A lightweight blockchain based two factor authentication mechanism for LoRaWAN join procedure

Securing the LoRaWAN Join Procedure using Blockchains

Syed Muhammad Danish[a,d,*], Marios Lestas[b,], Waqar Asif[c], Hassaan Khaliq Qureshi[a], Muttukrishnan Rajarajan[c], Kaiwen Zhang[d]

[a]*National University of Sciences & Technology (NUST), Islamabad, Pakistan*
[b]*Frederick University, Cyprus*
[c]*School of Engineering and Mathematical Sciences, City, University of London, UK.*
[d]*École de technologie supérieure, Canada*

Abstract

LoRaWAN, part of the long range Internet of Things (IoT) technologies, is a Low Power Wide Area (LPWA) protocol that enables communication between the battery operated resource constrained devices. Although LoRaWAN provides system level security, these networks are based on a basic assumption of trust between the network server and the user, which as a result empowers a network server with undue authorities. Moreover, the nature of the LoRaWAN join procedure, which is a key component for ensuring network operation, is highly susceptible to jamming and replay attacks through slight manipulation, thus making the network vulnerable. In order to address these deficiencies, we present a blockchain based distributed framework for the LoRaWAN join procedure, to develop a secure and trusted authentication system within LoRaWAN networks. The proposed framework eliminates the jamming and replay attack threats against the LoRaWAN join procedure and in addition builds trust among LoRa end devices and network servers. The framework is endorsed by a Proof of Concept (PoC) implementation using the Ethereum blockchain which is used to evaluate the proposed sys-

*Corresponding author
Email addresses: sdanish.msee16seecs@seecs.edu.pk, syed-muhammad.danish.1@ens.etsmtl.ca (Syed Muhammad Danish), eng.lm@frederick.ac.cy (Marios Lestas), waqar.asif@city.ac.uk (Waqar Asif), hassaan.khaliq@seecs.nust.edu.pk (Hassaan Khaliq Qureshi), r.muttukrishnan@city.ac.uk (Muttukrishnan Rajarajan), kaiwen.zhang@etsmtl.ca (Kaiwen Zhang)

tem in realistic LoRaWAN network scenarios. The considered performance metrics are the achieved latency and throughput at the join server, which are measured as the number of concurrent clients and join request messages increase. The simulations results indicate that the system achieves efficient system performance up to an upper bound on the load level which involves 30 join requests from 1000 concurrent clients. However, the security and trust advantages that the blockchain technology offers, comes at the cost of performance deterioration for loads exceeding that level. This can be resolved by introducing additional join servers. A lightweight, cost effective blockchain based two factor authentication framework can also be employed in LoRaWAN networks when these are characterized by small number of networks servers and LoRa end devices. We demonstrate through simulations that this approach incurs additional delays and can be preferred for systems with no strict requirements of throughput and latency.

Keywords: Blockchain, LoRaWAN network, Join Procedure, IoT security, Authentication

---

1. Introduction

According to Ericsson's mobility report [1], there will be 18 billion Internet of Things (IoT) devices by 2022 and to anticipate for this prospect, market spending for IoT related services is constantly rising. Presently, IoTs are experiencing major deployments both in consumer applications as well as in industry. With the increasing number of IoT devices, security and privacy of user's data have become a key concern [2], [3]. There is a need to introduce new mechanisms to fulfil the IoT security requirement but, due to the lack of available computational resources, these devices cannot make use of the traditional security mechanisms [3].

IoT devices are often power constrained and to account for the design challenges emanating from this characteristic, the Low Power Wide Area Network (LPWAN) has been introduced [4]. LoRaWAN is an LPWAN IoT technology which enables bidirectional, long range communication between battery operated, low power and low memory IoT devices. For secure communication in LoRaWAN networks, integrity and confidentiality is ensured by end-to-end encryption and the LoRaWAN join procedure is used to provide authentication to LoRa end devices [5].

The LoRaWAN join procedure, which is also referred to as the Over

The Air Activation (OTAA) procedure, involves an exchange of a set of authentication messages between the network server and LoRa end devices. The OTAA procedure has been shown to be vulnerable to both jamming [6], [7], [8] and replay attacks [9]. These attacks are directly associated with the DevNonce value in the join request message. DevNonce values are unique in nature, associated with every end device and generated for every new join request. An adversary can trigger a jamming attack by simply limiting the randomness of the generated DevNonce value, thus resulting in a denial of Service (DoS) attack for the end device [8]. As the newly generated device is not unique, the network server would not register the device, thus jamming operation. On the other hand, in case of the replay attack, an eavesdropping adversary can capture the DevNonce of a legitimate user and replay it to confuse the server, thus interfering with the operation of a legitimate end device [9]. This DevNonce value also becomes an issue due to the centralized nature of the network servers. A network server has the complete authority to change the DevNonce value for any device. Therefore, this becomes an issue of trust between the end device and the server and gives the network server an undue authority to reject legitimate users for personal gains.

LoRaWAN networks are developed to interconnect multiple IoT devices and in order to make this successful, a unified distributed authority system needs to be implemented that can overcome the known jamming and replay attacks. This paper proposes a new LoRaWAN join procedure architecture, that makes use of the well known blockchain network solution. The blockchain framework is used among the server nodes who have the required computational capabilities and it allows the formulation of a distributed network validation and authority system. In contrast to the current LoRaWAN join procedure protocol [5], the proposed blockchain based solution offers the following advantages to the LoRaWAN join procedure:

- *end-to-end encryption:* in the current LoRaWAN specifications [5], join and re-join request messages are not encrypted. In our solution, join request messages will be encrypted; contents of the join request message will not be exposed to other entities which will provide extra security to the LoRaWAN join procedure.

- *lightweight:* no modifications are required in the LoRa end devices to adopt the proposed solution as network and application session keys will be generated by the existing LoRaWAN specification. However,

small changes need to be incurred to the join request message format to meet the specifications of the proposed solution.

