LightCert4IoTs: Blockchain-Based Lightweight Certificates Authentication for IoT Applications

Abba Garba, David Khoury, Patrick Balian, Samir Haddad, Jinane Sayah, Zhong Chen, Zhi Guan, Hani Hamdan, Jinan Charafeddine, Khalid Al-Mutib

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 February 2023, accepted 10 March 2023, date of publication 20 March 2023, date of current version 24 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3259068

RESEARCH ARTICLE

LightCert4IoTs: Blockchain-Based Lightweight Certificates Authentication for IoT Applications

ABBA GARBA[1], DAVID KHOURY[2], (Member, IEEE), PATRICK BALIAN[2], SAMIR HADDAD[3], JINANE SAYAH[4] ZHONG CHEN[1], ZHI GUAN[1], HANI HAMDAN[5], JINAN CHARAFEDDINE[6], AND KHALID AL-MUTIB[7]

[1]School of ECCS, Peking University, Beijing 100871, China
[2]Department of CS and ICT, American University AUST, Beirut 6452, Lebanon
[3]Department of Computer Science and Mathematics, Faculty of Arts and Sciences, University of Balamand, Koura 1177, Lebanon
[4]Department of Telecom and Networks, Issam Fares Faculty of Technology, University of Balamand, Koura 1177, Lebanon
[5]Laboratoire des Signaux et Systèmes (L2S UMR CNRS 8506), CentraleSupélec, CNRS, Université Paris-Saclay, 91190 Gif-sur-Yvette, France
[6]Laboratoire d'Ingénierie des Systèmes de Versailles (LISV EA 4048), Pôle scientifique et technologique de Vélizy, Université Paris-Saclay, 78140 Vélizy, France
[7]Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11495, Saudi Arabia

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

ABSTRACT The proliferation of the internet of things (IoT) within the emergence of five-generation (5G) networks has received a huge attention in both industrial and academic domains. A 5G network is a cornerstone of realizing the full potential of the IoT, which interconnects billions of devices wirelessly. However, wireless communication in IoT devices reveals tremendous security risks in different dimensions and precisely in the distribution of the user certificates. The existing X.509 PKI, or the proposed decentralized PKI based on blockchain solutions have lacked practicality, and continue to have security flaws, or have not yet gained widespread acceptance owing to complexity and deployment issues. We present a lightweight certificate in size (LightCert4IoTs) that is not issued by Certification Authorities (CAs) due to the cost and complexity of the assignment of a signed certificate. In LightCert4IoTs first, an end-user (i.e., mobile and IoT devices) issues a self-signed certificate and lets Local Registration Authorities (LRAs)/EDGE nodes to verify and validate the binding identity-self signed certificate of the users through the Ethereum blockchain where The Ethereum network is used as the global notary for the IoT light certificates by saving them in the blockchain immutable ledger. The LightCert4IoTs leverages the advantages of blockchain technology and smart contracts to address the existing challenges of PKI certificates in IoT devices, which neatly achieve certificate issuance, update, and revocation more securely and efficiently. Finally, the LightCert4IoTs experimental results show that LightCert, as compared to relevant solutions/baselines, achieves reasonable overheads and is suitable for use in low- constrained IoT devices where the memory and processor power are optimized.

INDEX TERMS IoTs, certificate, blockchain, PKI, authentication, 5G.

I. INTRODUCTION

For the past decades, the communications mean of the various IoT devices were through wireless means, which is subject to vulnerabilities and security threats [1], [2]. 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 a [3]. The device authentication could rely on a Certificate Authority (CA) by assigning a public certificate to an IoT device [4]. It is the similar method applied for the network servers using certificate-based authentication defined by X.509 PKI standard. However, the existing certificate-based PKIs is not suitable for the IoT devices in general due to the complexity and the cost of the assignment of a signed certificate adding to that the memory and processor power requirements in case of the IoT constrained devices.

To address these problems, various proposals have been considered like the so-called Concise Binary Object Representation (CBOR) encoding to suitably design lightweight

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

X.509 profiles for IoT-constrained devices [5], [6]. Another proposal is using the Elliptic Curve Qu-Van (ECQV) to provide lightweight implicit certificate solutions for IoTs devices [6], [7], [8], [9]. Although these implicit certificates require a small size and memory footprint of the certificates as compared to other proposals such as X.509 certificates, [6]. However, few weak points have been identified with the ECQV in terms of security and efficiency related to the certificate issuance protocol [6], [9]. Moreover, IoT devices can be more easily attacked and hijacked as they are deployed in open environment. On another hand the certificate revocation list (CRL) and Online Certificate Status Protocol (OCSP) efficiently issued by CA needs a considerable amount of memory space, and it is not easy to manage in case of various CAs are involved [2], [10], [11], [12], [13].

By observing the comparability between the existing TLS/SSL X.509 PKI certificate proposal [4], [6], [14] and other proposals to address IoTs constraint devices authentication [6], [7], [8], [9] we leverage the advantages of Blockchain technology to design a lightweight certificate in size LightCert4IoTs which is not issued by conventional PKI/CA system due to energy consumption, memory footprints, and deployment complexity. The idea is based on the self-certificate principle [15], [17] that promotes the use of end-user certificates. The overall goal of LightCert4IoTs is to provide certificates to end-users by addressing the well-known bottlenecks of X.509 PKIs certificates used on low-constrained devices (e.g., high energy consumption, memory footprints, certificate cost, storage cost, and deployment complexity).

We present LightCert4IoTs, a blockchain-based digital certificate system for IoT devices. 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).

The proposed solution is based on the paper referenced: [18] Local PKI: An Interoperable and IoT Friendly PKI where the LRA were introduced and on the paper of Filip Forsby ''Digital Certificates for the Internet of Things'' document [19] where a smaller size IoT certificate was suggested. The solution is based on the previous work and published papers by the author of this paper referenced in [20], [21], and [22] where a blockchain domain control verification method was introduced. The proposed solution in this paper is an extension of the previous proposals described in the cited papers for the constrained device certificate called lightCert4IoT.

The LRA could be a local configuration or registration server for IoT devices or mobile devices for example. It could represent an EDGE node in 5G networks, LRA could be a software module inside the EDGE node to handle the registration, authentication, and verification of the IoT devices.

This work presents an alternative to current PKI infrastructure for IoT devices using lightweight certificates and blockchain technology. The new lightweight certificate Lightcert4IoT follow the X509 standard where some further extraneous information related to the CA were taken off based on the paper [6]. The use of blockchain has eliminated the need of trusted authorities CA. The LRA takes on the role of the local registration and configuration and verification server of the IoT devices. The LRA is not a CA, it could be an EDGE node; it is part of the system used for configuration of the IoT devices.

The main novel part of the proposed solution that the blockchain is used as the global notary for the IoT light certificates through saving and storing them in the immutable ledger of Blockchain technology.

It should be noted that the implementation is related to certificate less authentication without using the PKI/CA certificate assignment. The solution does not propose any new cryptography algorithm but applying the existing cryptographic algorithm. The advantages of the certificate less as it is mentioned in the paper is the simplification and the cost of assigning a certificate in general. The Ethereum blockchain is the corner stone of this solution because it stores and validates the certificate. The solution is independent of the cryptographic algorithm and continue to be applicable when the post quantum encryption is adopted. We think that redactable blockchain is outside of our system design it is related to an improvement and upgrade of the blockchain.

One of the advantages of the LightCert4IoT is that it solves the complexity of the assignment of a signed certificate for the IoT devices based on the current CA/PKI method. Consequently, LightCert4IoT is smaller in size since most of the information in X509 is not needed or relevant for the IoT case. On the other hand, it is a secure, automated method for IoT devices to control their public and private keys.

Thus, our contributions to the paper are summarized as follows:

- We proposed a lightweight certificate (LightCert4IoT) in size with smaller data not according to the X509 structure and not issued by PKI/ Certification Authorities (CAs) infrastructure.
- LightCert4IoT is self-signed certificates leverages the advantages of blockchain technology to verify and approve the certificate.
- The LightCert4IoT is an optimized solution for the IoT constrained devices where the memory and processor power are limited.
- The Ethereum network is used as the global notary for the IoT light certificates by saving them in the blockchain immutable ledger
- By discussing different relevant security threats and countermeasures, we analyzed the security implications of the proposed scheme.
- The security proof, instantiation, and evaluation indicated that our solutions are solid and extensible for IoTs applications.
- The prototype of LightCert4IoTs is implemented, and we have conducted a performance evaluation analysis to substantiate the proposed system using the Ethereum solidity smart contract.

