Leveraging blockchain for a robust and scalable device identification in LoRaWAN

Lounès Meddahi[1,2], Fen Zhou[1],
[1]IMT Nord Europe, Institut Mines-Télécom, Lille, France
[2]Université de Lille, Lille, France

***Abstract*—In LoRaWAN 1.1 networks, end devices must be activated via Over-The-Air Activation (OTAA) to securely send and receive data. Activation involves a two-step process: identification using a unique identifier (DevEUI) and authentication through a Message Integrity Code computed with a pre-shared key. Both DevEUI and pre-shared keys must be provisioned beforehand. We propose a scalable blockchain-based decentralized model to improve the identification of untrusted end devices during LoRaWAN's OTAA without altering the protocol, thus allowing immediate deployment in existing networks. Our approach accommodates corrupted devices and stores data in a distributed, permanent, and publicly auditable manner while facilitating rapid detection and identification of untrusted devices. Adapting blockchain technology to LoRaWAN resource-constrained environment, we design a specific data block structure combined with a "Proof of Identification" (PoI), while maintaining distributed consensus. This approach ensures a robust and scalable LoRaWAN join procedure while enhancing device identification and traceability. Through simulation models that combine our blockchain proposal with LoRaWAN 1.1 OTAA specification, we show improved performance for various metrics, e.g., for 10,000 devices, identification is two times faster using the blockchain.**

***Index Terms*—IoT, LoRaWAN 1.1, Blockchain, Proof of Identification, Security.**

I. Introduction

The rapid growth and deployment of IoT technologies play a crucial role in everyday life, notably in Industry [1]. Yet, they present specific challenges such as resource constraints, device density, scalability, data volume and heterogeneity, but also security and privacy issues. Mobility and performance are also limited, representing further obstacles.

Blockchain technology has been introduced to IoT environments to address some of these challenges, with its decentralized model enhancing IoT device security (identification, access control, traceability). However, traditional blockchain must be adapted for optimal integration with specific IoT technologies, such as with the Long Range Wide Area Network (LoRaWAN), a popular low-power, long-range communication technology.

LoRaWAN faces challenges in achieving long lifetimes for battery-powered devices and minimizing interaction complexity due to resource constraints. As a result, sophisticated security mechanisms may not be applicable, particularly for end devices with limited capacities. Existing proposals for blockchain-based LoRaWAN, characterized by a centralized join procedure, are limited.

In this paper, we propose a new approach for blockchain-based LoRaWAN 1.1, leveraging the specific proof and distributed consensus of traditional blockchain for a robust and resilient join procedure and device identification. We introduce a "proof of identification" and a "trust_index" for network servers. Performance evaluation using our custom simulation tool yields satisfactory results. Our approach demonstrates a higher device identification rate and significantly reduces detection time of corrupted devices compared to a system without blockchain. Additionally, our solution dynamically bans corrupted devices or network servers through the traceability and immutability features of the blockchain technology.

The paper is structured as follows: an overview of LoRaWAN 1.1 and traditional blockchain technology is proposed in section 2. A review of the related work on blockchain for IoT security is presented in section 3. A description of our proposed lightweight and efficient blockchain-based LoRaWAN join procedure is proposed in section 4. A comparison of different types of proofs with our new "Proof of Identification," and a performance evaluation under various scenarios using simulation are presented in section 5 before the conclusion.

II. Preliminaries

We first describe the LoRaWAN Over the Air Activation procedure (OTAA) with a focus on the identification and authentication process, called join procedure.

A. LoRaWAN join procedure and security model

LoRaWAN protocol is used for transmitting data from low power sensors to application servers over large areas. This paper focuses on LoRaWAN 1.1 [2]. The architecture comprises end-devices, gateways, network servers, join servers, and application servers. End-devices and join servers have pre-provisioned 64-bit identifiers, DevEUI and JoinEUI, respectively, and share two pre-shared 128-bit root keys, AppKey and NwkKey, for securing communications. The join procedure occurs during Over The Air Activation (OTAA) in three steps, as illustrated in Fig. 1. First, the end device sends a cleartext Join-request message to its associated join server, containing the DevEUI, JoinEUI, DevNonce, and MIC. The request is forwarded by gateways to network servers, which direct it to the specified join server. Secondly, join servers perform the following actions:

1) Identify and authenticate the device using received DevEUI and MIC by verifying the MIC based on the DevEUI's NwkKey.