- *accessibility:* as a result of all information being distributed, failure of some network servers will not prevent access to the device authentication information saved in the blockchain network.

- *tamper-proof:* encrypted device information will be saved in the blockchain network and thus, no single network server will be allowed to delete or change this information without the consent of the other network servers.

- *confidentiality:* the proposed architecture will hide the LoRa end device information since join request messages will be encrypted.

In particular, the main contribution of this work is to design a new decentralized LoRaWAN join procedure framework using the blockchain technology. In the proposed solution, the blockchain technology is not integrated in the LoRa end devices which increases its utility in large scale LoRaWAN networks. In addition, to simplify the implemented processes within the blockchain network, our solution utilizes a smart contract to store the LoRa end device information in the blockchain network. Moreover, the join accept messages from the join server, after fetching information from the blockchain network, forward it to the LoRa end devices in real time. To summarize, the proposed solution is specifically designed to provide more security and trust to the LoRaWAN join procedure as well as to provide scalability.

The framework is endorsed by a Proof of Concept (PoC) implementation using the Ethereum blockchain which is used to evaluate the proposed system in realistic LoRaWAN network scenarios. The considered performance metrics are the achieved latency and throughput at the join server, which are measured as the number of concurrent clients and join request messages increase. The simulations results indicate that the system achieves efficient system performance up to an upper bound on the load level which involves 30 join requests from 1000 concurrent clients. However, the security and trust advantages that the blockchain technology offers, comes at the cost of performance deterioration for loads exceeding that level. This can be resolved by introducing additional join servers. It must be noted that the simulation scenarios involve relatively high loads of join request messages at the network server per client, which are considered to test the limits of

the proposed system. However, in practice, every LoRa end device performs the join procedure once or twice a day [8], rendering our system suitable for implementation in LoRaWAN networks to enhance security and trust. When the implementation cost is of primary concern, a lightweight, cost effective blockchain based two factor authentication framework may also be considered. We demonstrate through simulations that this approach incurs additional delays and can be preferred for systems with no strict requirements of throughput and latency.

The rest of the paper is organized as follows. In Section II, prior work related to blockchains and the LoRaWAN join procedure is presented. Section III describes the architecture of the proposed blockchain based solution for the LoRaWAN join procedure while, Section IV presents the security analysis of our proposed framework. Implementation details of the system are provided in section V while Section VI presents the proof of concept Ethereum based implementation which is used to evaluate the proposed solution. The main evaluation results are summarized in the same section. Section VII summarizes the key conclusions of our paper.

2. Related Work

LoRaWAN is a relatively new technology and security features of the LoRaWAN protocol are being defined by the LoRa Alliance. The LoRaWAN protocol is susceptible to bit flipping attacks [11], wormhole attacks [12], jamming attacks [6], [7], [8], replay attacks [9], [13], [14] and battery exhaustion attacks [13]. The authors in [8] explain that the LoRaWAN join procedure is vulnerable to jamming attacks where the jammer limit the ability of the LoRa end device to connect to the LoRaWAN server. In [6], an intrusion detection mechanism to detect jamming attack against the LoRaWAN join procedure is proposed. Similarly, the disruption in the LoRaWAN communication as a result of the jammer operating near the LoRa end device is explained in [12]. Moreover, authors in [14] set up an attack scenario and demonstrate that the LoRaWAN join request is vulnerable to replay attacks.

Blockchain has been recently considered as an effective tool with which to improve security in IoT networks. It was introduced with Bitcoin [15] to solve the double-spending problem and is currently widely adopted for a large number of crypto-currencies, such as Ethereum, Ripple, EOS, Zcash etc. [16]. The nodes on the blockchain network interact with each other via a pair of private/public keys [17]. One key emerging use case of blockchain technology

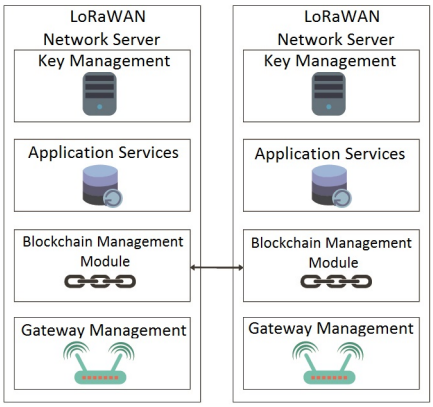


Figure 1: Blockchain functionality in LoRaWAN network servers

involves "smart contracts" [18]. Nodes on the blockchain network interact with smart contracts through the smart contract address. Authors in [19] provide an extensive literature review of how blockchains have been used in IoT networks. In [20], a blockchain based authentication mechanism is proposed for IoT networks. Authors in [21] propose a zero knowledge proof based solution to provide authentication for smart meters. A blockchain based identity framework for IoT is proposed for smart home's IoT devices using the appliance signature [22].

To the best of our knowledge, the only previous work related to the proposed solution is the work in [10], which proposes a blockchain based framework for the LoRaWAN network to build trust among the LoRaWAN network entities. However, there are several differences between the work in [10] and ours. First, our work focuses on the authentication mechanism in the LoRaWAN network. We propose a blockchain based solution to develop a secure and trusted authentication system. In contrast, the work in [10] focuses on data integrity and provides the indisputable mechanism to verify that the data of a transaction has existed at a specific time within the LoRaWAN network. Secondly, the proposed framework differs from the work in [10] as multiple join servers have been used to provide scalability and a smart contract has been deployed to save device information in the blockchain network. The work in [10] fails to provide the required amount of security for the LoRaWAN join procedure, which still remains an open issue. The proposed work address this key issue and unlike [10], highlights the success of the proposed approach using simulations.

3. Blockchain based LoRaWAN Join Procedure

The proposed framework presents a new decentralized authentication mechanism for LoRaWAN networks where the LoRa end device information is authenticated and stored using blockchain technology.