The remainder of this paper is organized as follows. Section II provides background information for the paper, including existing challenges of PKI/CA in the context of IoT application, and a general overview of blockchain technology. We give a detailed explanation of the proposed system, which comprises entities and its functionalities of the proposed system in III, we give an informal discussion of the security analysis in IV, implementation and evaluation V, finally conclusion and future work in section VII.

II. BACKGROUD AND PRELIMINARIES

A. IoT OVERVIEW AND APPLICATIONS

IoT is a general term that refers to the billions of physical objects or ''things'' connected to the Internet, all collecting and exchanging data with other devices and systems over the Internet. IoT devices are hardware devices, such as sensors, gadgets, appliances, and other machines that collect and exchange data over the Internet. They are programmed for certain applications and can be embedded into other IoT devices.

The are many types of IoT applications based on their usage. Here are some of the most common ones:

- Consumer IoT - Eg: home appliances, voice assistance, and light fixtures.
- Commercial IoT - used in the healthcare and transport industries. Eg: smart pacemakers and monitoring systems.
- Military Things (IoMT) - used for the application of IoT technologies in the military field. Eg: surveillance robots and human-wearable biometrics for combat.
- Industrial Internet of Things (IIoT) - used with industrial applications, such as in the manufacturing and energy sectors. Eg: Digital control systems, smart agriculture, and industrial big data.
- Infrastructure IoT - used for connectivity in smart cities. Eg: infrastructure sensors and management systems

B. OVERVIEW OF LRA SOLUTION

The concept of LRA was mentioned in the paper [18] LocalPKI: An Interoperable and IoT Friendly PKI. The LRA can be considered as part of a local PKI. A public-key infrastructure (PKI) binds public keys to identities of entities. Usually, this binding is established through a process of registration and issuance of certificates by a certificate authority (CA) where the validation of the registration is performed by a registration authority. The LRA instead is performing the binding by a local authority and the issuance is left to the end user or to the local authority. The role of a third entity is then to register this binding and to provide up-to-date status information on this registration.

Three different entities were defined in LOCALPKI: the Electronic Notaries (EN), the Local Registration Authorities (LRA) and the users (or End Entities). The EN is equivalent to the root CA in a classical PKI system and manages the databases containing registered users. The LRA represents the intermediate entity between the user and the EN like a Registration Authority in a classical PKI. The LRA could be practically a registration server for IoT device close to the user. The LRA is registered by some EN, and the identity checks are performed during the recording of a new user.

A user first needs to be registered in the system. The phase starts by a new key pair generated by the user. After that the user interacts with a LRA for the registration process. the user gives his public key to the LRA. Next, the authority containing at least: the user's ID, the user public key, a Serial Number (SN), a validity period and the URL of the notary associated with the LRA. The SN has been obtained by the LRA from previous exchanges with the EN. The latter oversees the SN generation and communicates to each supervised LRA a specific range of SN that is generating the certificate.

C. EXISTING CHALLENGES OF PKI/CA IN THE CONTEXT OF IOT APPLICATIONS

PKI is a set of organized, collection of networked, hardware, software, entities, procedures, and policies that are required to generate, distribute, store, and revoke digital certificates and provide public-key encryption [10]. A PKI certificate is a digitally signed statement that binds the public key to the identity of the user issued by CAs. Certificate-based authentication uses the X.509 PKI standard [15], to provide a strong level of client authentication. X.509 certificate is the main standard format of the PKI developed by the IETF PKIX working group. This is the most commonly accepted used standard of internet protocol [4]. However, in practice, the existing certificate-based PKIs are not designed for constrained IoT environments.

Additionally, the cost of verifying the certificate, size, storage, and complexity of interactions poses tremendous challenges for resource constrained IoT devices. In particular, the conventional techniques used in the SSL/TLS PKI system for certificate revocation are conducted through certificate revocation list (CRL) and Online Certificate Status Protocol (OCSP) efficiently issued by CAs. Thus, it needs a considerable amount of memory space, and it is not easy to manage in case of various CAs are involved [2], [6]. The main aim of any PKI is to guarantee data exchange confidentiality, integrity, and authenticity together with a strong level of authentication of the entities. To build a fully functional PKI, we consider the following basic principles must be satisfied based on the following project initiatives [15], [16].

D. OVERVIEW OF IOTS WITHIN THE 5G NETWORK

1) MEC SERVER IN 5G

The 5G networks are needed for IoT to offer real-time, low-latency communications in mission-critical applications. The introduction of Multi-Access Edge Computing (MEC), called EDGE, in the 5G system have enhanced the IoT applications and services, The services are being delivered to customers by running a powerful compute capability at the edge of

the network closer to users. Mobile Edge Computing was introduced by the European Telecommunications Standards Institute (ETSI) Industry Specification Group (ISG) as a means of extending intelligence to the edge of the network along with higher processing and storage capabilities [39]. In 2017, the ETSI industry group renamed MEC. The MEC allows for the data generated to be processed locally, unlocking a wealth of potential for new and enhanced enterprise and consumer applications that require faster response times, greater resilience, enhanced privacy, and a better customer experience. The paradigm shift is predicted such that %75 or more of the data processing and analytics will run at the edge of the network where it is most efficient to create the ''Internet of Intelligent Things''. The increased speeds and reduced delays enable novel applications such as connected vehicles, large-scale IoT, video streaming, and industrial robotics. Machine Learning (ML) is leveraged within mobile edge computing to enable seamless automation of network management to reduce operational costs and enhance user experience. ML within mobile edge computing and the advances needed in automating adaptive resource allocation, mobility modeling, security, and energy efficiency for 5G networks.

2) 5G AUTHENTICATION AND SECURITY

Different types of credentials are considered in 5G:

- SIM Cards / eSIM
- Certificates CA
- Pre-Shared Key
- Username / password

New authentication framework has been standardized in 5G and in all generations of 3GPP networks using the EAP (Extensible Authentication Protocol) (Applied for IoT device with a SIM or eSIM). This method is performed during initial registration known as initial attach when a device is turned on for the first time. The successful of the authentication procedure leads to the creation of sessions keys. These keys are used to protect the session between the device and the network. The authentication method has been designed as a framework to support the extensible authentication protocol (EAP) a security protocol specified by the Internet Engineering Task Force (IETF) organization. This protocol is widely used in IT environments. The main advantage of this protocol is that it allows the use of different types of credentials besides the ones commonly used in mobile networks and typically stored in the SIM card, such as certificates, pre-shared keys, and username/password. This authentication method has the flexibility to be applied for use cases applications outside the telecom industry [40], [41].

E. BLOCKCHAIN AND SMART CONTRACT

The emergence of a decentralized global ledger called blockchain has been introduced initially in the first concept of Bitcoin digital currency [24]. Other blockchain platforms such as Ethereum followed which realized the smart contract [27]. Blockchain is a continually growing list of blocks that are linked together. Each block comprises a list of transactions that occurred at a regular time interval. Thereby arriving at the current state of the system by processing transactions in order from the first block to the latest block. The immutability of Data in blockchains is achieved based on cryptographically signed statements and information in conjunction with the distributed and decentralized consensus mechanism as well as redundant and transparent P2P data storage.

Blockchain uses the classical cryptographic mechanism or algorithm, but it is mainly an immutable ledger where transactions or data cannot be modified by a third party. The classical cryptographic algorithms are implemented in the Ethereum blockchain for security purposes (Authentication, confidentiality, and Integrity) of the transactions.

A smart Contract is an application that runs on top of an underlying blockchain that allows parties to write a contract and allow execution results between participants in the peer-to-peer network [25]. Ethereum is a decentralized platform that allows computer programs to run in a decentralized form by mutually distrustful nodes without the need for central authority [26]. In Ethereum Blockchain, constrained IoT hardware devices or applications interface with the full blockchain network using LES. Light clients use the Light Ethereum subprotocol (LES), which only downloads a subset of block headers at first and then fetches the rest from the blockchain network as needed [27].

