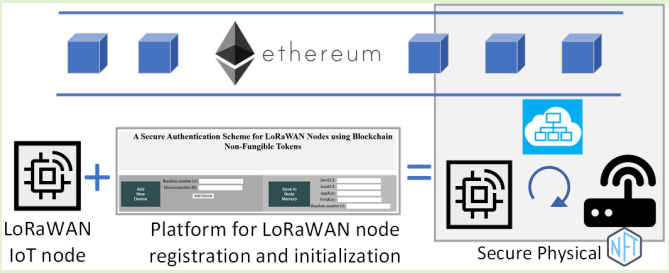
This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2024.3431432

A Lightweight Authentication Scheme for LoRaWAN Nodes Represented as On-Chain Non-Fungible Tokens

M. Sidorov, *Member*, *IEEE*, J.H. Khor, A. C. H. Wong, Y. Y. Lee, and J. Li, *Member*, *IEEE*

***Abstract* — QR codes, Near-Field Communication (NFC), and Radio Frequency Identification (RFID) tags have been widely used for supply chain tracking purposes. By binding valuable physical goods with digital Non-Fungible Tokens (NFTs) a Physical NFT is created, preventing counterfeit products from penetrating the supply chain. However, QR codes, NFC, and RFID tags are not suitable for real-time monitoring, especially in long-distance tracking applications. As an alternative, Long Range Wide Area Network (LoRaWAN) nodes have gained widespread adoption in smart supply chains for their ability to facilitate long-distance monitoring and tracking, while consuming low amounts of power. However, challenges to bind LoRaWAN nodes with NFTs exist in the areas of energy consumption and security. Therefore, this paper proposes a secure authentication scheme for LoRaWAN nodes, binding them to on-chain NFTs for proof-of-authenticity purposes and LoRaWAN security enhancement. The scheme is compatible with existing LoRaWAN specifications, where microcontroller ID, additional computed information together with the sensor data, temperature in this case, are included as a payload of the uplink message. This scheme is developed using the SHA256 hash function and exclusive-OR operations. The proof of concept demonstrates that the scheme is fully supported by resource-constrained LoRaWAN nodes. The security of the proposed protocol has been analyzed and is proven to be secure from replay, man-in-the-middle, and cloning attacks. The performance of the scheme has been evaluated, and with a computation cost of 1.432 ms, a storage cost of 896-bit, and a gas consumption of 22,984 it shows to outperform other related works.**



***Index Terms* — Authentication, Blockchain, LoRaWAN, NFT**

I. Introduction

Physical Non-Fungible Tokens (NFTs) are digital tokens associated with real-world assets and linked together using unique identifiers, such as QR codes, Near-Field Communication (NFC) or Radio Frequency Identification (RFID) tags, etc. They serve as proof of ownership for valuable items, prevent counterfeiting, and enhance transparency in the supply chain which allows backtracking to their origin, e.g. Cryptokicks iRL sneakers created by RTFKT and Nike [1], or Helium Internet of Things (IoT) and 5G Mobile hotspots [2]. Physical NFTs have two parts – a combination of digital and physical components. The digital component is stored on a public blockchain and is accessed via smart contract. The physical one is represented by a real-world asset and is linked to the NFT using the aforementioned QR code, NFC tag, or RFID tag. All of these identifiers require short proximity for scanning, ranging from 1 cm to 10 m and reaching 100 m for active RFID tags. This range, however, is not suitable for real-time monitoring of moving assets in supply chains [3], hence another solution is needed.

The adoption of Long Range (LoRa) technology for tracking valuable assets has recently emerged as one of the trends in smart supply chains. LoRa technology provides long-distance coverage and combined with the energy efficiency it distinguishes itself from Global Positioning System-based (GPS) or cellular-based tracking systems [4]. The integration of sensor data for monitoring, e.g. humidity, temperature, etc., enhances the monitoring capabilities of the LoRa-based tracking system, enabling real-time monitoring of temperature-sensitive goods throughout the supply chain. In addition, a robust LoRaWAN MAC layer is provided that offers a set of security features for the network, such as two layers of 128-bit Advanced Encryption Standard (AES) encryption [5]. This added layer of security makes LoRa an attractive choice for applications where asset tracking and data integrity are paramount. However, no prior research that combines LoRaWAN nodes as a part of physical NFTs securely has been conducted.

Several approaches to store NFTs on a blockchain exist,

"This work was carried out during the tenure of an ERCIM 'Alain Bensoussan' Fellowship Programme." (Corresponding author: M. Sidorov).

M. Sidorov is with the University of Southampton Malaysia, 79100 Iskandar Puteri, Johor, MY (e-mail: michail.sidorov.cl@tut.jp).

J. H. Khor is with the Connected Intelligence Research Group, University of Southampton Malaysia, 79100 Iskandar Puteri, Johor, MY (e-mail: j.khor@soton.ac.uk).

A. C. H. Wong is with the University of Southampton, Southampton, SO17 1BJ Hampshire, UK (e-mail: achw1n20@soton.ac.uk).

Y. Y. Li is with the University of Southampton, Southampton, SO17 1BJ Hampshire, UK (e-mail: yyl1c20@soton.ac.uk).

J. Li is with the Norwegian University of Science and Technology, 7034 Trondheim, NO (e-mail: jingyue.li@ntnu.no).

with the main difference being the way data is stored. Hence, NFTs are classified into off-chain and on-chain ones. Off-chain NFTs are generally implemented where digital assets, such as images, are hosted on the InterPlanetary File System (IPFS) – a protocol and a file-sharing peer-to-peer network in a distributed file system. The metadata, however, for off-chain NFTs, such as the content identifier from the IPFS, is stored on the blockchain [6]. This practice is vulnerable to data availability and data consistency. In contrast, on-chain NFTs store everything entirely on a blockchain, which is more secure. However, this introduces other drawbacks such as limitation in blockchain storage, and additional privacy issues due to public blockchain being transparent [7].

Combining LoRaWAN nodes as part of physical NFTs poses technical challenges, particularly in the areas of energy consumption and security. LoRaWAN nodes are known for being energy-efficient low-power IoT devices. They are battery-powered and have limited processing, storage, and energy resources. In contrast, blockchain nodes, which play a crucial role in NFTs, are more powerful and require more energy to perform complex cryptographical algorithms [8]. NFTs are created through minting processes that require validation steps. These steps consume energy with the amount dependent on the consensus algorithm employed. Ethereum blockchain, which has transitioned from the resource-intensive Proof of Work (PoW) to energy-efficient Proof of Stake (PoS), dominates in hosting most NFTs due to its extensive popularity [9]. While PoS is less energy-intensive, there is a need to seamlessly link NFTs with the LoRaWAN network without imposing an additional burden of higher energy consumption on LoRaWAN nodes.

While combining LoRaWAN nodes with NFTs poses a technical challenge, the security of LoRaWAN nodes is a concern on its own. Key disclosure may arise from weak security employed in data storage mediums. This is valid for both sensor nodes and network providers not utilizing secure elements or trusted execution environments [10]. Additionally, the use of hardcoded keys from open-source firmware poses a risk, as does the possibility of the device itself being compromised on a manufacturing level. Authors in [10] demonstrated three offline cracking methods involving dictionary and brute-force attacks, highlighting the need for enhanced security measures. Key disclosure can lead edge nodes to be susceptible to a plethora of security attacks, with the most common ones being replay [11], man-in-the-middle [12], and cloning [13].