Since LoRa end devices and gateways are resource constrained outdoor deployed IoT devices, they cannot accommodate complex blockchain functionalities. Therefore, in the proposed framework, blockchain functionality is added in peer-to-peer (P2P) LoRaWAN network servers using the management module through which different network servers will communicate to fulfill blockchain functionalities as shown in Fig. 1. Also, each network server works as a miner to save LoRa end device information within the blockchain network.

Moreover, a single smart contract is deployed on the blockchain network that defines the functions to store the LoRa end device information. Join servers interact with the smart contract to store and retrieve LoRa end device information to perform the LoRaWAN join procedure. Fig. 2 shows the proposed framework for the system under consideration.

3.1. Threat Model

In our threat model, it is assumed that all the cryptographic primitives and security protocols are inherently secure and cannot be compromised. Moreover, a global adversary $\mathcal{A}$ is considered, which can be internal or external. The adversary $\mathcal{A}$ is global if he has full information of the LoRaWAN network i.e. authentication and data communication information. Internal indicates that the adversary has control over the LoRaWAN network server and can modify the authentication information of the LoRa end nodes while, external adversary can capture all the authentication and communication messages between the LoRa end device and the LoRaWAN network server. Specifically, it is considered that the internal and external $\mathcal{A}$ can launch the following attacks in the LoRaWAN network.

- An external adversary $\mathcal{A}$ can capture the authentication messages between the LoRa end device and the LoRaWAN network server and can launch the jamming and replay attack to stop LoRa end node from joining the LoRaWAN network server.
- An internal adversary $\mathcal{A}$ can modify the authentication information in the network server to stop LoRa end device to join the LoRaWAN

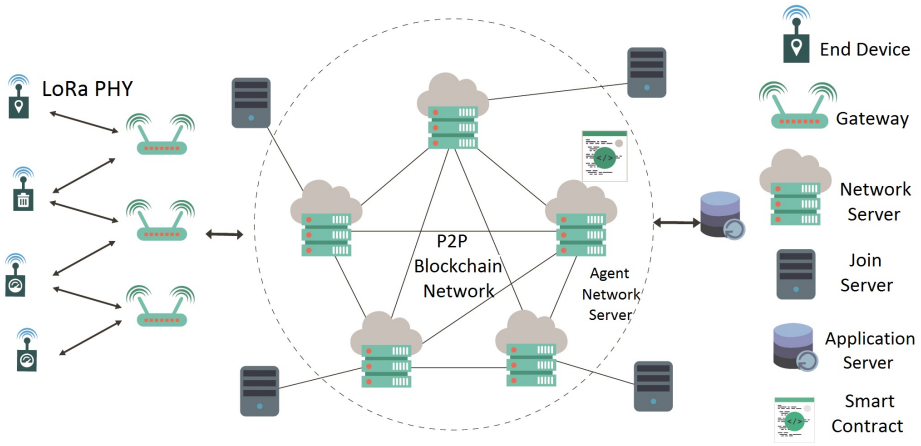


Figure 2: Blockchain architecture for the LoRaWAN join procedure

network server. This attack can also be launched by external adversary if he compromises the LoRaWAN network server.

3.2. System Architecture

The framework comprises of the following components:

3.2.1. LoRa end devices & Gateways

LoRa end devices are low power, resource and memory constrained IoT devices which communicate with the network server to perform specific tasks via the gateway. Each LoRa end node is uniquely identified at the network server via a 128 bit NwkKey (AppKey in LoRa alliance specification v1.0) [23]. The gateway then transfers the LoRa end device's data through TCP/IP connections to the network server. The gateway functions as a relay between the network server & LoRa end device and does not perform any security functionalities.

3.2.2. Join Server

The Join server communicates with the network server over the IP connections and produces the application and network session keys and distributes them to the application and network servers respectively. In the proposed framework, the main function of join server is to save or retrieve the authentication information from blockchain network. When a join server receives a join request message, it forwards it to the smart contract deployed in the

blockchain network to save or retrieve the device information based on the join request message content.

3.2.3. Smart Contract

The main function of smart contract is to read and write the authentication information in the blockchain network. The smart contract defined in this framework has two functions i.e. read and write. When join server wants to save LoRa end device information in the blockchain network, it calls the write function. Similarly, when a join server calls the read function of smart contract, it sends the block number as an input which contains LoRa end device authentication information and the corresponding smart contract fetches this from the blockchain network.

3.2.4. Agent Network Server

In the proposed framework, the main function of the agent network server is to deploy the smart contract within the blockchain network. After deploying the smart contract in the blockchain network, agent network server distributes the smart contract address in the network. This address is then used by join server to interact with the smart contract.

3.2.5. Blockchain network

Within the proposed framework, P2P connected network servers, equipped with the blockchain module, interact to form the blockchain network. The Network server's functionality is identical to the one defined for the LoRaWAN network however, blockchain modules are introduced to add the blockchain functionality in the network servers i.e. mining, transaction, smart contract etc. The miners render the network stable and secure by approving the blocks and keeping copies of the distributed ledger. The main function of this P2P blockchain network is to store the LoRa end device authentication information so that no single network server can modify it. Since the LoRaWAN network servers have high computational power, Proof of Work (PoW) consensus algorithm is employed to make blockchain networks more secure. Since, LoRaWAN network servers are public nodes and can show byzantine behavior therefore, the PoW consensus mechanism will stop network servers to register a specific LoRa end device in multiple blocks as well as fake LoRa end device identities. PoW ensures high level of security at the expense of high computational effort however, it is performed only once for a single device registration, it is not expected to incur high

computational burden and a registering device will pay a transaction fee to the mining network server.

3.3. System Interactions

The interactions between different components of the framework are shown in Fig. 3. The system interactions can be encompassed in 3 different network functions or phases:

- Bootstrapping blockchain network, in which LoRaWAN network servers create a P2P blockchain network.
- Device registration, in which a new LoRa end device is registered in the blockchain network.
- Device authentication, in which the registered LoRa end device is authenticated with the help of information saved in the blockchain network.