2) Establish session keys by generating a random JoinNonce value, used with AppKey and NwkKey to derive session keys (AppSKey, NwkSEncKey, FNwkSIntKey, SNwkIntKey).
3) Distribute information to network servers and application servers, including JoinNonce, NwkSEncKey, SNwkSIntKey, FNwkSIntKey, and AppSKey.

Finally, network servers generate a join accept message using the JoinNonce value. This Join Accept message is encrypted using AES-128 and sent back to the end device, allowing it to generate session keys and other parameters from the JoinNonce. In LoRaWAN 1.1, the identification process relies on a centralized model, assuming the DevEUI is registered and stored in the join server.

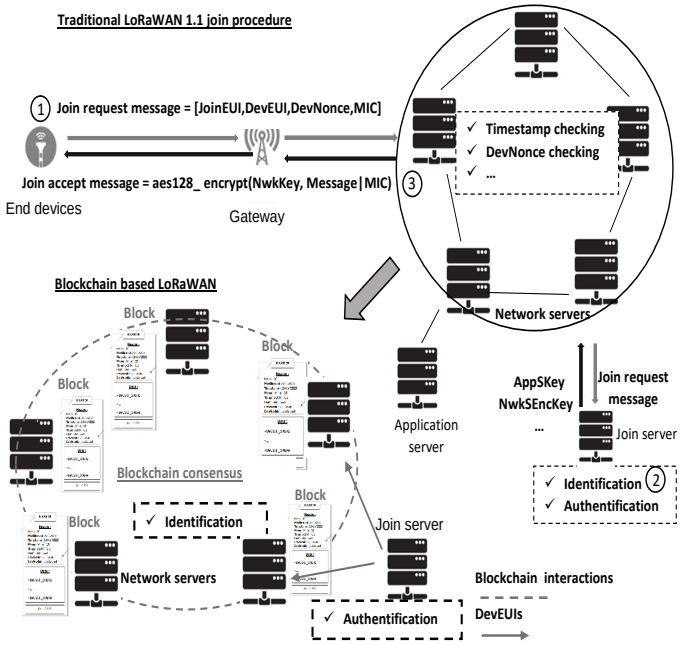

Fig. 1: Traditional vs blockchain-based LoRaWAN join procedure.

B. The blockchain technology

S. Stornetta and S. Haber [3] first proposed a solution for timestamping and securing digital documents in 1991, improved in 1992 to timestamp multiple digital documents at once [4]. Their work laid the foundation for Satoshi Nakamoto's blockchain, Bitcoin, introduced in 2008 [5]. A blockchain is a data structure composed of blocks, each characterized by a header, data section, and proof (Fig. 2).
The header contains various information fields: index (block location), Merkle root value (unique hash), timestamp of the block, miner identity, difficulty (expressed in Nonce), and hash value of the current and previous blocks.The Merkle root value is unique and generated from information in the data field, by using a hash function such as SHA256. Miners are the entities responsible for validating and adding new blocks to the blockchain. The hash value is computed by using all header values, including the hash of the previous block, which ensures block integrity and maintains the correct block order. The data field stores information, such as user transactions in the case of Bitcoin. A proof is required to add a new block to the blockchain, such as Proof of Work (PoW) [6] or Proof of Stake (PoS) [7]. The blockchain consensus is achieved by all participants adhering to the same proof and rules.
In summary, blockchain technology offers a secure, traceable, and distributed way to store and manage information. The next section discusses blockchain-based IoT systems, focusing on addressing vulnerabilities in LoRaWAN 1.1 [8] [9] join procedure by employing a decentralized blockchain-based identification process.

