IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 February 2023, accepted 15 March 2023, date of publication 24 March 2023, date of current version 13 April 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3261666*

RESEARCH ARTICLE

# L-ECQV: Lightweight ECQV Implicit Certificates for Authentication in the Internet of Things

**MANISHA MALIK[1], KAMALDEEP[1], MAITREYEE DUTTA[1], AND JORGE GRANJAL[2], (Member, IEEE)**

[1]National Institute of Technical Teachers Training and Research, Chandigarh 160019, India
[2]Centre for Informatics and Systems, University of Coimbra, 3030-290 Coimbra, Portugal

Corresponding author: Manisha Malik (manisha.cse@nitttrchd.ac.in)

This work was supported by the Foundation for Science and Technology—FCT, I.P./MCTES through National Funds (PIDDAC) within the scope of CISUC Research and Development Unit under Project UIDB/00326/2020 and Project UIDP/00326/2020.

**ABSTRACT** The vast expansion of the Internet of Things (IoT) devices and related applications has bridged the gap between the physical and digital world. Unfortunately, security remains a major challenge and the lack of secure links have fueled the increased attacks on IoT devices and networks. Due to its inherent scalability, Public Key Infrastructure (PKI) is the well-known and classic approach to bring public-key certificate based security to IoT. Even though the standard X.509 explicit certificates can be viable solution, they are inefficient and too large for resource constrained IoT networks and therefore, smaller, faster and more efficient Elliptic Curve Qu Vanstone (ECQV) implicit certificates can be employed for establishing authenticated connections in IoT. Moreover, the existing certificate-based authentication proposals in standardized IoT networks have either been deployed at the transport or physical layers. Thus, these proposals fail to provide true end-to-end security to messages at the application layer in the presence of intermediate CoAP proxies. This challenging aspect is addressed in this proposal by focusing on the certificate-based authentication at the application layer to ensure true end-to-end security of messages. Additionally, IoT application layer security protocols like EDHOC lacks mechanism for authenticated distribution of public keys and thus, there is a need for lightweight authentication based cryptographic primitive for establishing secure key agreement in IoT. This paper introduces a design and implementation of a lightweight ECQV implicit certificate and use them for authenticated key exchange in EDHOC at the application layer. We also design a lightweight profile with a novel encoding mechanism for ECQV implicit certificate, called L-ECQV. To prove its viability, L-ECQV has been implemented and evaluated on Contiki operating system. Our evaluation results show that the proposed L-ECQV certificate approach reduces energy consumption by 27%, message overhead of EDHOC handshake by 52%, and shows improvements in certificate validation time. The security analysis demonstrates that proposed L-ECQV certificates for EDHOC protocol is secure against a number of attack vectors present in the IoT network. This novel combination of ECQV certificates with EDHOC key exchange leads to a secure and lightweight authenticated key agreement in IoT networks.

**INDEX TERMS** Cryptographic primitive, authentication, key agreement, Internet of Things (IoT), elliptic curve Qu Vanstone (ECQV), ephemeral Diffie-Hellman over COSE (EDHOC).

## I. INTRODUCTION

The Internet of things (IoT) materializes a network of connected devices addressable over the Internet and equipped with identification, sensing and networking capabilities to communicate to each other. Popularly recognized as one of the key technologies in the next digital revolution, IoT has led to new sensing application areas including smart homes, intelligent transportation systems, smart buildings and smart environment monitoring system, among others. By 2030,

The associate editor coordinating the review of this manuscript and approving it for publication was Fung Po Tso.

the IoT will see an explosive growth, where the total number of Internet connected smart devices is forecasted to be 125 million [1]. Factors motivating the design and adoption of IoT devices include the low costs of processors, development of lightweight wireless protocols, the digital revolution and growth in the IoT application and software. The IoT is setting up itself and expanding its domains in the physical world. However, due to the direct influence of IoT on the physical world, issues related to privacy and security are also rising and there is a need for efficient and reliable security mechanisms optimized for constrained IoT devices.

Despite such advancements in IoT applications and services, security for real world IoT deployments is still in its nascent stage and research is mostly pivoted along two domains: (i) adaptation of conventional Internet security protocols for IoT, and (ii) the development of new lightweight security protocols specifically for IoT. The central theme of both domains is that the security mechanisms must be resource-efficient i.e. use the minimum amount of resources of IoT device like energy, memory (RAM and ROM) and CPU and at the same time be robust against attacks by an adversary by protecting the privacy and confidentiality of the data exchanged between devices. Such solutions much also be scalable and be implemented using low-cost hardware and require fewer software resources, making them cost-effective for IoT applications. This allows for the development of more sophisticated and secure IoT applications and services, such as smart homes, healthcare monitoring, and industrial automation. For instance, the development of Datagram Transport Layer Security (DTLS) [2] protocol at the transport layer, lightweight versions of the Internet Protocol Security (IPSec) and Internet Key Exchange (IKE) at the network layer, and IEEE 802.15.4 security at the link layer are of principal value in this regard. Thus, using lightweight and efficient cryptographic solutions in the IoT domain is essential.

Regardless of the maturity of research in these domains, a major security challenge that continues to exist at each of these layers and one that plays an important role in the support of all the security services is the key management. Key management is a cross-layer security aspect, often associated with authentication of communicating parties involved in the negotiation of both long term and short term keys for various security mechanisms. Clearly, the classic approach to manage millions of IoT devices lies in the usage of Public Key Cryptography (PKC) whereby the private and public keys are mathematically related to each other and it is computationally infeasible to compute the private key from the public counterpart. However, PKC demands that public keys must be authenticated to ensure that key belongs to a particular device.

The Public Key Infrastructure (PKI) provides a feasible and scalable solution for the authentication of public keys. PKI is essentially the creation, storage, management, verification and revocation of certificates. These operations are managed by a trusted third party known as the Certificate Authority (CA). A certificate consists of three components: a public key, identification data, and a digital signature that binds the public key to the user's identity. Certificates, in general, are managed either implicitly or explicitly. Explicit certificates follow the X.509 standard and involves explicit verification of the CA's signature, the certificate validity and the public key. However, explicit certificates are large and incur overhead on the IoT devices for their storage, computation and communication. By contrast, the implicit certificate introduced in [3] and [4] consists of public key reconstruction data where the digital signatures are superimposed on the public key associated with an identifier. The size of an implicit certificate is extremely small when compared to X.509 certificates and thus, they are well suited to resource-constrained IoT environments.

Researchers have proposed various optimizations to PKI based key management in IoT but they are more resource demanding in terms of memory footprints and size of messages exchanged. The transmission of long messages increase the computation and communication complexity thereby increasing the power consumption of the device. On the other hand, most of these proposals have been evaluated on transport layer security protocols which fails to provide e2e security when transporting the payload through multiple proxies. The use of application layer security protocols would ensure e2e security between client and server and make IoT devices independent of the various transport protocols. Lately, the Internet Engineering Task Force (IETF), the Internet standards body, too has acknowledged this problem by standardizing a range of lightweight protocols including the Object Security for Constrained Restful Environments (OSCORE, [RFC 8613]) [5] to individual messages at the application layer and Ephemeral Diffie-Hellman Over COSE (EDHOC) [6] to ensure authenticated key exchange mechanism for establishing an OSCORE security context. However, EDHOC lacks a mechanism for authenticated distribution of public keys. Thus, there is need to develop optimized certificate-based proposals for authenticated key exchange in application layer security protocols to make them lightweight and robust at the same time.