III. RELATED STUDIES

Various proposals using different mechanisms have been presented:

PKI4T [6] was built using X.509 digital certificates together with Concise Binary Object Representation (CBOR) encoding, dubbed XIOT. The XIOT profile can be used to minimize certificate sizes without breaking X.509 compatibility. CAs can now issue standard X.509 certificates suitable for IoT devices. On limited networks, edge devices transform these to and from the XIOT format. As a result, the integrity of the original CA signature is preserved, and the certificate conversion edge device does not need to be trusted. Although the PKI4oT addresses important security challenges for low constraint devices by automatically acquiring the certificate, the proposed system itself does not protect against sophisticated denial-of-service attacks.

Raza et al. [36] have attempted to build a key management architecture for the resource constrained IoT. They propose a key server, called a '' trust anchor,'' to distribute pre-shared symmetric or raw asymmetric keys among the communicating parties. The proposed interactions with the trust anchor minimize message exchange compared to Kerberos, Radius or PKI mechanisms. They argue that PKI is not always suitable for constrained environments due to the size of standardized certificates and the lack of economic methods for developing PKI with globally trusted certificates. They built an experimentation environment based on the CoAP,

DTLS, UDP, IPv6, 6LoWPAN, and IEEE 802.15.4 protocols for performance measurements. They analyzed energy consumption, memory, and time overhead related to symmetric key management and cryptography operations.

Forsby et al. [5] proposed a lightweight certificate for resources for IoT devices. The proposed approach employed an X.509 profile for IoT that only incorporated the fields required by IoT devices while maintaining certificate security. Additionally, this proposal uses the CBOR encoding method to condense X.509 profiled data. This means that profiled and compressed X.509 certificates for IoT may be enrolled, verified, and revoked without requiring changes to the existing X.509 standard or PKI implementations. One of the essential features of this proposal is its compatibility with the X.509 standard, which means the lightweight certificate can be used in any existing PKI solution. Singla and Bertino [37] Authors deploy three different blockchain solutions as alternatives to CA-based PKI for certificate management. The authors describe how these approaches address the flaws while preserving or enhancing certificate verification performance. The proposed solution can be used in the context of low-resource IoT devices concerning computational power and storage capabilities.

Similarly, [38] another study offers a new certificateless public key cryptography (CL-PKC)-based authenticated key agreement scheme. Furthermore, the suggested technique is supported by the enhanced Canetti-Krawczyk (eCK) security model. ECQV various proposals using the so-called Elliptic Curve Qu-Van (ECQV) have been proposed to provide lightweight implicit certificate solutions for IoTs devices [7] [8] [9] [6]. Although these implicit certificates require a very small size and memory footprint of the certificates as compared to other proposals such as X.509 certificates [10]. However, several weak points have been identified with the ECQV in terms of security and efficiency related to the certificate issuance protocol [6], [9]. Moreover, when IoT devices are put in an open environment, they can be more easily attacked and hijacked.

Although various security standard mechanisms [6], [7], [16], [17], [31] have been proposed to satisfy the requirements of the IoT's systems, Thus, many vulnerabilities have been highlighted in the literature [1], [6], [20], [42]. These vulnerabilities allow a malicious entity to attack the IoT devices and threaten their security goals.

IV. PROPOSED MODEL LightCert4IoT

The solution is based on the previous work and published papers by the author referenced in [19], [20], and [21] adapted and modified for the constrained certificate called lightCert4IoT.

A. SUMMARY OF THE PROPOSED SYSTEM

We introduced the LightCert4IoTs that allow end-users and IoT constraint devices to acquire certificates easily securely and cost-effectively. LightCert4IoT is built on the X509 certificate format with reduced the size of many fields. The

TABLE 1. Description of the notations used in the paper.

SYMBOL	DEFINITION
MEC/EDGE	Multi Access Computing
UUID	Universally Unique Identifier randomly generated and assigned by the LRA server for client to create a self-signed certificate.
H	Hash function that maps the length of message
HTTPS/GET	is a technique for retrieving data from a specific URL
Message(m)	encrypted message for the key pair
PK	Public key pair of the certificate
SK	Private key pair of the certificate
EM1	Encrypted message 1
EM2	Encrypted message 1
LP_K	Local registration authority public key
IP	an IP address for the LRA certificate for secure communications with the IP address provided.
N1	is generated with the help of a Provable service, as generating random numbers in a deterministic machine (EVM)
N2	is generated with the help of a Provable service, as generating random numbers in a deterministic machine (EVM).

LightCert4IoT certificate is not issued by CAs; instead, the end-user device generates a self-signed certificate after LRAs vouch for its identification. The client generates a self-signed certificate. It is the LRA server's task to maintain a mapping between the client's identity and its corresponding token and/or Ethereum wallet address. The LRA server can be a full Ethereum node, acting as a miner, storage node, or validator in the Ethereum blockchain, or it can access the blockchain via a light client LES. The LightCert smart contract module in the Ethereum blockchain acts as a certifying authority that verifies the public key of end-users mapped to the user identity once submitted by LRA.

The LightCer4IoT certificate format contains mainly the following data:

- UUID (User Identity),
- Public Key of the constraint device,
- Expiry date,
- LRA domain name,
- User information (Optional)
- Validity

B. SYSTEM MODEL

Fig. 1 presents the system model, which illustrates entities that are involved in our proposed system and their interactions to manage the LightCert4IoTs. End-users (i.e., IoT devices) submit their credentials to LRAs which verify the users' identity. The identity verification is based on a unique Token assigned to the IoT device and provided by the IoT application during configuration, or the Token could be a unique SN (Serial Number) assigned to each device during the production and configured in the LRA. Our understanding that every HW device is identified by a unique identity number called SN. This could be verified by the EN or the LRA during the configuration of the Devices.

The end-user device generates a self-signed certificate after LRAs vouch for its identification. The LRA function could be part of the EDGE/MEC node in case of 5G. The interaction with blockchain is done through the LRA server. The LRA

server communicates directly with the blockchain and executes the lightcert4IoT storage of several constrained devices as one transaction. Another alternative that the end-user can interact directly with the blockchain through the LES, but this approach requires high processing power and memory which could not be applicable for the constrained devices.

Components of the solution:

a) End-Users: The end-user is an entity that is linked to IoTs. Initially, end-users must present themselves within the LRA to apply for the light certificate. End-user user-self sign certificate sent to LRA for vouching, after the LRA verifies the binding identity and makes sure the user's detail matches with information in the certificate subject. IoT applications e.g connected cars, Hospitals, mobile devices, etc…The IoT device could be uniquely identified by its HW serial number SN which is unique globally or by a token assigned by the (Electronic Notary) EN. The binding between the IoT device user id and its Token or SN is done during the registration of the IoT device in the LRA. The Binding could be done at the EN during the configuration of the IoT device [18]

b) Edge Node Zone/ LRAs): LRA serves as an intermediary between the end-user and the Blockchain. LRA can also serve as Edge nodes serve which are closed to the end-user and can serve as a voucher for the public key certificate. LRA can be part of the 5G blockchain network as an edge node in the 5G network EDGE node or server called multi-access edge computing (MEC). The MEC node could contain the LRA function to handle the security and authentication of the IoT devices. The end-user can submit his credential for the certificate request to the LRA. In practical terms, the LRA could be composed of mobile operator banks, IT firms, Intelligence manufacturing firms, or a component in a 5G and associated with IoT applications or an agency close to the end-user, such as the user's insurance company, his bank, the postal office, etc. Those agencies usually already can check identities. The LRA server communicates directly with the blockchain and executes the storage of several constrained devices as one transaction.

c) Blockchain Network: LightCert platform consists mainly of a smart contract running over the Ethereum blockchain network. The LightCert module is used as a decentralized key store that accepts the public keys and other data of the devices mapped to their identities. In our proposal, we use the platform to store the LightCert4IoT. The goal is to incorporate Blockchain, a trustless, decentralized network, into the domain verification process.