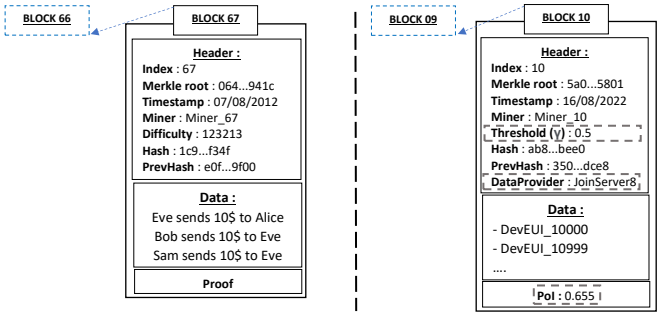

Fig. 2: Traditional Vs PoI based based Blockchain

III. Related work

The literature review indicates limited contributions on the integration of blockchain with IoT systems, particularly LoRaWAN 1.1 systems. To our knowledge, [10] is the primary study focused on leveraging blockchain for enhancing LoRaWAN connection performance, while other references consider general IoT systems.

- In [11], Sadawi et al. survey blockchain integration with IoT. They propose an architecture that eliminates central authority, resulting in a system with reduced reliance on a central trust authority. This approach enhances fault tolerance by distributing and sharing data storage among several independent servers, governed by the same blockchain consensus rules. Scalability is achieved through the native distributed architecture of the blockchain network, and security is improved via the cryptographic structure that ensures the secure evolution of the blockchain and its associated data. They also analyze interactions between IoT network objects and the blockchain.
- Puthal and Mohanty [12] introduce a new type of proof named "Proof of Authentication" (PoAh), for combining blockchain with IoT authentication. They argue that this approach improves network security and performance in terms of delay and computing power. An implementation of PoAh [13] demonstrates that it is 20 times faster and better suited for device authentication than Proof of Stake (PoS) blockchains like Ethereum [14].
- Danish et al. [10] examine the integration of blockchain with LoRaWAN 1.1 for improving the join procedure. They propose a two-factor authentication mechanism

utilizing a blockchain network deployed through the network servers. The end devices register their information in the blockchain before initiating the join procedure. The authors' evaluation suggests that this approach enhances authentication security and builds trust between end devices and network servers. However, their implementation has some limitations, such as privacy concerns and the use of Ethereum, which is slower than PoAh-based blockchains.

IV. Blockchain based LoRaWAN 1.1 join procedure and identification

Our proposal focuses on integrating a specific blockchain architecture with the LoRaWAN 1.1 join procedure, ensuring minimal impact on performance (e.g. connection rate) and addressing key challenges such as scalability and untrusted end devices with limited capacities. The blockchain distributes device identification from join servers to network servers. We discuss key design aspects for transitioning from traditional LoRaWAN to a blockchain-based LoRaWAN. Existing works consider general IoT and not specific technologies like LoRaWAN 1.1. Since each IoT technology has unique characteristics, the blockchain should be tightly coupled with the IoT network environment.