Therefore, there is a need to design a secure lightweight authentication scheme for LoRaWAN nodes that link to on-chain NFTs. The proposed scheme is designed to prevent the aforementioned three security attacks even when LoRaWAN root keys are disclosed through a key disclosure attack. This scheme can be adopted and applied to any existing LoRaWAN network server, i.e. The Things Network, Loriot, Senet, etc.

The main contributions of this paper are as follows:

1. A lightweight authentication scheme for LoRaWAN nodes bound with NFTs and designed using bitwise XOR operation and hash function to enhance the security of LoRaWAN nodes.
2. This work enables secure proof of authenticity for LoRaWAN nodes associated with on-chain NFTs without added energy burden.
3. A proof-of-concept designed to justify the validity of the proposed authentication scheme for using LoRaWAN nodes as on-chain physical NFTs.
4. Security analysis of the authentication scheme was performed to validate its safety against replay, man-in-the-middle, and cloning attacks.

The remainder of this paper is structured as follows: Section II reviews the related work done on improving the security protection of the LoRaWAN network. Section III presents the new authentication scheme using on-chain NFTs. A proof of concept of the proposed scheme is described in Section IV, and its security analysis is described in Section V. In Section VI, a performance analysis of the proposed scheme is discussed. Section VII concludes this paper.

II. Related Work

Several research works that focus on improving the security of the LoRaWAN network via key management [11, 14-18] or authentication protocols [19-22] exist. The key management approach consists of three components: a key management system, a device key provisioning, and the on-chip security of the end device [23]. Key management is essential for the LoRaWAN network as each LoRaWAN end node is equipped with an Application Key (AppKey) for authentication and a Network Key (NwkKey) for network access control purposes. Two session keys are used, where a Network Session Key (NwkSKey) is used to guarantee the data integrity, and an Application Session Key (AppSKey) is used to ensure data confidentiality [24]. These two keys are generated using AppKey and NwkKey static root keys. As mentioned in Section I, static root keys are vulnerable to key disclosure attacks. Thus, researchers proposed secure and efficient root key [14] and session key update mechanisms to strengthen security [15, 16]. LoRaWAN v1.1 specification describes the network protocol between the network server and end devices [5]. However, the communication channel between the network server and the join server or an application server is not covered. Thus, Tsai *et al.* proposed a secure session key generation for LoRaWAN servers [17]. However, secure key generation alone does not guarantee protection against compromised keys or unauthorized access.

Researchers have further proposed blockchain technology to strengthen LoRaWAN security. Blockchain features such as decentralization, transparency, and immutability can solve trust issues in centralized networks [25]. In [18], a permissioned blockchain was used to store keys for key availability purposes instead of using the centralized join servers. Danish *et al.* proposed a lightweight two-factor authentication mechanism for the LoRaWAN join procedure using Ethereum private blockchain to add a layer of security [19]. However, both permissioned and private blockchains are not fully decentralized. Thus, they are susceptible to data modification. Therefore, public blockchains, which provide full decentralization and transparency, should be used to improve the security of LoRaWAN networks.

Several companies such as Wisekey [26] and SmartAxiom [27] bind NFTs to IoT devices for authenticity purposes withholding information regarding its implementation. There

are works on using off-chain NFTs for proof of ownership using the Ethereum Request for Comments (ERC)-721 [28-31] standard. ERC-721 is one of the standards used for minting NFTs, i.e. NFT creation. This process is executed based on the smart contracts written following those standards. As a feature, NFTs inherit public blockchain characteristics making NFT metadata stored on the blockchain immutable and transparent. NFTs further provide extra functionality that includes ownership verification and transferability. Off-chain NFTs have been proposed to be integrated with physical IoT devices for traceability and ownership management in [29, 30]. NFTs have further been proposed to represent the digital twin of medical devices for efficient medical device traceability and ownership management [29]. However, the proposals solely concentrated on smart contract implementation for NFT verification and authentication purposes without deploying the solution on real IoT devices. Turki *et al.* presented an off-chain storage drug traceability solution by binding NFTs with IoT devices [30]. However, a resource-rich Raspberry Pi 3B+ board was used for testing and validation purposes. This device is costly, energy-hungry, and very powerful compared to typical low-power IoT devices. Arcenegui *et al.* on the other hand verified their proposal via proof-of-concept using popular ESP32 microcontroller boards. Physical Unclonable Function (PUF) was used to generate a private key that is used in authentication processes to prevent key disclosure attacks [28, 31].

Although there are works describing physical NFTs for proof of ownership purposes [28, 31], this does not apply to low-power LoRaWAN end nodes. Therefore, to fill the research gap, this work proposes an authentication scheme that fuses NFTs and LoRaWAN nodes together both for the purpose of enhancing the security of the LoRaWAN network and offering enhanced proof of ownership features.

III. Authentication Scheme for LoRaWAN Nodes Represented as On-chain NFTs

LoRaWAN v1.1 provides secure communication between end nodes and network servers, with the condition that root keys are secure from adversaries. As described in Section I, these 128-bit root keys are vulnerable to disclosure attacks, enabling network servers to authenticate malicious nodes. Thus, the proposed authentication scheme acts as an additional authentication layer on top of the original one listed in the specifications. Two 256-bit secret values are used to enhance the security of the proposed protocol from brute force attacks, as the security of 256-bit secret values is much stronger compared to 128-bit ones [32]. The proposed authentication scheme is compatible with the existing LoRaWAN specifications by including additional payload data in the encrypted message. For earlier LoRaWAN standard v1.0.x, the NwkKey is treated as a variable but not used as a root key.

The proposed authentication scheme focuses on the network server and assigns additional roles that facilitate the usage of NFTs in the authentication scheme. However, the complete LoRaWAN system involves three additional backend servers: identity, join, and application. The roles of these servers are left unchanged with the full server description provided in Table 1. The additional roles of the network server include:

1. LoRaWAN node registration on the Ethereum blockchain
2. Smart contract deployment on the Ethereum blockchain
3. NFT minting to bind to LoRaWAN nodes
4. Integrity verification of the received message from the LoRaWAN nodes
5. LoRaWAN node authentication by computing a signature using the received message and verifying the computed Signature with the one stored in the metadata of NFT

The ERC-721 standard is used in this project to design the NFTs since it supports static metadata stored directly on a smart contract, which cannot be provided by the ERC-1155 standard. NFTs are generally used for proof of authenticity and ownership because their identities are written on smart contracts and stored on the blockchain. Their identity is immutable and transparent; thus, building trust is easier compared to a centralized database. In addition, the token ownership can be identified and tracked via blockchain. Adversaries can compromise the authentication scheme mentioned in the LoRaWAN v1.1 if the root keys are successfully disclosed through security attacks.

The proposed authentication scheme consists of two phases: initial phase and authentication phase. The authentication phase is also known as the communication phase. As the proposed protocol mainly focuses on the authentication of the on-chain physical NFTs with a LoRaWAN network server, the revocation phase is not included in this paper. The scheme utilizes bitwise XOR operations to encrypt the outputs of two SHA256 hash functions. Although bitwise XOR encryption is susceptible to known-plaintext attacks, the proposed protocol is immune to this since XOR operations are used to encrypt two hash outputs, instead of two plaintext messages. Notations used in the design of the proposed authentication scheme are illustrated in Table II.