Based on these premises, lightweight digital certificates are required which are smaller in size, need lesser memory footprint and their communication overhead is as minimal as possible. Due to the constrained nature of IoT devices, the smaller and lightweight implicit certificates have proven to be highly effective for reducing the communication overhead and power consumption. Therefore, developing implicit certificate-based solutions seem to be a promising solution to secure and authenticate IoT communications. Certificates, however need to be encoded in a particular format before transmission and thus, there is a need for optimizations in the encoding format of implicit certificates also. The integration of implicit certificates with application layer security protocols to support mutual authentication for key establishment and their analysis for certificate enrollment and management must be researched.

Our consideration in this paper is on key exchange using lightweight implicit certificates for application layer security in IoT. Based on EDHOC protocol, in this work, we design, implement and evaluate a lightweight implicit certificate for resource constrained IoT devices. We develop a lightweight ECQV profile for IoT by including only the necessary fields and removing some of the implied fields specifically for constrained IoT networks. We fixed the value of certain fields for constrained IoT networks and removed them from ECQV certificate considering them to be implicitly initialized to certain values. Additionally, we also introduce a novel compression of the ECQV profile fields using the Concise Binary Object Representation (CBOR) [7] encoding mechanism. We provide an integration of proposed profile with the EDHOC security protocol to ensure authenticated key exchange in the IoT.

In summary, the contributions of this work are as follows:

- To reduce the memory and communication overhead, we propose L-ECQV, a lightweight profile of implicit ECQV certificates for mutual authentication of communicating entities in IoT.
- A novel encoding of ECQV certificates using CBOR format to further reduce the certificate size before transmission over the constrained IoT network.
- An integration of L-ECQV with EDHOC application layer key establishment protocol that ensure secure end-to-end communication even in the presence of proxies.
- The proposed authenticated EDHOC key establishment using L-ECQV is proven to be secure against various attacks e.g. key compromise impersonation, replay and man-in-the-middle attacks.
- A comprehensive performance evaluation of the proposed approach and its comparison with the conventional EDHOC message exchange and other state-of-the art proposals.

The remainder of the paper is organized as follows: Section II presents the relevant technical background and Section III discusses the state of the art proposals in this area. Section IV presents the design of certificate profile and details the CBOR encodings of the ECQV certificate. In Section V, we give detailed description of EDHOC message handshake and propose L-ECQV for the EDHOC message exchange necessary for key establishment in IoT. Section VI discusses further security consideration of proposed L-ECQV. Section VII provides the details of implementation and experimental evaluation and Section VIII concludes the paper and discusses future work.

## II. BACKGROUND

The exponential growth in IoT devices and fast pace of digitization has led to considerable attention towards IoT security paradigm. Securing the IoT will require a multi-faceted approach, in particular, securing the network, the communications and the data. For securing the communications, security mechanisms must ensure confidentiality, integrity, authentication and non-repudiation amongst the communicating entities. To satisfy these security services, the existing paradigm of IoT security is restricted to the development of new security protocols for IoT, but these often lack mechanisms to configure cryptographic keys used by security protocols.

### A. SECURITY GOALS

Like any other network, the following security goals are essential to ensure secure key establishment in an IoT network:

1) *Mutual Authentication:* It includes two-way authentication of sensor nodes with another sensor node or border router, before revealing any critical information in the context of a given IoT application.
2) *Identity Protection:* Prevents the disclosure of the identities over the network and provides a way to communicate the identity of each participant in the protocol to its session peer [8].
3) *Perfect Forward Secrecy:* The compromise of a long-term key should not result in the compromise of future session key.
4) *Lightweightness:* Since the devices in IoT are resource constrained, overhead must be reduced during authentication and key establishment phase.
5) *Attack Resistance:* The proposed key establishment must be resistant to popular attacks like replay, impersonation, node compromise and man-in-the-middle attacks.

### B. APPLICATION LAYER SECURITY IN THE IoT

The conventional Internet relies on the end-to-end (e2e) security paradigm for sessions between datagram-based applications augmented by stream-oriented Transport Layer Security (TLS), commonly referred to as the DTLS protocol [2]. DTLS along with application layer gateways and transport support, however, break the e2e security paradigm. Presently, the IoT involves large scale deployments of resource-constrained IoT devices that are protected behind these application layer gateways or proxies. CoAP [9] protocol is a web transfer protocol that supports application layer communications in constrained IoT nodes and networks. For ensuring security, CoAP references DTLS [2] protocol for establishing a secure channel at the transport layer (below CoAP) over unreliable datagram protocols such as UDP [10]. DTLS provides hop-by-hop security by protecting entire CoAP messages. However, the proxies are unable to read encrypted CoAP messages and thus, DTLS connection needs to be terminated at the proxy. These proxies act as intermediaries between client and server for processing of CoAP messages on the DTLS channel. Therefore, a single DTLS communication cannot be established between the client and server. Hence, implementation of DTLS for secure communications over CoAP protocol suffers from following limitations: (i) the large protocol overhead of DTLS headers;

(ii) works only with UDP transport; (iii) fails to provide end-to-end security in the presence of proxies; and (iv) requires trusted proxies to ensure privacy of communication. Despite extensive research, hop-by-hop security associations are terminated at intermediate proxies and fail to provide end-to-end security associations. This issue has been addressed in [11], which examines vulnerabilities to CoAP message exchange via proxies and thereby develop security requirements to mitigate such vulnerabilities. This work proposed to implement object-based security mechanisms in addition to or in place of existing network and transport security protocols. From the standardization viewpoint, object-based security has been standardized through CBOR Object Signing and Encryption (COSE) [12] which is based on the CBOR [7] format, a data format developed for smaller code and message sizes. COSE explains how to use CBOR format for representing cryptographic keys and also deals with creation and processing of digital signatures, encryption and MAC codes using CBOR for serialization.

Based on COSE, IETF has recently standardized Object Security for Constrained RESTful Environments (OSCORE) [5], a new protocol based on object security which works in conjunction with CoAP and ensures e2e security on the application layer. In contrast to DTLS, OSCORE provides selective encryption, authentication and integrity protection on different parts of CoAP messages, thereby enabling e2e security even in the presence of intermediate (untrusted) proxies. OSCORE has the ability to work in constrained nodes and networks due to its small message size and limited memory requirements. Another advantage of OSCORE is that it makes use of features from CoAP, COSE and CBOR. In this context, Gundogan et al. [13] revisited the IoT protocol stack and presented a comparative analysis of security protocols over CoAP. In their findings, the authors indicated that OSCORE omits the overhead of maintaining headers at endpoints and thus it is an improvement over DTLS in IoT networks. However, unlike DTLS, OSCORE does not support any inherent key exchange protocol and depends solely on pre-shared keys. Hence, to establish an e2e security context, OSCORE needs a lightweight and authenticated key exchange mechanism. In this direction as a standardization effort, Ephemeral Diffie-Hellman Over COSE (EDHOC) [6] has been lately proposed for authenticated key establishment between two endpoints.

### C. EDHOC

EDHOC [6] is a compact and lightweight authenticated key establishment protocol based on the well known SIGMA (SIGn-and-MAc) protocol construct [8], particularly the SIGMA-I variant known as ''mac-then-sign''. Its main aim is to provide lightweight authenticated and session key establishment for OSCORE. Like OSCORE, it is based on CBOR encoding and COSE cryptography. In addition to providing mutual authentication, EDHOC ensures identity protection and perfect forward secrecy (PFS). EDHOC is based on the elliptic curve ephemeral Diffie-Hellman (ECDHE) and to ensure PFS it generates a new ephemeral key pair on every protocol run. Authentication can be done via PSK, raw public keys (RPK) or public key certificates, and the application must offer input on how to check that endpoints are trustworthy. On the basis of selected authentication mode, EDHOC specification identifies the COSE algorithms that are compulsory to be implemented by endpoints. These typically include ECDH, HKDF-256, AES-CCM-64-64-128 and EdDSA [14] if the authentication mode is RPK or certificates and just the first three if PSK is used for authentication.