A. Proposed blockchain architecture

Our proposed blockchain is closely integrated with the LoRaWAN architecture, considering the constraints and limited resources of different entities such as untrusted end devices, join servers for security authentication and data isolation, and application servers supporting user applications and services. In our approach, end devices are considered "untrusted" as they can potentially be corrupted, while network and join servers are seen as "trusted" entities offering a certain level of isolation. We assume that network servers can detect and identify corrupted devices or data through a trusted third party or middlebox [15]. If the number of corrupted devices connected via the network server exceeds a predefined threshold, the network servier becomes untrusted and is banned from the blockchain network. Thus, we address the issue of traditional LoRaWAN lacking procedures for revoking or renewing corrupted DevEUIs. We opt for network servers to host and support the blockchain, enabling real-time updates with the dynamic evolution of the LoRaWAN architecture, minimal impact on network architecture and protocol, and flexibility for new future LoRaWAN 1.1 specification updates.
The join server (a trusted entity) becomes a data provider for the blockchain : To identify the end devices rapidly and dynamically, we exploit the network servers instead of the join servers. Indeed, according the different types of blockchain objects interactions described in [11], we choose entities that are not interconnected (join servers) and can only send predefined data to the blockchain (e.g. the DevEUIs). So, join servers as a trusted entity will play the role of the data provider in our architecture.
Node entities for the blockchain : The main design aspect is identifying entities with their trust levels to act as nodes for the blockchain. Network servers are ideal candidates for hosting and supporting blockchain execution (full nodes) due to their accessibility, storage, and computing capacities. These nodes store the blockchain, verify data from join servers, mine, and share new blocks. The network server layer is optimal for supporting our decentralized, blockchain-based approach, ensuring efficient and robust device identification and access control. However, for an effective approach, join servers must share specific information and parameters with network servers. These network servers, which store and manage the blockchain, maintain the integrity of the immutable ledger for entities involved in the identification process. The integrated blockchain with network servers provides a secure, decentralized ledger for device identification and access control, minimizing the impact on LoRaWAN entities and resources. 1 illustrates the blockchain integration with LoRaWAN 1.1 and interactions with network and join servers. If a network server grants access to many corrupted devices (above a certain threshold), it is considered untrusted, leading to its ban from the blockchain.
In the next section, we define a specific blockchain architecture and a new type of proof called Proof of Identification (PoI) for blockchain consensus.

B. Blockchain for device Identification

Defining and exploiting an adaptated blockchain [13] is relevant and efficient compared to the main existing ones (PoW or PoS). [13] allows the tracking of blockchain's nodes participants. However, a specific blockchain is required to take into consideration both blockchain's nodes tracking and devices identification. Thus, we propose a *trust_index* metric for Network servers as a function of time and of the total number of DevEUIs listed in the blockchain (both legitimate and corrupted). We assume that a corrupted device, as opposed to a legitimate one, is a device for which the DevEUI is revoked from the join server or identified as corrupted by the network server. We also introduce a *Proof of Identification* which relies on a specific function γ. Given this, we define the PoI-based blockchain with the following rules:

- The information stored in the data blocks are the DevEUIs obtained from the join servers. For security purposes, we also extend the information contained in the header block, with the JoinEUI obtained from the join servers (DataProvider field in Fig. 2)
- A *trust_index* value from 1 (default value) to 0 is defined for each network server. A *trust_index* is equal to 1 if all the DevEUIs in the blocks mined by the network servers correspond to legitimate devices. A *trust_index* is equal to 0 if all the DevEUIs in the blocks mined by the network server correspond to corrupted devices.
- The *trust_index* of network servers is a function of time as defined by equation (1). This *trust_index* is computed

by all networks servers.

$$f : [0,\infty[\rightarrow [0,1] \quad : t \rightarrow 1 - \frac{m(t)}{n(t)} \quad (1)$$

$$\gamma : [0,\infty[\rightarrow [0,1] \quad : t \rightarrow \gamma(t) \quad (2)$$

Where: t =time; $m(t)$ =number of corrupted DevEUI at t; $n(t)$ =total DevEUIs listed in the blockchain at t.

- The network server is periodically updated with the list of DevEUIs registered in the join server or tagged as corrupted. Note that even if a corrupted device uses a particular block for its identification, as the DevEUI is listed in different blocks all the associated network servers will have their *trust_index* decrease.
- The network server's *trust_index* is included in the PoI field when adding a block to the blockchain, enabling block rejection if the *trust_index* is below the trust threshold $\gamma(t)$.
- To represent the difficulty level, we use a threshold parameter indicating the minimum trust level for adding a new block, defined by equation (2).
- The threshold γ is provided to the blockchain at initialisation, and is defined as a constant for simulation purposes.
- The threshold reflecting the trust level is computed for each block. Therefore, this threshold can be verified by all network servers in real time.
- The network servers with a *trust_index* under a pre defined trust level (threshold) listed in the last block header, are banned from the blockchain network. This is proved by the following threshold Lemma :

$\forall \gamma : [0,\infty[\rightarrow [0,1]$, if $\exists t_0 \in [0,\infty[$ such that $f(t_0) < \gamma(t_0)$, then $\nexists t_1 \in [t_0,\infty[$ such that $f(t_1) \geq \gamma(t_0)$.