A. Initial Phase

Similarly to existing LoRaWAN node registration procedures, this phase relies on the owner of the LoRaWAN node to initiate and save network credential data inside the memory of the nodes. The following describes the initial steps required to be done before verifying the authenticity of the LoRaWAN nodes by the network server. Steps include creating an Ethereum account, generating and storing data, minting corresponding NFT, etc., as seen in Fig. 1. Detailed procedure is as follows:

1. The owner of the LoRaWAN node registers an account on the Ethereum network and deposits some credit to the platform. The *DevEUI*, *JoinEUI* (known as *AppEUI* in the earlier specifications), *AppKey*, *NwkKey*, and a secret key, *t* are generated by the network server for the new LoRaWAN node.
2. The LoRaWAN node owner saves the *DevEUI*, *AppEUI*, *AppKey*, and *NwkKey* to the node flash memory

TABLE I
LoRaWAN Servers and Their Original Roles

Server	Role
Identity	Manages access control of end devices and gateways
Network	Manages and monitors the connectivity of LoRaWAN nodes' and gateways' magnetic induction
Join	Stores root keys and generates session keys
Application	Manages a web interface

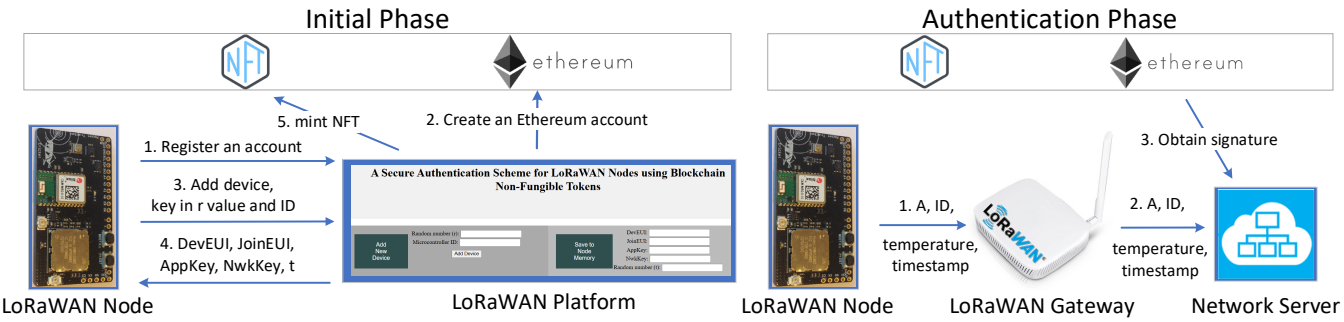


Fig. 1. Initial phase and Authentication phase of the proposed protocol

TABLE II
PROTOCOL NOTATIONS AND DESCRIPTION

Notation	Description
Hash	SHA256 Hash Function
||	Concatenation
ID	88-bit unique microcontroller ID generated by the LoRaWAN node manufacturer
t	256-bit secret value generated by LoRaWAN network server
r	256-bit secret value generated by LoRaWAN node
A	256-bit, $A = Hash(r\|\|ID) \oplus Hash(t\|\|timestamp)$
AppKey	128-bit Application Key
NwkKey	128-bit Network Key
DevEUI	64-bit Device Extended Unique Identifier
AppEUI JoinEUI	64-bit Application Extended Unique Identifier
DevNonce	16-bit Device nonce
Signature	256-bit, $Hash(Hash(r\|\|ID)\|\|AppKey\|\|NwkKey)$
Timestamp	40-bit current timestamp of the transmitted message

3. An Ethereum account is created for the owner of the LoRaWAN node by the network server using its pre-deployed smart contract. The generated wallet address is stored in a database of the network server for making future transactions.
4. The LoRaWAN node generates *r* value and stores it in its memory. The owner of the LoRaWAN node adds a new device to the platform by keying the *ID* and *r*. A *Signature*, *Hash(Hash(r||ID)||AppKey||NwkKey)* is then created by the network server using the keyed-in *r* and *ID* values and its stored *AppKey* and *NwkKey* values. The *ID* is stored in its database. The *r* value, however, is discarded by the network server as it is no longer needed.
5. The network server mints an NFT for the LoRaWAN node. A transaction is made on the Ethereum blockchain, where the wallet address of the owner, *ID,* and *Signature* are included as the input of the transaction.

B. Authentication Phase

This phase describes the steps required to enable the LoRaWAN network server to authenticate a legitimate LoRaWAN node by verifying the received message and the Signature of the NFT stored on the Ethereum blockchain, as seen in Fig. 1. Detailed steps are as follows:

1. LoRaWAN node sends a payload message consisting of timestamp, sensor data (temperature in this case), *ID,* and *A* to the gateway.
$$A = Hash(r||ID) \oplus Hash(t||timestamp)$$
2. The payload message is forwarded to the network server.
3. The network server obtains the LoRaWAN node data in its local database based on the obtained *ID* value. The network server obtains the *Signature*, *Hash(r||ID ||AppKey||NwkKey)* on the blockchain based on the *ID* value.
4. The network server computes *Hash(t||timestamp)'* using the timestamp found in the message and *t* stored in its memory. It then extracts *Hash(r||ID)* by XOR-ing message *A* with the computed *Hash(t||timestamp)'*. It proceeds to compute the Signature, *Hash(Hash(r||ID)||AppKey||NwkKey)*. If the computed *Signature* equals the one stored on the blockchain, the network authenticates the legitimate LoRaWAN node. The network server then stores message *A* in its database to prevent future replay attacks.

IV. PROOF OF CONCEPT

A. NFT minting contract design

A lorawanNFT.sol smart contract was written using Solidity language. This contract is ERC-721-based and allows to mint NFTs for the LoRaWAN nodes. ERC721.sol file describing this standard is imported from the OpenZeppelins GitHub repository allowing lorawanNFT.sol to inherit all the functions from the ERC721.sol. A *mint()* function with three parameters was created to mint a token. This *mint()* function calls the *_safeMint()* function from ERC-721 to link a token owner's address with a device *ID* and *Signature*, as shown in Fig. 2.

B. Development of a physical LoRaWAN node

A low-power LoRaWAN Cricket-L082CZ development board manufactured by Tlera was used as a LoRaWAN node. It is equipped with a Murata CMWX1ZZABZ-078 all-in-one LoRaWAN module containing an ARM Cortex-M0+ Micro Controller Unit (MCU) with 192 Kbytes of flash memory, 20 Kbytes of RAM, and 20 Kbytes of EEPROM. The board further is equipped with a U-BLOX CAM M8Q Global Navigation Satellite System (GNSS) receiver module.

The LoRaWAN node was programmed to transmit temperature, *A*, and *ID* messages to a LoRaWAN gateway. In order to perform the OTAA activation, *DevEUI*, *JoinEUI*, *AppKey*, and *NwkKey* were stored in the LoRaWAN node. In addition, the LoRaWAN node stores *r* and *t* in its flash memory for authentication purposes. The microcontroller *ID*, temperature, and timestamp are obtained using the member functions from libraries listed in Table III. The microcontroller *ID* can be read from the system flash of STM32L082 MCU,

Algorithm 1 Pseudo-code of *mint()*.

Input:
_to: address
_tokenId: uint256
_signature: bytes

1. **if** (*_to* is not equal to zero) **then**
2. **if** (*_tokenId* has not already been minted) **then**
3. call the *_safeMint*(address *to*, uint256 *tokenId*, bytes *data*) to safely mint a new token
4. **else** revert the transaction
5. **else** revert the transaction
6. **end**

Fig. 2. Pseudocode for binding an NFT with a LoRaWAN device

which is a read-only memory. The ID is split into three components and stored at the memory address that starts from 0x1FF80050. In addition, the temperature can be read using the onboard Bosch BME280 pressure, temperature, and humidity sensor. The timestamp is obtained using the U-BLOX GNSS receiver. The *ID*, temperature, timestamp, and message *A* are then included in the payload.

