TDDD17 - Blockchain Security for IoT

Joakim Kahlström Johan Hedlin
Email: {joaka568, johhe911}@student.liu.se
Supervisor: Andrei Gurtov, andrei.gurtov@liu.se
Project Report for Information Security Course
Linköping University, Sweden

***Abstract*—Home appliances and devices such as lights, thermostats, and air conditioners are gradually becoming more and more connected with their environment in a phenomenon known as the Internet of Things (IoT). Traditionally, these are connected to servers controlled by the manufacturer and are usually controlled through a companion app on a smartphone or through a voice assistant. With this development in connected devices, security seems to have fallen behind, as evidenced by several attacks targeting smart devices to create botnets or even ransom attacks where the user is locked out of their home or have their fridge be turned off until the ransom is paid. In this work, we study the practicality of applying blockchain technology to IoT in an effort to improve security through the authentication and verification methods present in a blockchain network. Our findings indicate that such a system is feasible to implement using low-cost single-board computers, and that it could provide secure storage of system state and log data in a decentralized fashion.**

I. Introduction

The Internet of Things (IoT) is an area that is rapidly growing worldwide. Internet-connected ”smart” devices aim to increase the functionality of previously isolated devices such as light bulbs and washing machines by allowing them to interact with each other or a user's smartphone. IoT is used both on large scale as in cities, big companies, and infrastructure and on a smaller scale such as in smart homes. IoT devices often consist of sensors or actuators of different kinds such as cameras, power regulators, etc, which could become a serious security or privacy issue. Especially because one of the main parts of IoT is that it is all connected to a network of devices and possibly accessible through the Internet.

Even data which at first glance might not seem that sensitive, such as timestamps or the presence of a mobile device, might reveal sensitive data in combination with other information, or reveal a pattern of when a person is at home or away [1].

Due to limited hardware resources, it might be infeasible to protect them the same way as a normal computer. As the focus of IoT devices is typically low cost, supporting an antivirus or a firewall would lead to more expensive hardware and physically larger devices. With these low-cost devices, security may often be neglected when vendor-specific software is developed, as efforts may be focused on features and efficiency instead of robustness and security.

For IoT systems that handle more sensitive functions such as alarms and door locks, having a log of when a door was opened and who disabled the alarm is also an important feature. If the security measures fail or if an account is compromised, actions can be taken afterwards to secure the system from future attacks. This log would then have to be stored where an attacker cannot access it, as they would otherwise simply be able to erase any trace of their presence.

One technique that has good potential in increasing security for IoT devices is blockchain technology, which has become popular as it is the main technique used in cryptocurrency. However, the technique has much more potential than only for cryptocurrency. As a blockchain consists of a sequence of data units, called blocks, and it is only possible to write new data at the end of the chain, it maintains a permanent record of all previous blocks. These cannot be modified without invalidating newer blocks, which makes the blockchain a good candidate for storing the types of sensitive logs mentioned above. All blocks are also cryptographically signed, which enables a device to verify that the data originates from a trusted source.

Currently, the Hyperledger project, created by the Linux Foundation, is working on multiple open-source blockchain projects making the technology available for businesses to use the frameworks in their own infrastructure [2]. These projects have slightly different focuses and are based on different languages, but all part of a blockchain ecosystem.

This project will explore some of the possibilities to integrate blockchain techniques with IoT devices, and how it may improve security. By using the resources from Hyperledger, a prototype will be constructed to try one of the existing framework in order to see how applicable it is in the home IoT network scenario.

We also provide precompiled binaries of Hyperledger Fabric for the AArch64/ARM64 architecture that are compatible with the Raspberry Pi 2/3/4 models of single-board computers, as well as instructions on how to set up the test network provided by Hyperledger. Both the binaries and the instructions are available at https://github.com/busan15/fabric-binaries-pi.

II. Background

A. Blockchain

Blockchain technology first appeared with the creation of Bitcoin in 2009. It consists of a chain of blocks, each containing a reference to the previous block and some data [3]. The reference to the previous block also consists of a cryptographic hash, which will not be valid if the previous block has been modified. This protects both against corrupted data and intentional modification, and essentially makes the blockchain an append-only data store.

