IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 August 2024, accepted 25 August 2024, date of publication 28 August 2024, date of current version 10 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3451134

RESEARCH ARTICLE

Performance Evaluation and Analysis of LightCert4IoT Using Cooja-Contiki Simulator

DAVID KHOURY[1], (Member, IEEE), SAMIR HADDAD[2], PATRICK SONDI[3], GABY ABOU HAIDAR[4], (Senior Member, IEEE), DAVID SEMAAN[4], (Member, IEEE), AND JINANE SAYAH[5]

[1]Laboratoire d'Informatique Signal et Image de la Côte d'Opale (LISIC UR 4491), Université du Littoral Côte d'Opale, 62100 Calais, France
[2]Department of Computer Science and Mathematics, Faculty of Arts and Sciences, University of Balamand, Koura, Tripoli, Lebanon
[3]Center for Digital Systems, Institut Mines–Télécom, IMT Nord Europe, 59650 Lille, France
[4]Computer Science Department, American University of Science and Technology (AUST), Beirut 1107-2020, Lebanon
[5]Department of Telecom and Networks, Issam Fares Faculty of Technology, University of Balamand, Koura, Tripoli, Lebanon

Corresponding author: Samir Haddad (Samir.haddad@balamand.edu.lb)

ABSTRACT Recently, the focus on the Internet of Things (IoT) has rocketed in parallel with the evolution of mobile fifth-generation (5G) networks. To fully utilize IoT, which wirelessly connects billions of devices, a 5G network is essential. However, the provision of user's certificates exposes the network to serious security threats. The LightCert4IoT leverages the advantages of EDGE nodes with blockchain technology and smart contracts to address the existing challenges of PKI (Public Key Infrastructure) certificates in IoT devices, which neatly achieves certificate issuance, update, and revocation more securely and efficiently. An end-user issues a self-signed certificate and lets Local Registration Authorities (LRAs)/EDGE nodes to verify and validate the binding identity signed certificate of the users through a blockchain such as Ethereum. This work analyzes the performance of LightCert4IoT in IoT devices by utilizing the Cooja-Contiki simulator. The results show reduced energy consumption and memory size when compared to the conventional X509 certificate. In conclusion, the LightCert4IoT meets the requirements of major IoT device constraints. The paper also addresses the security threats with the widespread use of IoT devices and analyses the adversary on security level from the authentication based on hardware token and blockchain and deploying firewall to protect servers against malicious attacks like DDOS. Scalability, resource limitations, and privacy concerns are analyzed as IoT devices handle sensitive information, and the transparent and immutable nature of blockchain can raise data protection. Analysis of Network partitioning challenge and communication overhead

INDEX TERMS Certificate, blockchain, public key infrastructure, authentication, IoT-5G, simulation.

I. INTRODUCTION

For the past decades, the communications of the various IoT devices were wireless, which is subject to vulnerabilities and security threats [1], [2]. Their massive adoption in many domains, including smart factories [3], [4], infrastructure monitoring [5], domotics [6], and sensor networks in health care [37] imposes efficient solutions that address these security issues. More particularly, authentication is the most critical and challenging security requirement for the IoT environment, where external entities directly access the information from remote devices [2]. Device authentication could rely on a Certificate Authority (CA), which assigns a public certificate to an IoT device [7]. A similar method is applied for the network servers using certificate-based authentication defined by the X.509 Public Key Infrastructure (PKI) standard.

To address these problems, various proposals have been considered like the so-called Concise Binary Object Representation (CBOR) encoding to suitably design lightweight X.509 profiles for IoT-constrained devices [8], [9], [10].

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Sookhak.

LightCert4IoTs is a blockchain-based digital certificate system for IoT devices [11]. The proposed system lets end-users (i.e., IoT devices) generate self-signed certificates which are then stored in the blockchain by a Local Registration Authority (LRA).

This paper's primary objective is to validate the simulation findings of the LightCert4IoTs certificate using Contiki OS, Cooja. The assessment covers the certificate processor energy consumption and memory size with the complexity of deployment in comparison with current systems.

The main contribution of this work is the following:

- Simulation with analyzed results from different performance aspects of the LightCert4IoT. The test was conducted within the Cooja-Contiki network simulator, [22] with an emphasis on analyzing the energy consumption and memory size.
- The paper also lists and explores the security threats with the widespread use of IoT devices and analyses the benefit of adopting LightCert4IoT on the security level.

The rest of this paper is structured as follows: The paper's background material is included in Section II, along with an overview of blockchain technology in general and current PKI/CA problems in the context of IoT applications. A thorough explanation of the proposed system is provided, which includes entities and their functionalities, in Section III, followed by sections on Simulation, Security Analysis, Conclusion, and Future Work.

This paper does not focus on the hardware attacks including side channel and physical attacks on IoT. Nevertheless, in the Security Analysis chapter, we explain these threats.

II. BACKGROUND AND RELATED WORK

A. OVERVIEW OF LightCert4IoT METHOD

Lightcert4IoTs [11] allows the end user IoT devices to generate self-signed certificates stored in the blockchain after authentication and verification by an LRA.

Lightcert4IoT is implemented using the Ethereum solidity smart contract platform. This solution targets the client-side certificate intended for an IoT device or any other client like a mobile terminal, a Tablet, etc., not the server-side. It is a new method to authenticate a client by assigning a light certificate without the need for PKI/CA. Lightcert4IoT allows end-users to create a self-signed certificate and let the local registration authorities (LRAs) or edge nodes verify and validate the identity of the users and store the LightCert4IoT in the Ethereum blockchain platform.

The authentication is based on hardware tokens and blockchain. The globally unique HW serial number (SN) or a token issued by the Electronic Notary (EN) may be used to uniquely identify an IoT device. During the IoT device's registration in the LRA, the binding between the user ID and its token or SN is completed.

LRA server must keep track of a mapping between the client's Universal User Identity (UUID) and the token and/or client wallet address. The advantages of the LightCert4IoT are the following:

- Solves the complexity of the assignment of a signed certificate to an IoT device in comparison with the current PKI/CA method.
- LightCert4IoT is smaller in size since most of the information in X509 is not needed or relevant for the IoT case.
- The authentication of the IoT device is HW based through SN or Token and approved by the LRA.
- The verification of the LightCert4IoT certificate is conducted via LRA in the Blockchain network.
- The solution is scalable and has no single point of failure on the network level, thanks to Blockchain and EDGE nodes identified by LRA (see security analysis chapter).

See below the sequence diagram in Figure 1 extracted from reference [11], which illustrates the registration process and storage of an IoT device LightCert4IoT in Blockchain:

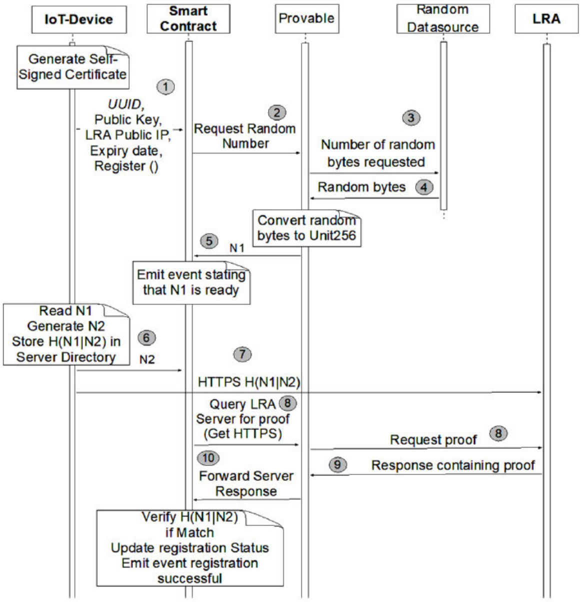

IoT-Device
Smart Contract
Provable
Random Datasource
LRA
Generate Self-Signed Certificate
1
UUID, Public Key, LRA Public IP, Expiry date, Register ()
2
Request Random Number
3
Number of random bytes requested
Random bytes 4
Convert random bytes to Unit256
5
N1
Emit event stating that N1 is ready
Read N1 Generate N2 Store H(N1|N2) in Server Directory
6
N2
7
HTTPS H(N1|N2)
Query LRA 8 Server for proof (Get HTTPS)
Request proof 8
9
Response containing proof
10
Forward Server Response
Verify H(N1|N2) if Match Update registration Status Emit event registration successful

FIGURE 1. Blockchain-based LightCert4IoT client's devices storage and control validation.