C. Development of a Decentralized Web Application Platform

A decentralized web application platform was created to enable the owner of LoRaWAN nodes to add new devices to the LoRaWAN network by providing *r* and *ID* values, as shown in Fig. 3. The newly added LoRaWAN node is bound to an NFT by including the owner's wallet address, *ID*, and a *Signature* in a transaction made to the blockchain. The platform also enables the network server to retrieve the LoRaWAN node *Signature* from the Ethereum Goerli testnet blockchain and authenticate the LoRaWAN node by comparing it with the computed *Signature*, as shown in Fig. 3. The decentralized web application was designed using HTML and Javascript, where Web3.js is used to interact with the NFT smart contract. Interaction with Ethereum blockchain is done via Infura, a remote Ethereum node. The details of a successful transaction to mint a token can be found in [33].

V. Proposed Authentication Protocol Security Analysis

This paper does not include experimental security analysis, as the time needed to crack the SHA256 hash function to get its preimage is around 10^{32} years [34]. Thus, theoretical and formal security analysis (in Section VI.D) were performed for this work as they are well accepted by the related research works [35]. In order to analyze the security of the proposed protocol, the following assumptions are therefore made where:

1. An adversary can obtain the DevEUI and AppEUI/JoinEUI because they are not secret and visible to the public. An adversary can compute *AppSKey* and *NwkSKey* based on the LoRaWAN root keys.
2. An adversary can obtain the *AppKey* and *NwkKey* through key disclosure attacks, subsequently modifying the *DevNonce* and computing a valid MIC.
3. An adversary cannot access and modify the LoRaWAN nodes and storage memory of the server.
4. An adversary is not interested in performing denial of service attacks but instead would like to enable successful authentication of counterfeit tags.

A. Replay Attack

In the LoRaWAN v1.1 standard, the *DevNonce* is used to prevent replay attacks. If a reused *DevNonce* is detected, then the transmitted message will be considered invalid. Hence, the network server will not authenticate the LoRaWAN node. As described in [10], the adversary can obtain or crack the LoRaWAN root keys, *AppKey* and *NwkKey*. Once the adversary obtains the root keys, the *DevNonce* can be easily changed. Thus, the proposed authentication protocol provides an additional security layer for the existing LoRaWAN standard.

Replay attack can be prevented using message *A*, *Hash(Hash(r|| ID)* $\oplus$ *Hash(t||timestamp)*. The message *A* varies for each session because the timestamp is included as a preimage of the hash function. If the adversary replays the previous session's message *A*, the network server will be able to detect it by comparing message *A* with its database. Therefore, the adversary cannot perform the replay attack by replaying the captured message *A*.

B. Man-in-the-middle Attack

In order to perform the man-in-the-middle attack on the original LoRaWAN v1.1 standard, the adversary eavesdrops and captures the transmitted message on the communication channel between the LoRaWAN nodes and network servers. The adversary then modifies the *DevNonce* and computes a valid MIC, and the network server will then be able to authenticate the adversary's LoRaWAN node.

This paper provides additional security protection from the man-in-the-middle attack using the *Signature* stored on the blockchain. During the authentication phase, the network server obtains *H(r||ID)'* by XOR-ing the message *Hash(Hash(r||ID)*$\oplus$*Hash(t||timestamp)* sent by the node. Next, the network server computes the *Signature*, *Hash(Hash(r|| ID)||AppKey||NwkKey)*, using the computed *H(r||ID)'* and its stored *AppKey* and *NwkKey*. Since the adversary is unable to compute valid *H(r|| ID)* and *t* values, if the adversary randomly modifies the transmitted messages, then the computed *Signature* will be different from the one stored on the blockchain, and the authentication will fail.

C. Cloning Attack

Since the Ethereum public blockchain is transparent, adversaries might be able to obtain the *ID* and *Signature*. Adversaries might perform cloning attacks by duplicating the microcontroller *ID* and *Signature* to a new LoRaWAN device. However, the adversaries are unable to access the memory of LoRaWAN nodes and thus unable to access *r* and *t* values, as the Memory Protection Unit is used in the IoT device, further explained in Section VI.F. The adversaries are thus unable to compute valid *Hash(r||ID)* and *Hash(Hash(r|| ID)*$\oplus$*Hash(t||timestamp)* values.

Most join servers and network servers use centralized

TABLE III
Library and its Member Functions

Parameter	Library
ID	STM32L0.h
Temperature	BME280.h
Timestamp	RTC.h and GNSS.h

databases, e.g. either local servers or cloud servers, that can be hacked and modified by adversaries. The adversaries might hack the server database, modify the *Signature*, and force the network server to authenticate counterfeit LoRaWAN nodes. However, this attack is excluded in this paper, as stated in the aforementioned assumptions. Although the server database modification attack by adversaries is excluded, there is a possibility that malicious servers might change their stored data. Thus, the proposed authentication scheme stores the *Signature* on the Ethereum blockchain. Since the *Signature* stored on the blockchain is immutable, the adversaries cannot modify it. Hence, the binding of the LoRaWAN nodes with NFTs enables the LoRaWAN servers to authenticate the nodes throughout their lifetime. If the adversaries transmit a message using counterfeit LoRaWAN nodes, the network server will not authenticate them due to the invalid *Signature*.

VI. Comparative Performance Analysis

As described in Section II.B, no academic research was found on using on-chain NFTs to improve the security of the LoRaWAN communication channel. This section analyzes the performance of the proposed authentication scheme in terms of storage, payload size, security, and transaction gas fee. Ethereum blockchain security analysis is omitted in this paper as it has been proven to be secure by other researchers in [8].

A. Storage Cost

In addition to the original 64-bit *DevEUI*, 64-bit *JoinEUI*, 128-bit *AppKey,* and 128-bit *NwkKey* defined by the LoRaWAN standard, the proposed protocol requires IoT devices to store r and t in their memory. This requires an additional 512 bits of space, 256 bits for r and 256 bits for t, respectively. Therefore, the total storage cost increases to 896 bits, as presented in Fig. 4.

Arcenegui *et al.* proposed the use of SRAMs PUF to keep the private key safe from attackers [28, 31]. The private key is not stored but is reconstructed from the 2048-bit Helper Data (HD_{DEV}) stored in the non-volatile memory of the IoT device and several masks called ID and RND masks generated from the start-up response of the SRAM. A public key is reconstructed using additional steps. Additionally, Zero-Stage Bootloader (ZSB), public key of manufacturer $PK_{MAN,}$ and 256-bit *tokenID* are stored in One Time Programmable memory of the IoT device for [28], but the former two with unknown sizes are omitted for [31]. In addition, storage cost information is unavailable for [29, 30]. Gebreab *et al.* introduced the concept of binding physical medical devices with NFTs to enable interoperable authentication and device verification in [29]. NFTs have also been utilized in smart

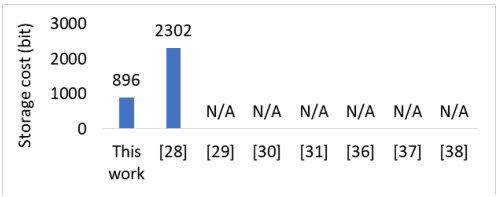


Fig. 4. Comparative analysis of storage costs

cities [36, 37] and for Unmanned Aerial Vehicle (UAV) [38] authentication and ownership management. However, these solutions have primarily focused on smart contract design for verification and device authentication, omitting the implementation using real IoT devices. Turki *et al.*, in contrast, utilized a Raspberry Pi 3B+ board as a target IoT device for integration with NFTs [30]. However, similar to [29], the solution focused on designing smart contracts for authentication and authorization, omitting storage costs.