Some cryptocurrencies, such as Ethereum[1], have added support for smart contracts, the ability to run code in a distributed fashion in the blockchain network. These could be used in conjunction with the currency to provide an independently auditable way of providing a service (through the code deployed to the blockchain) in exchange for a set amount of currency [3].

While their initial usage was limited to cryptocurrencies, blockchains have found use in other areas where a decentralized, peer-to-peer data store is needed, such as for banking information [4], radio bandwidth allocation [5] and supply chain management for businesses [6]. A demonstration of a slightly less business-oriented use of blockchains is the virtual cat trading game *CryptoKitties*[2], which amusingly managed to amass enough users to impact the performance of the Ethereum blockchain network backing the service [7].

B. The Hyperledger project

As mentioned earlier, the Hyperledger project is a collaborative open-source blockchain project lead by the Linux Foundation. It consists of multiple sub-projects targeted towards, for example, digital identities, credential storage, publicly accessible networks, and private networks [8].

As of the beginning of 2020, the main projects for distributed ledger systems (which a blockchain is) are [8]:

- ***Hyperledger Besu***
  An Ethereum client for enterprises, public/private networks with different choices for verification algorithms.
- ***Hyperledger Burrow***
  Blockchain solution focused on speed, simplicity and portability. Mainly targeted towards public networks.
- ***Hyperledger Fabric***
  A general purpose solution for developing modular applications. The method used to achieve consensus in the network is stated to be privacy-preserving while maintaining performance.
- ***Hyperledger Indy***
  Tools and components for managing digital identities.
- ***Hyperledger Iroha***
  Stated to be an easy to use and modular system for blockchain applications.
- ***Hyperledger Sawtooth***
  Flexible and modular solution with a separation between the blockchain network and developed applications.

Some of these will be examined in more depth in the prestudy section.

III. Prestudy

A. Existing work

As both blockchain and IoT are currently popular topics, there are quite a few existing works combining them as well. Although not all of them in regards to smart home networks. Lombardi et al. investigate how you could use blockchain in an IoT-aided smart grid to enable prosumers to directly trade energy between themselves without a centralized third party [9]. By using blockchain technology and smart contracts, it is possible to highly increase security and make a fully decentralized smart grid. Whereas this paper includes both IoT devices, blockchain, and smart contracts it is different from the smart home scenario.

There has been some unofficial work done as well, people who try out this technique and document it to their private blog or website. One person who has done this is Joe Motacek, who experimented with Hyperledger Fabric on Raspberry Pies together with Docker swarm two years ago [10]. Since then Fabric has been further developed, but this could have been some of the earlier attempts to get Fabric working on Raspberry Pi.

Most people have used Hyperledger Fabric together with Docker which makes it easier to set up and get started. This has resulted in multiple docker images for Fabric with different modifications and where some of them where compiled towards Raspberry Pi. Because Raspberry Pi has an ARM processor architecture, only some of them worked on Raspberry Pi.

B. Choosing the Hyperledger platform to use

The first Hyperledger platform we looked at was Iroha which seemed to have good potential. It was the suggested platform for this project although not explicitly required to use. Iroha is one of the furthest developed platforms in the Hyperledger ecosystem. It is based on C++ and has a role-based access control which is something that could be beneficial for this project. It has a modular design and support for assets and identity management. While these features might be useful it might not be the best match for our given project.

Nevertheless, an attempt was made at the beginning of the project to compile Iroha for the ARM architecture of the Raspberry Pi in order to compare Iroha and Fabric. However, Iroha depended on several dependencies that also had to be compiled, and many of the configuration scripts were written assuming that a more common x86 architecture was used. This created many issues, and after 1-2 weeks it still only compiled partially. At this point, an old build of Iroha was found for the Raspberry Pi, but the documentation around this build was very lacking.

According to Iroha's documentation, some of the areas where it could be used are managing digital assets or identity for applications such as payment systems, national IDs, or logistics, just to mention a few.