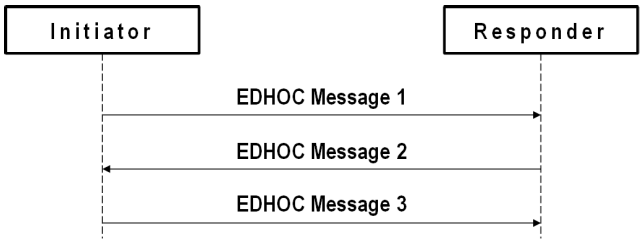


**FIGURE 1.** EDHOC message exchange.

As shown in Figure 1, EDHOC in its minimal form, requires three-message exchange for negotiating the security parameters, ensuring mutual authentication and establishing the shared secret. A discretionary fourth message is used to provide explicit key confirmation to initiator when no additional data needs to be sent from responder to initiator. The initiator starts the communication by sending message 1, which includes its ephemeral public key. On receiving message 1, responder extracts the initiator's ephemeral public key and sends its own ephemeral public key in message 2. Finally, the initiator concludes the EDHOC message exchange by sending message 3. To reduce the communication overhead, these EDHOC messages are encoded as CBOR arrays. After successful exchange of EDHOC messages, the communicating parties are able to establish the same secret key which can be used for securing subsequent application layer communications like OSCORE.

### D. KEY MANAGEMENT IN IoT

The configuration and management of cryptographic keys is known as key management and it plays a crucial role in the support of security mechanisms for Internet-integrated IoT networks. Key management involves secure bootstrapping, exchange, storage, utilization and replacement of keys. As emphasized in our previous works in [15], bootstrapping of security keys required to secure e2e communication is a cross-layer challenge and many IoT security proposals do not specify how keys will be bootstrapped. In general, the bootstrapping process involves the transition in the state of device, system or a network from being non-operational to operational. With security bootstrapping, a security association and trust is formed between devices which are unknown to each other. The works in [16] present a survey of secure

bootstrapping mechanisms for IoT emphasizing the lack of definite standards and definitions.

The configuration of keys in existing IoT security infrastructure is mostly based on Pre-Shared Keys (PSK) where devices are pre-programmed with symmetric keys. Symmetric keys avoid the higher computational cost of public key cryptography but they are not scalable and are prone to vulnerabilities. For instance, a shared secret if leaked will require updating all devices in the network. However, this is not the case with public key cryptography (PKC) which uses two types of keys: private and public key. The public key is known to all and the private key is kept secret. In addition, PKC is based on expensive mathematical operations like prime factorization, exponentiation, modulus, discrete logarithm problem, etc. Apart from being computationally resource demanding, PKC needs to ensure that the public keys are authentic and that they belong to a particular device or user. The authentication can be via some out-of-band mechanisms like in raw public keys (RPK) or with certificates such as in Public Key Infrastructure (PKI), one of the research challenges in realizing robust security in the IoT. The PKI involves the policies to manage public key encryption and to create, distribute, manage, store and revoke digital certificates. PKI ensures authentication of public keys of devices by associating them with their identities. For this purpose, a Trusted Third Party (TTP), termed as the Certificate Authority (CA) registers, issues, stores and revokes certificates of various users.

### E. PKI IN IoT

Security researchers have recognized the need for a PKI in creating trusted ecosystems for IoT devices. PKI allows devices to be provisioned with identities that determine their access to a particular application or service. PKI has become a popular choice for authentication of public keys in IoT due to its scalability, flexibility and systematic management of compromised or misused certificates even after the device has left the secure premises of manufacturer's environment. A digital certificate generally consists of three parts: the identification data, public key and the digital signature binding the public key to the identity of the user. A certificate may be administered explicitly or implicitly, as we proceed to discuss. The existence of a CA is essential for both types of certificates in PKI. We must also note that PKI deployment in IoT is expensive, particularly the resource constraints in terms of computation, memory and energy.

#### 1) X.509 EXPLICIT CERTIFICATE

Explicit or conventional certificates are based on the popular X.509 standard [17] where the trusted CA binds the public key to the identity of the user in the form of a digital certificate. As shown in Figure 2, an explicit certificate consists of identification data, public key and digital signature as three distinct elements. In the standard specification [17], these elements are encoded in binary format using the Abstract Syntax Notation One (ASN.1) Distinguished Encoding Rules (DER) [18] encoding format which is a tag, length, value encoding system. In this system, the X.509 certificate is considered to be a hierarchical structure of three fields: to-be-signed Certificate (tbsCertificate), signature algorithm identifier and the signature value. The tbsCertificate field consists of the subject and issuer name, the subject's public key, a validity period, and other associated information. The second field, the algorithm identifier includes the unique signature algorithm identifier and other parameters, if any. Lastly, the signature value is a primitive field which contains a digital signature computed upon the ASN.1 DER encoded tbsCertificate. A detailed structure of X.509 certificate with compound and primitive fields is given in [19]. A thorough explanation of ASN.1 DER encoding format is given in Section II-G.

The explicit certificate contains the CA's signature over public information of a user, to assure that the data enclosed in the certificate is indeed correct. X.509 certificates were designed back in the 90s with broad specifications having typical size of approximately 1 kB. Introduced in 1985, Elliptic Curve Cryptography (ECC) acts as an alternative to established PKC systems like DSA and RSA. ECC has gained a lot of attention mainly due to significantly smaller parameters than RSA and DSA, while with equivalent security levels. For example, to achieve a security level of 80 bits, a field size of 160 bits is sufficient for ECC whereas for RSA solutions need 1024 bits to achieve the same security level [20]. The smaller key sizes of ECC allow faster computations and reduced processing power, memory space and bandwidth. However, the problem with all types of explicit certificates is that they are large in size which makes them unsuitable for resource constrained IoT devices (for instance, class 0 and class 2 devices [21]) and networks to be used in their natural form. Additionally, a full sized PKI certificate leads to unnecessary fragmentation and re-assembly of packets in an IoT network. Thus, there is a need to either prune down the size of X.509 certificates or look for smaller and lightweight alternatives like the implicit certificates.

#### 2) IMPLICIT CERTIFICATES

Implicit certificates are another type of certificates in PKI, where all the parts of certificate, i.e. identification data, public key and digital signatures, are superimposed on one another, in such a way that the size of the certificate is equal to the size of the public key [3]. In contrast to an explicit certificate where all parts of the certificate are distinct elements, implicit certificate is smaller in size and allows the recipient to extract and verify the public key from the signature part of the certificate. Hence, there is no need for explicit validation of the signature of the CA available in the certificate. This strategy lowers the bandwidth consumption required as there is no need to transmit both the certificate and the verification key [22]. Figure 2 illustrates the main differences between explicit and implicit certificates, in what respects its

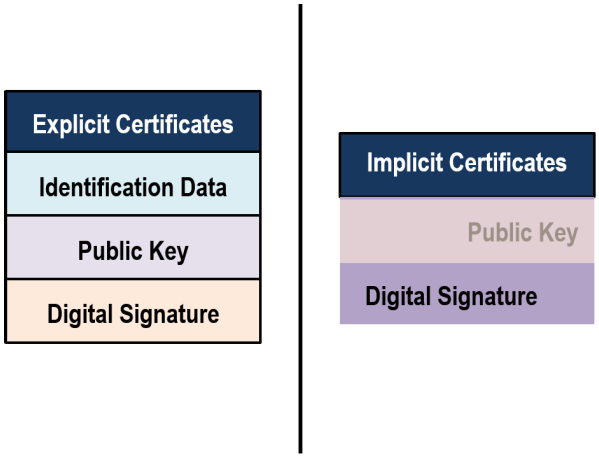


**FIGURE 2.** Explicit v/s implicit certificates [15].

**TABLE 1.** Notations used by ECQV.

| Notation | Meaning |
|---|---|
| A | Certificate Requester |
| CA | Certificate Authority |
| $Q_A$ | User A's public key |
| $Q_{CA}$ | CA's public key |
| $Cert_A$ | Implicit Certificate of user A signed by Certificate Authority |
| H | Cryptographic Hash Function (modulo n) |
| $R_A$ | Certificate request value created by A, an elliptic curve point |
| $d_A$ | User A's private key |
| $d_{CA}$ | ÃĞA's private key |
| $k_A$ | Random positive integer generated by A |
| k | Random positive integer generated by CA |
| $P_A$ | To-be signed Certificate Data Text |
| G | Elliptic Base point generator |
| r | Private Key Contribution Data |
| e | Hash of Implicit Certificate |
| Encode | Certificate Encoding Method |

main forming components. Following are the advantages of implicit certificates over explicit ones:

- **Faster processing:** The derivation of a public key in implicit certificates is faster than verification of a digital signatures in explicit certificates. That is, a user can directly extract the public key from implicit certificate and use it for desired operations of key agreement or signing [23].
- **Less Space Required:** The footprint of an implicit certificate is smaller than explicit counterpart. For a security level of 112 bits, an ECDSA-signed explicit certificate requires 680 bits in addition to the ID data while RSA requires 4096 bits in addition to ID data whereas a comparable implicit certificate is only 232 bits plus the ID data.
- **More Flexible:** Implicit certificates are flexible in terms of authority delegation, that is, for different team members with different responsibilities, implicit certificates can be issued according to their responsibilities.
- **No revocation:** Since the issuance of implicit certificate is frequent, its revocation becomes unnecessary and the CA no longer refreshes the user's certificate.

### F. ECQV BASED IMPLICIT CERTIFICATES

With an intention to provide an efficient alternative to traditional certificates, the Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV) [4] was initially proposed in 2014 by Certicom Research (under the Standards for Efficient Cryptography Group (SECG) to develop commercial standards for inter-operable and efficient cryptography based on ECC). ECQV certificates are well suited for environments which have constraints on bandwidth, computation and memory. The setup of an ECQV certificate involves three entities - a Certificate Authority (CA), a certificate requester (A) and a certificate processor (B). The requester A obtains an ECQV certificate from a CA, certifying A's identity and enabling the processor B to obtain A's public key. As a pre-requisite, the CA establishes EC domain parameters, a hash function, certificate encoding format and all parties select random number generator. We assume that the CA has already generated its public and private key pair and all parties have received authentic copies of CA's public key and other domain parameters. This standard also assumes that each entity has its own unique identifier. The detailed process of ECQV setup presented in Figure 3 is discussed below:

1) The requester A sends a request for an ECQV certificate from the CA along with its ECC public key.
2) The CA generates an EC pair and computes the elliptic curve point $P_A$. After this, the CA invokes a certificate encoding method on $P_A$ and other necessary fields and computes the hash of encoded certificate. Finally, it sends $r$, the private key reconstruction data and the implicit certificate $Cert_A$ to the requester A.
3) The requester now extracts its public key $Q_A$ which is binded to the certificate using CA's public key. The requester computes its private key from the public key output from certificate extraction. It is important to note that this step does not require any secret information and can be performed by any user having access to the CA's public key and $Cert_A$. The requester finally validates its private and public key pair ($d_A$,$Q_A$) upon receiving the ECQV certificate.

The employment of implicit certificates is indeed a viable approach for use in constrained IoT environments. In the next section, we review state of the art proposals that implement compressed explicit and implicit certificates in key agreement protocols in the context of IoT security on the application layer particularly revolving around open standardized protocols of CoAP, DTLS and OSCORE i.e the focus of our discussion will be only on standardized protocol stack in the IoT [24].

### G. ENCODING OF CERTIFICATES IN PKI

The PKI certificates (both implicit and explicit) are represented in binary format by using certain encoding formats. These encoding formats enable the transport and exchange

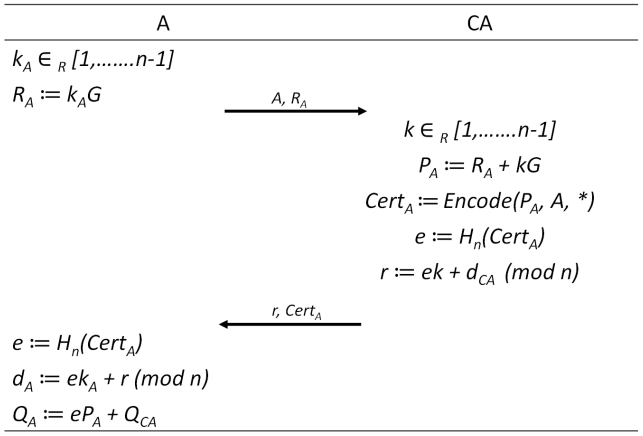


**FIGURE 3.** Description of ECQV certificate issuance process.

of certificates in binary format and provide text-based communication in human-readable format. In the IETF, one of the most popular standard is the ASN.1 [18] which has been used widely for defining the data type, its values and their combination to form various data structures. The encoding of ASN.1 data structures into binary format for transmission is done by rules defined in the DER. In ASN.1, the data is encoded using the type-length-value data fields encapsulated in a hierarchical structure.

The ASN.1 DER has been widely used for the encoding of X.509 certificates on the Internet. However, there are certain challenges in the implementation of ASN.1 DER encoding for certificates in the IoT networks. These challenges include the complex implementations of the ASN.1 encoder and decoder as well as the large size of the encoded certificate. Thus, there is a need for more compact encoding which reduces the certificate size. Therefore, CBOR [7] data format was emerged from work on the IETF. The design goals of CBOR include its small code and message size, and flexibility without requiring version negotiation. The CBOR notation is an extended version of the JavaScript Object Notation (JSON) and provides full support to data models in JSON. The description of CBOR structures is specified by the Concise Data Definition Language (CDDL) [25]. In the realm of digital certificates, CBOR and CDDL is pivotal to specify the structure of the certificates. CBOR drastically reduces the certificate size, which is known to improve performance in terms of reduced communication overhead, power consumption, latency, storage, etc. It also reduces the possibility of packet fragmentation as packets travel from source to destination. To optimize the X.509 certificates for resource constrained environments, CBOR profiles for X.509 explicit certificates have been discussed in [26] which reduces their size by 50%. On the other hand, the implicit certificates were developed deliberately to reduce bandwidth overhead and therefore, extremely flexible and lightweight encoding scheme supporting binary data like CBOR is needed for the implicit certificates.

## III. RELATED WORK

Researchers have identified the need for certificate-based authentication with PKI as essential for the largely scalable IoT [15]. However, existing solutions are severely restricted in terms of certificate sizes. The larger size of the certificate increases the message overhead in communication and memory requirement for storage in IoT devices and networks. Thus, there is need to reduce the certificate size in constrained IoT devices. The researchers stated that compression of Internet and IoT protocol headers is restrictive and that the research should now shift towards novel approaches like compression and profiling of certificates with only required fields. The researchers also emphasized on the need to replace existing X.509 explicit certificate solutions with other candidate solutions like implicit certificates, self-certified and certificateless schemes. Though explicit certificates have been extensively used for both key transport and key agreement but since we are unable to achieve forward secrecy with key transport therefore, our focus in this review will be only on certificates based key agreement protocols. The scope of this proposal is only on certificate-based authentication strategies in standardized IoT networks employing 6LoWPAN communication stack [24] and thus, only those works which are focused on certificate-based authentication for standardized IoT environments have been included in the literature review.