C. SYSTEM ARCHITECTURE

1) IOT REGISTRATION AND CONFIGURATION

An end-user-constraint IoT device in the proposed system refers to any device that has no addressable identity. the IoT

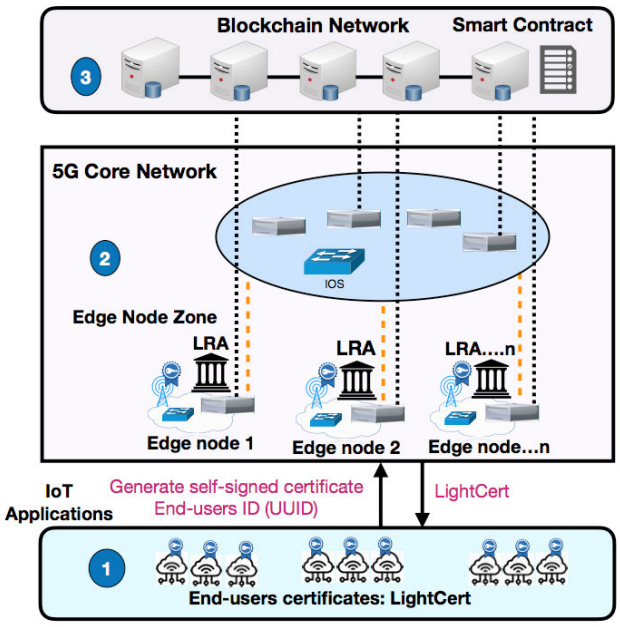


FIGURE 1. A general framework for the proposed system is LightCert4IoTs.

device starts by registering its identity including its token or SN in the LRA server. it should be noted that the identity of the device is assigned to the device during the configuration with its token in the LRA or in the en. the SN is assigned to the IoT device as a unique global identity and the token generated randomly by the LRA. the IoT device (client) then generates a self-signed certificate. The IoT device could contain a Ethereum wallet address in the alternative of accessing the Blockchain directly. The wallet address with the needed ether provided by the LRA server enables the device to access the Blockchain network. in this regard, it is the LRA server's task to maintain a mapping between the client's UUID and its corresponding token and/or wallet address. in this regard, the LRA has two choices: to store the lightcert4iot in Blockchain directly or to store it through the LRA.

The necessary configuration data for a device can be preset or acquired from the LRA server on-demand using https. see Fig. 2. it is important to note that the IoT device can be behind a network address translation (NAT) and the ip address known by the device is not ip addressable. that is why the public ip address is not an accurate indicator of the ip address of the device.

2) IOT DEVICE AUTHENTICATION

The client is authenticated after verifying its self-signed certificate against the public key mapped to its User Identity (UUID) in the Blockchain, aiming at facilitating addressing and management challenges. LRA servers running full nodes would be considered validators, making them the only entities able to validate transactions in Ethereum 2.0. The IoT service would be running a node in light sync mode, in other words, they would be light nodes.

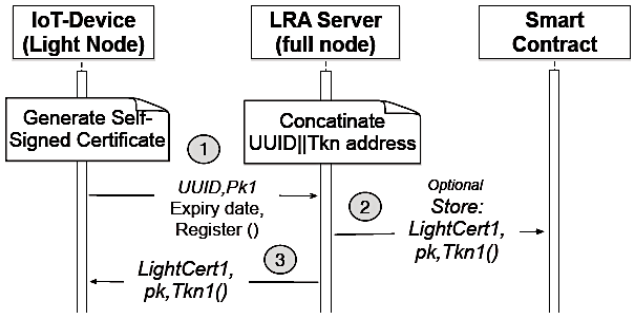


FIGURE 2. IoT device registration and configuration.

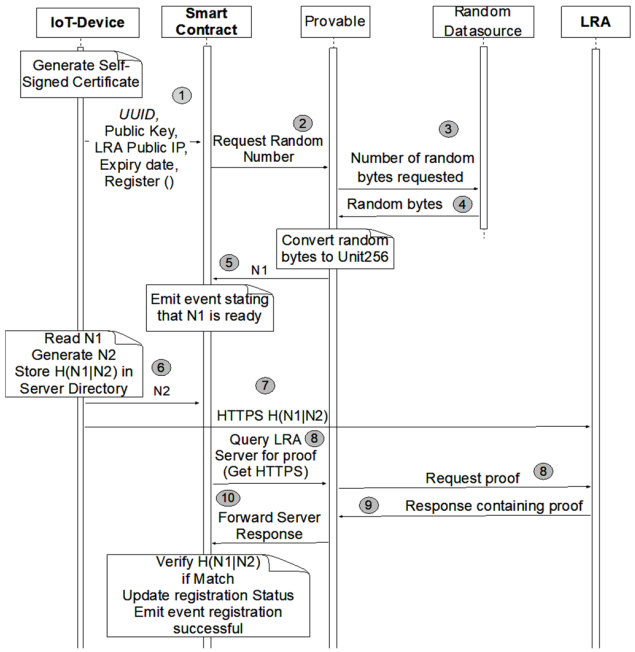


FIGURE 3. Blockchain-based lightCert4Iot client's devices control validation.

3) IOT DEVICE INTERFACE TO THE ETHEREUM BLOCKCHAIN

The download or installation of the user's LightCert client module automatically generates the public/private keys and creates an empty Ethereum wallet. 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. The LRA performs the wallet management functionality by sending Ether to the client (IoT) and approving the storage transaction in the blockchain. The LRA server module could contain an Ethereum full node (it means part of the blockchain network) to interface with the blockchain. In this regard, we provide the following alternatives: These alternatives described below were based on the following papers [19], [20], [21]. The main contribution that these alternatives were adapted for the Lightcert4IoT.

Alternative 1 - Blockchain-based lightCert4IoT clients devices Control Validation.

The IoT device generates a self-signed light certificate. The LightCert module inside the IoT device sends to the smart contract the device identity UUID, and the generated LightCert4IoT contents. Ref. [19], [20], [21]. The IoT device generates a random number, N2, and stores the hashing function H (N1, N2) in the LRA root directory (well-known/PKI-validation), where H is a hashing function. Afterward, it sends N2 to the smart contract, which in turn, initiates an HTTPS GET request to the server requesting the file hosted on the root directory. When the result arrives at the smart contract, it verifies the hash, the result's proof, and approves the server's record. Upon approving the server's identity, other devices can retrieve its public key and establish a secure session (see Fig. 3).

Alternative 2 - Blockchain-based lightCert4Iot client's devices without Provable.

Approval of the IoT public key or light certificate key through the LRA configuration without the need for provable. The client is authenticated after verifying its self-signed certificate against the public key mapped to its device identity (UUID) in the Blockchain, aiming at facilitating addressing and management challenges. LRA servers running full nodes would be considered validators, making them the only entities able to validate transactions. IoT service would be running a node in light sync- mode. In other words, they would be light nodes. The LRA server (after configuration) sends to the smart contract the list of the IoT LightCert module UUID mapped to the wallet address. These are wallets users as accepted by the system before executing the public keys or light certificate storage transactions. The benefits of this proposal include preventing any external user with an Ethereum wallet from initiating the execution of the storage of the public keys and consuming gas in the smart contract before the real user stores the application public keys. These are wallets users accept as accepted by the system before executing the public keys or light certificate storage transactions. The Wallets IoT users' addresses are already configured with the necessary ether. The benefits of this proposal include preventing any external user with an Ethereum wallet from initiating the execution of the storage of the public keys and consuming gas in the smart contract before the real user stores the application public keys. The storage of the LightCert4IoT contents in Blockchain is according to the interface defined in the LightCert platform.

4) IOT DEVICES PEER-TO-PEER COMMUNICATIONS

Communication between two devices is conducted through the following process: We assume that both devices saved their public keys or registered with the LightCer4IoT platform, as shown in Fig. 4.

1) Lightcert IoT Device 1 (light node) requests a public key pk of the device UUID of the responder Lightcert Device 2 from the blockchain smart contract.

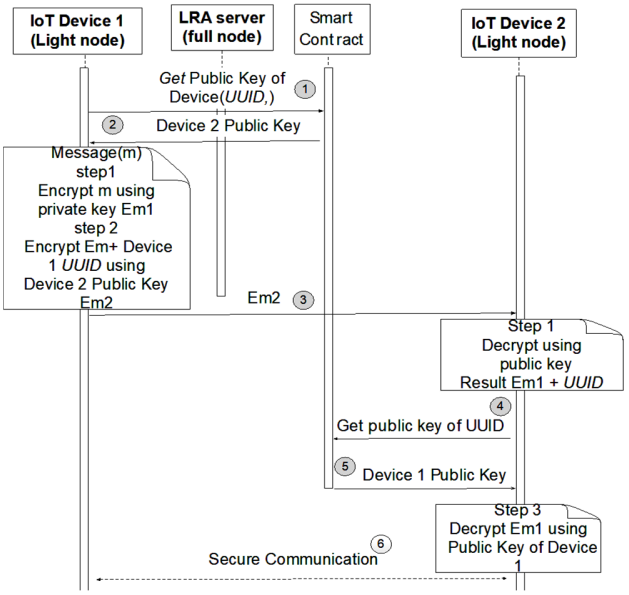


FIGURE 4. IoT devices peer to peer communication.