3.3.1. Network Setup

The first phase is to bootstrap the blockchain network through publicly available LoRaWAN network servers. In this phase, the blockchain network is created between the P2P network servers with the help of blockchain module in the LoRaWAN network servers. It should be noted that LoRaWAN network server will perform normal functionalities along with the blockchain functionalities. Once the blockchain network is created, the agent network server deploys the smart contract. After the smart contract is deployed, the agent network server retrieves the smart contract address from the blockchain network. This contract address is used by the blockchain network entities to interact with the smart contract. All the join servers interact with the contract using the contract address, to store, re-register or retrieve the LoRa end device information from the blockchain network. This step is performed only once during the lifetime of blockchain based LoRaWAN join procedure.

3.3.2. Device Registration

When a new LoRa end device joins the LoRaWAN network, its information is not saved in the blockchain network. At this point, the LoRa end device has to follow the current join procedure specification to register itself in the blockchain network. Therefore, for the first time, the LoRa end device will follow the current specification of join procedure. To join the LoRaWAN

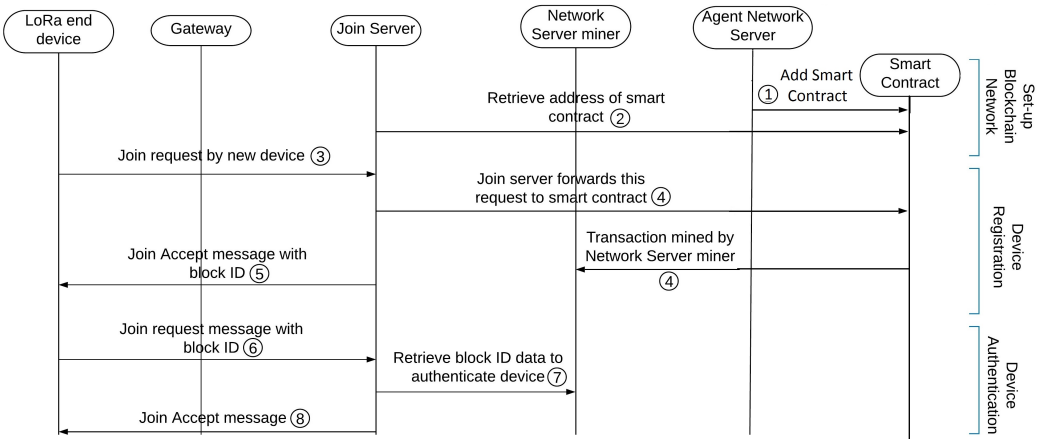


Figure 3: Blockchain network setup, device registration, device authentication

network, LoRa end device sends the join request message to the network server through gateway, by following the LoRaWAN join procedure. At this point, the join request message is not encrypted [5]. This join request reaches the join server and after checking the validity of the join request, the join server encrypts the device information using the device AES 128 bit NwkKey (AppKey) and forwards this encrypted request to the agent network server using the smart contract address. The main reason to encrypt the authentication information before writing it in the blockchain network is that since, the blockchain network is public and anyone can read the block content. Therefore, once the information is encrypted, no one will be able to see the corresponding block number of the registered LoRa end device. After the encrypted information reachers smart contract, it is then executed, mining is performed by the network servers and the LoRa end device information is stored in the blockchain network. The agent network server replies to the join server with the block-id associated with the device information. The Join server replies with an join accept message along with the block-id in which the information is saved. This join request message is encrypted with a pre-shared 128 bit AES key. Thus, the newly connected LoRa end device is registered within the blockchain network.

3.3.3. Device Authentication

This is the final step for the LoRa end device who has registered itself in the blockchain network. After following the normal LoRaWAN join procedure to register itself in the blockchain network, LoRa end node is now

ready to authenticate itself to the LoRaWAN network server through the authentication information saved in the blockchain network. Now, when the LoRa end device wants to perform the LoRaWAN join procedure for the second time, it will use the block-id along with the timestamp instead of the nonce/counter value, which was acquired in step 2 in the join accept message. In order to perform the join procedure, the LoRa end device will encrypt the block-id along with the time stamp with the AES 128 bit pre-shared key as a join request message and send it directly to the network server. After checking the block-id field in join request message, network server will forward this join request to join server and the join server will fetch the information from the blockchain network in real time and match the content of the join request message with the device information stored in the blockchain network in the associated block-id. Thus, once the LoRa end device's authentication information has been registered in the blockchain network, the device will use the block-id in the join request message every time it wants to connect to the LoRaWAN network server. The use of timestamp mitigates the replay attacks while, the absence of nonce in join request message mitigates the jamming attack. Moreover, the encrypted authentication information is saved in the blockchain network, no single network server can change this information thus, making the LoRaWAN join procedure more secure.

3.4. System Limitations

The security of LoRaWAN join procedure is improved by the introduction of blockchain technology in the LoRaWAN network, however, some technical specifications of the blockchain network pose performance limitations to the proposed solution.

Network servers have to integrate the blockchain management module within the current LoRaWAN network servers. So in order to process a transaction, network servers may include a transaction fee to mine the device information in the blockchain network. Since, LoRaWAN network servers are publicly owned networks and LoRa end nodes will be the subscriber of different publicly owned network server. Therefore, the main reason for the transaction fee is to compensate for the computational mining of LoRaWAN network server for any given LoRa end device. After the device information has been saved in the blockchain network, fetching the information using call functions by the join servers, will not incur any fee. A newly connected device has to pay a transaction fee only once to get the device information added in the blockchain network. However, when the device information is

saved, the join servers can simply use the call function to retrieve information for authentication purpose from any network server, free of charge.

Also, miners take time to add transactions in the blockchain network so when the device connects to a LoRaWAN network for the first time, the join server might incur long delays in replying to the join request message since the device information has to be mined by any of the network servers. After this, every time the LoRa end device wants to connect to a network server, the join server will reply with the join accept message on real time. Our proposed architecture authenticates LoRa end devices in real time even if the LoRa end devices are joining or leaving the LoRaWAN network rapidly. This is because of the fact that once the information is stored on the blockchain network, the information will then be fetched using the read function of the smart contract in real time, irrespective of the number of times the LoRa end device performs the join procedure.