Proof. Let t_0 be the time the *trust_index* is lower than $\gamma(t_0)$. From t_0, the blockchain of the banned network server is not updated. So, $\forall t \in [t_0,\infty[\ n(t) = n(t_0)$ and the threshold γ is equal to $\gamma(t_0)$. As the network server is periodically kept updated with the corrupted devices, there are two possibilities for function $m(t)$:

- $m(t)$ increases with limit $n(t_0)$ if new corrupted devices are identified.
- $m(t)$ is constant as no new devices are identified as corrupted.

Note that $m(t) \in [n(t_0)(1-\gamma(t_0)), n(t_0)]$ for $t \in [t_0,\infty[$. So, $\forall t \in [t_0,\infty[\ : 0 < 1 - \frac{m(t)}{n(t)} \leq f(t_0) < \gamma(t) \Leftrightarrow 0 < f(t) \leq f(t_0) < \gamma(t_0)$ as γ is constant and greater than $f(t_0)$.

No matter if the function γ is constant or not, once the *trust_index* goes under the threshold, the network server is banned from the blockchain. Compared with the PoW, PoS and PoAh, our identification based blockchain combined with the PoI is efficient, thanks to its properties such as: low computation requirements (computation of the *trust_index* and addition of blocks is less intensive), a limited search space size as the blocks are larger than for the PoS but smaller than the PoW, and a limited information look up delay in the blockchain (low latency). Also, as the proposed approach has a limited impact on the core network and particularly no impact for the end devices, we assume that the impact on energy consumption is low compared to the other blockchains (e.g. PoW). Fig. 3 and 4 show a concrete illustration and example of PoI usage. This example corresponds to the blokchain given by fig. 2. Fig. 3, shows the evolution of the *trust_index* for two network servers (NS1 and NS2) as a function of the number of devices (fig. 4).

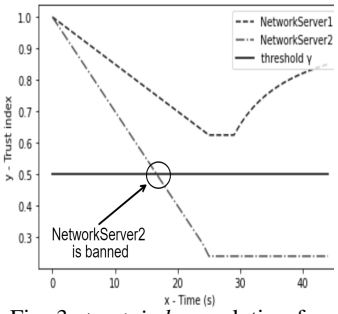

Fig. 3: *trust_index* evolution for NS1 and NS2

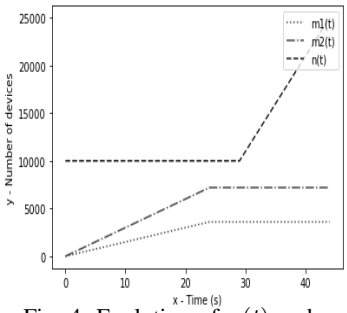

Fig. 4: Evolution of $n(t)$ and corrupted devices for NS1, NS2

For this scenario, both network servers NS1 and NS2 connect the same number of end devices (both legitimate and corrupted). We also consider that the identified corrupted end devices are different for NS1 and NS2. Fig. 4 gives $n(t)$ as the total number of connected devices over time. The functions $m1(t)$ and $m2(t)$ correspond to the number of corrupted devices identified by NS1 and NS2 respectively. As the number of corrupted devices identified by NS2 increases (Fig. 4 for $m2(t)$), the *trust_index* for NS2 goes under the pre defined threshold $\gamma(t) = 0.5$ and remains under this threshold, even if new devices are added to the network (see $n(t)$ from 30 sec). Thus, the NS2 is banned from around 16sec (fig. 3). For NS1, the new devices joining the network are identified as legitimate from 30sec and lead to an increases of its *trust_index*. This NS1 *trust_index* remains greater than $\gamma(t) = 0.5$ (fig. 3).

In the next section, we provide a performance evaluation of our approach based on our custom simulator[1].

V. Performance evaluation

We assess various properties via different metrics and provide a performance evaluation of the proposed blockchain in terms of robustness, efficiency, and lookup delay. A detailed performance comparison with traditional LoRaWAN 1.1 is also presented. In the next section, we explain the motivation for developing a specific simulation tool and offer a comprehensive description of the simulator and its parameters.