2) Lightcert IoT device 1 (light node) get the public key pk for the Lightcert IoT device 2. Lightcert IoT device 1 Encrypt the message m of the public key belong to the Lightcert device 2 using her privet key SK. Message(m) step1 Encrypt m using private key Em1 step 2 Encrypt Em+ Device 1 UUID using Device 2 Public Key Em2A randomly generated. This Em1 helps device 2 ensure that the correspondent is device 2, as only device 2 can decrypt the message Em1.
3) Afterward, encrypt message Em2 is sent to the Lightcert device 2
4) Device 2 decrypts the message (Em1+UUIID) using her pk
5) request pk of UUID from the smart contract (i.e., device 1 pk)
6) Device 1 encrypt Em1 using pk device 1, to establish the secure channel.

5) IoT DEVICES' SESSION ESTABLISHMENT WITH THE APPLICATION SERVERS

In this subsection, we showed how the session can be established between the IoT device and the server using lightcert4iot/lra. Initially, IoT devices connected and transmitted the public key pk and UUID to the application server. the application server obtains the LRA's IoT public key pk. Note that the LRA server communicates directly with the Blockchain and executes the storage of several constrained devices as one transaction. In this regard, retrieve IoT pk from the Blockchain network. The IoT pk key is sent to the application server for verification. The application server will verify the LRA with the one connected by the IoT device to establish a secure session. Fig.5 shows the end-user application session establishment with the application servers.

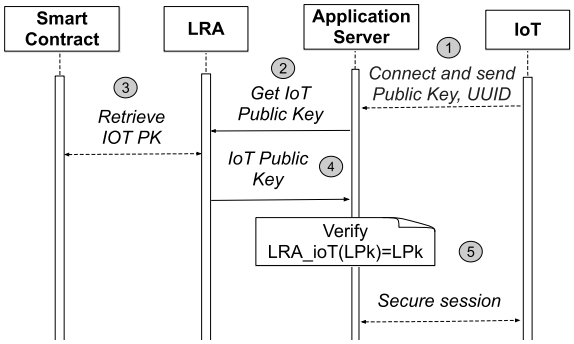


FIGURE 5. Describe IoT devices session establishment with the application servers.

6) UPDATE AND REVOCATION OF THE LIGHTCERT

When traditional x.509 certificates are issued to the domains, they are expected to be valid within their validity period. despite that, the certificates are revoked whenever they are considered untrustworthy. for example, they have a validity period that specifies their expiration date and time; Certificates that have passed their expiration date (i.e., certificates used after the expiration date) are considered invalid and are not trusted by peers. another crucial reason for revocation is if the encryption keys of the certificate have been compromised. certificate revocation in lightcert4iots is done by the IoT device by updating the record in the smart contract Ethereum Blockchain after approval from the LRA. Each record in the smart contract contains the device's wallet address. To update a record, for instance, in the revocation process, a client is restricted from initiating an update request from the wallet used initially for creating its identity without approval from the LRA.

7) SMART CONTRACT ENTITIES, EVENTS, AND METHODS

Entities

```
struct device {
    address certOwner; // host of the certifiact (its owner)
    string publicKey; // public key of the device
    string LRADomain;
    string LRAIpAddress; // ip address of the LRA
    //uint portNumber;
    uint expiryDate; // unix format epoch time
    bool isValid;
    bool registered;
  }
struct numbers {
    int randomNumber1; // N1 generated by provable defaults to -1
    int randomNumber2; // N2 generated by device defaults to -1
  }

  struct validQuery {
    uint queryType; // an integer representing the type of the query
    bool isValid; // boolean value that represents query validity
    string uuid; // ID of the device concerned
  }

  //mapping for device uuid : device
  mapping(string => device) public devices;


  //mapping for host address : numbers
  mapping(string => numbers) private LRARandomNumbers;

  //mapping for query id : validQuery
  mapping(bytes32 => validQuery) private queries;

  address private owner;
  uint256 constant MAX_INT_FROM_BYTE = 256;
  uint256 constant NUM_RANDOM_BYTES_REQUESTED = 7;
```

Log events

```
event LogNewProvableQuery(string description);
event LogRegistrationOperation(string description);
event generatedRandomNumber(string indexed uuid,bool isN1Ready);
```

Methods

```
function __callback(bytes32 _queryId,string memory _result,bytes memory _proof) public
function getRandN1(string memory _uuid) public view returns(int)
function provideSecondRandomNumber(string memory _uuid,int _secondNumber) payable public
function requestRandomNumber(string memory _uuid)  payable public
// Get the device struct corresponding to _uuid
function getDevice(string memory _uuid) public returns(device memory)
// Register a new device
function registerDevice(string memory _uuid, string memory _LRADomain,string memory _publicKey, string memory _LRAipAddress, uint _expiryDate) public
// Enables only the host address to modify the device configuration
  function modifyDeviceConfig(string memory _uuid,string memory _publicKey, string memory _newPublicKey, uint _expiryDate) public
function revokeCertificate(string memory _uuid) public
```

V. SECURITY ANALYSIS

This section contains an analysis of the proposed system's security. We begin by discussing the security analysis of the proposed system including how attackers can compromise LRA services, as well as how to secure data retrieval from the Blockchain, especially with the light client.

The proliferation of billions of IoT devices opens the door to new sorts of attacks, such as man-in-the-middle attacks and DDoS attacks, including IoT nodes as the attackers. We consider an attacker can eavesdrop and temper network traffic to perform a man-in-the-middle attack against the LRA and client. Moreover, LightCert is a significant step toward providing strong security to the Internet of Things and preventing nodes from becoming compromised. These devices, however, might be physically replicated and hacked. To limit attacks on the global Internet from IoT devices and vice versa, a new and enhanced firewall or CDN compatible with IoT protocols may still be required. In addition, to restrict the impact of an active attacker, servers should utilize fundamental prevention methods such as a back-off after possibly dangerous connection attempts.

In general, blockchain is faced with various security concerns. For example, a 50% attack is the most significant type of attack, which may happen when an attacker manages to control more than 50% of the computational power, therefore blockchain network would become under the attacker's control which may lead to possible attacks. For this reason, the attacker arbitrarily excludes and modifies the blockchain information, such as tempering the transactions in the blockchain and impeding normal mining operations of other miners from mining legit blocks. Notwithstanding that, it is difficult to conduct more than 50% attacks in practice [33] whether in bitcoin or Ethereum blockchain since both Bitcoin and Ethereum use the POW algorithm to reach consensus, the Bitcoin consensus algorithm applies the concept of a simple majority of the chain protocol [28]. Furthermore, several research works including [29], [31], and [32], provide theoretically proves of how 50% of attacks can affect the public blockchain network to make the platform unsteady. Moreover, in our proposed system we assume a blockchain network is controlled by full nodes, therefore it is computationally difficult for an attacker to control the network and launch more than 50% attack.

In the LightCert system client can validate the light certificate via block header-only but cannot verify the correctness of the incoming certificate transaction. Because the block header is small, therefore, we can only synchronize the summary of the block from the block header. This can be addressed Merkle tree, which offers evidence of data included in a large dataset without downloading and storing the entire dataset, which is an essential component of light clients. Merkle trees make considerable use of one-way hash functions, which must be collision-free. The inner node value is a one-way function of its children's values [33]. The client can validate the integrity of the leaf's value by retrieving the leaf node's corresponding pairs up to the root. Because the lite client is unable to download all the blockchain's data, it must communicate with a full node, which will supply the block header and transaction tree path.