The initial references to compression of X.509 certificates dates back to 2010 when Pritikin et al. [27] proposed Compressed X.509 Format (CXF) for certificates and revocation lists. The proposed format uses a pre-configured dictionary and implements the DEFLATE algorithm [28]. The authors defined translations between standard and compressed certificate formats. The CXF format was later extended by Graham Edgecombe [29] who developed a new dictionary of certificates. However, the protocol bindings were designed only for TLS and IKE making it unsuitable for IoT networks. The work in [30] replaces the X.509 certificates with smaller variants like self-descriptive card verifiable (CV) certificates. Additionally. the authors propose to utilize extension fields to support authorization services.

With reference to IoT, Forsby et al. [31] designed an X.509 certificate profile for the IoT scenario by removing the unnecessary fields of the certificate and compressing the remaining using the CBOR encoding. The proposed profile maintained compatibility with existing PKI solutions based on X.509 standard. Using this profile, the authors extended their work in [32] and proposed a lightweight certificate enrollment protocol, Indraj for IoT by leveraging the conventional Enrollment over Secure Transport(EST) protocol. Similar works by Kwon et al. [33], reduced the size of typical X.509 certificates by nearly 30%. *LightCert* proposed by [34] exploits the concepts of CBOR encoding and apply 6LoWPAN header compression method to compress X.509 certificate fields. More precisely, the fields common to the whole IoT network were removed, and therefore, inherently understandable to IoT devices. The 6LoWPAN border router was delegated the task

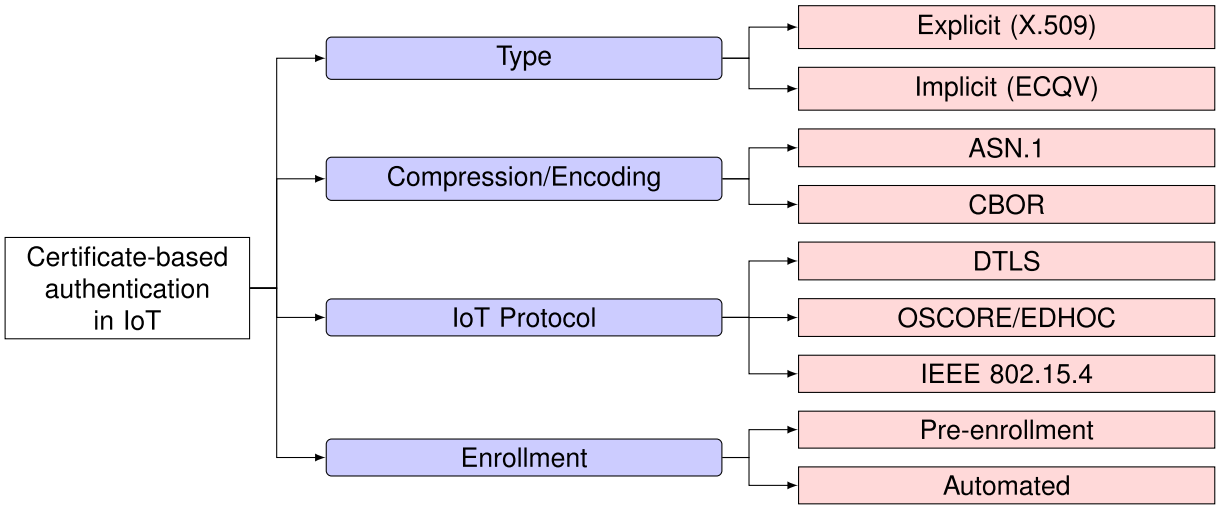


**FIGURE 4.** A taxonomy for classifying related work.

of compression and decompression when the conventional certificate enters the IoT network and vice versa. In [35], authors introduce new type of certificate, PKIoT which consisted of only the link to the original certificate which the PKIoT resource rich server later obtained for verification. However, the certificate format is not standardized and is compatible only with PKIoT architecture. Hoglund et al. [19] improved the Indraj protocol [32] and called it PKI4IoT with compressed certificate structure termed as XIOT. They integrated XIOT with EST protocol for certificate enrollment. However, the authors admitted that further investigations on CA discovery and querying are required for more efficient solutions. With an intention to reduce the computational burden of multiple DTLS handshakes, a group-oriented e2e security association between a single CoAP client (Internet host) and multiple CoAP servers (IoT devices) was proposed by [36].

Goworko and Wytrebowicz [37] proposed IoT-Crypto, a secure communication system based on a custom self-signed certificate format inspired by both X.509 and OpenPGP trust models. The authors claimed that proposed certificate is lighter than both X.509 and OpenPGP but at the same time includes all the necessary information required to build a trust relationship. Their proposed system was hinged on Internet Protocol Support Profile (IPSP) of 6LoWPAN for Bluetooth Low Energy (BLE) networks. IoT-Crypto consists of two public keys: one for encryption and other for digital signatures which require longer processing time compared to its symmetric counterparts for same level of security. Also, the proposed certificate format is incompatible with existing X.509 infrastructure and requires 8 messages of 1639 bytes reflecting the high computation and communication overheads. Similarly, Gupta and Varshney [38] proposed a novel BLE profiled certificate for authentication of BLE devices based on the Just Works model. The proposed certificate profile supplemented the existing pairing mechanism and could also be incorporated with other associative models utilized in BLE networks.

In general, so far the state of the art proposals allow to reduce the size of X.509 certificates either by profiling or compressing the certificate. Even though, X.509 certificate was extensively used as a viable solution for IoT security. its implementation on the IEEE 802.15.4 network is inefficient since it is extremely large to fit into the 127 bytes frame of IEEE 802.15.4. Alternatively, a lot of research is being carried out to inculcate lightweight implicit certificates in IoT. The purpose of implicit certificates in IoT is on ensuring unauthorized access and authentication of public keys which act as starting point for effective key agreement mechanisms. This process involves validating the certificate request, extracting the public key and finally deriving the session key to secure the communication. The validation of certificate request is performed with shared secrets either network-wise [39], [41], [42], [47], [48] or pair-wise [43]. For verification of request, authenticated identities have also been used in [40] and [49]. As already discussed, DTLS has been the de facto standard security protocol for establishing security context between CoAP endpoints. As a consequence, most of the literature reports are focused on key establishment in DTLS and therefore, initial research on implicit certificates in IoT networks was introduced with DTLS protocol in [47] and [48]. In [47], implicit certificates were used as replacement for X.509 certificates in securing CoAP based IoT networks with DTLS. The 6LBR acts as the trusted root i.e. CA and all other entities are pre-configured with authentic identities. This work implements the AES-MMO (Matyas-Meyer-Oseas) hash algorithm for reduced code size, less power and computational cost while trying to maintain strong security. However, these designs are not fully compliant with the existing DTLS standard [2] and requires careful consideration during real-time implementation. A new cipher suite for DTLS with ECQV certificate has been proposed in [40]

**TABLE 2.** Comparison of proposals on certificate based authenticated key establishment in IoT.

| Ref. | Year | Type | Compression | IoT Protocol | Enrollment |
|---|---|---|---|---|---|
| [27], [29] | 2010 | X.509 | ASN.1 DER | TLS/IKE | - |
| [30] | 2015 | X.509 | Self-descriptive DER | TLS | - |
| [31] | 2017 | X.509 | CBOR | DTLS | Pre-enrollment |
| [34] | 2019 | X.509 | Hashing, CBOR | DTLS | - |
| [35] | 2019 | X.509 | CBOR | DTLS | |
| [32] | 2019 | X.509 | ASN.1 | DTLS | Automated |
| [19] | 2020 | X.509 | CBOR | DTLS | Automated |
| [37] | 2021 | X.509 | CBOR | BLE-IPSP | - |
| [38] | 2022 | X.509 | - | BLE | Pre-enrollment |
| [39] | 2014 | ECQV | - | DTLS | - |
| [40] | 2016 | ECQV | ASN.1 | DTLS | - |
| [41] | 2015 | ECQV | - | IEEE 802.15.4 | - |
| [42] | 2017 | ECQV, X.509 | ASN.1 | CoAP | - |
| [43] | 2017 | ECQV | - | IEEE 802.15.4 | |
| [36] | 2018 | ECQV | - | DTLS | - |
| [44] | 2020 | ECQV | - | - | - |
| [45] | 2019 | ECQV,X.509 | - | | |
| [46] | 2019 | ECQV | - | WAVE-VANETs | - |

-:Missing evaluation/ Not applicable

for enrollment and authenticated key exchange. Although time efficient, the certificate registration phase in this work requires pre-shared keys for authentication.