B. CRYPTOGRAPHY PROTOCOL IN LightCert4IoT

Elliptic Curve Cryptography (ECC) is a method of public-key encryption that is built on the algebraic structure of elliptic curves. Compared to non-EC encryption, ECC enables smaller keys while offering comparable security [32], [33], [34]. Elliptic Curve Cryptography (ECC) is a widely used public key encryption technique, several studies have shown that ECC provides superior security and requires smaller key sizes to provide the same level of security as RSA. Elliptic curve Diffie–Hellman (ECDH) and Digital Signature Algorithm (ECDSA), optimized for the IoT have been implemented and tested by Pinol et al. [35]. When compared to homogeneous and affine coordinate systems, the test demonstrates that the use of the Jacobian coordinate system provides higher performance and has smaller memory footprints. For a 256-bit key size, key generation takes about 5000 ms and requires 76 mJ of energy. Additionally, signature verification and generation take approximately 11 and 5 seconds, 154 and 76 mJ respectively.

C. IoT SECURITY SYSTEMS BASED ON BLOCKCHAIN

Several recent works propose to use of the blockchain for various security purposes in IoT systems. The authors of [5] have shown that the blockchain can contribute to accelerating the identification of fake IoT devices in large-scale IoT systems. Another recent work investigates the potential of the blockchain as a digital ledger to support decentralized audits in open and mobile IoT-sensitive systems such as vehicle networks [6].

Won et al. [12] present a PKI called IoT-PKI that is decentralized and built on a blockchain network. IoT-PKI uses distributed nodes to handle scalability.

Högland [13] made DECKIN, a PKI solution on top of an existing blockchain protocol. It provides a practical key-management solution. It employs Physical Unclonable Functions (PUFs) to address the system's key management issues.

Three distinct blockchain-based alternatives to conventional CA-PKI for certificate administration are proposed and analyzed by Singla and Bertino [14]. The first proposal makes use of the Emercoin blockchain that provides Name Value Storage (NVS). The second approach employs Ethereum smart contracts, and the final proposal uses Ethereum Light Sync mode, which does not require a remote blockchain node, unlike the first two proposals.

Magnusson [15] evaluated the performance of an existing PKI that uses smart contracts on the Ethereum blockchain, by deploying it on a Raspberry Pi 2. The author found that deploying the PKI to this IoT-like device required over 20 GB of storage to store the blockchain.

Mustafa Al-Bassam [16] proposes SCPKI, a smart contract-based PKI and identity system.

Pranav Gangwani et al [38] proposes a technique for IoT identity management called PUF-based Device Identity Management (PUF-DIM) that employs Physical Unclonable Function (PUF) to perform device identity management to establish trust in the data associated with each device and the device's unique identifier. Moreover, a review of the major security problems with IoT and how blockchain plays a significant role in tackling those issues is discussed.

Pranav Gangwani et al [39] deal with the integration of Data Science and IoT with Blockchain for Industry 4.0, and overcomes the major challenges of the traditional blockchain to enable seamless integration with IoT devices. Three IIoT applications are proposed and elaborated, consisting of (1) Device Identity Management, (2) Sensor data anomaly detection using Artificial Intelligence (AI), and (3) Security of IIoT data.

The following recent and specific articles on blockchain security and scalability issues must be considered:

[40] This paper introduces a novel blockchain-based framework to ensure the security and integrity aspects of IoT data in smart cities.

[41] The paper introduces a lightweight scalable blockchain for IoT with trust-based consensus to boost scalability and throughput without sacrificing the security and privacy of IoT data, sidechaining with the efficient Grey Wolf Optimization (GWO) algorithm, allowing the construction of parallel chains connected to the main blockchain.

[42] The paper deals with detecting Security Breaches in Smart Contracts through techniques and tools.

D. CONTIKI-COOJA SIMULATION

Contiki is an Operating System (OS) designed for the Internet of Things. The most important aspects that should be concentrated are the memory size, power of the constrained devices, and their processing capabilities. Contiki is a free and open-source operating system for IoT development that was built on the C programming language. With wireless networks, IoT devices can communicate quickly and securely thanks to Contiki. It is built with an event-driven kernel that supports pre-emptive multithreading.

Cooja is a network simulator, used in Contiki for simulation purposes. Cooja has been developed in JAVA. An interface represents a sensor node, and the plugin is used to cooperate with the simulation. Java Native Interface (JNI) is used to link the simulator with Contiki thereby allowing applications to run in Contiki. This approach has laid the foundations for applications to run on a real sensor node.

The underlying libraries of RFID chips and sensors are provided in C in the Cooja network simulator, which is used to program Contiki. The back-end C programs and associated header files can be modified and recompiled to achieve the required results for programming, controlling, and monitoring the remote IoT devices. Contiki integrates lightweight protocols into IPv4 and IPv6 networking so that radio frequency and low-power processors can be connected without experiencing performance concerns [22].

A recommended work on Cooja-Contiki for reference is "OSCAR: Object Security Architecture for the Internet of Things". [36]

III. SYSTEM MODEL AND IMPLEMENTATION

A. SUMMARY OF THE PROPOSED SYSTEM

The introduction of LightCert4IoTs provides the opportunity for end users and IoT restriction devices to get certifications safely and affordably. Although the LightCert4IoT certificate format is based on the X509 certificate format, several fields have been condensed. The LightCert4IoT certificate is not created by CAs; rather, the end-user device creates a self-signed certificate when LRAs confirm its identity. A self-signed certificate is created by the client. The identity of the client and the related token are tracked by the

LRA server. The LRA server through the Ethereum wallet and WEB3 can access the Ethereum blockchain via a light client LES [31] or interworks with the Ethereum blockchain as a complete node, operating as a miner, storage node, or validator. The LightCert4IoT smart contract module on the Ethereum blockchain serves as a certifying authority, validating the end-users public keys once LRA has provided the user identity.

The LightCer4IoT certificate format contains mainly the following data:

- **UUID (User Identity)** is a unique and universal device identity derived from the Token/SN assigned to each IoT device.
- **Public Key** generated by the IoT device plus identifier of the algorithm for which this key is to be used, together with any associated parameters.
- **Expiry date**: period of validity, consists of two dates: the first and last on which the certificate is valid. (This Data could be optional if there is no expiration date considered for the IoT device).
- **LRA domain name** or public IP address where IoT device identity has been verified.
- **Signature** of the self-signed certificate

B. SYSTEM MODEL USING CONTIKI-COOJA

The system model presented in Figure 2 shows the various players who work together to manage the LightCert4IoTs. LRAs authenticate user identities using credentials provided by end users' clients (IoT devices). A unique Token assigned to the IoT device and supplied by the IoT application during configuration, or a unique SN (Serial Number) assigned to each device during production and configured in the LRA are the bases for the identity verification. Each HW device has a special identification number known as an SN thanks to which it is confirmed during Device configuration by the LRA.

The end-user device issues a self-signed certificate after LRAs attest to its identification. The blockchain is communicated using the LRA server. The lightcert4IoT storing numerous restricted devices is carried out as a single transaction by the LRA server with direct communication with the blockchain. The end user can also communicate with the blockchain directly using the Light Ethereum Subprotocol (LES). However, this option may not be suitable for devices with limited processing and memory.

The Light Ethereum Subprotocol (LES) is the protocol for use by "light" clients, which only download block headers. They provide full functionality in terms of safely accessing the blockchain, but do not mine and therefore do not take part in the consensus process. Light clients can be in any type of device where the memory is limited but has the advantage that the device can have direct access to Ethereum.

The simulation setup involves authenticating IoT devices using LRAs, issuing self-signed certificates, and storing them on the Ethereum blockchain. COOJA tools are used to visualize, control, and analyze the simulation, providing insights into network performance and device interactions.