4. Security Analysis

In order to check the security vulnerabilities of our proposed approach we employ STRIDE [24] mnemonic: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. In our proposed framework, it is assumed that all the cryptographic primitives and security protocols are secure. The reason to perform security analysis through STRIDE is to check the wide range of attacks against each and every individual entity of the proposed framework. Although, the blockchain part of our architecture provides data integrity and trust, LoRa end devices and join servers are not the part of the blockchain network and are vulnerable to multiple security threats. These threats mainly emanate from the join mechanism of the end devices, where a malicious server can spoof (imitate the join server), repudiate (refuse to generate keys), tamper (modify join request message contents), DoS or disclose LoRa end device information. Since these join servers are directly connected to the blockchain based LoRaWAN network servers, these join servers can get signed certificates from Certification Authority (CA) and network servers can verify these certificates to authenticate join servers. Similarly, a LoRa end device can be spoofed by replication of the 128-bit Nwkkey (hardcoded in the LoRa end device hardware), which can then be used to register a new device with the network servers. The distributed nature of the blockchain architecture would eliminate the possibility of such an attack by making the Nwkkey to DevNonce value mapping

available to all LoRaWAN network servers.

In addition to the above weaknesses, our proposed architecture is expected to completely eliminate the jamming attacks as well as replay attacks. Jamming attacks reduce the randomness in the DevNonce which may lead to a DoS attack however, in our framework, the DevNonce is only used for the key generation which eliminates the jamming attack threat. Moreover, the proposed system relies on a timeout field that mitigate the possibility of overhearing and then replaying the DevNonce value of a legitimate user, thus eliminating the possibility of a replay attack.

5. Implementation

A PoC implementation is developed for the blockchain based LoRaWAN join procedure framework. This implementation is used to test and evaluate the proposed framework. A brief overview regarding the PoC implementation of the LoRa end device, join server and blockchain network, is provided in this section.

5.1. Blockchain Network

In the proposed framework, P2P network servers are performing the blockchain functionalities. The Ethereum blockchain platform is selected for the PoC implementation. To keep the solution simple, a private blockchain has been chosen because it provides more reliable results while evaluating the system. However, a public blockchain, instead of a private blockchain, should be used in a realistic scenario since, LoRaWAN is a public network and device information should be open to all the join servers in LoRaWAN network.

Ethereum is a blockchain platform with a built-in Turing complete programming language called Solidity, which allows anyone to create decentralized applications and smart contracts, specifying own transaction formats and ownership [25]. Ethereum defines two types of accounts, externally owned accounts and contract accounts. Externally owned accounts are controlled by a private key, while contract accounts are controlled by the contract code. In addition to the sendTransaction function, Ethereum also defines the call function which can be invoked in a smart contract to retrieve the data stored in the blockchain free of cost, with no transaction fee. Join servers in the reference scenario use this function to retrieve authentication information of the specific LoRa end device from blockchain network.

5.1.1. Smart Contracts

In the Ethereum blockchain, smart contract is a script, written using the Solidity programming language. We have implemented a single smart contract in the blockchain network. A mapping data structure is used to store the LoRa end device authentication information in the blockchain smart contract. Mappings can be seen as hash tables which are virtually initialized such that every possible key exists and is mapped to a value whose byte-representation is all zeros.

5.1.2. Join Server

The Join server is a JavaScript interface which connects the LoRa end devices with the blockchain P2P network servers. This interface communicates with the blockchain network through JSON RPC calls which are generated via the python program serving as LoRa end device client.

5.2. LoRa End Devices

The join procedure protocol is implemented using the python programming language in which the LoRa end device generates the join request as a JSON packet. A 128 bit NwkKey (AppKey in LoRa specification v1.0) functionality was added to uniquely identify each LoRa end device's join request. The generated join request packet for a specific LoRa end device containing the block number, is sent to the join server which checks the authentication information stored in the blockchain and replies with a join accept or reject message.

6. Performance Evaluation

In this work, we evaluate the performance of the proposed approach by implementing a LoRaWAN network which uses an Ethereum based blockchain architecture [26], [27] for the distributed servers. We measure the performance based on the response time of the join server and the end devices. As the blockchain network is not undergone any changes, we choose to limit our results to the Join server and end device interaction only. In addition, since the payload delay on the LoRa physical layer depends on the spreading factor, we assume the prior arrival of the payload at the gateway. In the rest of this section, we examine the effects of introducing the blockchain capability on the LoRaWAN servers, by evaluating the performance of the resulting system in terms of the achieved throughput and overall delay.

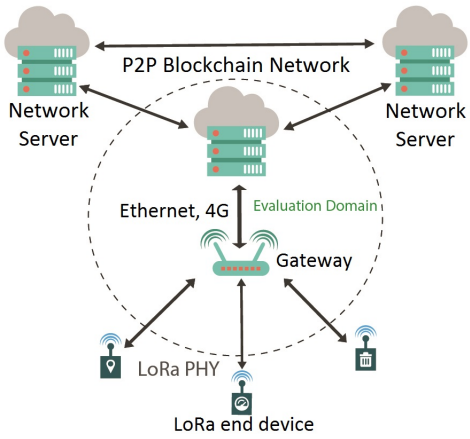


Figure 4: Evaluation Domain

6.1. System Setup

The experiments were performed on a UBUNTU 16.04 desktop with Intel(R) Core(TM) i7-6700 @ 3.40 Ghz and 16 GB RAM. A geth Ethereum client, a Go language implementation of the Ethereum protocol, was used to run the private blockchain network and to deploy the smart contract. For the purpose of easily scaling of the conducted experiments, a benchmark tool was developed which implements concurrent virtual client functionality to send join requests to the blockchain network. The time out, for the join requests is set to 15 seconds. Any delays beyond this timeout time to complete the request results in a join request failure.