The second platform that was looked at was Fabric. Similar to Iroha, Fabric is also one of the most developed platforms in the Hyperledger ecosystem. It was designed for enterprise use with privacy and confidentiality of transactions in mind already during the design phase instead of adapting an existing platform. With the ability to support "chaincode" (Fabric's variant of smart contracts) that are written in general-purpose programming languages like Java, Go and Node.js it makes developing easier as you don't need to learn a new language like Ethereum's "Solidity". And unlike Bitcoin and Ethereum that

[1]https://ethereum.org/

[2]https://www.cryptokitties.co/

are public permissionless networks, Fabric is a permissioned network meaning that the participants are known to each other. Although the participants don't necessarily have to trust each other completely, like between business competitors, but can still operate safely over the same network with knowledge of one another. Hyperledger also state that Fabric should have low latency to meet the needs for enterprise usage, although not closely specified. It should also be highly modular and configurable, which allows it to be adapted towards IoT.

A beneficial feature that Fabric has is that it can work without a native cryptocurrency and therefore avoid costly mining execution, which is very good for low resource devices like in an IoT network.

Ban et al. did a good comparison between some of Hyperledger's platforms [11]. The platforms they compared were Fabric, Iroha, Sawtooth, Indy, and Burrow, which at the time were the existing frameworks. However, due to the fact that not all platforms were developed enough for testing, they could only do their own evaluation on two of them; Fabric and Sawtooth. They did, however, include all of them in their feature comparison. By the looks of it, Sawtooth could also be a good platform as it is highly customizable. Although, one of the major drawbacks it had was the lack of good documentation which makes it a lot harder to get working.

In the end, Fabric was the framework that seemed to be the most suited for this project.

IV. Method

A. Hardware

The hardware that was used in this project was two Raspberry Pi 4 to represent IoT devices. A few Raspberry Pi Zeros were also available, but these had a different architecture (only supporting 32-bit applications) from the Raspberry Pi 4 which would have resulted in more time spent on making the platform run on all devices rather than the project itself.

B. Constructing the prototype

The first step of constructing the prototype was to look through the documentation[3] for Hyperledger Fabric to see how it was intended to be used.

When attempting to follow the installation instructions in the documentation, it appeared that there were no official binaries available for the ARM architecture that the Raspberry Pi is based on, a similar situation to the one with Iroha mentioned in the prestudy. Some unofficial builds were available[4], but these were from version 1.2, which was several years old. The Python client libraries[5] from Hyperledger only supported version 1.4 and newer, which meant that another language with support for older versions would have had to be chosen. As that would have lead to spending time to get familiar with a new language and missing out on features released after version 1.2, an effort was made to compile Fabric from source for the AArch64/ARM64 architecture supported by the Raspberry Pi.

This was not a very well documented process, and like for Iroha, many parts of the compilation process assumed that the x86 architecture was used. After fixing several configuration files and trying to get things to compile for a couple of days, a point was reached where all of the components of Fabric seemed to be working on the Raspberry Pi[6].

To ensure that the compiled programs were working as intended, the test network[7] provided by Hyperledger was set up and tested according to the instructions provided. This also did not work due to the switch to ARM, but after a few hours of troubleshooting, the test blockchain was up and running. This network made use of several Docker containers simulating network nodes (peers) and an orderer node, which determines the order in which incoming transactions are written to the blockchain [12]. The prototype setup is shown in Figure 1. While these are currently running on the same physical host, the containers communicate using standard TCP and UDP protocols. Thus, the prototype blockchain network will also work with physically separate hosts, as long as these can communicate over a network link. However, the instructions provided by Hyperledger assumes that only one host is used, so the setup scripts would have to be modified to work with additional hosts.