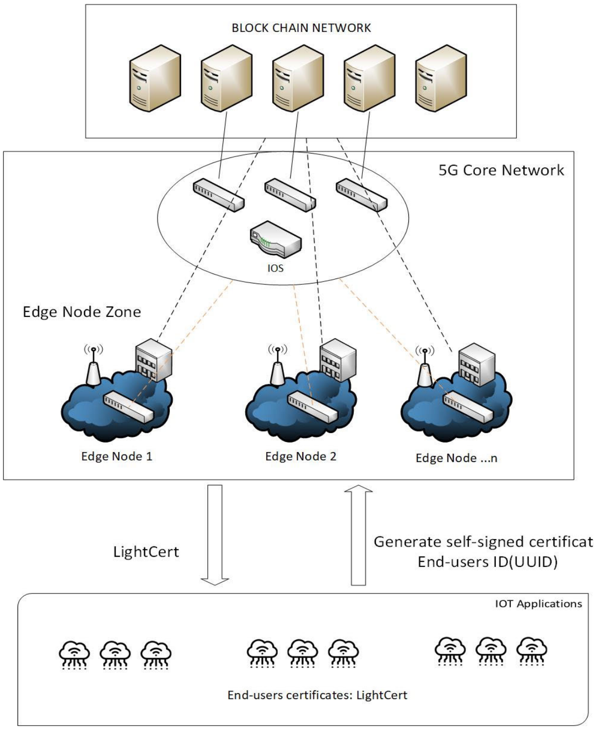


FIGURE 2. A general framework for the proposed system LightCert4IoT.

C. COMPONENTS OF THE SOLUTION

a) Client: The Client is a user identity connected to IoTs. The client must first appear within the LRA to apply for the light certificate. After the LRA confirms the binding identification and ensures that the user's information corresponds with that in the certificate topic, the end-user self-signed certificate is sent to the LRA for vouching. IoT applications, such as those for mobile devices, connected autos, hospitals, etc. The globally unique HW serial number SN or a token issued by the Electronic Notary (EN) may be used to uniquely identify an IoT device. During the IoT device's registration in the LRA, the binding between the user ID and its token or SN is completed. When configuring the IoT device, the binding is completed at the LRA.

b) LRAs: An LRA serves as a bridge between the Blockchain and the Client (end-user). LRA can also function as a public key certificate voucher and as an Edge Node. LRA server function can be integrated inside the Multi-access Edge Computing (MEC) node in the 5G network to handle IoT device security and authentication. The end user can provide the LRA with its credentials to request a certificate. Practically speaking, the LRA could consist of a mobile operator bank, an IT company, an intelligence manufacturing company, a part of a 5G network connected to IoT

TABLE 1. System model parameters.

Players and Roles:
LRAs (Local Registration Authorities): Authenticate user identities based on credentials from IoT devices.
IoT Devices: Each has a unique Token or Serial Number (SN) for identity verification.
End-User Device: Issues a self-signed certificate after LRA authentication.
Blockchain: Used for storing LightCert4IoT transactions via the LRA server.
LES (Light Ethereum Subprotocol): Protocol for light clients to access the blockchain.
Solution Components
Client (IoT Device):
- Connected to IoTs and must apply for a light certificate from the LRA.
- Identified by a globally unique HW serial number SN or a token issued by the EN.
- Registers in the LRA, binding user ID with token/SN.
LRAs:
- Bridge between Blockchain and Client.
- Serve as a public key certificate voucher.
- Function as Multi-access Edge Computing (MEC) servers in a 5G blockchain network.
- Authenticate IoT devices and manage wallets, providing necessary ether for transactions.
LightCert4IoT:
- Integrated into client mode, though simulated separately due to COOJA constraints.
- Converts LightCert4IoT into client mode.
d) **Blockchain Network:**
- Ethereum-based smart contract platform for LightCert.
- Stores public keys and device identities in a decentralized manner

applications, or an organization near the end-user, like their insurance provider, bank, postal service, etc. Typically, these organizations are already competent to confirm identities. The storing of numerous restricted devices is carried out as a single transaction by the LRA server, which interfaces directly with the blockchain.

c) LightCert4IoT: This mote ought to be integrated into client one, however, because the Cooja simulation was unable to run both codes concurrently, it was split into two independent motes that coexist as one. Therefore, the primary goal of this node is to switch lightCert4IoT into client mode.

d) Blockchain Network: A smart contract that functions on the Ethereum blockchain network makes up the majority of the LightCert platform. The LightCert module accepts the public keys and other information related to the identities of the devices, acting as a decentralized key store. We store the LightCert4IoT on the platform in our proposal. The goal is to incorporate Blockchain, a decentralized network with no central authority, into the domain verification procedure.

D. SYSTEM IMPLEMENTATION IN CONTIKI-COOJA

For this research work, the simulation results are obtained by starting the Contiki OS and the Cooja Simulator. In the following system, the end-user is referred to as the client Mote (IoT Device) Figure 3. The client's light certificate establishes the public and private keys automatically, creates a blank Ethereum wallet, and begins by registering its identification, including its public key, with the LRA server. The main role of the LRA is to authenticate the LightCert client module in constrained devices like IoT and send the necessary ether to the IoT device to be able to store the IoT device's public key in the LightCert smart contract on the Ethereum Blockchain network. By providing ether to the client (IoT) and approving the storage transaction in the blockchain, the LRA carries out the functionality of managing wallets. A random token is assigned by the LRA, and the Public Key is assigned to the IoT device as a distinct global identity. The IoT device (client) creates a self-signed certificate, which in our project is the Lightcert4IoT. Instead of directly accessing the Blockchain, the IoT gadget might store an Ethereum wallet address. The gadget can connect to the blockchain network thanks to the wallet address and required Ether that the LRA server provides. In this regard, it is the responsibility of the LRA server to keep track of a mapping between the client's UUID and the token and/or wallet address that corresponds to it. There are two options available to the IoT in this regard: either directly storing the lightCert4IoT in the blockchain or storing it via the LRA.

1) COOJA SIMULATION TOOLS

The primary simulating tools that play a significant part in assisting us in obtaining everything we may need in the transaction between IoT devices are all present in the COOJA simulation window. There are five key tools in it.

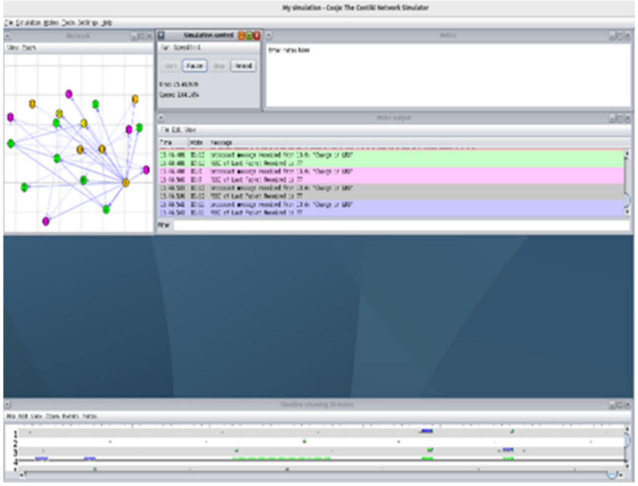
- Network - Displays where each node in the network is located. It is used to visualize each node's status. In our example, client-client, client-LRA, and client-blockchain were the parties involved in the transaction and communication.
- Simulation Control - Steps in the simulation can be started, paused, reloaded, or carried out using this panel. It displays the execution time and simulation speed.

TABLE 2. System implementation in Contiki COOJA.

