blockchain) and authentication via ID of the presented wallet. I will also be aiming to merge the efforts of Aberdeen's scientist into the project, enhancing trust of captured data.

Preliminary research revealed similar efforts by scientist in Khon Kaen University [3], although focusing mostly on Hyperledger Fabric and optimization of the configuration through Genetic Algorithms. The paper doesn't make any mention of logging and provenance functionality, which I would like to put more attention towards - along with focusing on Ethereum platform instead.

Goals

The project can be divided onto main goals and extras, leaving some field for maneuvering in case of road blocks or difficulties resulting from the challenges faced in the dissertation. By having flexible targets, I will be able to stop sooner in case of overestimating the schedule, or carrying on with extra work, should I find myself meeting the targets quicker than expected. Moreover, I would like to also put some attention to listing the things that the project is **not** trying to be, to avoid sidetracking.

Main Goals:

- Design Blockchain network and deploy on Ethereum platform, capable of recording transactions and allowing for modification of ACLs, i.e. which wallet ID is permitted to access given topics.
- Design containerised software acting as a secure proxy between brokers and connecting clients.
- Deploy and test on at least one MQTT implementation (e.g. Moquette, Mosquitto), simulated with software.

Extra Goals:

- Explore and implement implementation of the architecture to any, arbitrary broker.
- Connect physical LoRaWAN gateways to the software.

No-goals:

- Design a new blockchain platform from scratch.

Methodology

This section should describe *how* you will conduct your project. You should explain in general terms the activities you will be carrying out during your project, such as:

- reading about related work – either to get ideas on how to proceed, or to compare your approach with what was done before;
- learing a new programming language or API;
- learning about relevant technologies;
- developing prototypes to test ideas;
- testing and debugging early design choices.

The above examples are purely suggestions. You should try to think of what would be appropriate for your specific project.

Resources Required

For the purposes of the project, laptop and a workstation should be sufficient, as the MQTT brokers can be simulated via software, without the need of physical IoT devices sending the messages. For the second stage of the project, I will be considering putting the software on a physical LoRaWAN gateway, enhancing the current broker implementation. TrustLens team from University of Aberdeen recently acquired such gateway, which I will be able to use given the need. Moreover, the team also owns several PuckJS sensors and Arduino boards, which would allow to deploy the whole system and test its behaviour in the real world.

To summarize the requirements:

- Linux PC with the following software:
  - Golang 1.12+
  - Docker 19.03.5+
  - Suitable code editor (Vim)
- Additionally, for tests in the field:
  - LoRaWAN Gateway
  - IoT sensors (PuckJS, Arduino)

Risk Assessment

Risk	Mitigation	Level
• Ethereum doesn't meet my needs • I face a block relating to knowledge of Ehtereum effectively stopping me from reaching my goals.	• Switch to an implementation that I am more familiar with, although with slightly different tradebacks, that is, Hyperledger Fabric.	Medium
• Ethereum doesn't meet my needs • I face a block relating to knowledge of Ehtereum effectively stopping me from reaching my goals.	• Switch to an implementation that I am more familiar with, although with slightly different tradebacks, that is, Hyperledger Fabric.	Low
• Unexpected hardware failure or theft. • Any hazards resulting from the natural wear and tear of utilized hardware	• All code, documentation and report will be frequently backed up and versioned using git (and thus, uploaded to GitHub), allowing for data recovery. • If LoRaWAN gateway or other IoT equipment was to become faulty or infeasible, focus on virtualised solutions.	Low

Try to describe possible circumstances (e.g. a particular piece of technology doesn't work or is too expensive) that might cause the project to become become infeasible. What would you have to do or change to recover your project?

Timetable

This section should describe the *schedule* for your project. You should describe the various activities you expect to perform and their durations, along with any deadlines and deliverables. It is often useful to collect all of this information in a *Gantt chart*, as shown in Figure 2 below:

Don't forget to add time at the end of your project for evaluation and writing-up! This could easily require 2-3 weeks.

Figure 2: Main Project Activities

References

[1] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 129–142, May 2008.

[2] Milan Markovic and Peter Edwards. Enhancing transparency of mqtt brokers for iot applications through provenance streams. In *Proceedings of the 6th International Workshop on Middleware and Applications for the Internet of Things*, M4IoT '19, page 17–20, New York, NY, USA, 2019. Association for Computing Machinery.

[3] N. Klaokliang, P. Teawtim, P. Aimtongkham, C. So-In, and A. Niruntasukrat. A novel iot authorization architecture on hyperledger fabric with optimal consensus using genetic algorithm. In *2018 Seventh ICT International Student Project Conference (ICT-ISPC)*, pages 1–5, July 2018.