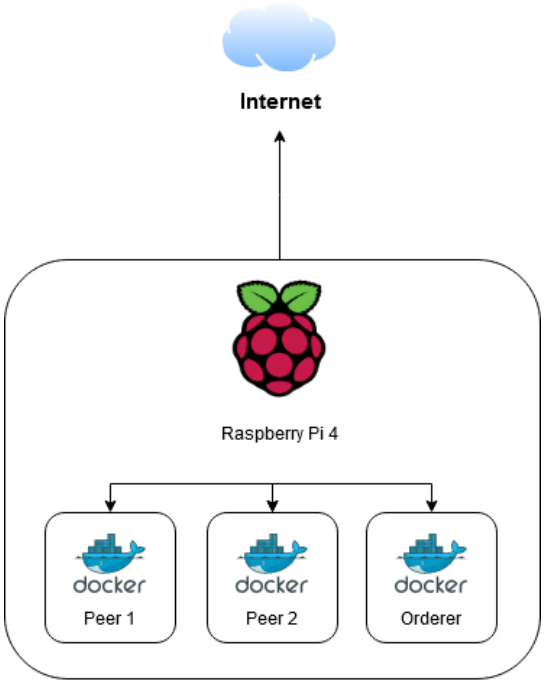

Fig. 1. Prototype setup using Docker on a Raspberry Pi

To interact with this network, the Python client libraries were used to create a program that could send requests to a peer in the blockchain network, and trigger a function to be called whenever a new transaction occurred in the network. However, these libraries were very poorly documented, which

[3]https://hyperledger-fabric.readthedocs.io/en/release-2.0/
[4]https://hub.docker.com/r/pesicsasa/fabric-peer
[5]https://github.com/hyperledger/fabric-sdk-py

[6]The compiled binaries and Docker images are made available at https://github.com/busan15/fabric-binaries-pi
[7]https://hyperledger-fabric.readthedocs.io/en/release-2.1/test_network.html

made it difficult to construct more complex interactions with the blockchain.

C. Research questions

Using the prototype and published resources such as documentation and papers, the following questions will be answered:

1) Securing against unauthorized access: Examine how the system can be secured from an attacker with access to large amounts of processing power, or malicious guests on the same network. A known issue with the Bitcoin network (and other public blockchains) is that a node which controls more than 50% of the network's processing power can block new transactions from taking place, as well as reverse previously verified transactions in an attack known as the 51%-attack [13].

2) Identifying data leaks: If possible, identify which data can leak to other parties regarding events and timestamps from devices. This could be done from the perspective of an outside user by examining what data can be accessed by a client that has not been authorized for inclusion in the blockchain network.

3) The need for Internet access: Does the system require access to the Internet to function? If so, what restrictions are placed on the system when the Internet connection is nonfunctional? If not, would there be any benefits of using the Internet to augment the system in any way?

V. Result

A. Constructing the prototype

With the issues mentioned in the method section, there was not enough time to construct a prototype that resembles a complete IoT solution. Our prototype supports adding new devices such as lights or sensors, as long as they can be represented by key-value pairs (e.g. {"light_1_status": "on"}). These are stored as part of the blockchain network through the distributed database system built into Fabric. The devices can then be controlled through functions that change the value of these key-value pairs for a given device. Whenever such a change occurs, all the other clients on the network are notified of this and can take action to, for example, turn off a connected light bulb or update a user interface. While no extensive performance testing was done, the delay from issuing a command to when the other client displays the new data was around 1-2 seconds. Each command sent required approximately 5 kB of disk space when stored in the blockchain, with the initial block taking up about 90 kB.

All these functions are implemented as a smart contract, which gives the benefit of not having to set up a central server for managing the state of the devices in the network. Authentication and protection against unauthorized changes to the data store are also provided by the blockchain network thanks to the consensus system built into Fabric, which ensures that a majority of the connected clients have to agree on the output of every command sent to the network.

An overview of the network architecture is shown in Figure 2. As described in the method section, the peers interact with each other and the orderer node through the blockchain network, which is built on top of TCP and UDP. All peers have access to a decentralized blockchain database, which is used to store data about the state of each IoT device that is connected. The peers can then be connected to these IoT devices over a radio link, acting as a bridge to the rest of the network, or over Ethernet for devices that support this. A client can then interact with a peer over TCP+TLS to forward commands to a device.