System Implementation in Contiki COOJA:
Client Mote (IoT Device):
• Establishes public/private keys, creates Ethereum wallet, and registers identity with LRA.
• LRA authenticates and sends ether to store the IoT device's public key on the blockchain.
• IoT device creates a self-signed certificate (LightCert4IoT).
COOJA Simulation Tools:
• **Network:** Visualizes node status and transaction/communication paths (client-client, client-LRA, client-blockchain).
• **Simulation Control:** Controls simulation execution (start, pause, reload).
• **Mote Output:** Displays serial interface output of nodes, used for transaction sniffing.
• **Timeline:** Shows simulation messages/events (channel changes, LED changes, log outputs).
• **Notes:** Records thoughts on the simulation.
• **Additional Tools:** Breakpoints, Radio messages, Script editor, Buffer view, Mote duty cycle.
Simulation Results:
• Divided into four parts: client, LRA, Lightcert4IoT, and blockchain.
• Each mode has a distinct code for maintaining functionality.
• Parameters like CPU usage, bandwidth consumption, ETX, network latency, packet counts, beacon interval, radio duty cycle, and average power are computed.
Data Collection and Analysis:
• **Command Sending:** Start data collection by sending commands to nodes.
• **Data Collected:** Includes topological graph, sensor data (temperature, battery life), network data, and energy usage. • **ETX Calculation:** ETX = 1/(df x dr), helps determine network latency and other metrics.

It means that we can execute the events much more quickly than real-time execution would allow. Mote output - Shows all output of the serial interface of the nodes. It is possible to enable one window of Mote output for each node in the simulation. We used it as a sniffing tool to see the transaction.

- Timeline - Simulation messages and events, such as channel changes, LED changes, and log outputs, are displayed on this timeline. Our simulation uses it to determine when the client begins the exchange process.
- Notes - Used as a straightforward notebook to record thoughts on the simulation.



Legend: Green motes: client
Orange motes: LRA
Purple motes: LightCert4IoTs certification
Yellow motes: Blockchain

FIGURE 3. Implementation of LightCert4IoTs in Contiki-Cooja.

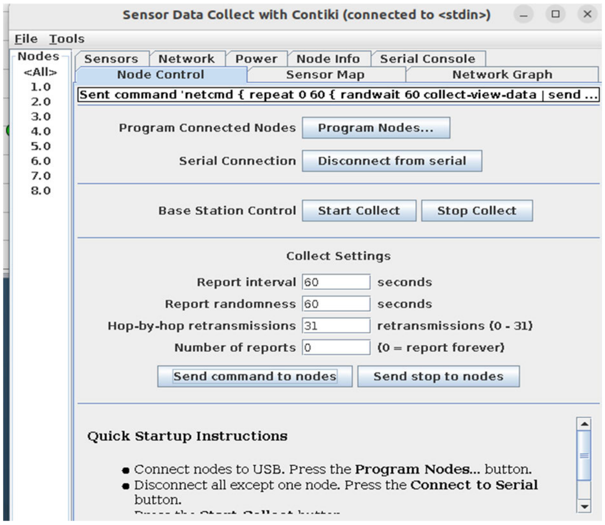
In addition to the standard tools, the menu allows you to enable additional tools like Breakpoints, Radio messages, Script editor, Buffer view, and Mote duty cycle.

IV. SIMULATION RESULTS

This research consists of four main parts: the client, the LRA, the Lightcert4IoT, and the blockchain. Each mode has a distinct code that oversees maintaining the project's functionality. Cooja was used to begin gathering opinions regarding the exchange between clients after configuring each mote and starting the experiment. By setting the CPU usage parameters in the collect view, we can compute the CPU and bandwidth consumption figures. Sending commands to the nodes should come first before beginning the collection (Figure 4).

We can start gathering data from nodes once the command has been sent to them. The information gathered from sensors includes a topological graph (Network graph, Sensor Map), as well as information about temperature and battery life. Additionally, we can gather the network data and energy usage that are most important to our project.

The Expected Transition count (ETX), which is frequently used in wireless routing to distinguish between paths that require a large number of packet transmissions from those that require a smaller number of packet transmissions for successful packet delivery and acknowledgment, can be extracted from the network information collected by Cooja and it is calculated using the forward packet delivery ratio (denoted df), i.e. ETX = 1/(df x dr). Additionally, it aids in determining network latency, the quantity of received and dropped packets, and the beacon interval, or the period



Sent command 'netcmd { repeat 0 60 { randwait 60 collect-view-data | send ...

Program Connected Nodes

Serial Connection

Base Station Control

Collect Settings

Report interval 60 seconds

Report randomness 60 seconds

Hop-by-hop retransmissions 31 retransmissions (0 - 31)

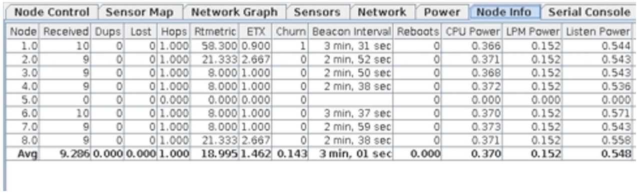
Number of reports 0 (0 = report forever)

Quick Startup Instructions

- Connect nodes to USB. Press the **Program Nodes...** button.
- Disconnect all except one node. Press the **Connect to Serial** button.

FIGURE 4. Sending command to node and start collection.

between beacon frames sent by an access point. Furthermore, we can determine the radio duty cycle and average power for each node.



Node	Received	Dups	Lost	Hops	Rtmetric	ETX	Churn	Beacon Interval	Reboots	CPU Power	LPM Power	Listen Power
1.0	10	0	0	1.000	58.300	0.900	1	3 min, 31 sec	0	0.366	0.152	0.544
2.0	9	0	0	1.000	21.333	2.667	0	2 min, 52 sec	0	0.371	0.152	0.543
3.0	9	0	0	1.000	8.000	1.000	0	2 min, 50 sec	0	0.368	0.152	0.543
4.0	9	0	0	1.000	8.000	1.000	0	2 min, 38 sec	0	0.372	0.152	0.536
5.0	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	0.000
6.0	10	0	0	1.000	8.000	1.000	0	3 min, 37 sec	0	0.370	0.152	0.571
7.0	9	0	0	1.000	8.000	1.000	0	2 min, 59 sec	0	0.373	0.152	0.543
8.0	9	0	0	1.000	21.333	2.667	0	2 min, 38 sec	0	0.371	0.152	0.558
Avg	**9.286**	**0.000**	**0.000**	**1.000**	**18.995**	**1.462**	**0.143**	**3 min, 01 sec**	**0.000**	**0.370**	**0.152**	**0.548**

FIGURE 5. Collected data by the sensors.

A. POWER CONSUMPTION

Before and after utilizing LightCert4IoT, the amount of energy consumed by each component of the certificate handling process was measured. Figures 6 and 7 show the average power consumption for X509 and LightCert4IoT respectively.

The numbers in the table are the percentage of total power consumption for each function in IoT device. The Power is measured in (mW).

The results show that the LightCert4IoT certificate uses less power when compared to the conventional X509 certificate. However, for the Live Partition Mobility (LPM) it is the same.

The average consumption of the X509 certificate is higher than that of LightCert4IoT, and there are significant differences in several important factors such as CPU usage, radio listening, and radio transmission. The X509 certificate consumes 0.38 units of power when using the CPU, compared to 0.35 units for LightCert4IoT, an increase of 0.03 units. Although it may seem small, it can accumulate over time in devices that are constantly running, increasing energy consumption and shortening battery life.

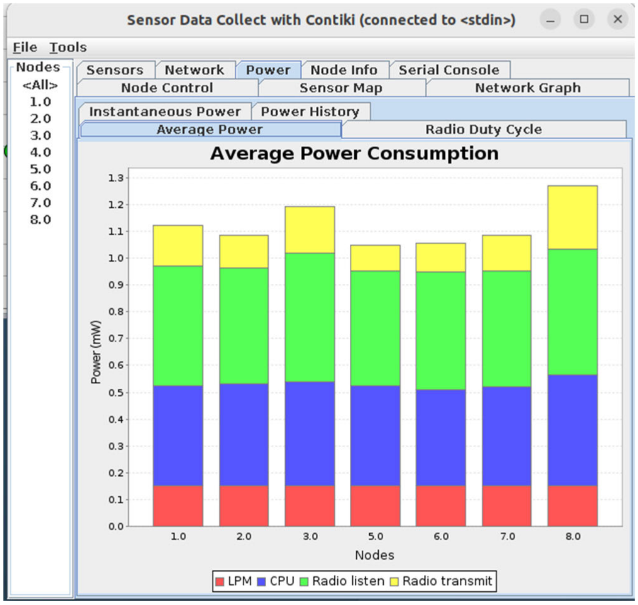


FIGURE 6. Average power consumption for the X509 certification.