B. Computation Cost

To minimize the added power consumption burden on a low-power LoRaWAN node, the energy-intensive minting process is offloaded to a resource-rich network server. For authentication purposes, the LoRaWAN node encrypts the hash values of its unique microcontroller ID, random numbers, and timestamp. The average time to compute message A which consists of two SHA256 hash functions and one XOR operation using the LoRaWAN Cricket-L082CZ is 1.432 ms. The computation cost has been compared with other related works solely based on the algorithms used. Different IoT devices were utilized for the computational cost analysis, with the ESP32 used in [28, 31] and Raspberry Pi 3+ used in [30]. Solutions by Arcenegui *et al.* require 52.65 ms for a complete cycle, i.e. private key obfuscation, shared secret generation, zero-stage boot, private key reconstruction, account address generation, and making a transaction on a blockchain [28, 31]. As shown in Fig. 5, the computation cost for the proposed solution is insignificant compared to the solutions presented by Arcenegui et al. The computation cost, however, is not specified for [29, 30, 36-38] as their primary focus was designing smart contracts for the authentication of NFTs.

C. Payload Size

LoRa operates in an unlicensed radio spectrum. However, it is region-dependent. Hence different frequency bands are used depending on the region, e.g., EU868 band in Europe,US915 in the US, AS923 in Asia, etc. Each frequency band supports different maximum payload sizes for each data rate [39]. Malaysia supports the AS923 frequency plan, and its maximum payload size for each Spreading Factor (SF) has been analyzed and listed in Table IV. SF is a set of parameters

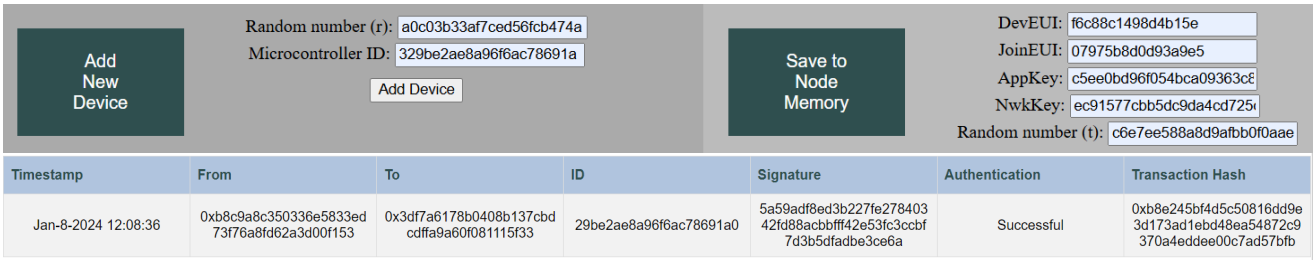


Fig. 3. Decentralized Web Application excerpt for adding new LoRaWAN nodes, minting NFTs, and authentication confirmation.

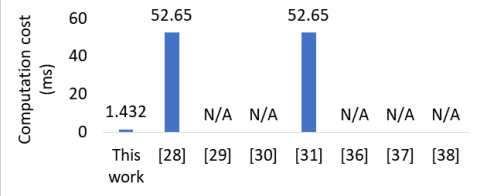


Fig. 5. Comparative analysis of computation costs

used to control the speed of data transmission and communication range. This maximum payload size is measured using the following setting, where the Adaptive Data Rate (ADR) is set to false to prevent the network server from controlling the data rate and transmit power of the LoRa device. If the ADR is set to true, the network server can change these settings and optimize the data rate and energy consumption. The AS923 frequency band limits maximum dwell time (time needed to transmit a LoRaWAN message) to 400 ms. As the time on air for a transmitted message is longer for a higher SF, LoRa supports SF of 7 to 10 only if the UplinkDwellTime parameter is set to true (dwell time limit is enabled). If a wider signal coverage is required, this parameter can be set to false in order to use SF of 11 and 12. However, a LoRa node with a higher SF requires more power to transmit, while offering a smaller payload size compared to a lower SF. The proposed authentication protocol includes a timestamp, ID, message $A = Hash(r||ID) \oplus hash(t||timestamp)$, and temperature data in a payload of the uplink messages. Table V shows the payload message and its size. The total payload size of an uplink message is 392 bits, which is represented using a hexadecimal string, converted to a 98-byte character string that is supported by data rates of 4 and 5. These data rates can support uplink message sizes of up to 125 bytes and 242 bytes respectively. Related works focus on binding IoT devices with NFTs, however, they do not specifically address the linking of low-power LoRaWAN nodes with NFTs. Therefore, a comparative analysis of LoRaWAN payload sizes in transmitted messages for an authentication scheme is not applicable to [28-31, 36-38].

D. NFT Data Persistency Analysis

Data persistence is vital for off-chain NFTs. Generally, a digital asset such as an image or a file is stored on the IPFS due to the limited storage capacity of blockchains. IPFS is a distributed file storage protocol that allows hypermedia and files to be shared in a peer-to-peer network. IPFS provides a Uniform Resource Identifier (URI) to link the NFTs metadata to its stored digital asset on the IPFS. The IPFS URI is useful to guarantee the integrity of NFT digital assets. Thus, the URI metadata of digital assets will also be stored on the IPFS. The metadata is further used to describe the contents of the stored digital asset. The URI of the metadata will be stored on the blockchain. However, the data stored on the IPFS, an approach that was used in [29, 30, 38], might be lost or unavailable if it is accidentally or purposely removed from the IPFS nodes. Thus, pinning a content identifier is vital for data storage on IPFS to avoid data removal. Long-term data persistence on IPFS can be achieved by additionally storing the data on a decentralized storage network, such as Filecoin. Due to the complexity of implementing off-chain NFTs, on-chain NFTs are emerging to solve the off-chain NFT data

TABLE IV
SPREADING FACTOR VS MAXIMUM PAYLOAD SIZE FOR AS923

Data Rate	SF	Maximum Payload Size (byte)	
		Uplink Dwell Time (False)	Uplink Dwell Time (True)
0	12	51	NA
1	11	51	NA
2	10	51	11
3	9	115	53
4	8	242	125
5	7	242	242

persistency issues [28, 31, 36, 37]. This paper implements on-chain NFTs to permanently store the nodes *ID* and unique *Signature* on the blockchain making LoRaWAN node information immutable.

E. Smart Contract Vulnerability Analysis

NFTs are minted using a smart contract. Thus, same as with other smart contracts deployed on blockchains, NFTs with vulnerable smart contracts are susceptible to security attacks. Several NFTs incidents occurred where attackers exploited the vulnerabilities of the NFT smart contract to gain benefits [40]. Thus, it is crucial to analyze the NFT smart contract before deploying it on the blockchain. The *mint()* function in the designed lorawanNFT.sol inherits the available *_safeMint()* function from the ERC-721 smart contract developed by OpenZeppelin. Thus, the lorawanNFT.sol is secure since OpenZeppelin provides smart contracts using libraries that have been verified to be safe from security attacks. Gebread *et al.* implemented two smart contracts, one using custom functions for device verification and approval, and another one for minting tokens, which inherits predefined functions from the ERC-721 smart contract [29]. Turki *et al.* created three functions for NFTs creation, namely *CreateLotNFT (drugName)*, *CreateOrderNFT(Lot IDs)*, and *CreateVehicle NFT(vehicle address)*. The first two functions use the *_safeMint()* function from the ERC-721 smart contract. However, the *CreateVehicleNFT(vehicle address)* function is a custom one [30]. Solutions by Arcenegui *et al.* used in [31] require nine custom functions for smart contracts, whereas [28] extended the existing ERC-721 standard with an additional eight attributes, 15 functions, and four events. In [36, 37], the ERC-721 standard has been extended to support the digitization of IoT devices in smart city applications. Unlike [29, 30, 38], these additional codes are not analyzed and might be susceptible to security attacks.