Key establishment with implicit certificate has also been integrated with other IoT protocols like IEEE 802.15.4 and SMQV. The authors in [39] propose a new PAuthKey (Pervasive Authentication protocol and Key establishment) protocol depending on the security available in the IEEE 802.15.4 protocol at the link layer. This work is based on the pre-shared network key and 6LoWPAN identities where the size of the certificate is reduced to 44 bytes and the derivation of session key requires 12 messages, 6 each in the two phases of registration and authentication. Likewise, a fixed Diffie Hellman key management protocol for IEEE 802.15.4 is also proposed in [41]. In this work, the number of messages exchanged have been reduced to 4 assuming that implicit certificates are already pre-loaded into IoT device. The authors implement specific optimizations on scalar multiplication and modular arithmetic to further reduce the memory overhead on IoT device. In [42], this work was further enhanced for DTLS protocol by using hardware accelerators for execution of ECC based cryptographic operations. The fixed DH key exchange was improved to ephemeral one by implementing nonces per session. The implementation and evaluation on the OpenMote platform showed improvements in airtime consumption up to 86.7%. Instead of using a single shared network key for authentication of identities of certificate requester, the author in [43] modified the PAuthKey protocol in [39] and proposed to use hashes obtained using Cryptographically Generated Addresses (CGA) to obtain authenticated identities. However, this work was based on the underlying assumption that each device in the network knows the MAC address of every other device. Implicit certificates were also used for public key generation in the SMQV (Strengthened MQV) [50] protocol for IoT in [49]. The authors argue that high security of SMQV with reduced communication overhead of implicit certificate achieves lightweight and escrow-free authenticated key agreement. Further, the communication overhead is also reduced to two messages each in certificate allotment and authenticated key agreement stages. ECQV implicit certificates have been used extensively in IoT-related technologies like vehicular ad-hoc networks (VANETs) [45], [46], [51], [52], smart grid [53], [54], 5G [55], WBANs [56] etc. For instance, Farooq et al. [45] evaluated the performance of explicit and implicit certificates over IEEE 1609.2 WAVE protocol based VANETs. The authors analyzed the transmission and verification time of the certificate and assessed that implicit certificates were well suited for lightweight authentication in VANETs. Qi and Chen [53] proposed key agreement scheme for smart grid using ECQV certificates which used only two passes for mutual authentication.

Some proposals based on ECQV certificates like [44], [57], and [58] are independent of any IoT protocol. Lee and Lee [44] compare lightweight authentication and key agreement (AKA) schemes for IoT: one based on ECQV implicit certificates and other based on CL-PKC. Their proposed implementation makes communication faster and reduces the amount of transmitted data. The authors deduce that the ECQV certificate based AKA is faster and efficient than CL-PKC. Similarly, Gaba et al. [57], [58] evaluate the ECQV certificates for trust building and performed formal and informal security analysis. However, the integration of these schemes with IoT protocols is missing in their work. As previously discussed, so far the research proposals ensure hop-by-hop security which must be terminated at each proxy and we know that IoT communications are mostly performed

through proxies for enhancing efficiency and scalability [59]. However, such proposals do not provide e2e security on the application layer in the presence of intermediate entities like proxies. Instead, OSCORE protocol [5], presented in Section II, provides true e2e security even in the presence of proxies. Therefore, in order to achieve e2e security, some recent proposals based on object security have also been proposed. In [60], the authors evaluate the performance of OSCORE on real IoT hardware and compare it with CoAP and CoAP over DTLS. However, the OSCORE protocol too requires an efficient key agreement protocol. For this, the IETF formed the LAKE working group in 2019 and a deliverable of this group was the standard EDHOC protocol. The formal analysis and verification of EDHOC is given in [61], [62], and [63]. Since its development, EDHOC has also become popular having many independent implementations so far. There are sufficient evidences in the literature [11], [64], [65], [66] wherein EDHOC-based proposals have been proved as an effective and efficient approach for key establishment in IoT-constrained scenarios. In particular, the works in [65] analyzed EDHOC for certificate enrollment on the application layer. However, a detailed evaluation was missing in this work. Recently, the authors in [67] proposed LICE, an application layer enrollment protocol for IoT. In this work, the EST enrollment protocol running over DTLS was optimized by encoding messages with CBOR. The authors pointed out that such encoding of protocol messages is essential before certificate-based solutions are deployed with new IoT standards like OSCORE and EDHOC. A preliminary evaluation of EDHOC for key management in LoRAWAN networks is performed in [66]. In this work, the authors compare three alternative security solutions DTLS, IKE and EDHOC for LoRAWAN networks and conclude that EDHOC acts as a viable solution for key updating with smaller message sizes. Later in [11], the authors extend their work by designing and developing CompactEDHOC, a lightweight version of EDHOC to further reduce the network overhead. The authors compare the performance of EDHOC and CompactEDHOC over PSK and RPK based authentication modes of DTLS protocol. However, the only first implementation of embedded EDHOC on Contiki OS is performed in [68] and with support of hardware acceleration, the authors reduce the execution time of EDHOC by 37%.

### A. LIMITATIONS OF EXISTING APPROACHES

The aforementioned studies present several certificate based proposals for secure and authenticated key establishment on the application layer in IoT. Each of these studies used a different certificate type, compression method, and different communication and security protocols. Table 2 compares the relevant studies on the basis of security functionality features and common performance evaluation metrics. Although the above discussed SoA proposals enable IoT nodes to employ certificates for authenticated key exchange in IoT, most of them are based on the explicit usage of X.509 certificates. However, the direct applicability of X.509 on resource constrained IoT devices is challenging as they are heavy and incur on a lot of overhead. Existing work [42], [43], [44] has shown that the ECQV certificates are more efficient in comparison to native or compressed implementation of X.509 certificates. Though ECQV certificates have been evaluated with DTLS [36], [39], [40], these proposals implement ECQV certificates in the ASN.1 format with DER and BER encodings. However, their output is not compact, and the code can be complex for resource constrained IoT devices [21]. We also note that proposals implementing ECQV implicit certificates on application layer in IoT are few [36], [39], [40], [42] and therefore, we also consider in our analysis, some independent [57], [58] and similar ECQV implementations on the different layers of the protocol stack. To emphasize, previous work in this area of ECQV implicit certificates in IoT only consider its uncompressed implementations and focuses on matured IoT protocols like DTLS and IEEE 802.15.4.

There are alternatives to ECQV implicit certificates for key management in IoT, the so called self-certified and certificateless schemes [15]. Key management based on self-certified schemes have been implemented to authenticate and establish a key for secure communication in IoT [69]. However, the repudiability and forgery associated with self-certified schemes currently make them an infeasible approach for secure communications in IoT. In our previous work, we have discussed certificateless schemes [70] which have implemented in [71]. These schemes seem a viable approach for key management, although currently these are not interoperable with the outside Internet that does use certificates and may require the intervention of the border gateway to perform translations. This emphasized the need to keep certificate compatibility, which is expected to stay valid for several years. Apart from specific compression mechanisms, there are many other on-the-fly general purpose compression schemes for exchanging certificates in the initial handshake [72] and thus, additional compression mechanisms would incur overhead on the IoT device. However, within the domain of OSCORE, the compact representation of CBOR already exists and therefore, this encoding will not incur overhead on the IoT device.