6.2. Performance

The latency introduced in the join server, as a result of the blockchain system performing the LoRaWAN join procedure, together with the achieved throughput, are used as performance metrics in the conducted experiments, as shown in Fig. 4.

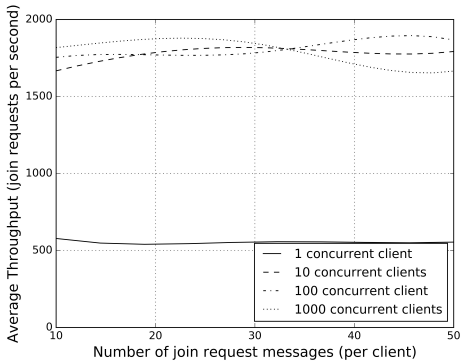
The conducted experiments involve using the benchmark tool to connect a number of virtual end devices to the join server whose performance, when connecting to the blockchain network, is then examined. In this scenario, the virtual LoRa end device clients are configured to send join request message to the join server. After the join server has received the join request messages, it uses the RPC call function to retrieve the authentication information of

the LoRa end device from blockchain network. Based on this retrieved information, it then returns the join accept or reject message to the virtual LoRa end device clients.

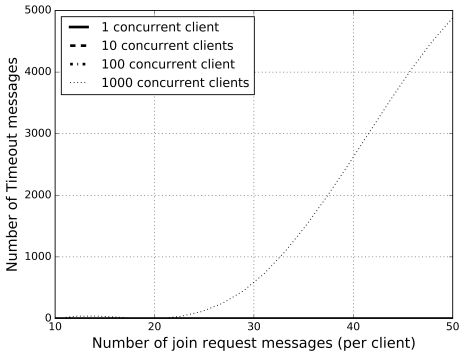
In the aforementioned scenario, the join server issues join accept message to all the virtual LoRa end device clients. Different number of concurrent clients were considered in each experiment, in the range from 1 to 1000. The presented results are averaged over 5 executions for the same experiments.

Fig. 5a shows the throughput achieved at the join server connected to blockchain network as a function of the number of join request messages per client, for different number of concurrent clients. The throughput is defined as the number of join requests processed by the join server per second. It can be seen from the graph that for 1 concurrent client, the throughput is almost constant, despite increasing the number of join request messages. The average throughput for 10 concurrent clients, generating 10 join requests is around 1666 requests per second. This increases to around 1818 requests per seconds for each client generating 30 join requests messages and remaining almost the same for 50 join requests. The average throughput for 100 concurrent clients shows a similar trend experiencing maximum throughput for 50 join requests per client. The average throughput for 1000 concurrent client was maximum for 30 join requests messages per client. However, it experiences a significant drop for more than 30 join requests per client and this decline is associated with the number of timeout messages for 1000 concurrent clients since, large number of requests requires more time to be resolved and thus, we have timeout messages as shown in Fig. 5b. Fig. 5c, presents the latency experienced at the join server in response to the same join request messages. It can be seen that for 1 and 10 concurrent clients, the join server response time is less than 1 second. However, as the concurrent clients are increased to 100, the latency at the join server increases to around 3 seconds for 50 join requests per client. For 1000 concurrent clients, the delay increases significantly as the number of join request messages generated per client increases. For more than 30 join requests per client, the delay is more than 15 seconds, resulting in join request timeouts.

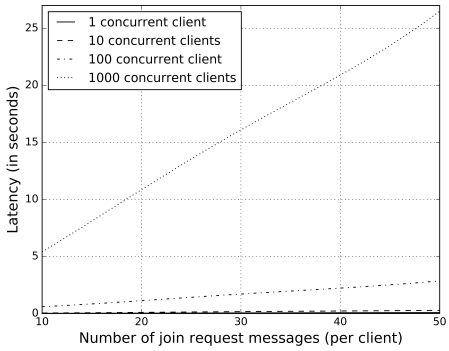
Summarizing the above findings, we can conclude that up to 1000 concurrent clients, generating 30 join request each, the blockchain network enhancement on the join server comes at minimum cost, achieving high throughput and low latency. However, beyond this limit, performance degrades significantly due to excessive delays leading to timeouts and bad use of the available resources. This problematic behavior can be resolved by introducing multi-



(a) Join server throughput



(b) Join server timeout messages



(c) Join server latency

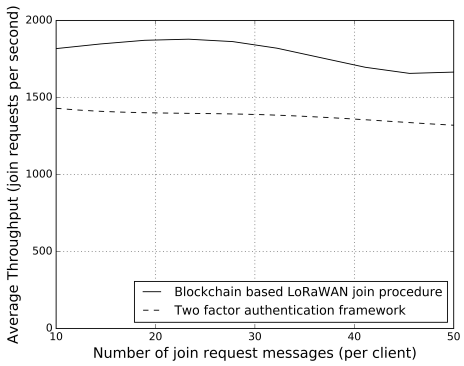
Figure 5: Performance analysis of blockchain based LoRaWAN join procedure

ple join servers. In the current PoC implementation, we have considered one join server working concurrently with LoRaWAN network server however, in practical LoRaWAN network, multiple join servers are operated concurrently with LoRaWAN network servers [29] to increase the scalability and reliability, which make our system design feasible for large scale LoRaWAN network. This study can be used as a baseline to decide as to when such multiple servers deployment is necessary to meet the design requirements i.e. throughput and latency, of blockchain based LoRaWAN network framework. This decision can be made to optimize the design. It must be noted that the simulation scenarios involve relatively high loads of join request messages at the network server per client, which are considered to test the limits of the proposed system. However, in practice, every LoRa end device performs the join procedure once or twice a day [8], rendering our system suitable for implementation in LoRaWAN networks to enhance security and trust.