VI. IMPLEMENTATION AND EVALUATION/WWWWW/SIMULATION RESULTS

A. SMART CONTRACT'S TRANSCATION EXECUTION TIME AND COST

Lightcert4IoT is implemented using various technology tools. Ethereum solidity smart contract for writing the contract [26], Node.js server-side scripting for managing wallet server and the LES client [33]. The Ethereum 2.0 is considered in the testing of the proposal. We have used Ethereum test network "Goerli" adopting the consensus algorithm Proof of Stack (PoS).

On September 15-2022 the Ethereum has undergone major upgrade, on that time was triggered by the blockchain passing the most notable changes was the deactivation of the proof of work consensus algorithm and switching to the proof of stake instead, that in turn required multiple changes to the internal API's.

The Ethereum layer 2 upgrade will improve the scalability. The main objective of scalability is to increase transaction speed and throughput and at the same time reduce the transaction fees. The following are the main features layer 2:

Sharding: is the process of splitting a database horizontally, to spread the load. In the Ethereum network, sharding will work together with layer 2 rollups 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 are currently the preferred layer 2 solution for scaling Ethereum. By using rollups, users can reduce gas fees by up to 100× compared to layer 1.

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.

The fees of assigning a lightCert4IoT will drop dramatically compared to the existing cost when the Ethereum layer 2 is adopted in the proposed solution.

We list the technologies used for the testing of the solution and by measuring the time needed to establish secure sessions, and the costs required to execute the transactions in the EVM.

To test the performance of our proposed system, we have conducted experiments on the following components:

- Laptop CPU: intel core i54460 LGA1150, 6M Cache, up to 3.40 GHz running Ubuntu 16.04.
- RAM: 16 GB DDR3 in dual channel configuration mode
- GPU: Nvidia GTX 970 4GB
- Motherboard: GIGABYTE G1 P85 3
- Wallet: Browser Injected web3 using the MetaMask plugin on Google Chrome.
- Solidity: Smart contract programming language.
- Web3j: Lightweight Java application for interfacing the Ethereum Blockchain.
- Environment Remix IDE (Chrome) with an injected Web3 object provided by Metamask
- My Ether Wallet: This is the wallet to hold the cryptocurrency.
- Metamask: MetaMask is a software cryptocurrency wallet used to interact with the Ethereum blockchain. It allows users to access their Ethereum wallet through a browser extension or mobile app, which can then be used to interact with decentralized applications.
- Goerli test network.
- Network speed: 2Mpbs
- Compiler version: 5.17

To validate the client-server approach, we used the HTTPS protocol. The time used for registering devices in the smart contract is related to the Ethereum block mining time. Evaluating a smart contract involves estimating the amount of GAS required for executing its transactions. The main objective of scalability is to increase transaction speed and throughput and at the same time reduce the transaction fees. The fees of assigning a lightCert4IoT will drop dramatically compared to the existing cost when the Ethereum layer 2 is adopted in the proposed solution.

Table 2 represents the estimated costs in both Ether and USD for the smart contract's task execution. As of October 2022, the Ether price ≈ $1,472 USD

As we see in table 2, we can calculate the transaction time, Gas fee in ETH, and the cost of every function of smart contract task execution.

B. LightCert4IoT MEMORY SIZE AND ENERGY CONSUMPTION

The evaluation is conducted to show the feasibility of the proposed system in terms of memory size, energy consumption. We provide a detailed evaluation of the LightCert size.

Certificate Size: LightCert4IoT

TABLE 2. Estimated cost for smart contract's task execution.

Function	Transaction time (s)	Gas (ETH)	Provable fee (ETH)	Total (ETH)	Amount (USD)
Deploy smart contract	13.53	0.01360	0	0.01360	17.816
Register device	4.30	0.000780		0.000780	1.0218
HTTP Request & Response	13.56	0.000203	0.004035	0.004238	5.55178
Modify device configuration	12.45	0.000100	0	0.000100	0.131
Get device info	5.23	0.000143	0	0.000143	0.18733
Revoke certificate	11.38	0.000068	0	0.000068	0.08908
Update certificate	13.20	0.000066	0	0.000066	0.08646
Generate Random Number	5.40	0.000147	0.004176	0.004323	5.66313

We estimate the size of the LightCert4IoT by going over the details and attribute fields of the X509 standard. The calculation is based on the Filip Forsby "Digital Certificates for the Internet of Things" document [18] and we also use the guidelines from the Datagram Transport Layer Security (DTLS) profile for the IoT [35] standard. We have adopted the same analysis as in the referenced paper by calculating the X509 fields needed and their size for the proposed lightcert4IoT. The results showed that the LightCert4IoT is smaller in memory size than the IoT profile based on X509 and contains mainly the following data: User Identity should be a Token or a serial number SN, Public Key of the constraint device, Expiry date, LRA domain name, and others.

(See section III).

Device UUID (User Identity): The device is identified by a user identity (UUID) created during configuration in the LRA server and consequently the client generates a self-signed certificate. The LRA server's task to maintain a mapping between the client's SN or Token to its corresponding UUID and or Ethereum wallet address.

Public key: constrained device contains the public key in a bit string and the cryptography algorithm is used with. For our lightCert4IoT certificate the ecdsaWithSHA256 will be adopted

Expiry Date: represents the duration of validity of the certificate such as starting date and ending date of the certificate.

LRA domain name: represent the domain name of the local registration authority.

LRA IP address: represent the IP address of the LRA most IP address is 32 bits which is equivalent to 0.004kb.

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

- Version IoT Profile: This field will be restricted to version 3 only. Certificates with a version different than 3 will be rejected. While there is no gain in size in this field, restricting the field to one value enables compressing to be done, by omitting the field completely.
- Serial Number: according to the x509 specification, all certificates issued by the same CA must have a unique serial number. In our proposal the certificate is issued by the IoT device, it is self-signed. In this case the serial number can the identity of the IoT device which could be the serial number SN, or the Token assigned by the

TABLE 3. Certificate size.

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	12
Issuer	114	20	20
Validity	32	32	32
Subject	168	36	36
Subject public key info	294	91	91
Issuer and subject unique ID	0	0	0
Extensions	596	31	31
Signature algorithm	15	12	12
Signature	26	75	0
Total	1526	324	249

LRA. In this case the serial number field represents the unique identifier of a certificate.

- Signature: No additional restrictions are added to this field, and therefore only follow the X.509 specification restrictions. However, in this profile the signature algorithm will be restricted to one algorithm it is a self-signed certificate not done by a CA. The ecdsaWithSHA256 will be adopted.
- Issuer: This field will be restricted to only contain a common name (CN) of the UTF8String type of the LRA name
- Validity To represent a date in this profile, the ASN.1 UTC Time is used, with the format UTC Time in format YYMMDD. While this format will be obsolete after the year 2049, it would be bad to break compatibility with the DTLS Profiles for IoT and since this is a much wider problem there might be a solution later. If the certificate is used with devices with no source of absolute time, the time can be set to an arbitrary value.
- Subject: this field is not needed for IoT Profile: The subject field consists of on CN structure with either the EUI-64 if the subject is an IoT device, or the name of the CA if the subject is a CA
- Subject public Key: X.509 specification: This field contains the public key in a bit string and identifies which algorithm the key is used with the knowledge from above, the only solution following the given design goal would be to restrict the cryptographic algorithm to 256 bits ECC keys from the curve prime256v1.
- Extension: Any extension
- Issuer and subject: No
- Signature algorithms: There is no reason to support stronger hashing algorithms than SHA256 since it is assumed to be secure, and the use of a 256-bit ECC curve makes a longer hash pointless. ECDSA is the elliptic curve version of the Digital Signature Algorithm (DSA), and the differences are like the ones of ECC and RSA. For example, an ECDSA signature produces a smaller signature and uses smaller keys than DSA. An important factor when deciding the signature algorithm is also the support from hardware. Hardware solutions that support ECC public key cryptography are also very likely to support ECDSA signatures. For the reasons above, the signature algorithm in this profile is restricted to ECDSA with SHA256, and thus the ASN.1 OID ecdsaWithSHA256.
- Signature: There is no need of this field

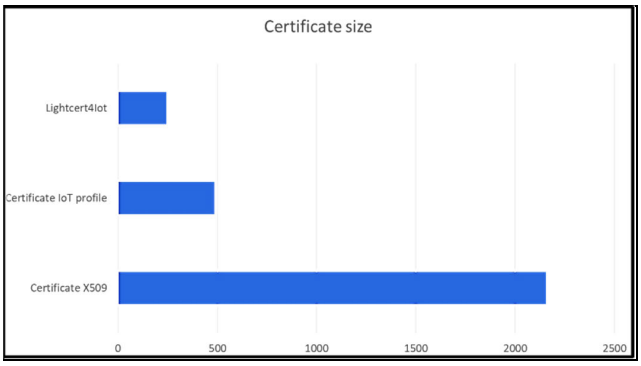


FIGURE 6. Different certificate size.