A. Blockchain-based LoRaWAN 1.1 Simulation Tool

To the best of our knowledge, no open-source simulator, including NS-3, implements the complete LoRaWAN 1.1 specifications, covering security primitives, full join procedure involving all LoRa entities, and allowing the integration

of the PoI blockchain. While ARMmbed is primarily for device simulation, we developed a specific LoRaWAN 1.1 simulation tool[1] to address this gap. Our simulator, compliant with LoRaWAN 1.1 specifications, enables various testing scenarios and measurements. The performance evaluation considers different scenarios based on our simulator, such as : number of crashed network servers, number of connected end devices, type of topology, device or gateway density and spatial distribution. The implementation choice for our open-source simulator allows quick and easy integration with other simulators, such as an extension module for NS-3. The simulation parameters are :

- Perfect transmission without interference, message loss.
- 28 messages per second for the gateways [16].
- 15 kilometers coverage for the gateways.
- Uniform distribution of the end devices on a map of 150 square kms (containing a total of 100 gateways, distributed uniformly according a grid topology).
- 15km between gateways to minimize black spots.

Fig. 5 shows the topology given by the simulation tool where 10000 devices uniformly distributed with a low spreading factor (SF7_DR6) lead to a limited signal range but a high frequency of sending (up to 1 message every 399ms). [17].

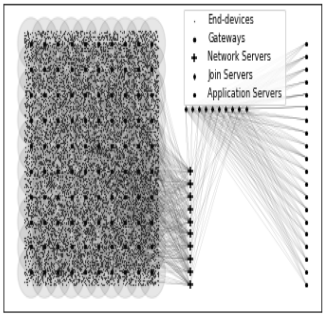

Fig. 5: Simulated LoRaWAN topology for 10000 devices.

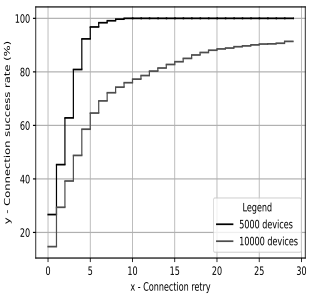

Fig. 6: Connection success vs. connection retries for SF7_DR6.

The figures Fig.6 and Fig.7 give the performance results for the traditional LoRaWAN (without blockchain) when considering 2 metrics: the "connection success rate" and the "connection delay", for a spreading factor SF7_DR6. SF7_DR6 provides a low Airtime with a low gateway congestion rate that accelerates simulation results with a large number of concurrent clients and join requests. We evaluate the maximum number of join requests required for a device to connect to the network. For this purpose, we proceed by connection iteration. For each iteration, each device sends a join request message to its associated join server, until the connection succeeds. Fig. 6 gives this connection success rate versus the number of join request retries, for different number of devices (5000 and 10000). We note that with only 5 join request messages sent per device, 96,8% of devices succeed to join the network for 5000 devices, while it is only 64,6% for 10000 devices. These results are due to the limited capacity of the gateways. Indeed, if the number n of devices increases by 2 and requests are uniformly distributed over the number m of gateways (n/m requests per gateway), the gateways are 2 times more congested ($2n/m$ requests per gateway). Fig. 7 gives the connection delay (delay between the first join request message and the final join accept message) for different connection rates and number of devices (5000 and 10000). The figure Fig. 7 shows for 5000 devices that 100% of these devices are connected with a connection delay of 14 sec and 25% with a maximum connection delay of around 1 sec. For 10000 devices, the limit is 90% with 40 sec delay and 10% with a connection delay of almost 2 sec. These simulation shows the relevance and consistency of the results regarding the defined topology and networks characteristics. This simulation tool measures the effectiveness of our PoI-based blockchain through different performance metrics.