F. Formal Security Analysis

The proposed protocol is analyzed using AVISPA, a formal security verification tool. The proposed protocol is modeled using High-Level Protocol Specification Language (HLPSL). The proposed protocol is analyzed using the On-the-Fly Model Checker (OFMC) backend in the AVISPA tool. The intruders under this model can control the entire network, including intercepting, modifying, and replaying messages. The OFMC backend uses symbolic techniques to reduce the search space using a falsification approach to identify attack traces and a verification approach to prove the correctness of protocols. The OFMC result shows that the proposed protocol is safe from replay and MITM attacks, as seen in Fig. 6.

Arcenegui *et al.* implemented unique PUFs, making IoT devices resistant to replay, man-in-the-middle, and cloning

TABLE V
PAYLOAD MESSAGE SIZE

Message	Size (bit)	Size (byte)
Hash(r||ID)⊕hash(t||timestamp)	256	64
ID	84	21
Temperature Data	16	4
Timestamp	36	9

attacks [28, 31]. However, the security analysis in terms of replay and man-in-the-middle attacks for the transmission data in [29, 30, 36-38] is not possible due to missing transmitted message information.

G. Hardware Security

Hardware security is crucial to ensure device is secure from plethora of attack vectors. In [29], authors used smart labels based on PUF to store unique identifiers with the aim of preventing medical devices from being counterfeited. IoT devices with PUF features have been proposed to be used in smart city applications, as discussed in [36, 37]. However, similarly to [29] the description of the implementation was omitted. In [30, 38] hardware security was not considered at all. Given that the proposed solution targets low-power LoRaWAN nodes, addressing the hardware security of LoRaWAN nodes is crucial.