TABLE 4. Energy consumption.

Operation	Energy Consumption in mJ		
	Uncompressed	Compressed	LighCert4IoT
Receive	6.55	4.69	3.96
Transmit	23.20	15.11	9.68
Decode	0.01	1.26	1.12
Verify (SW)	305.00	306.40	303.20
Verify (HW)	16.13	16.14	15.9
Total			
Software	334.76	327.46	317.96
Hardware	45.89	37.20	30.66

We can also calculate the size of individual fields of the LightCert4IoT certificate based on [18]. The individual size of the lightCert4IoT is the smallest.

Memory Usage

There are three different kinds of certificates that have been compared: a regular X.509 certificate, a certificate conforming to the X.509 Profile for IoT (Certificates for IoT devices) according to [18].

From Fig 6, we can conclude the size of the lightCert4IoT is smallest among other profiles.

Energy Consumption

The amount of energy where each part of the LightCert4IoT certificate handling consumes has been measured. These figures are based on the similar calculations done in [18].

We extrapolate the energy consumption of the LightCert4IoT based on the calculation done for the IoT certificated based on X509. When no hardware support for ECC operations, the verification step is by far the most dominant consumer. In this case, the gain from smaller size is not as evident as when hardware cryptography is used see the table 4.

Not only the comparison between uncompressed and compressed certificates was introduced but also the comparison between verifying the certificate by software or hardware was performed. As we can see in table 4, verifying by hardware consumes less power 45.89 mJ (millijoule)than verifying by software 334.76 mJ. We added a new column for the lightCert4IoT certificate and concluded by extrapolation that LightCert4IoT consumes less energy; 317.96 mJ if we verify the cryptography by software and 30.66 mJ if we verify it by hardware.

VII. CONCLUSION AND FUTURE WORKS

In this work, we proposed an alternative to the PKI model, for assigning certificate in general and precisely to the IoT devices namely Lightcert4IoTs. The solution is implemented using the Ethereum solidity smart contract platform. It is targeting the client-side certificate running in the IoT or any other device like a mobile device, browser, etc., It is a new and simple method to authenticate a client by assigning the light certificate without the need for a trusted PKI/CA. Lightcert4IoT allows end-users to use self-sign to create a certificate and let local registration authorities (LRAs) or edge nodes to verify the binding identity of the users and validate the light certificate through the Ethereum blockchain. This method solved the complexity of certificates' assignment on the client side especially in the IoT cases when a system contains a big number of devices. Nonetheless, we do not consider the size of the certificate to pose a tremendous challenge, but the more complicated part is the assignment of this certificate for the IoT application. Consequently, LightCert4IoT is smaller in size since most of the information in X509 is not needed or relevant for the IoT case. Another advantage is that the verification and the signature of the LightCert4IoT certificate is conducted via LRA in the Blockchain network. The LRA is responsible for approving the certificate instead of an external trusted CA. On the other hand, it is a secure automated method for IoT devices to control their public and private keys.

For our future work, we will confirm our simulation results for by testing LightCert4IoT on different mobile and low-constrained IoT devices using Contiki OS or any other alternative tools suitable for our proposed system. The Contiki-cooja simulation is planned: Cooja is a sensor network simulator, the Contiki OS is a convenient operating system designed for a limited number of devices such as sensor nodes, and it is built on an event-driven kernel. The main purpose is to evaluate the LightCert4IoT in reducing the power consumption. The project consists in cooja of 4 motes: Client, LRA, LightCert4IoT, and Blockchain. We intend to investigate the strict formal security approach of the LightCert4IoT while at the same time developing LightCert4IoT instances on other types of blockchain and comparing their performances. Also, we aim to consider scalability aspects of LightCert4IoT by considering fast lightweight algorithms used in the blockchain IoT applications. The tools such as Syther, attack-tree and Tamarin will be considered to perform security analysis. We also plan to evaluate our proposed system in terms of the memory usage and energy consumption complexity of deployment and compare it with existing systems. The 5G authentication could support those use cases by including additional security enhancements and other authentication methods. Besides, LightCert4IoT can be extended and applied as a generic security method for 5G radio access based on ESP- TLS.

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] 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.

[4] 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," Tech. Rep. RFC 5280, 2008.

[5] F. Forsby, M. Furuhed, P. Papadimitratos, and S. Raza, "Lightweight X.509 digital certificates for the Internet of Things," in *Interoperability, Safety, and Security in IoT*. Berlin, Germany: Springer, 2017, pp. 123–133.

[6] 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.

[7] *SEC 4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)*, Standards Efficient Cryptogr., Jan. 2013.

[8] T. Hewa, A. Bracken, M. Ylianttila, and M. Liyanage, "Blockchain-based automated certificate revocation for 5G IoT," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.

[9] C.-S. Park, "A secure and efficient ECQV implicit certificate issuance protocol for the Internet of Things applications," *IEEE Sensors J.*, vol. 17, no. 7, pp. 2215–2223, Apr. 2017.

[10] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 internet public key infrastructure online certificate status protocol-OCSP," Tech. Rep., 2560, 1999.

[11] 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.

[12] 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.

[13] 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*, 2019, pp. 345–352, doi: 10.5220/0007366803450352.

[14] G. Schmitt, A. Mladenow, C. Strauss, and M. Schaffhauser-Linzatti, "Smart contracts and Internet of Things: A qualitative content analysis using the technology-organization-environment framework to identify key-determinants," *Proc. Comput. Sci.*, vol. 160, pp. 189–196, 2019.

[15] P. Hallam-Baker, "X.509 v3 transport layer security (TLS) feature extension," Tech. Rep., 2015.

[16] Ł. Krzywiecki, P. Kubiak, M. Kutyłowski, M. Tabor, and D. Wachnik, "Lightweight certificates—Towards a practical model for PKI," in *Proc. Int. Conf. Bus. Inf. Syst.*, in Lecture Notes in Business Information Processing, vol. 117. Berlin, Germany: Springer, 2012, doi: 10.1007/978-3-642-30359-3_26.