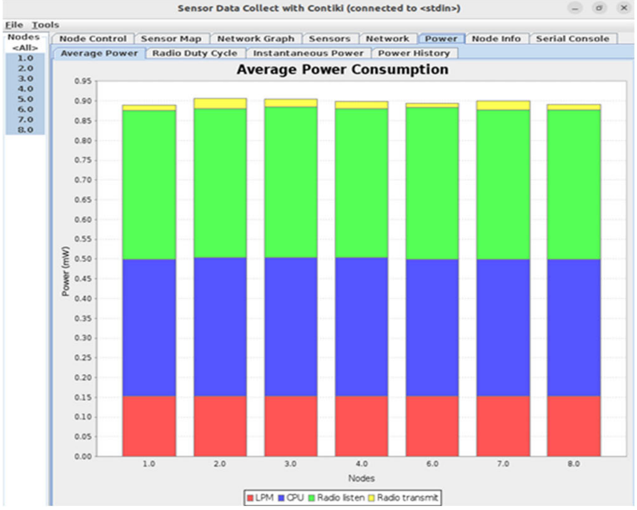


FIGURE 7. Average power consumption for the LightCert4IoT.

In Radio Listening, the X509 certificate needs 0.47 units of power, but the LightCert4IoT needs only 0.35 units. This 0.12-unit difference makes sense because IoT devices spend a fair amount of time listening to incoming signals. Lower power consumption with LightCert4IoT extends the device‘s operating time between charges, which is critical in battery-powered or remote applications where battery replacement or charging is difficult.

The most obvious difference is in the radio transmission requirements. where LightCert4IoT needs only 0.01 units, while the X509 certificate uses 0.15 units - a difference of 0.14 units. This reduction is necessary for systems that transmit data frequently because transmitting often consumes more power than listening. LightCert4IoT promotes more

TABLE 3. Energy Consumption.

Power	X509 certificate	LightCert4IoT	Difference
LPM	0.15	0.15	0
CPU	0.38	0.35	0.03
Radio listen	0.47	0.35	0.12
Radio Transmit	0.15	0.01	0.14

frequent data transmission without reducing battery life by reducing the required transmission.

The benefits of reduced power consumption in these areas are multifaceted:

- **Extended Battery Life**: Lower power usage leads to a longer battery life, by reducing the frequency of recharges or replacements. This is essential for IoT devices in remote or inaccessible locations.
- **Cost Efficiency**: Reduced energy consumption lowers operating costs, especially in large-scale deployments where power costs can accumulate significantly.
- **Environmental Impact**: Lower power usage results in a smaller carbon footprint, supporting greener and more sustainable technology solutions.
- **Enhanced Performance**: Devices with lower power requirements operate more reliably and efficiently, which leads to minimizing the risk of power-related disruptions.
- **Scalability**: Lower power needs enable more devices to be supported within the same power budget, facilitating the scalability of IoT networks.

In summary, LightCert4IoT's lower power consumption in CPU usage, radio listening, and radio transmission make it a superior choice for IoT applications. It offers substantial advantages in terms of battery life, cost efficiency, environmental impact, performance, and scalability, making it an ideal solution for modern IoT deployments.

B. ANALYSIS OF THE RESULTS

Using LightCert4IoT certificates can lead to reduced energy consumption and memory usage in IoT devices due to several key factors.

1. The Lightweight Cryptographic Algorithms (ECC) provide high security with smaller key sizes compared to traditional RSA. Smaller keys require less computational power and memory. This algorithm is optimized for low-power devices, reducing the number of operations and, consequently, energy consumption.
2. LightCert4IoT certificates are designed to be smaller in size, reducing the amount of memory required to store them and the bandwidth needed for transmission. In addition, Efficient encoding schemes, such as ASN.1 DER or compressed formats, minimize the certificate's footprint. Moreover, the certificates are simplified to include only essential information, making parsing and validation processes faster and less resource-intensive.
3. Techniques like session resumption can avoid repeated full handshake processes, saving energy and memory. LightCert4IoT is often used with protocols specifically designed for IoT, such as CoAP (Constrained Application Protocol) and DTLS (Datagram Transport Layer Security), which are optimized for low power and limited resources. Furthermore, devices can enter low-power sleep modes when not in use, waking up only for necessary cryptographic operations. Also, smart scheduling of cryptographic operations to coincide with periods of high-power availability or low device activity can help manage energy consumption efficiently.
4. The alternative to using dedicated hardware like secure elements or cryptographic accelerators can offload processing from the main CPU, reducing energy consumption.

C. CERTIFICATE SIZE: LightCert4IoT

By examining the features and attribute fields of the X509 standard, we can determine the size of the LightCert4IoT. Datagram Transport Layer Security (DTLS) profile for the Internet of Things standard [24] and Filip Forsby's "Digital Certificates for the Internet of Things" are the foundations for the computation. Results showed that the LightCert4IoT profile largely contains the following data and uses less memory than the X509-based IoT profile: According to the findings, the LightCert4IoT has a smaller memory footprint than the IoT profile based on X509 and mostly stores the following information: User Identity should be a Token or serial number SN, Public Key of the constrained device, Expiry date, LRA domain name, and others.

Device UUID (User Identity): The client generates a self-signed certificate in response to the device being identified by a user identity (UUID) created during configuration on the LRA server. Maintaining a mapping between the client's SN or token and its matching UUID in the LRA server.

Public key: The cryptography procedure is applied, and the enclosed device carries the public key as a bit string. Our lightCert4IoT certificate will use the ECDSA with the SHA256 algorithm.

The expiration date, which includes the certificate's beginning and ending dates, indicates how long the certificate will be valid.

LRA domain name: depicts the local registration authority's domain name. LRA IP address: This is a representation of the LRA's primary IP address, which is 32 bits, or 0.004 kilobytes.

We estimate below the size of each field as defined in the X509 standard:

- Version IoT Profile: Version 3 is the only supported version for this field. Versions of certificates other than 3 will be denied. Although there is no size increase in this field, limiting it to a single value makes it possible to condense the data by altogether removing the field.
- Serial Number: The x509 specification mandates that every certificate issued by the same CA has a distinct serial number. In our idea, the certificate is self-signed and issued by the IoT device. In this scenario, the serial number may serve as the IoT device's unique identifier. It may either be the SN or the Token given by the LRA. In this instance, a certificate's unique identification is represented by the serial number field.
- Signature: The X.509 specification's limitations are the only ones that apply to this field because no other restrictions have been introduced. A self-signed certificate that was not completed by a CA. **The algorithm is used to sign the certificate,** together with any associated parameters. Because this information is repeated in the Signature field at the end of the certificate, this field has little if any, utility. The value is Zero.
- Issuer: **X.500 name of the certificate authority (CA) that created and** signed this certificate. This field will be used for the LRA domain name or IP address, limited to a common name (CN) of the LRA name's UTF8String type.
- Validity: In this profile, dates are represented using the format YYMMDD of the ASN.1 UTC Time. Even if this format becomes outdated in 2049, it would be undesirable to lose compatibility with the DTLS Profiles for IoT, and as this is a much bigger issue, a solution may come. The time can be changed to any value if the certificate is applied to gadgets without a source of absolute time. This field could be optional if the validity is not required.
- Subject: IoT Profile does not require this field. If the subject is an IoT device, the subject field contains the EUI-64, otherwise it contains the name of the CA if the subject is a CA.
- Subject Public Key: The only way to achieve the stated design goal would be to limit the cryptographic technique to 256 bits ECC keys from the curve prime256v1 according to the X.509 specification, which states that this field holds the public key in a bit string and defines which algorithm the key is used.
- Extension: Any extension
- Issuer and subject: No for the LightCert4IoT
- Signature algorithms: Since SHA256 is secure, there is no reason to support it, and using a longer hash would be useless due to the use of a 256-bit ECC curve. The elliptic curve variant of the Digital Signature Algorithm (DSA), or ECDSA, differs from ECC and RSA in the same ways. For instance, compared to DSA, an ECDSA signature generates a smaller signature and utilizes smaller keys. The support from hardware is also a key consideration when choosing a signature method. ECC public key cryptography-compatible hardware is quite likely to enable ECDSA signatures as well. Due to the aforementioned factors, this profile's signature method is limited to ECDSA with SHA256, giving rise to the ASN.1 OID ecdsaWithSHA256
- Signature: We can also calculate the size of individual fields of the LightCert4IoT certificate based on [9]. The individual size of the lightCert4IoT is the smallest.