Several methods can be used to ensure the LoRaWAN root keys are stored securely from being accessed by adversaries. These methods include using trusted execution environments and hardware security modules at the backend servers and LoRaWAN nodes with secure elements. For example, The Things Industries LoRaWAN Join server and Network server are integrated with a Trusted Execution Environment to ensure the secure key generation, storing, and management for the Application server and network server in a Hardware Security Module (HSM) [41]. A number of manufacturers produce Secure Elements for LoRaWAN nodes, e.g. Microchip ATECC608 series [42, 43], STSAFE-A110 from STMicroelectronics [44], etc. Some of them integrate HSM directly into the System-on-Chip (SoC), i.e., Cypress Programmable SoC 6 [45]. These secure elements are vital in ensuring the LoRaWAN nodes are pre-provisioned with root keys, and these root keys are locked from being accessed in the LoRaWAN nodes. The security of the LoRaWAN node used in this paper can be ensured using its Memory Protection Unit integrated within the Cortex M0+ MCU.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/lora.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.00s
  visitedNodes: 28 nodes
  depth: 4 plies
```

Fig. 6. OFMC security result

H. Gas Consumption to Mint NFTs

When a transaction is made on the Ethereum blockchain, the gas needed to perform the transaction is calculated based on the operations that are going to be executed [46]. High transaction fees are typically associated with using the Ethereum blockchain. However, these can be mitigated by using a second-layer scaling solution, e.g. Polygon blockchain.

As described in Section IV, a *mint()* function is created to bind NFTs to LoRaWAN devices. The gas consumption to perform this function is 22,460 as shown in Fig 7. In [29], a *mintNFT()* function is called, consuming 73,997 gas to mint NFTs after verifying the device IDs do not exist in a mapping that associates device IDs with the NFT token IDs. The gas consumption for the *mintNFT()* function in both [36, 37] is 32,303. In [30], three different smart contracts are used to mint three different NFTs, each with its unique gas consumption: LoT NFTs consume 557,238 gas, Order NFTs with a gas consumption of 351,656, and Vehicle NFTs consume 350,752 gas. These NFTs have various purposes, including tracking drug lots, managing orders containing multiple drug lots, and associating vehicles with drug transportation.

Same as [30], three different ERC-721 smart contracts have been used in [38] to mint different NFTs - Component, LoT, and Assembled UAV NFTs. However, only gas consumption of minting the Component and LoT NFTs is specified, which are 139,902 and 607,853, respectively. Whereas the gas consumption to mint a new token using the *createToken()* function for [28] is 167,263 and 112,510 for [31] respectively. Table VI provides a comparative summary of each solution. It shows the proposed solution not only can enhance LoRaWAN

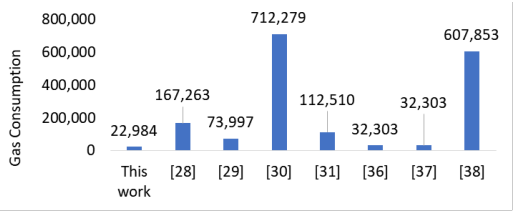


Fig. 7. Comparative gas consumption analysis

TABLE VI
COMPARATIVE ANALYSIS WITH RELATED WORK

	Description	This work	[28]	[29]	[30]	[31]	[36]	[37]	[38]
Security	Protection against Replay	Yes	Yes	N/A	N/A	Yes	Yes	Yes	N/A
	Protection against Man-in-the-middle	Yes	Yes	N/A	N/A	Yes	N/A	N/A	N/A
	Protection against Cloning	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Performance	Enhances LoRaWAN security	Yes	No	No	No	No	No	No	No
	Low-cost IoT device	Yes	Yes	N/A	No	Yes	N/A	N/A	N/A
	Computation cost (ms)	1.43	52.65	N/A	N/A	52.65	N/A	N/A	N/A
	Storage cost (bit)	896	2302	N/A	N/A	N/A	N/A	N/A	N/A
	NFT Data Persistence	Yes	Yes	No	No	Yes	Yes	Yes	No
	Smart contract vulnerability analysis performed	Yes	No	Yes	Yes	No	No	No	Yes
	Direct compatibility with ERC721 standard	Yes	No[*1]	No	No	No[*2]	No	No	No
	Link blockchain account address of a device to	Device ID Signature	Tkn ID[*3]	Tkn ID	Tkn ID	Tkn ID	TknID	Tkn ID	Tkn ID
	Gas consumption for token creation	22,984	167,263	73,997	≥35,072	112,510	32,303	32,303	≥139,902

*1 – 8 additional attributes, 15 functions, and four events are required; *2 – 9 additional custom functions are required; *3 – Token ID

security but also outperform related work in terms of computation cost, storage cost, and gas consumption.

VII. Conclusion

This paper addressed the research gap by binding LoRaWAN nodes to on-chain NFTs for the purpose of secure proof-of-authenticity means without adding extra energy burden. The proposed lightweight authentication scheme is compatible with the existing LoRaWAN specifications and is fully supported by the resource-constrained LoRaWAN nodes as shown by the proof-of-concept. The proposed scheme's security was analyzed and proven to be secure from replay, man-in-the-middle, and cloning attacks. A performance analysis was conducted, and a thorough comparative analysis was done with related works. The proposed scheme has been demonstrated to outperform other comparable approaches, exhibiting a computation cost of 1.432 ms, a storage cost of 896 bits, and gas consumption totaling 22,984.

The future work includes allowing ownership transfer of NFTs bound with LoRaWAN nodes.

References

[1] "Cryptokicks iRL Is Now Here." RTFKT. https://cirl-lookbook.rtfkt.com (accessed 29 December, 2023).

[2] "Hotspot Owner Migration." Helium. https://docs.helium.com/solana/migration/hotspot-owner/#hotspots-become-nfts (accessed 29 December, 2023).

[3] F. Chiacchio, D. D'Urso, L. M. Oliveri, A. Spitaleri, C. Spampinato, and D. Giordano, "A Non-Fungible Token Solution for the Track and Trace of Pharmaceutical Supply Chain," *Applied Sciences,* vol. 12, no. 8, p. 4019, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/8/4019.

[4] S. Gerard. "How LoRaWAN Helps Improve Smart Supply Chain and Logistics." Moko LoRa. https://www.mokolora.com/smart-supply-chain-and-logistics/ (accessed 8 October, 2023).

[5] A. Bertolaud *et al.*, "LoRaWAN 1.1 Specification," LoRa Alliance, 2017. [Online]. Available: https://resources.lora-alliance.org/technical-specifications/lorawan-specification-v1-1

[6] J. Bellagarda and A. M. Abu-Mahfouz, "Connect2NFT: A Web-Based, Blockchain Enabled NFT Application with the Aim of Reducing Fraud and Ensuring Authenticated Social, Non-Human Verified Digital Identity," *Mathematics,* vol. 10, no. 21, p. 3934, 2022, doi: https://www.mdpi.com/2227-7390/10/21/3934.

[7] B. Hammi, S. Zeadally, and A. J. Perez, "Non-Fungible Tokens: A Review," *IEEE Internet of Things Magazine,* vol. 6, no. 1, pp. 46-50, 2023, doi: 10.1109/IOTM.001.2200244.

[8] J. H. Khor, M. Sidorov, and P. Y. Woon, "Public Blockchains for Resource-constrained IoT Devices -A State of the Art Survey," *IEEE Internet of Things Journal,* pp. 11960-11982, 2021, doi: 10.1109/JIOT.2021.3069120.

[9] "Blockchians Vie For NFT Market, But Ethereum Still Dominates - Report." Cointelegraph Research. https://cointelegraph.com/news/blockchains-vie-for-nft-market-but-ethereum-still-dominates-report (accessed 21st of February 2022).

[10] C. Cerrudo, E. M. Fayo, and M. Sequeira, "LoRaWAN Networks Susceptible to Hacking: Common Cyber Security Problems, How to Detect and Prevent Them," IoActive, 2020. [Online]. Available: https://act-on.ioactive.com/acton/attachment/34793/f-87b45f5f-f181-44fc-82a8-8e53c501dc4e/1/-/-/-/-/LoRaWAN%20Networks%20Susceptible%20to%20Hacking.pdf

[11] N. Hayati, K. Ramli, S. Windarta, and M. Suryanegara, "A Novel Secure Root Key Updating Scheme for LoRaWANs Based on CTR_AES DRBG 128," *IEEE Access,* vol. 10, pp. 18807-18819, 2022, doi: 10.1109/ACCESS.2022.3150281.

[12] J. Thomas, S. Cherian, S. Chandran, and V. Pavithran, "Man in the Middle Attack Mitigation in LoRaWAN," in *International Conference on Inventive Computation Technologies*, 26-28 February 2020, pp. 353-358, doi: 10.1109/ICICT48043.2020.9112391.

[13] J. Navarro-Ortiz, N. Chinchilla-Romero, J. J. Ramos-Munoz, and P. Munoz-Luengo, "Improving Hardware Security for LoRaWAN," in *IEEE Conference on Standards for Communications and Networking*, 28-30 October 2019, pp. 1-6, doi: 10.1109/CSCN.2019.8931397.

[14] J. Han and J. Wang, "An Enhanced Key Management Scheme for LoRaWAN," *Cryptography,* vol. 2, no. 4, 2018, doi: 10.3390/cryptography2040034.

[15] X. Chen, M. Lech, and L. Wang, "A Complete Key Management Scheme for LoRaWAN v1.1," *Sensors,* vol. 21, no. 9, p. 2962, 2021, doi: 10.3390/s21092962.

[16] S. A. A. Hakeem, S. M. A. El-Kader, and H. Kim, "A Key Management Protocol Based on the Hash Chain Key Generation for Securing LoRaWAN Networks," *Sensors,* vol. 21, no. 17, 2021, doi: 10.3390/s21175838.