B. Simulation conditions and parameters for the blockchain

The simulation results aim to evaluate the performance of the proposed PoI combined with the LoRaWAN architecture. For a traditional LoRaWAN network, we assume all join servers are pre-configured with the authorized DevEUIs. In the blockchain-based join procedure, each join server provides the DevEUIs of legitimate devices to their associated network servers. All network servers are connected to the blockchain network at the beginning of the simulation.
Each network server is capable of creating blocks associated with the received DevEUI, adding the DevEUI to the blockchain, and sharing the blockchain with other network servers. Consequently, data verification (DevEUIs listed in the blocks) can be performed efficiently.
Network servers with a *trust_index* below the predefined threshold are banned, ensuring efficient data verification. The join servers providing DevEUIs are listed in the block header, DataProvider, allowing network servers to check legitimate and corrupted DevEUIs by matching the JoinEUI with the DataProvider field. To prevent banned network servers from creating fake join servers, a minimum number of verifications can be defined before adding a block. We assume that it's possible to check this information for all network server sets. We evaluate the following average metrics: connection time, time to detect a corrupted device, connection rate versus the number of devices (scalability property), and identification success rate versus the number of servers down (robustness). Simulations were conducted on a PC with an AMD Ryzen 9 16-Core Processor 3.40 GHz and 128 GB RAM using our simulator[1]. An implementation of our blockchain is available[1].

C. Numerical Results

First, we measure the successful connection rate versus connection retries for 5000 and 10000 devices. Fig. 8 shows that the impact on the connection rate when adding a blockchain to the LoRaWAN is significantly low compared to the traditional architecture without blockchain. Indeed, for both 5000 and 10000 devices, 90% of these devices connect after 10 retries, indicating that adding the blockchain has minimal impact on

[1]https://github.com/LounesMD/LoraSIM

the connection rate. Secondly, to measure the robustness of the blockchain based distributed architecture, we consider in our numerical simulation that a certain number of network servers can go down during the join procedure operation. For the traditional architecture devices identification is done by the join servers, accessible only via network servers. As opposed, for the blockchain based LoRaWAN the device identification is done directly by the network servers through its blockchain. Fig. 9 reveals that for 5,000 and 10,000 devices, the identification success rate declines after 6 network servers crash with the blockchain, compared to 2 network server crashes without the blockchain. For these simulations we consider that each gateway is connected to 2 or 3 network servers. For this reason, it is only from 2 network servers down that the identification rate starts to decrease (for both 5000 and 10000 devices). This is also due to the network servers that are no longer available for sending the join request to the join server for authentication, in the traditional based architecture. For the blockchain based architecture, it starts to decrease from 6 network servers, as the number of network servers is not sufficient to support the number of join requests. Fig. 10 represents the detection rate of requests sent from

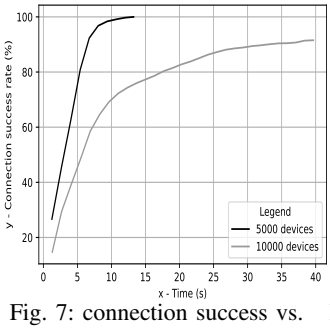

Fig. 7: connection success vs. connection delays for SF7_DR6

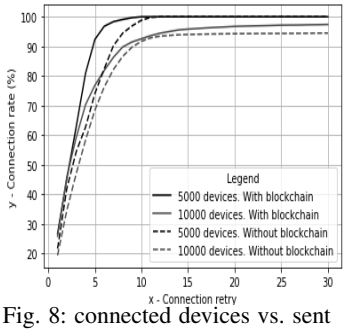

Fig. 8: connected devices vs. sent requests (w./w.o. blockchain)

the corrupted devices versus time. These corrupted devices perform corrupted join procedure as their DEvEUI is no more registered in any join server or tagged as corrupted [15]. The figure demonstrates that with 50% corrupted devices (out of 10,000), the detection time for the blockchain-based architecture is significantly better. In fact, 50% of corrupted devices are detected within 2s using blockchain, compared to over 6s without. On average, a corrupted device is detected within 201ms with blockchain and 710ms without. The 509ms difference is due to network delay between the network server and join server, highlighting the blockchain's rapid response and detection capabilities for corrupted devices.