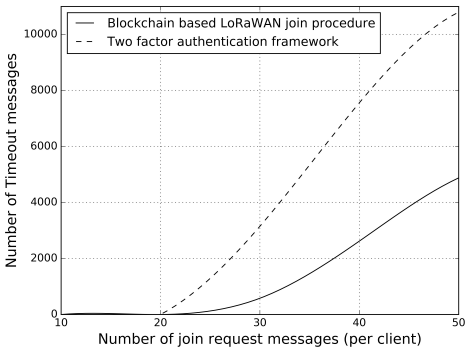
6.3. Two factor authentication framework

The aforementioned proposed framework is specifically designed for large scale LoRaWAN networks with large number of LoRa end devices and network servers, to fulfill strict throughput and latency requirements in LoRaWAN network. However, when the LoRaWAN network is characterized by small number of network servers and LoRa end devices with no strict requirements of throughput and latency, a lightweight cost effective blockchain based solution might be meaningful.

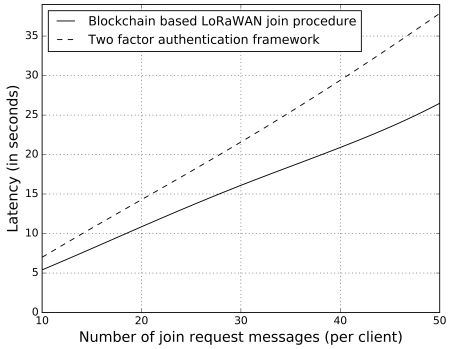
We refer to such an approach as a two-factor authentication framework due to the fact that, in addition to the normal LoRaWAN join procedure protocol, a device is also authenticated based on the information stored in the blockchain network. In the proposed two factor authentication framework, the LoRa end device follows the LoRa Alliance specified join procedure and also save the authentication information in the blockchain network. Therefore, whenever the device wants to connect to the LoRaWAN network server, it follows the join procedure. In addition to this, it also checks the authentication information saved in the blockchain network as a second factor of authentication. This provides extra security to the LoRaWAN join procedure since the authentication information will be written on the blockchain network and the network server cannot tamper this information. More details of this architecture can be found at [28]. The main difference between this work and the two factor authentication mechanism is that in this paper, we are presenting a blockchain based authentication mechanism for the LoRaWAN



(a) Throughput comparison for 1000 concurrent clients



(b) Timeout messages comparison for 1000 concurrent clients



(c) Latency comparison for 1000 concurrent clients

Figure 6: Performance comparison of the two frameworks

network and we are modifying the current LoRaWAN join procedure protocol to completely eliminate jamming and replay attacks along with ensuring trust among LoRaWAN network entities. However, in [28], the join procedure remains the same but an extra authentication factor is added which is based on the blockchain to ensure trust in the LoRaWAN network. The approach has been implemented using the same software tools as in previous sections however, the LoRa end devices and the network servers are now communicating using the Python client server socket programming while the network server is communicating with the blockchain network through RPC calls.

Fig. 6 demonstrates the performance comparison between the two proposed frameworks for 1000 concurrent clients. It can be seen from Fig. 6a that, for 1000 concurrent clients, the throughput is high for the blockchain based LoRaWAN join procedure as compared to the two factor authentication framework. Similarly, in Fig. 6b and Fig. 6c, the blockchain based LoRaWAN join procedure performs better in terms of timeout messages and latency respectively. This is due to the fact that in the two factor authentication framework, the join request messages incur additional delays due to the introduction of the independent blockchain network which also leads to degradation in the throughput and timeout of LoRa end device join request messages.

The above results demonstrate that the blockchain based LoRaWAN join procedure framework reports better performance and can be used for large scale LoRaWAN networks with multiple network servers, to provide high system throughput and low delay. However, such a performance comes at the expense of high implementation cost of the join servers. The two factor authentication framework reduces this cost to provide a lightweight cost effective authentication solution to the LoRaWAN join procedure with few number of network components and no strict requirements on system performance.

6.4. Discussion

In this work, we have proposed a blockchain based authentication mechanism for LoRaWAN network. In our proposed framework, the LoRa end device is authenticated by the information saved in the blockchain network in real time. In the current LoRaWAN specification, LoRa end device is authenticated by network server, based on the nonce/counter value in join request message, which makes it vulnerable to jamming and replay attacks.

Moreover, this information for a specific LoRa end device is saved in the centralized network server, which makes it vulnerable to the modification of authentication information. In our proposed framework, the authentication is independent of the nonce/counter value in LoRaWAN join request packet, which makes the join request secure from jamming and replay attacks. Moreover, this information is saved in multiple network servers i.e. blockchain network therefore, no single entity can modify the authentication information. Thus, our system provides protection against jamming and replay attacks in LoRaWAN join procedure as well as builds trust among LoRaWAN network entities.

Moreover, the performance of the proposed architecture is represented in terms of throughput and latency. The results indicate that our system achieves average throughput of 1800 join requests per second with the maximum load of 1000 concurrent clients with 30 join requests each. With the same maximum load, it takes approximately 15 seconds by the system to respond to all these 1000 concurrent clients with 30 join requests each. These parameters are chosen to check the performance of the system for massive authentication workload. Additional join servers can be added to scale our proposed architecture. However, in practice, every LoRa end device performs the join procedure (authentication) once or twice a day [8], and each device generates 1 request per authentication. Thus, the join server will be able to respond with high throughput. In addition, our system allows LoRa end devices to authenticate themselves in the trust-less environment.

7. Conclusion

This paper addresses the existing security problems in the LoRaWAN join procedure as a result of its vulnerability to jamming and replay attacks. Moreover, the network server, being the centralized entity, brings trust issues among network entities and customers. In this paper, we present a blockchain based distributed framework for the LoRaWAN join procedure, to develop a secure and trusted authentication system within LoRaWAN networks. The proposed framework eliminates the jamming and replay attack threats against the LoRaWAN join procedure and in addition builds trust among LoRa end devices and network servers. We propose a scalable, generalized and easy to manage authentication system for the LoRaWAN join procedure, endorsed by a PoC prototype, to validate the proposed framework. The simulations results indicate that the system achieves efficient

system performance up to an upper bound on the load level which involves 30 join requests from 1000 concurrent clients. However, the security and trust advantages that the blockchain technology offers, comes at the cost of performance deterioration for loads exceeding that level. This can be resolved by introducing additional join servers. When the implementation cost is of primary concern, a lightweight, cost effective blockchain based two factor authentication framework may also be considered. We demonstrate through simulations that this approach incurs additional delays and can be preferred for systems with no strict requirements of throughput and latency. In the future, we are planning to implement our system on a LoRaWAN hardware testbed to check the feasibility of the proposed framework in practice.