[17] K. L. Tsai, F. Y. Leu, L. L. Hung, and C. Y. Ko, "Secure Session Key Generation Method for LoRaWAN Servers," *IEEE Access,* vol. 8, pp. 54631-54640, 2020, doi: 10.1109/ACCESS.2020.2978100.

[18] V. Ribeiro, R. Holanda, A. Ramos, and J. J. P. C. Rodrigues, "Enhancing Key Management in LoRaWAN with Permissioned Blockchain," *Sensors,* vol. 20, no. 11, 2020, doi: 10.3390/s20113068.

[19] S. M. Danish, M. Lestas, W. Asif, H. K. Qureshi, and M. Rajarajan, "A Lightweight Blockchain Based Two Factor Authentication Mechanism for LoRaWAN Join Procedure," in *IEEE International Conference on Communications Workshops*, 20-24 May 2019, pp. 1-6, doi: 10.1109/ICCW.2019.8756673.

[20] K. Mikhaylov, J. Petäjäjärvi, J. Haapola, and A. Pouttu, "D2D communications in LoRaWAN Low Power Wide Area Network: From idea to empirical validation," in *IEEE International Conference on Communications Workshops*, 21-25 May 2017, pp. 737-742, doi: 10.1109/ICCW.2017.7962746.

[21] J. Kim and J. Song, "A Secure Device-to-Device Link Establishment Scheme for LoRaWAN," *IEEE Sensors Journal,* vol. 18, no. 5, pp. 2153-2160, 2018, doi: 10.1109/JSEN.2017.2789121.

[22] A. Jabbari and J. B. Mohasefi, "A Secure and LoRaWAN Compatible User Authentication Protocol for Critical Applications in the IoT Environment," *IEEE Transactions on Industrial Informatics,* vol. 18, no. 1, pp. 56-65, 2022, doi: 10.1109/TII.2021.3075440.

[23] T. Rosati and E. Wood, "End-to-End Secure LoRaWAN: Secure Device with Key Management from Provisioning to Operations," Cypress, 2020. [Online]. Available: https://lora-alliance.org/wp-content/uploads/2020/11/cypress-escrypt-member-security-whitepaper_web-opt.pdf

[24] "The Current State of LoRaWAN Security - Technical Brief," 2021. [Online]. Available: https://documents.trendmicro.com/assets/pdf/The%20Current%20State%20of%20LoRaWAN%20Security.pdf

[25] J. Lin, Z. Shen, C. Miao, and S. Liu, "Using blockchain to build trusted LoRaWAN sharing server," *International Journal of Crowd Science,* vol. 1, no. 3, pp. 270-280, 2017, doi: 10.1108/IJCS-08-2017-0010.

[26] "WiSeKey Semiconductors NFTs and Blockchain Securing IoT Devices and the Supply Chain." GlobeNewswire. https://www.globenewswire.com/news-release/2022/02/18/2388012/0/en/WISeKey-Semiconductors-NFTs-and-Blockchain-Securing-IoT-Devices-and-the-Supply-Chain.html (accessed 10th of March, 2022).

[27] "SmartAxiom NFT Solution Secures the Identity and Ownership of Digital Assets." SmartAxiom. https://www.smartaxiom.com/solutions/nft/ (accessed 10th of March, 2022).

[28] J. Arcenegui, R. Arjona, R. Román, and I. Baturone, "Secure Combination of IoT and Blockchain by Physically Binding IoT Devices to Smart Non-Fungible Tokens Using PUFs," *Sensors,* vol. 21, no. 9, p. 3119, 2021, doi: https://doi.org/10.3390/s21093119.

[29] S. A. Gebreab, H. R. Hasan, K. Salah, and R. Jayaraman, "NFT-Based Traceability and Ownership Management of Medical Devices," *IEEE Access,* vol. 10, pp. 126394-126411, 2022, doi: 10.1109/ACCESS.2022.3226128.

[30] M. Turki, S. Cheikhrouhou, B. Dammak, M. Baklouti, R. Mars, and A. Dhahbi, "NFT-IoT Pharma Chain : IoT Drug traceability system based on Blockchain and Non Fungible Tokens (NFTs)," *Journal of King Saud University - Computer and Information Sciences,* vol. 35, no. 2, pp. 527-543, 2023, doi: https://doi.org/10.1016/j.jksuci.2022.12.016.

[31] J. Arcenegui, R. Arjona, and I. Baturone, "Secure Management of IoT Devices Based on Blockchain Non-fungible Tokens and Physical Unclonable Functions," Cham, 2020: Springer International Publishing, in Applied Cryptography and Network Security Workshops, pp. 24-40.

[32] M. M. Barhoush, B. H. Abed-Alguni, R. Hammad, M. A.-. Fawa'reh, and R. N. Hassan, "DES22: DES-Based Algorithm with Improved Security," *Jordanian Journal of Computers and Information*

Technology, vol. 8, no. 1, pp. 18-32, 2022, doi: 10.5455/jjcit.71-1632868199.

[33] "Rinkeby Etherscan - Transaction Details." https://rinkeby.etherscan.io/tx/0xe6b2409a2447eefc53349689849029787b7ee70abdf954926a0b7bd8a29754d7 (accessed 19 April, 2022).

[34] M. Amy, O. D. Matteo, V. Gheorghiu, M. Mosca, A. Parent, and J. Schanck, "Estimating The Cost of Generic Quantum Pre-Image Attacks," *Cryptology ePrint Archive,* 2016. [Online]. Available: http://eprint.iacr.org/2016/992.

[35] M. A. Ferrag and L. Shu, "The Performance Evaluation of Blockchain-Based Security and Privacy Systems for the Internet of Things: A Tutorial," *IEEE Internet of Things Journal,* vol. 8, no. 24, pp. 17236-17260, 2021, doi: 10.1109/JIOT.2021.3078072.

[36] U. Khalil, O. A. Malik, O. W. Hong, and M. Uddin, "Leveraging a novel NFT-enabled blockchain architecture for the authentication of IoT assets in smart cities," *Scientific Reports,* vol. 13, no. 1, p. 19785, 2023/11/13 2023, doi: 10.1038/s41598-023-45212-1.

[37] U. Khalil, M. Uddin, O. A. Malik, and O. W. Hong, "A Novel NFT Solution for Assets Digitization and Authentication in Cyber-Physical Systems: Blueprint and Evaluation," *IEEE Open Journal of the Computer Society,* vol. 5, pp. 131-143, 2024, doi: 10.1109/OJCS.2024.3378424.

[38] D. Hawashin, M. Nemer, K. Salah, R. Jayaraman, D. Svetinovic, and E. Damiani, "Blockchain and NFT-based traceability and certification for UAV parts in manufacturing," *Journal of Industrial Information Integration,* vol. 39, p. 100597, 2024, doi: https://doi.org/10.1016/j.jii.2024.100597.

[39] "Packet Size Considerations." LoRa. https://lora-developers.semtech.com/documentation/tech-papers-and-guides/the-book/packet-size-considerations/ (accessed 19th May, 2022).

[40] G. LIkhitskaya. "Do NFT Loopholes Uncover NFT Security Issues?" Security Boulevard. https://securityboulevard.com/2022/02/nft-loopholes-uncover-nft-security-issues/ (accessed 9th of March, 2022).

[41] A. Lynn. "The Things Industries partnership to benefit LoRaWAN." Electronic Specifier. https://electronicspecifier.com/news/latest/the-things-industries-partnership-to-benefit-lorawan (accessed 19th January, 2022).

[42] *ATECC608A for LoRaWAN™ Data Sheet*, M. T. Inc, 2020. [Online]. Available: https://www.microchip.com/en-us/product/ATECC608A

[43] *ATECC608B-TNGLoRaWAN CryptoAuthentication™ Data Sheet*, M. T. Inc, 2020. [Online]. Available: https://www.microchip.com/en-us/product/ATECC608B

[44] "STSAFE-A110," 2019. [Online]. Available: https://www.st.com/resource/en/datasheet/stsafe-a110.pdf

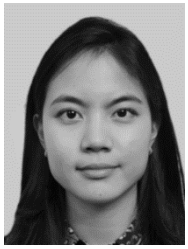
[45] "Cypress and Semtech Collaborate on Integrated LoRaWAN™ Solution for Smart City Applications." BusinessWire. https://www.businesswire.com/news/home/20180606005455/en/Cypress-and-Semtech-Collaborate-on-Integrated-LoRaWAN%E2%84%A2-Solution-for-Smart-City-Applications (accessed 19th January, 2022).

[46] J. Khor, M. A. Masama, M. Sidorov, W. Leong, and J. Lim, "An Improved Gas Efficient Library for Securing IoT Smart Contracts Against Arithmetic Vulnerabilities," in *Proceedings of the 9th International Conference on Software and Computer Applications*, Langkawi, Malaysia, 2020, pp. 326–330, doi: 10.1145/3384544.3384577.

Michail Sidorov received his Ph.D in Computer Science and Engineering from Toyohashi University of Technology (TUT), Japan in 2020. He was a Teaching Fellow at the University of Southampton Malaysia (UoSM), and worked as a Researcher after obtaining his PhD in TUT, subsequently he was an ERCIM postdoctoral fellow at the NTNU. He is an Assistant Professor with the UoSM. His research interests include sensor design, blockchain, and IoT.

Jing Huey Khor received her Ph.D degree at Universiti Sains Malaysia in 2013. She is an Assistant Professor with the University of Southampton Malaysia. Her interests include designing privacy preserving protocols for communication between IoT devices and blockchain, new consensus algorithms, and decentralized application for IoT purposes.

Alvin Chern Hao Wong is currently pursuing the Master of Engineering degree in Electrical and Electronics with the University of Southampton. His interest include digital systems, IoT, cryptography, as well as hardware and network security.



Ying Ying Lee is a third-year Electrical and Electronics Engineering student at the University of Southampton. Her interests include security in IoT, computer networking, cryptography, and blockchain technology.

Jingyue Li received the Ph.D. degree in software engineering from the Department of Computer Science, Norwegian University of Science and Technology (NTNU), in 2006. He is a Professor with the Computer Science Department, NTNU. His recent research interests include software engineering, software security, blockchain, and big data.