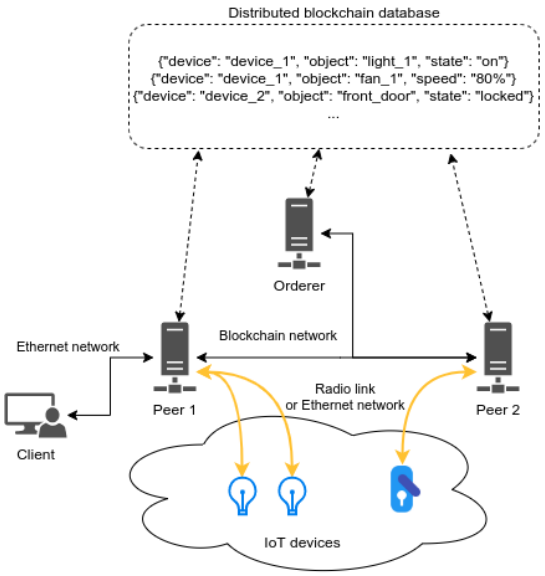

Fig. 2. Overview of blockchain network

B. Securing against unauthorized access

Fabric provides a way of ensuring that transactions submitted to the blockchain network originate from a trusted client, namely through certificates issued by a local certificate authority (CA) [14]. This certificate infrastructure forms a Public Key Infrastructure (PKI) which allows clients to verify that another device has been authorized by the CA. The PKI infrastructure requires some setup in the form of having to copy the issued certificates to the target device over a secure channel, but when set up it should provide a way of verifying a device's identity without having to update all existing devices whenever a new client device is added to the network.

Since Fabric networks are permissioned and private, as opposed to cryptocurrencies that use unpermissioned and public networks, an attacker would have to brute-force the private key of a client's certificate in order to be able to interact with the network. The CA provided with Fabric supports Elliptic Curve Digital Signature Algorithm (ECDSA) signatures of up to 521 bits with SHA512 [15], which should provide an adequate level of security. ECDSA is also an alternative for securing HTTPS websites, in the form of Transport Layer Security (TLS), and is generally considered to be resistant to attacks if implemented properly [16].

C. Identifying data leaks

While no experiments were performed, the structure of the PKI as discussed above should provide protection against

unauthorized requests leading to the disclosure of private information. The nodes in the network can also be configured to use TLS when communicating with each other or a client application [17]. This provides security against attackers eavesdropping on the network traffic, as it will all be encrypted.

Optionally, the network nodes can require a client application to present a certificate as well, ensuring that an attacker without a CA issued certificate cannot imitate a client application and gain access to the network through a trusted network peer.

D. The need for Internet access

The prototype constructed does not rely on Internet access, and does not currently send any data to the Internet. When implementing a real IoT solution, it may be desirable to control devices in the home when traveling, and this could be realized by opening up access to the blockchain network from the Internet. This will increase the attack surface of the system, but the authentication built into the blockchain peers should prevent any unauthorized access. To simplify the process of enabling access to the network for users behind a firewall or Network Address Translation (NAT), a relay server could be used to forward traffic between the blockchain network and the user. If desired, this relay could also participate in the blockchain network to provide redundancy if another node goes offline, or as a backup for data storage.

VI. Evaluation

A. Performance and resource use

As mentioned in the prototype section of the results, the latency when sending a command from the sender to the receiver through the blockchain network was around 1-2 seconds. This was for a small network with only two devices, so the latency may increase when more devices are added, and/or when more data has been added to the blockchain. This delay may be acceptable for simple interactions such as turning off a light, but if multiple messages need to be sent to, for example, turn off multiple lights or messages are sent in quick succession like when dimming a light gradually, then the performance may fall below an acceptable level. Further testing may be needed to conclude if the delay is per message or if there is an overhead to establish a communication channel from the client application to the network node. A method for combining multiple messages into one could also be developed, which would help in the case where multiple devices need to be controlled at once.

Storage-wise, Fabric seems to have very modest requirements for each block added to the blockchain. While we do not know how this will scale with more devices and nodes in the network, a few kilobytes per event will mean that even devices with smaller storage capacity will be able to store a fairly long sequence of events. One issue though is what actions that can be taken when that storage eventually fills up. Longer chains of blocks could maybe lead to longer processing time for new transactions, and with every command sent the available storage diminishes. If new transactions are to be verified, enough nodes to form a consensus might have to store the entire blockchain. Since all blocks refer to the previous block, deleting blocks at the beginning of the chain to reclaim some storage space might create issues since the chain between the initial block (genesis block) and the current one has been broken. This could perhaps be solved through some mechanism where a new genesis block can be created, or if this is not possible, nodes with fewer resources might decide to only store a part of the blockchain. However, these will then not be able to verify transactions, so enough nodes with more storage must be available in order to achieve consensus.