References

1. "Ericsson mobility report: On the pulse of the networked society," Ericsson, Tech. Rep., November 2017. [Online]. Available: https://www.ericsson.com/mobility-report

2. F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi. Internet of things security: A survey. *Journal of Network and Computer Applications*, 2017.

3. D. Airehrour, J. Gutierrez, and S. K. Ray, "Secure routing for internet of things: A survey," *Journal of Network and Computer Applications*, vol. 66, pp. 198–213, 2016.

4. Barrachina-Munoz, S., Bellalta, B., Adame, T., & Bel, A. Multi-hop communication in the uplink for LPWANs. *Computer Networks*. pp. 153-168, 2017

5. LoRa Alliance. 2017. LoRaWAN Specification V1.1. [Online]. Available: https://lora-alliance.org/resource-hub/lorawantm-specification-v11

6. S. M. Danish, A. Nasir, H. K. Qureshi, A. B. Ashfaq, S. Mumtaz, J. Rodriguez. Network Intrusion Detection System for Jamming Attack in LoRaWAN Join Procedure. *Proceedings of the 54th IEEE International Conference on Communications (ICC)*. pp. , May, 2018

7. S. M. Danish, H. K. Qureshi, S. Jangsher, M. Lestas. Effects of Wireless Power Transfer on LoRaWAN Join Procedure. *Proceedings of the*

14th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC). pp. , June, 2018

8. Stefano Tomasin, Simone Zulian and Lorenzo Vangelista,. Security Analysis of LoRaWANTM Join Procedure for Internet of Things Networks. *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 1-6, 2017

9. Butun, I., Pereira, N., & Gidlund, M. Analysis of LoRaWAN v1. 1 security. *In Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart ObjectsSensors*, p. 5, 2018

10. Jun Lin, Zhiqi Shen, and Chunyan Miao. Using Blockchain Technology to Build Trust in Sharing LoRaWAN IoT. *In Proceedings of the ACM 2nd International Conference on Crowd Science and Engineering (ICCSE'17)*, pp. 38-43, 2017

11. JungWoon Lee, DongYeop Hwang, JiHong Park, and Ki-Hyung Kim. Risk Analysis and Countermeasure for Bit-Flipping Attack in LoRaWAN. *International Conference on Information Networking (ICOIN)*, pp. 549-551, 2017

12. Emekcan Aras, Gowri Sankar Ramachandran, Piers Lawrence and Danny Hughes. Exploring The Security Vulnerabilities of LoRa. *International Conference on Cybernetics (CYBCONF)*, pp. 1-6, 2017

13. Yang. Xueying, Evgenios. Karampatzakis, Christian. Doerr, and Fernando. Kuipers. Security Vulnerabilities in LoRaWAN. *IEEE/ACM Third International Conference on In Internet-of-Things Design and Implementation (IoTDI)*, pp. 129-140, 2018.

14. SeungJae Na, DongYeop Hwang, WoonSeob Shin, and Ki-Hyung Kim. Kuipers. Scenario and Countermeasure for Replay Attack Using join request Messages in LoRaWAN. *International Conference on Information Networking (ICOIN)*, pp. 718-720, 2017

15. S. Nakamoto. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. [Online]. Available: https://bitcoin.org/bitcoin.pdf

16. Liu, Ziyao, et al (2019). "A Survey on Applications of Game Theory in Blockchain." arXiv preprint arXiv:1902.10865.

17. Understanding Public Key Cryptography (2005). [Online]. Available: https://technet.microsoft.com/en-us/library/aa998077(v=exchg.65).aspx

18. BUTERIN, V. A next generation smart contract & decentralized application platform. https://www.ethereum.org/pdfs/EthereumWhitePaper.pdf/, retrieved Feb. 2015.

19. Conoscenti, Marco, Antonio Vetro, and Juan Carlos De Martin. Blockchain for the Internet of Things: A systematic literature review. *In IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1-6, 2016

20. Hammi, Mohamed Tahar, Patrick Bellot, and Ahmed Serhrouchni. BC-Trust: A decentralized authentication blockchain-based mechanism. *In IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-6. 2018

21. Lee, Chan Hyeok, and Ki-Hyung Kim. Implementation of IoT system using block chain with authentication and data protection. *In IEEE International Conference on Information Networking (ICOIN)*, pp. 936-940, 2018.

22. Zhu, Xiaoyang, Youakim Badr, Jesus Pacheco, and Salim Hariri. Autonomic Identity Framework for the Internet of Things. *In IEEE International Conference on Cloud and Autonomic Computing (ICCAC)*, pp. 69-79, 2017

23. Lavric. Alexandru, and Valentin. Popa. "LoRa™ wide-area networks from an Internet of Things perspective. *In IEEE 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1-4, 2017

24. S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Uncover security design flaws using the STRIDE approach," MSDN Magazine, Nov. 2006.

25. Buterin. Vitalik A next-generation smart contract and decentralized application platform. *white paper*, 2014

26. James. Ray 2015, [Online]. Available:https://github.com/ethereum/wiki/wiki/Benchmarks

27. Winsvega 2015, [Online]. Available: https://github.com/ethereum/tests

28. S. M. Danish , M. Lestas, W. Asif, H. K. Qureshi, M. Rajarajan , "A Lightweight Blockchain based Two Factor Authentication Mechanism for LoRaWAN Join Procedure", Proceedings of the IEEE 53rd International Conference on Communications (ICC), May, 2019, Shangai , China.

29. https://www.thethingsnetwork.org/tech-stack#section1