Additionally, the absolute implementation of conventional or unprofiled Internet certificates on IoT devices is complex and requires profiling for the IoT similar to other Web protocols and technologies. For example, the CBOR protocol is lightweight version of JSON format of the web and CoAP is lighter alternative of HTTP protocol. The advantage of such profiling is the reduced code and message size, lesser overhead and simplified implementations specifically for resource constrained IoT devices. Such profiles for IoT devices also ensure interoperability with the Internet and remove the need to completely re-engineer the traditional protocols. Therefore, in this work, we build an IoT profile for ECQV implicit certificates based on the profiling of these technologies. Finally, the certificates on the Internet are issued in the ASN.1 format, which is not a compact encoding

method. For these reasons, our aim in this study is to propose an optimized and lightweight profile and encoding of ECQV implicit certificates for IoT. The proposed profile alongwith CBOR encoding reduces the certificate size and hence reduced computation, memory and computational overheads. We develop and evaluate the integration of designed certificate with OSCORE security protocol for securing CoAP communications in IoT. To the best of our knowledge, our work is the first to propose a novel compressed ECQV certificate profile for constrained RESTful IoT environments.

## IV. L-ECQV - LIGHTWEIGHT ECQV IMPLICIT CERTIFICATES FOR IoT

In this section, we address the issue of secure and lightweight certificates for IoT in two ways: Firstly, a solution for secure bootstrapping is proposed, followed by a mechanism for reducing the overhead of certificates. To make the PKI practically usable for IoT devices and networks, this section discusses how the proposed mechanism reduces the size of certificates and make them lightweight. This work proposes to use ECQV implicit certificate format with a new profile and formatting using CBOR rather than ASN.1. The proposed certificates remove unnecessary fields followed by encoding with CBOR which effectively reduces its size.

The entities involved in this process are shown in Figure 5 and described as follows:

- *Factory Server/Certificate Authority (CA):* The entity that issues factory-installed certificate for the IoT device. In our work, this certificate is our profiled and compressed ECQV certificate, L-ECQV.
- *IoT Device:* In our scenario, we consider the IoT device is configured with a pre-installed L-ECQV certificate from the CA. In the context of EDHOC, this IoT device acts as an initiator of the communication.
- *6LoWPAN Border Router (6LBR):* A comparatively resource rich device which acts as responder in the EDHOC communication. This device also acts as an intermediary between the IoT device and the outer Internet. 6LBR is also assumed to be pre-configured with an authentic L-ECQV certificate received from the CA.

### A. SECURE BOOTSTRAPPING IN MANUFACTURER DOMAIN

In our work, we consider that to enable PKI by installing factory certificates in IoT device. These certificates are obtained from the CA and are installed in the manufacturer domain when the firmware is installed on the IoT device. This process is known as pre-enrollment in the manufacturer domain. The process in which users request CAs to provide them with certificates during deployment is known as enrollment. This enrollment process can be both automated as well as manual. However, in our discussion, we assume that the certificates have been configured in the IoT device during the pre-enrollment process and the design of such a certificate enrollment and re-enrollment protocol during the deployment phase is out of scope of this work.

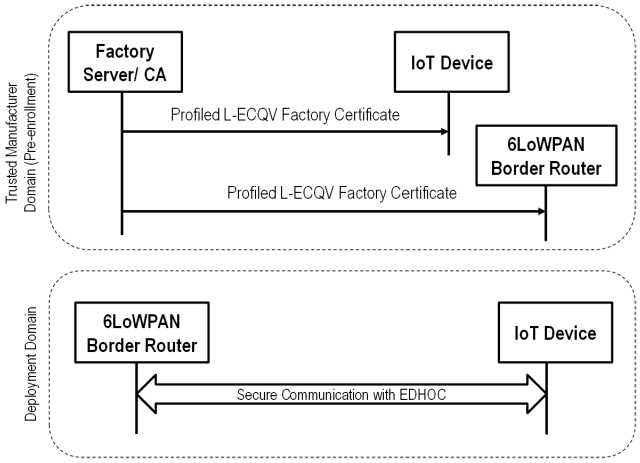


**FIGURE 5.** L-ECQV from a communication perspective; showing entities in respective domains.

### B. LIGHTWEIGHT CBOR ENCODED ECQV CERTIFICATES FOR THE IoT

The existing PKI for IoT is mainly centred around explicit certificates, particularly using the X.509 format with exceptions like X.301, X.25 and smart card certificates, CVC. These X.509 certificates are heavy and incur communication and computation overhead on the resource constrained IoT devices. Existing work show that optimized explicit and smaller implicit certificates can be used in IoT devices when naive X.509 certificates cause excessive overhead. Thus, there is a need to optimize the certificates for authenticated key establishment in IoT.

The structure of a typical ECQV certificate with field sizes and their description is presented in Table 3. The main components are: (i) Issuer (CA) and subject identifiers; (ii) information on curves and hashes for public key of subject and signature of CA; (iii) key usage and validity, and (iv) extensions (optional) [4]. The guidelines listed in [73] enumerate the usage recommendations of cipher suites and forms the basis of best practices. The representation of certificate structures as a series of bytes with standard formal notation ensures interoperability in the generation and processing of certificates. Such notations are independent of any language and subsequent physical representation of data. It provides a formalism to the abstract fields of certificates. The Abstract Syntax Notation One (ASN.1) Distinguished Encoding Rules (DER) [18] is the most popular ITU-T standard for encoding and decoding of certificates for storage and transmission on the conventional Internet. The encoding schemes denote rules linked with certificate elements. These rules often enforce standards related to PKI, such as key use, validity periods, issuer identifiers, and subject identifiers, among others. But, ASN.1 notation was not deemed light enough for IoT devices [31]. In the next subsections, we present measures to reduce the overhead of certificates: profiling, optimized encoding and deletion of tacit fields.

### 1) AN ECQV IMPLICIT CERTIFICATE PROFILE FOR THE IoT

Based on the existing challenges of heavy certificates in IoT, we propose a novel ECQV certificate profile, which can be implemented to reduce certificate overhead in IoT. We remove the fields with known values and delete the implied fields from a conventional ECQV certificate [4]. An encapsulation of a proposed LECQV profile independent of any encoding mechanism is given in Table 3.

- *Type:* This field indicates the type of certificate and defines its structure. It may assume two values; Type 1 (value=$0 \times 00$) certificate with no extensions and Type 2 certificates (value=$0 \times 01$) with extensions. Since an IoT network is already constrained and the information needed for authentic key exchange is already available in the certificate with no extensions, our profile fixes this type value to 0. The fields in such a certificate implies the fields as given in Table 3.
- *Serial Number:* A unique number assigned from the CA which allows unique identification of the certificate in the context of the issuer. It is impossible for two certificates to have same serial number even if they are issued by the same CA. Thus, we need an octet string of size 8 Bytes as the unique serial number.
- *Curve:* The named ECC curves for use with ECQV based key generation. For our profile, we fix this to secp256r1 and the corresponding value is $0 \times 05$.
- *Hash:* The cryptographic hash function employed for the ECQV based key generation. For our profile, we use SHA-256 with the corresponding value of $0 \times 01$.
- *Issuer ID:* This is an 8 byte identifier of the issuing CA. When checking the certificate chain, this information is necessary to locate the corresponding issuing certificate.
- *Valid from:* This 5 byte field represents the start of the certificate validity in Unix time and is denoted as the number of seconds passed since January 1, 1970.
- *Valid to:* A 4 byte field to denote the end of certificate validity in seconds since *Valid from* field.
- *Subject ID:* A unique 8 Byte identifier to identify the owner of private key relative to the public key in this certificate.
- *Key Usage:* The usage field defines the valid purpose of key included in the certificate as per RFC 5280 [17]. In the current context, we set its value to $0 \times 08$ denoting digital signature.
- *Public Key Reconstruction Data:* A value calculated by the ECQV algorithm. This field allows the reconstruction of public key and concurrent implicit verification of the certificate (at the time of using public keys for verifying signatures).The size depends on the type of ECC curve chosen. For this profile, since we use secp256r1 curve, the public key is of 33 bytes.