TABLE 4. Certificate Size from reference [1].

Field	Field Size (Bytes)		
	No Profile (X509)	IoT profile (X509)	LighCert4IoT
Overhead	8	7	7
Version	5	5	5
Serial number	18	3	3
Signature	15	12	0
Issuer	114	20	20
Validity	32	32	32or 0
Subject	168	36	36
Subject public key info	294	91	91
Issuer and subject unique ID	0	0	0
Extensions	596	31	31or 0
Signature algorithm	15	12	12
Signature	26	75	26
Total	1526	324	around 249

V. SECURITY ANALYSIS

The security of the suggested system is examined in this section. Due to the widespread use of IoT devices, new types of assaults, like DDoS and man-in-the-middle attacks that use IoT nodes as attackers, are now possible. We believe that a hacker might intercept and manipulate network communication to attack the LRA and client using a man-in-the-middle

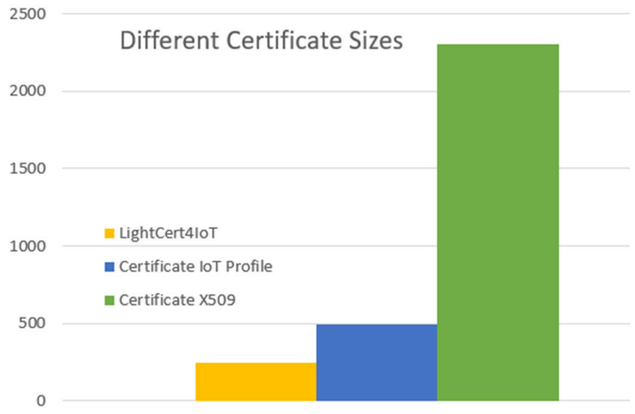


FIGURE 8. Different certificate sizes.

technique. The lightCert4IoT has the following adversaries on the security level:

1. LightCert4IoT implements Trusted Authentication of IoT devices. The authentication is based on Hardware tokens and blockchain. The globally unique HW serial number SN or a token issued by the (Electronic Notary) EN may be used to uniquely identify an IoT device. During the IoT device's registration in the LRA, the binding between the user ID and its token or SN is completed. These devices, however, might be physically replicated and hacked. In case of an IoT device is compromised, it will be identified by LRA and Blockchain.
2. To limit attacks on the network from IoT devices and vice versa, a new and enhanced firewall or Content Delivery Network (CDN) may also help to protect servers against some common malicious attacks, such as DDoS. Additionally, servers should employ fundamental preventative techniques such as a back-off following potentially risky connection attempts to limit the impact of an active attacker.
3. The most serious issue with blockchain is a sort of assault, such as a 50% attack, which occurs when an attacker gains control of more than 50% of the processing power. As a result, the blockchain network would come under the attacker's control and be vulnerable to attacks. Due to this, the attacker will fully alter the blockchain information, such as tampering with the transactions and preventing other miners from mining legitimate blocks as part of their routine operations. Despite this, since both Bitcoin and Ethereum employ the PoW algorithm to obtain agreement, it is challenging to carry out more than 50% of attacks in practice [27] whether on the Bitcoin or Ethereum blockchains. The PoW algorithm implements the idea of a simple majority of the chain protocol [27]. Additionally, other studies, including [28], [29], and [30], provide theoretical evidence of how 50% of attacks might impact the public blockchain network and make the platform unstable. A blockchain network with a proportion is assumed to be controlled by full nodes in our proposed system, making it computationally challenging for an attacker to take control of the network and launch a greater than 50% attack.
4. Scalability Challenge

The exponential development of IoT devices raises concerns about the scalability of blockchain systems for device registration and transaction handling. But we have considered in our design the following

Ethereum Layer 2: With the new release of blockchain Ethereum Sharding and Off-Chain. Execute sharding techniques and off-chain arrangements to decrease the load on the main blockchain network and create blockchain protocols and libraries optimized for IoT devices to minimize resource utilization. The Ethereum has undergone a major upgrade, the most notable changes were the deactivation of the proof of work consensus algorithm and switching to the proof of stake, the Ethereum layer 2 upgrade improves the scalability. The following are the main features of layer 2:

Sharding: is the process of splitting a database horizontally, to spread the load. In the Ethereum network, sharding by splitting up the burden of handling the large amount of data needed by rollups over the entire network. This will continue to reduce network congestion and increase transactions per second.

Rollups: Rollups bundle (or 'roll up') hundreds of transactions into a single transaction on layer 1. This distributes the L1 transaction fees across everyone in the rollup, making it cheaper for each user

Edge Computing: which is identified by LRA which offloads blockchain-related to more capable edge devices or gateways inside the IoT network to preserve assets on constrained devices

5. Resource Limitations Challenge:

IoT devices regularly have constrained computational power, memory, and bandwidth, making it challenging to run full blockchain hubs. But we built the network consisting of several LRA where IoT registration is handled by the EDGE node (LRA)

6. Private Information Handling Challenge:

IoT devices handle sensitive information, and the transparent and immutable nature of blockchain can raise information protection and privacy concerns.

1. Privacy-Enhancing Innovations: Utilize privacy-enhancing innovations like zero-knowledge proofs (ZKPs) and confidential exchanges to secure sensitive information on the blockchain.
2. Permissioned or Consortium Blockchains: Deploy permissioned or consortium blockchains for applications where information protection and limited access are fundamental.

7. Smart Contract Security Challenge:

Smart contracts utilized in blockchain frameworks can be vulnerable to bugs and security misuses, posturing risks to IoT applications.

1. Code Auditing and Testing: Thoroughly audit and test smart contracts to recognize and amend vulnerabilities before deployment.
2. Formal Confirmation: Actualize formal verification methods to guarantee the correctness and security of smart contract

8. Network Partitioning Challenge:
IoT devices work in situations with intermittent connectivity or network partitioning, making it troublesome to preserve a consistent view of the blockchain.
Organize Network Partitioning Arrangements
State Synchronization: Empower IoT devices to resynchronize with the blockchain when arranged connectivity is reestablished.

9. The analysis of communication overhead
The communication overhead (CoAP, MQTT, DTLS…) is improved as the computational overhead was decreased with LightCert4IoT. To reduce the overhead of certificate management and secure communication in IoT networks, lightweight communication protocols must be employed Constrained Application Protocol (CoAP) is a specific Internet application protocol for restricted hardware and MQTT: The publish-subscribe lightweight MQTT (Message Queues Telemetry Transport) network. It is recommended to employ lightweight cryptographic protocols like LightCert4IoT, making use of ECC allows for smaller key sizes, thus resulting in less communication overhead.

10. Side Channel attacks
This paper does not focus on the Hardware attacks including side channel attacks and physical attacks on IoT. Side-channel attacks are attempts to uncover secret information based on physical property (e.g., power consumption or EM radiation) of a cryptosystem, rather than exploiting the theoretical weaknesses in the implemented cryptographic algorithm. Timing attacks and power-analysis attacks are examples of side-channel attacks.