B. Data leaks and privacy

While data should not be able to leak to outside devices thanks to the encryption provided by TLS, all devices in the blockchain network still have access to all data in the network. A feature in Fabric that could help separate more sensitive data from certain users/nodes is the availability of different blockchain channels. These allow for private communication between a set of nodes, with each channel having a separate transaction system and blockchain [18]. Each node can be a member of multiple channels, but data cannot flow between different channels. As stated in the documentation, this enables sensitive and non-sensitive information to coexist within the same Fabric network. This could be useful in an IoT scenario where devices such as lights and sensors should be separated from devices managing more sensitive functions such as door locks and cameras, these could then be separated using channels and controlled from a smartphone which is a member of both channels. Access control could be implemented in a similar way, where a hotel guest might get access to a room-specific channel that only contains devices located in their room, or where students at a university can control the lights in study rooms but not the ventilation, etc.

VII. Related works

As mentioned in the prestudy, there are some existing works related to this project. However not many regarding IoT devices for home networks using Hyperledger's platform specifically. Dorri et al. investigates almost the exact use case as we do but without Hyperledger [19]. They create a smart home network consisting of IoT devices and a miner, which are in charge of processing and validating transactions. By having this miner that has more hardware resources, they manage to do delegate most of the blockchain calculations and processing that otherwise had to be done on each individual device. It's an interesting approach that by the looks of it solves problems regarding the limited hardware resources and can maintain a good security standard. However, this approach makes it less distributed and rely on a more centralized computer to do the actual blockchain calculation which is something that differs from our solution.

VIII. Conclusion

The project was meant to give an insight and explore in what possibilities and challenges there is using blockchain in

an IoT home network. There exists many different blockchain platforms that could be used to create a smart home network on, all with different pros and cons. This project only looked at a limited subset of platforms and only from Hyperledger's blockchain platforms. Based on those, we believe that Hyperledger Fabric is currently the one best suited for an IoT home network environment. But for it to reach greater potential, it needs to be further developed towards IoT devices as these devices often have limited hardware resources. As mentioned in the method, it was not straight forward to get it working on a Raspberry Pi 4 and will need more work to function well on even more limited hardware.

There are still questions that are relevant to answer, which we did not have time to address.

- Is the response time when interacting with devices acceptable? In order to be useful there can't be a noticeable delay when interacting with different products. Is there a limit in blockchain itself or could it be optimized further?
- If the blockchain is storing all history as a log, how could this be dealt with in order to preserve privacy?
- Is it possible to make a strong separation between data that should remain local and what might be shared outside the network? (Companies, other users etc.)
- How could data sharing and storing be transparent to let the owner know what data is shared while still keep data confidential to prevent unauthorized access?
- The owner of the devices should still be in control of what devices are added, removed and changed. How can the network differentiate a guest with good knowledge and resources from the owner?

For future projects, creating a prototype network based on Raspberry Pi and evaluating Hyperledgers blockchain framework might have to be separated into different projects. Since it takes quite a lot of time to set up the hardware, install the software, and then configure a blockchain network, a thesis project might have a large enough scope that a proper network can be set up and measurements or more in-depth analysis be performed. Using a pre-built network such as the Fabric test network on a platform which has binaries available might be a good starting point for measuring latency, analyzing the network traffic, or performing other measurements.

Even though we did not have time to test a fully operational IoT home network, we believe that blockchain has a good potential within this area.

References

[1] Pawani Porambage et al. "The quest for privacy in the internet of things". In: *IEEE Cloud Computing* 3.2 (2016), pp. 36–45.

[2] The Linux Foundation. *Hyperledger – Open Source Blockchain Technologies*. 2020. URL: https://www.hyperledger.org (visited on 2020-04-05).

[3] Melanie Swan. *Blockchain: Blueprint for a new economy*. "O'Reilly Media, Inc.", 2015.