VI. Conclusion

The integration of the blockchain with LoRaWAN 1.1 combined with a Proof of Identification, allows to improve the join procedure and devices identification in terms of robustness, efficiency and resiliency with a minimal impact on performances. These integration and combination improve also the scalability, thanks to the blockchain decentralised architecture. As the blockchain keeps track, traceability and

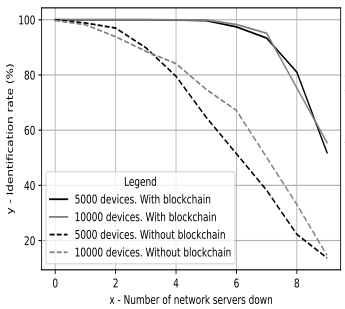

Fig. 9: conn. success vs. servers down (w./w.o. blockchain)

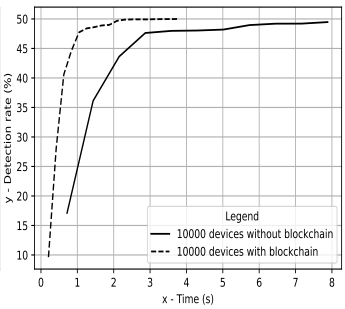

Fig. 10: corrupted DevEUIs detected vs t (w./w.o blockchain)

records of the join requests from the end devices, it provides a rapid look up of the legitimate and corrupted devices in the related blocks. The network servers that mine the "corrupted" blocks are banned from the network, depending on a pre-defined *trust_index*. The properties and characteristics of our proposed PoI based blockchain combined with a *trust_index* and the consistency results obtained from the performance evaluation show the relevance of the proposed approach. In future work, we plan to expand the security performance analysis by including physical layer security considerations in our model for constrained services types such as for Ultra Reliable Low Latency Communications slice in 5G+ networks.

References

[1] Ouafae Cohin, Patrick Sondi. Internet of things for smart factory. IEEE Communications Society Multimedia Communications Technical Committee E-Letter, 2015.

[2] LoRaWAN®1.1 specification, https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm_specification_-v1.1.pdf.

[3] Haber S., Stornetta W. How to time-stamp a digital doc. J. Crypto. 1991.

[4] Bayer D., Haber S., Stornetta W. Improving the Efficiency and Reliability of Digital Time-Stamping. Sequences II Springer book, 1993.

[5] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.

[6] A. Back. Hashcash: A DoS Counter-Measure. Tech. report, 2002.

[7] S. King, S. Nadal. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. Self published paper, 2012.

[8] X. Yang et al. Security Vulnerabilities in LoRaWAN, ACM Third Int. Conference IoTDI, 2018.

[9] K. Grover et al. Jamming and Anti-jamming Techniques in Wireless Networks: A Survey, International Journal of Ad Hoc and Ubiquitous Computing, vol. 17, no. 4, 2014.

[10] S. M. Danish et al., A Lightweight Blockchain Based Two Factor Authentication Mechanism for LoRaWAN Join Proc., IEEE ICC 2019.

[11] A. A. Sadawi, M. S. Hassan and M. Ndiaye, A Survey on the Integration of Blockchain With IoT to Enhance Performance and Eliminate Challenges, IEEE Access, vol. 9, 2021

[12] D. Puthal and S. P. Mohanty, Proof of Authentication: IoT-Friendly Blockchains, in IEEE Potentials, vol. 38, no. 1, Jan 2019.

[13] S. Maitra et al., Proof-of-Authentication Consensus Algorithm: Blockchain-based IoT Implementation, WF-IoT, 2020.

[14] V. Buterin. Ethereum: a next generation smart contract & decentralized application plateform, https://ethereum.org/en/whitepaper/, 2017.

[15] C. Dajiang et al. Privacy-Preserving Encrypted Traffic Inspection With Symmetric Cryptographic Techniques in IoT. IEEE Internet of Things journal, vol. 9, no. 18, Sept. 2022.

[16] LoRa® and LoRaWAN® technical documents, https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/

[17] T. Voigt, M. Bor, Spreading Factor : Airtime calculator for LoRaWAN. https://avbentem.github.io/airtime-calculator/ttn/eu868/222.