VI. CONCLUSION

This paper focused on the performance analysis of the LighCert4IoT by measuring the processing power consumption and memory size of the certificate in IoT devices which are usually classified as constrained devices. Lightcert4IoT was implemented using a different model than the standard PKI/CA. Our target is the client-side certificate, which could materialize by an IoT device, not the server-side. LightCert4IoT is a new method for issuing certificates for IoT devices called LightCert4IoTs, bypassing the need for a trusted PKI/CA. LightCert4IoT enables end users to generate self-signed certificates validated by local registration authorities (LRAs) or edge nodes and stored in the Ethereum blockchain. This approach overcomes the challenges associated with certificate client-side solutions, especially in broader IoT infrastructure.

The performance evaluation results of LightCert4IoT were achieved by utilizing the Cooja-Contiki network simulation. The results show that the energy consumption due to computation of the LightCert4IoT certificate uses less power when compared to the conventional X509 certificate, and the memory size needed is around 200 bytes far less than the X509 standard and IoT profile. In conclusion, the LightCert4IoT meets the requirements of major constraints IoT devices and could be standardized to be used in secure IoT protocols like CoAP and adopted as the authentication method for 4G and 5G mobile networks.

In our future work, we will investigate the adoption of the LightCert4IoT based on Blockchain as a general method for issuing certificates for IoT devices and used in constrained protocols like CoAP over DTLS and others. A new method in DTLS Handshake for IoT Authentication and device public key verification with the Applications server will be also designed, along with our effort to propose this solution at the standardization body.

REFERENCES

[1] M. Liyanage, A. Braeken, P. Kumar, and M. Ylianttila, *IoT Security: Advances in Authentication*. Hoboken, NJ, USA: Wiley, 2020.

[2] L. Duan, Y. Li, and L. Liao, "Flexible certificate revocation list for efficient authentication in IoT," in *Proc. 8th Int. Conf. Internet Things*, Oct. 2018, pp. 1–8.

[3] O. Cohin and P. Sondi, "Internet of Things for smart factory," in *Proc. IEEE COMSOC MMTC E-Lett.*, vol. 10, no. 5, Sep. 2015, p. 21.

[4] M. Soori, B. Arezoo, and R. Dastres, "Internet of Things for smart factories in Industry 4.0, a review," *Internet Things Cyber-Physical Syst.*, vol. 3, pp. 192–204, Jan. 2023, doi: 10.1016/j.iotcps.2023.04.006.

[5] L. Meddahi, A. Meddahi, P. Sondi, and F. Zhou, "Leveraging blockchain for a robust and scalable device identification in LoRaWAN," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Doha, Qatar, Oct. 2023, pp. 1–6, doi: 10.1109/isncc58260.2023.10323962.

[6] C. Zidi, P. Sondi, N. Mitton, M. Wahl, and A. Meddahi, "Review and perspectives on the audit of vehicle-to-everything communications," *IEEE Access*, vol. 11, pp. 81623–81645, 2023, doi: 10.1109/ACCESS.2023.3301182.

[7] W. Ejaz, A. Anpalagan, M. A. Imran, M. Jo, M. Naeem, S. B. Qaisar, and W. Wang, "Internet of Things (IoT) in 5G wireless communications," *IEEE Access*, vol. 4, pp. 10310–10314, 2016.

[8] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, document RFC 5280, 2008.

[9] F. Forsby, M. Furuhed, P. Papadimitriou, and S. Raza, "Lightweight X.509 digital certificates for the Internet of Things," in *Interoperability, Safety, and Security in IoT: Third International Conference, InterIoT 2017, and Fourth International Conference, SaSeIot 2017, Valencia, Spain, November 6–7, 2017, Proceedings*. Springer, 2017, pp. 123–133.

[10] J. Höglund, S. Lindemer, M. Furuhed, and S. Raza, "PKI4IoT: Towards public key infrastructure for the Internet of Things," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101658, doi: 10.1016/j.cose.2019.101658.

[11] A. Garba, D. Khoury, P. Balian, S. Haddad, J. Sayah, Z. Chen, Z. Guan, H. Hamdan, J. Charafeddine, and K. Al-Mutib, "LightCert4IoTs: Blockchain-based lightweight certificates authentication for IoT applications," *IEEE Access*, vol. 11, pp. 28370–28383, 2023.

[12] Y. Zhang and Z. Sun, "An IoT security framework based on the blockchain and elliptic curve cryptography," *Future Generat. Comput. Syst.*, vol. 96, pp. 518–527, Nov. 2019.

[13] J. Won, A. Singla, E. Bertino, and G. Bollella, "Decentralized public key infrastructure for Internet-of-Things," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2018, pp. 907–913.

[14] M. Hoogland, "A distributed public key infrastructure for the IoT," M.S. thesis, Fac. Elect. Eng., Math. Comput. Sci. (EEMCS), Delft Univ. Technol., Delft, The Netherlands, Jun. 2018.

[15] A. Singla and E. Bertino, "Blockchain-based PKI solutions for IoT," in *Proc. IEEE 4th Int. Conf. Collaboration Internet Comput. (CIC)*, Oct. 2018, pp. 9–15.

[16] S. Magnusson, "Evaluation of decentralized alternatives to PKI for IoT devices: A literature study and proof of concept implementation to explore the viability of replacing PKI with decentralized alternatives," M.S. thesis, Dept. Electron., KTH Inf. Commun. Technol., TRITA-EECS-EX-2018:13, 2018.

[17] M. Al-Bassam, "SCPKI: A smart contract-based PKI and identity system," in *Proc. ACM Workshop Blockchain, Cryptocurrencies Contracts*, Apr. 2017, pp. 35–40.

[18] E. F. Kfoury, D. Khoury, A. AlSabeh, J. Gomez, J. Crichigno, and E. Bou-Harb, "A blockchain-based method for decentralizing the ACME protocol to enhance trust in PKI," in *Proc. 43rd Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2020, pp. 461–465.

[19] A. Garba, Q. Hu, Z. Chen, and M. R. Asghar, "BB-PKI: Blockchain-based public key infrastructure certificate management," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Commun., IEEE 18th Int. Conf. Smart City, IEEE 6th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2020, pp. 824–829.

[20] A. Garba, Z. Chen, Z. Guan, and G. Srivastava, "LightLedger: A novel blockchain-based domain certificate authentication and validation scheme," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1698–1710, Apr. 2021.

[21] Z. Guan, A. Garba, A. Li, Z. Chen, and N. Kaaniche, "AuthLedger: A novel blockchain-based domain name authentication scheme," in *Proc. 5th Int. Conf. Inf. Syst. Secur. Privacy*, Prague, Czech Republic, 2019, pp. 345–352, doi: 10.5220/0007366803450352.

[22] A. Velinov and A. Mileva, "Running and testing applications for Contiki OS using Cooja simulator," in *Proc. Int. Conf. Inf. Technol. Develop. Educ. (ITRO)*, 2016, pp. 279–285.

[23] L. Krzywiecki, P. Kubiak, M. Kutyłowski, M. Tabor, and D. Wachnik, "Lightweight certificates—Towards a practical model for PKI," in *Business Information Systems: 15th International Conference, BIS 2012, Vilnius, Lithuania, May 21–23, 2012, Proceedings*. Springer, 2012, pp. 269–307.

[24] H. Tschofenig and T. Fossati, *Transport Layer Security (TLS)/Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things*, document RFC 7925, 2016.

[25] P. Hallam-Baker, *X. 509v3 Transport Layer Security (TLS) Feature Extension*, document RFC 7633, 2015.

[26] Accessed: Sep. 27, 2016. [Online]. Available: https://anrg.usc.edu/contiki/index.php/Collect_View#Using_Cooja_Simulator

[27] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on Bitcoin's peer-to-peer network," in *Proc. 24th USENIX Secur. Symp.*, 2015, pp. 129–144.

[28] Y. Marcus, E. Heilman, and S. Goldberg, "Low-resource eclipse attacks on Ethereum's peer-to-peer network," *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 236, Jan. 2018.

[29] W. Jiang, H. Li, G. Xu, M. Wen, G. Dong, and X. Lin, "PTAS: Privacy-preserving thin-client authentication scheme in blockchain-based PKI," *Future Gener. Comput. Syst.*, vol. 96, pp. 185–195, Jul. 2019, doi: 10.1016/j.future.2019.01.026.