[4] Jemima Kelly. "Banks adopting blockchain 'dramatically faster' than expected: IBM". In: *Reuters* (2016-09-28). URL: https://www.reuters.com/article/us-tech-blockchain-ibm-idUSKCN11Y28D (visited on 2020-04-28).

[5] Khashayar Kotobi and Sven G Bilen. "Secure blockchains for dynamic spectrum access: A decentralized database in moving cognitive radio networks enhances security and user access". In: *IEEE Vehicular Technology Magazine* 13.1 (2018), pp. 32–39.

[6] Hyperledger. *Announcing Hyperledger Grid, a new project to help build and deliver supply chain solutions!* 2019. URL: https://www.hyperledger.org/blog/2019/01/22/announcing-hyperledger-grid-a-new-project-to-help-build-and-deliver-supply-chain-solutions (visited on 2020-04-28).

[7] BBC. "CryptoKitties craze slows down transactions on Ethereum". In: *BBC.com* (2017-12-05). URL: https://www.bbc.com/news/technology-42237162 (visited on 2020-04-28).

[8] Hyperledger. *Blockchain Technology Projects – Hyperledger*. 2020. URL: https://www.hyperledger.org/projects (visited on 2020-04-28).

[9] Federico Lombardi et al. "A Blockchain-based Infrastructure for Reliable and Cost-effective IoT-aided Smart Grids". In: *Living in the Internet of Things Conference: Cybersecurity of the IoT - A PETRAS*. 2018-01. DOI: 10.1049/cp.2018.0042.

[10] Joe Motacek. *Hyperledger Fabric v1.0 on a Raspberry Pi Docker Swarm - Part 1 — Joe Motacek*. 2017. URL: https://www.joemotacek.com/hyperledger-fabric-v1-0-on-a-raspberry-pi-docker-swarm-part-1/ (visited on 2020-05-04).

[11] Tran Quy Ban et al. "Survey of Hyperledger Blockchain Frameworks: Case Study in FPT University's Cryptocurrency Wallets". In: *Proceedings of the 2019 8th International Conference on Software and Computer Applications*. ICSCA '19. Penang, Malaysia: Association for Computing Machinery, 2019, pp. 472–480. ISBN: 9781450365734. DOI: 10.1145/3316615.3316671. URL: https://doi.org/10.1145/3316615.3316671.

[12] Hyperledger. *The Ordering Service – Hyperledger Fabric documentation*. 2020. URL: https://hyperledger-fabric.readthedocs.io/en/release-2.1/orderer/ordering_service.html (visited on 2020-05-11).

[13] Congcong Ye et al. "Analysis of security in blockchain: Case study in 51%-attack detecting". In: *2018 5th International Conference on Dependable Systems and Their Applications (DSA)*. IEEE. 2018, pp. 15–24.

[14] Hyperledger. *Identity – Hyperledger Fabric documentation*. 2020. URL: https://hyperledger-fabric.readthedocs.io/en/release-2.1/identity/identity.html (visited on 2020-05-03).

[15] Hyperledger. *Fabric CA User's Guide*. 2020. URL: https://hyperledger-fabric-ca.readthedocs.io/en/release-1.4/users-guide.html (visited on 2020-05-03).

[16] Don Johnson, Alfred Menezes, and Scott Vanstone. "The elliptic curve digital signature algorithm (ECDSA)". In: *International journal of information security* 1.1 (2001), pp. 36–63.

[17] Hyperledger. *Securing Communication With Transport Layer Security (TLS) – Hyperledger Fabric documentation*. 2020. URL: https://hyperledger-fabric.readthedocs.io/en/release-2.0/enable_tls.html (visited on 2020-05-03).

[18] Hyperledger. *Channels – Hyperledger Fabric documentation*. 2020. URL: https://hyperledger-fabric.readthedocs.io/en/release-2.1/channels.html (visited on 2020-05-04).

[19] Ali Dorri et al. "Blockchain for IoT Security and Privacy: The Case Study of a Smart Home". In: *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 2017-03. DOI: 10.1109/PERCOMW.2017.7917634.