[17] *LocalPKI: An Interoperable and IoT Friendly PKI*. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-11039-0_11
[18] F. Forsby, M. Furuhed, P. Papadimitratos, and S. Raza, "Lightweight X.509 digital certificates for the Internet of Things," in *Proc. Int. Conf. Interoperability IoT*, Valencia, Spain, 2017, pp. 123–133, doi: 10.1007/978-3-319-93797-7_14.
[19] E. F. Kfoury, J. Gomez, J. Crichigno, E. Bou-Harb, and D. Khoury, "Decentralized distribution of PCP mappings over blockchain for end-to-end secure direct communications," *IEEE Access*, vol. 7, pp. 110159–110173, 2019.
[20] E. Kfoury, D. Khoury, A. AlSabeh, J. Gomez, and J. Crichigno, "A blockchain-based method for decentralizing the ACME protocol to enhance trust in PKI," in *Proc. 43rd Int. Conf. Telecommun. Signal Process. (TSP)*, 2020, pp. 461–465.
[21] D. Khoury, P. Balian, and E. Kfoury, "Implementation of blockchain domain control verification (B-DCV)," in *Proc. 45th Int. Conf. Telecommun. Signal Process. (TSP)*, 2022, pp. 17–22, doi: 10.1109/TSP55681.2022.9851252.
[22] S. Bouzefrane, K. Garri, and P. Thoniel, "A user-centric PKI based-protocol to manage FC^2 digital identities," *Int. J. Comput. Sci.*, vol. 8, no. 1, pp. 1694–0814, 2011.
[23] J.-G. Dumas, P. Lafourcade, F. Melemedjian, J.-B. Orfila, and P. Thoniel, "LocalPKI: A user-centric formally proven alternative to PKIX," in *Proc. 4th Int. Conf. Secur. Cryptogr.*, 2017, pp. 1–18.
[24] S. Nakamoto and A. Bitcoin. (2008). *A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[25] *Ethereum Yellow Pager: A Secure Decentralised Generalised Transaction Ledger Berlin Version Beacfbd*, Oct. 2022.
[26] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform (2013)," Tech. Rep., 2017.
[27] White Paper. (2019). *Ethereum Whitepaper*. [Online]. Available: https://ethereum.org/en/whitepaper/
[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] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *Proc. 24th USENIX Secur. Symp.*, vol. 2015, pp. 129–144.
[30] M. Mirkin, Y. Ji, J. Pang, A. Klages-Mundt, I. Eyal, and A. Juels, "BDoS: Blockchain denial-of-service," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* York, NY, USA: Association for Computing Machinery, Oct. 2020, pp. 601–619, doi: 10.1145/3372297.3417247.
[31] 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, 2019, doi: 10.1016/j.future.2019.01.026.
[32] K. Wüst and A. Gervais, "Ethereum eclipse attacks," Dept. Comput. Sci., ETH Zürich, Tech. Rep., 2016, doi: 10.3929/ethz-a-010724205.
[33] V. Buterin, "Light clients and proof of stake," Tech. Rep., 2015.
[34] Etherscan.io. *Ether Daily Price (USD) Chart*. [Online]. Available: https://etherscan.io/chart/etherprice
[35] H. Tschofenig and T. Fossati, "Transport layer security (TLS)/datagram transport layer security (DTLS) profiles for the Internet of Things," Tech. Rep., RFC 7925, 2016. [Online]. Available: https://etherscan.io/chart/etherprice
[36] S. Raza, L. Seitz, D. Sitenkov, and G. Selander, "S3K: Scalable security with symmetric keys—DTLS key establishment for the Internet of Things," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1270–1280, Jul. 2016.
[37] 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, doi: 10.1109/CIC.2018.00-45.
[38] W. Cui, R. Cheng, K. Wu, Y. Su, and Y. Lei, "A certificates authenticated key agreement scheme for the power IoT," *Energies*, vol. 14, no. 19, p. 6317, 2021, doi: 10.3390/en14196317.
[39] *Mobile Edge Computing (MEC); Technical Requirements*, ETSI GS MEC 002.
[40] J. Zhang, L. Yang, W. Cao, and Q. Wang, "Formal analysis of 5G EAP-TLS authentication protocol using proverif," *IEEE Access*, vol. 8, pp. 23674–23688, 2020, doi: 10.1109/ACCESS.2020.2969474.
[41] X. Zhang, A. Kunz, and S. Schroder, "Overview of 5G security in 3GPP," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Sep. 2017, pp. 181–186, doi: 10.1109/CSCN.2017.8088619.
[42] S. N. Swamy, D. Jadhav, and N. Kulkarni, "Security threats in the application layer in IoT applications," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Analytics Cloud) (I-SMAC)*, Palladam, India, Feb. 2017, pp. 477–480, doi: 10.1109/I-SMAC.2017.8058395.

ABBA GARBA received the B.Sc. degree (Hons.) in computing from the University of Portsmouth, U.K., the M.Sc. degree in information systems from Kampala International University, Uganda, and the M.B.A. degree from the University of Wales, U.K. He is currently pursuing the Ph.D. degree in computer software and theory with the Laboratory of Network and Information Security, Peking University, under the supervision of Prof. Zhong Chen. His current research interests include decentralized systems, network and information security, security and privacy in blockchain technologies, the Internet of Things (IoTs), and the Web3.0 ecosystem. He is also working on applying blockchain technology in the public key infrastructure (PKI) domain.

DAVID KHOURY (Member, IEEE) received the M.E. degree in telecommunications from ESIB, in 1983. He has more than 35 years of experience in telecommunications and technology. He held different positions with Matra and Ericsson, mainly in France and Sweden in research and development as well as 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 for 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. 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, for the past eight years. 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.

PATRICK BALIAN received the B.S. degree in computer science from the American University of Science and Technology (AUST) in 2021. He is a Research Assistant working with Dr. David Khoury. He is with the research and development of InMobiles, which provides services for global telecom operators. His research interests include distributed and decentralized systems and blockchain technology.

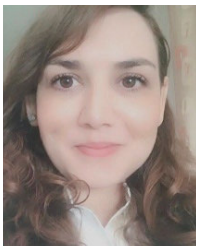
SAMIR HADDAD received the B.S. and M.S. degrees in computer engineering and the M.B.A. degree, the D.E.A. degree, and the Ph.D. degree in networking systems. He is an Assistant Professor and an IT specialist with the Department of Computer Science and Mathematics. He is undertaking research in computer 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, blockchain, security, and HCI. In the network science arena, he has focused on encoding and generic implementations.

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 from the Department of Telecommunications and Networking, Faculty of Technology. She is an Assistant Professor with the Department of Telecommunications and Networking, Faculty of Technology. In her research, she is working in the field of computer and networking, human-to-computer interaction (HCI), the Internet of Things (IoT), sensor networking, wireless technologies, blockchain, and bioinformatics.

ZHONG CHEN received the Ph.D. degree from the Computer Science and Technology Department, Peking University, in 1989. He was a Faculty Member of Peking University, where he became a Full Professor in 1995. He is a Professor with the School of Electronics Engineering and Computer Science (EECS), the Director of the MoE Key Laboratory of Network and Information Security, Software Assurance, Cloud Security, and the Director of the Financial Information Research Center, Peking University. His current research interests include blockchain, domain-specific software engineering, and network and information security.

ZHI GUAN received the Ph.D. degree in computer science from Peking University, in 2009. He has been a Faculty Member of the Ministry of Education (MoE) Key Laboratory of Network and Software Security Assurance, Peking University, and the National Engineering Research Center of Software Engineering, Peking University, since 2009. He is an Associate Professor. His current research interests include cryptography and blockchain.

HANI HAMDAN received the Engineering Diploma degree in electricity and electronics option computer science and telecommunication from the Faculté de Génie, Université Libanaise in Beirut, Lebanon, in 2000, the Master of Science degree in industrial control from the Faculté de Génie, Université Libanaise in collaboration with the Université de Technologie de Compiègne (UTC), France, in 2001, and the Ph.D. degree in systems and information technologies from UTC, in 2005. From 2002 to 2005, he was a Research Engineer of computer science with CETIM, Senlis, France. From 2005 to 2006, he was a Researcher with CNRS. From 2006 to 2008, he was an Assistant Professor with the Université Sorbonne-Paris-Nord. He was a Professor of the École Supérieure d'Électricité (Supélec), from 2008 to 2014. He is a Professor of electrical engineering and computer science with CentraleSupélec, L2S UMR CNRS 8506, Paris-Saclay University, Paris, France. He has authored over 95 peer-reviewed scientific publications. He has edited six scientific research books (ACM and IEEE). His current research interests include machine learning, signal processing, automatic control, and data analysis. He has received several excellence awards, such as the Excellence Scholarship for the Master of Science Internship, the Industrial Convention for Training through Research as Doctoral Thesis Scholarship, the Best Paper Award for his significant contribution to big data model-based clustering, and the Prize from the Downing College at the University of Cambridge for the excellent organization of the tenth International Conference on Developments in eSystems Engineering (DeSE 2017).



JINAN CHARAFEDDINE received the master's degrees in biomedical engineering and instrumentation and industrial computing from the Faculty of Engineering, Lebanese University, Beirut-Lebanon, in 2013, and the Ph.D. degree in motion science and control of mechatronic systems from Paris-Saclay University, Orsay, France, in 2021. Since November 2020, she has been a Lecturer with the Department of Mechatronics and Digital Systems of Engineering, Université Paris Saclay, and the Engineering School "Institut des Sciences et Techniques des Yvelines (ISTY), and a Researcher with the Laboratoire d'Ingénierie des Systèmes of Versailles (LISV).

KHALID AL-MUTIB received the B.Sc. and M.Sc. degrees in electrical and computer engineering from the University of Kansas, USA, in 1980 and 1985, respectively, and the Ph.D. degree in computer engineering from the University of Reading, U.K., in 1997. He is an Associate Professor of software engineering with King Saud University, where he is also a Principal Investigator of the Mobile Robot Research Group. His research interests include fuzzy logic, neural networks, artificial intelligence, and fuzzy logic control.