[30] K. Wüst and A. Gervais, "Ethereum eclipse attacks," Dept. Comput. Sci., ETH Zürich, Zürich, Switzerland, Tech. Rep., 2016, doi: 10.3929/ethz-a-010724205.

[31] V. Buterin, "Light clients and proof of stake," Tech. Rep., 2015. [Online]. Available: https://blog.ethereum.org/2015/01/10/light-clients-proof-stake

[32] F. Albalas, M. Al-Soud, O. Almomani, and A. Almomani, "Security-aware CoAP application layer protocol for the Internet of Things using elliptic-curve cryptography," *Power*, vol. 1333, p. 151, Apr. 2018.

[33] R. Qazi, K. N. Qureshi, F. Bashir, N. U. Islam, S. Iqbal, and A. Arshad, "Security protocol using elliptic curve cryptography algorithm for wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 547–566, Apr. 2020, doi: 10.1007/s12652-020-02020-z.

[34] T. D. P. Bai, K. M. Raj, and S. A. Rabara, "Elliptic curve cryptography based security framework for Internet of Things (IoT) enabled smart card," in *Proc. World Congr. Comput. Commun. Technol. (WCCCT)*, Feb. 2017, pp. 43–46.

[35] O. P. Piñol, S. Raza, J. Eriksson, and T. Voigt, "BSD-based elliptic curve cryptography for the open Internet of Things," in *Proc. 7th Int. Conf. New Technol., Mobility Secur. (NTMS)*, Jul. 2015, pp. 1–5.

[36] M. Vucinic, B. Tourancheau, F. Rousseau, A. Duda, L. Damon, and R. Guizzetti, "OSCAR: Object security architecture for the Internet of Things Grenoble Alps University," in *Proc. 15th IEEE Int. Symp. World Wireless, Mobile Multimedia Netw. (IEEE WoWMoM)*, 2014, doi: 10.1109/WoWMoM.2014.6918975.

[37] S. El-Haddad, M. G. Genet, and B. El-Hassan, "Mobile wireless sensor networks using MDSAP, model for a hospital application," in *Proc. 4th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Dalian, China, Oct. 2008, pp. 1–6, doi: 10.1109/wicom.2008.835.

[38] P. Gangwani, S. Joshi, H. Upadhyay, and L. Lagos, "IoT device identity management and blockchain for security and data integrity," *Int. J. Comput. Appl.*, vol. 184, no. 42, pp. 49–55, Jan. 2023.

[39] P. Gangwani, A. Perez-Pons, S. Joshi, H. Upadhyay, and L. Lagos, *Blockchain and Its Applications in Industry 4.0*, vol. 119. Singapore: Springer, 2023.

[40] A. Padma and M. Ramaiah, "Blockchain based an efficient and secure privacy preserved framework for smart cities," *IEEE Access*, vol. 12, pp. 21985–22002, 2024.

[41] A. Padma and M. Ramaiah, "GLSBIoT: GWO-based enhancement for lightweight scalable blockchain for IoT with trust based consensus," *Future Gener. Comput. Syst.*, vol. 159, pp. 64–76, Oct. 2024.

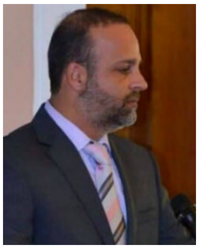
[42] A. Padma and M. Ramaiah, "Detecting security breaches on smart contracts through techniques and tools a brief review: Applications and challenges," in *Proc. Int. Conf. Inf. Manag. Eng.* Singapore: Springer, 2023, pp. 361–369.

DAVID KHOURY (Member, IEEE) received the M.E. degree in telecommunications from ESIB, in 1983. He is currently pursuing the Ph.D. with the Université du Littoral Côte d'Opale, Calais, France. He has more than 35 years of experience in the field of telecommunications and technology. He held different positions with Matra and Ericsson, mainly in France and Sweden in research and development and product and system management. He led a group to develop a generic ISDN platform in the Ericsson main exchange AXE. He was involved in early studies of the GSM and the evolution toward an IP-based network and contributed to the early studies of 3G/WCDMA, HSPA, and LTE. In 2005, he became a Technology and Business Consultant of the sales unit of the Middle East and Africa region driving new business opportunities and introducing new systems. In 2010, he has established his own start-up company (Secumobi) developing advanced military-grade secure communications systems and security solutions based on the Ethereum blockchain and backed by hardware encryption and trusted execution environments (TEE) in Stockholm. For the past eight years, he was a full-time Faculty Member with the Computer Science Department and a Research Fellow with the American University of Science and Technology (AUST), Beirut. Since 2018, he has been a Strategy Consultant for Wone, a startup located in Switzerland. He holds five U.S. patents and has published many research papers at international and local conferences. His research interests include the IoT, information security, and blockchain technology.

SAMIR HADDAD received the B.S. and M.S. degrees in computer engineering, the M.B.A. and D.E.A. degrees, and the Ph.D. degree in networking systems. He is an Assistant Professor with the Department of Computer Science and Mathematics and an IT specialist. He is undertaking research in the fields of computer and electronics networking and network devices, but also in improving the understanding, design, performance, and optimization of wireless sensor networks. In the networking field, he is currently working on multiple areas from the IoT, block-chain, security, and HCI. In the network science arena, he has focused on encoding and generic implementations.

PATRICK SONDI received the M.Sc. and Ph.D. degrees in computer science from the University of Valenciennes, in 2007 and 2010, respectively, and the Habilitation à Diriger des Recherches (H.D.R.) from the Université du Littoral Côte d'Opale, in 2020. He joined the Université du Littoral Côte d'Opale, as an Associate Professor, in 2013. He has been a Professor with IMT Nord Europe, since 2022. His research interests include protocol engineering, the quality of service, safety, security, and event-based simulation of wired and wireless networks, especially in their application in industrial and transportation systems.

GABY ABOU HAIDAR (Senior Member, IEEE) received the Ph.D. degree. He is the Coordinator of the Faculty of Engineering and the IT Manager with the American University of Science and Technology (AUST), Zahle. In addition, he is the Director of the MENA Program; and a member of the IEEE Communications Society, ASEE, and the Order of Engineers and Architects–Lebanon. He is a CISCO-certified instructor in routing and switching essentials and CISCO security. Moreover, he is a holder of the Microsoft Certified System Engineer (MCSE) and a Certified Trainer in the HCIA_AI Huawei Artificial Intelligence, HCIA–Routing and Switching, HCIA-5G, and HCIA–Datacom. He teaches communications systems, digital communications, wireless communications, embedded systems, virtual instrumentation systems, computer networking, and advanced computer networking courses. Added to that, he teaches Digital Systems Laboratory, Circuit Analysis I Laboratory, Circuit Analysis II Laboratory, Communications Systems Laboratory, Microprocessors Laboratory, Micro-Controller Laboratory, Digital Communications Systems Laboratory, Control Systems Laboratory, Electronics Laboratory, and Networking Laboratory. His research interests include in digital communications, networking, fractional calculus, embedded systems, control systems, and machine learning.



DAVID SEMAAN (Member, IEEE) received the master's degree in computer engineering from the University of Balamand, the M.B.A. degree from Notre Dame University, and the master's degree in international business from Bordeaux Management School. He is currently pursuing the Ph.D. degree in information engineering at city with the University of London. He is a full-time Faculty Member with the American University of Science and Technology and the Coordinator of the Computer Science Department. He has published several scientific research papers in international conferences and journals. His research includes a range of topics, such as artificial intelligence, gaming, and forecasting techniques. He is a member of the Lebanese Order of Engineers.

JINANE SAYAH received the B.S. and master's degrees in computer engineering, the D.E.A. degree in telecommunications, and the Ph.D. degree in networking and telecommunications. She is an Assistant Professor with the Department of Telecommunications and Networking, Faculty of Technology. In her undertaken research, she is working in the field of computer and networking and human to computer interaction (HCI), the Internet of Things (IoT), sensor networking, wireless technologies, block-chain, and bioinformatics.