### 2) CBOR ENCODING AND COMPRESSION

As already discussed, the existing PKI depend on ASN.1 encoding for description of certificate structures which is complex, bigger and not lightweight. CBOR, on the other hand, is easy to encode and decode, and produce smaller encoded strings. Thus, we propose to compress the ECQV certificate with CBOR encoding schema. By replacing ASN.1 encoding (see Figure 6) with CBOR encoding in ECQV certificates and integrating it with the proposed profile restrictions, we are able to reduce the certificate size by more than 20%. The final certificate structure is depicted in Figure 7, where all fields known by profiling have been eliminated and the remaining have been encoded in CBOR format.

**TABLE 3. Encapsulation of fields in LECQV profile.**

| Field | Byte(s) | Value |
|---|---|---|
| Type | 1 | Type 1(0) |
| Serial Number | 8 | Octet String - Unique Serial Number |
| Curve | 1 | secp256r1 (0x05) |
| Hash | 1 | SHA-256 (0x01) |
| Issuer ID | 8 | Octet String - Identifier of CA |
| valid from | 5 | Beginning of Validity in Unix Time |
| valid to | 4 | End of Certificate validity |
| Subject ID | 8 | EUID |
| Usage | 1 | DigitalSignature (0x08) |
| Public Key Reconstruction Data | 33 | An ECC point to reconstruct the public key |
| Total | 70 | |

```
ECQVCertificate ::= SEQUENCE {
type INTEGER,
serialNumber OCTET STRING (SIZE (8)),
Curve INTEGER,
Hash INTEGER,
issuerID OCTET STRING (SIZE (8)),
validFrom OCTET STRING (SIZE (5)),
validDuration OCTET STRING (SIZE (4)),
subjectID OCTET STRING (SIZE (8)),
Usage BIT STRING,
pubKey OCTET STRING
}
```

**FIGURE 6. The ECQV certificate structure expressed in ASN.1 DER [23].**

```
L-ECQVCertificate ::= [
serialNumber       : BYTE STRING,
issuerID           : BYTE STRING,
validFrom          : BYTE STRING,
validDuration      : BYTE STRING,
subjectID          : BYTE STRING,
pubKey             : BYTE STRING
]
```

**FIGURE 7. The LECQV CBOR profile of ECQV certificates expressed in CDDL.**

```
0x04      // Octet String
0x08      // Size 8
0x01 0x23 0x45 0x67 0x89 0xAB 0xCD 0xEF
// Value "01-23-45-67-89-AB-CD-EF"
```

**FIGURE 8.** The serial number field of an ECQV certificate encoded in ASN.1 DER (10 Bytes).

```
0x48                // Byte Array, Size 8
0x01 0x23 0x45 0x67 0x89 0xAB 0xCD 0xEF
// Value 0x0123456789ABCDEF
```

**FIGURE 9.** The serial number field of an ECQV certificate after profiling and CBOR encoding (9 Bytes).

In the subsequent sections, we discuss the expected size reductions in essential certificate fields obtained via CBOR compression.

*Type:* This field is removed and rebuilt as necessary, assuming an implied type 1 with value 0. This saves 1 byte.

*Serial Number:* Since this field is essential, the savings are derived solely through CBOR encoding. The overhead of encoding is reduced to one byte and the overall size of the field is reduced from 10 Bytes to 9 Bytes with CBOR as compared to ASN.1 encoding. Figure 8 and 9 show in detail the increased brevity in size of serial number field achieved through CBOR encoding.

*Curve and Hash:* The curve and hash fields are eliminated as a result of profile restriction to accept only fixed values of secp256r1 and SHA-256 respectively. This saves 2 bytes.

*Issuer ID:* Following the profile restrictions, the 10 byte octet string in ASN.1 encoding is compressed to 9 bytes with CBOR encoding. For self-signed certificates, this value is set to zero.

*Valid from and Valid to:* These are encoded as byte strings in CBOR which reduces the size from 13 bytes to 11 bytes.

*SubjectID:* The subject ID identified by organizationally unique identifier (EUI-64) based on unique 48-bit MAC address can be encoded using only 9 bytes with CBOR.

*Usage:* As this is fixed by the proposed profile restrictions, it is removed, thereby, saving 1 byte.

*Public Key Reconstruction Data:* An elliptic curve point embedded in the implicit certificate for reconstruction of the public key. This depends on the type of curve chosen and the CBOR encoding of this field for secp256r1 is 35 bytes.

The next section presents a detailed description of EDHOC interactions, as well as its integration with the proposed profile approach by exploiting CoAP for transport of the corresponding EDHOC messages.

## V. PROPOSED L-ECQV BASED EDHOC AUTHENTICATED MESSAGE EXCHANGE FOR KEY ESTABLISHMENT

In this section, the L-ECQV certificates (proposed in Section IV) have been implemented and evaluated as primitives for authenticated key exchange in EDHOC protocol. The evaluation results show that the proposed implicit certificate approach reduces energy, memory and message overhead of EDHOC handshake. This work is the first step towards securing IoT using lightweight implicit certificates for authenticated key exchange in EDHOC protocol at the application layer to support e2e secure communication in IoT. We begin our discussion by explaining the adversarial model used for evaluation of proposed framework of secure communication with L-ECQV certificates.

### A. ADVERSARIAL MODEL

The proposed implicit certificate-based key establishment using EDHOC protocol has been designed to be resistant against attacks defined in the Dolev-Yao (DY) [74] and the Canetti- Krawczyk (CK) [75] threat models.

- According to the DY model, an adversary ($\mathcal{A}$) can stealthily listen to all the communication between the entities involved over an insecure (open) communication channel. To get hold of the secret message being exchanged, the adversary can eavesdrop, manipulate or delete the message in transit or store and even replay the messages by pretending as a trusted party.
- The adversary in CK model is similar to the one in DY model. However, in addition, $\mathcal{A}$ can also extract critical information like secret credentials, keys and other information related to states of current session if they are stored in memory of communicating entities by physically capturing the device by power analysis attack [76] or by initiating session hijacking attacks [75]. Once credentials are obtained, $\mathcal{A}$ can launch replay, device impersonation and man-in-the-middle attacks.

Based on the information given in [77], we also believe that the border router is completely trustworthy and cannot be compromised. Otherwise, the network as a whole would be jeopardised if the border router were compromised. For such cases, we follow the approach discussed in [78] whereby the border routers are secured with tamper resistant hardware technologies. Additionally, the border router may be secured with a physical locking system as well. Finally, the provisioning of certificates is out of scope of this proposal, and an automated or manual enrollment protocol could be implemented through some out-of-band mechanism either by manufacturers or users. Thus, they are stored in the devices memory before network deployment.

### B. FRAMEWORK

The framework of secure communication with proposed factory installed L-ECQV certificates is given in Figure 10 and described as follows:

1) Initially, a root CA pre-installs a factory certificate (L-ECQV) in an IoT device and the 6LBR which will uniquely identify them for the entirety of their existence. This bootstrapping of L-ECQV certificates happen in the secure premises of the manufacturer.
2) In the next phase, the service initialization phase, the pre-installed factory certificates (L-ECQV) are used for

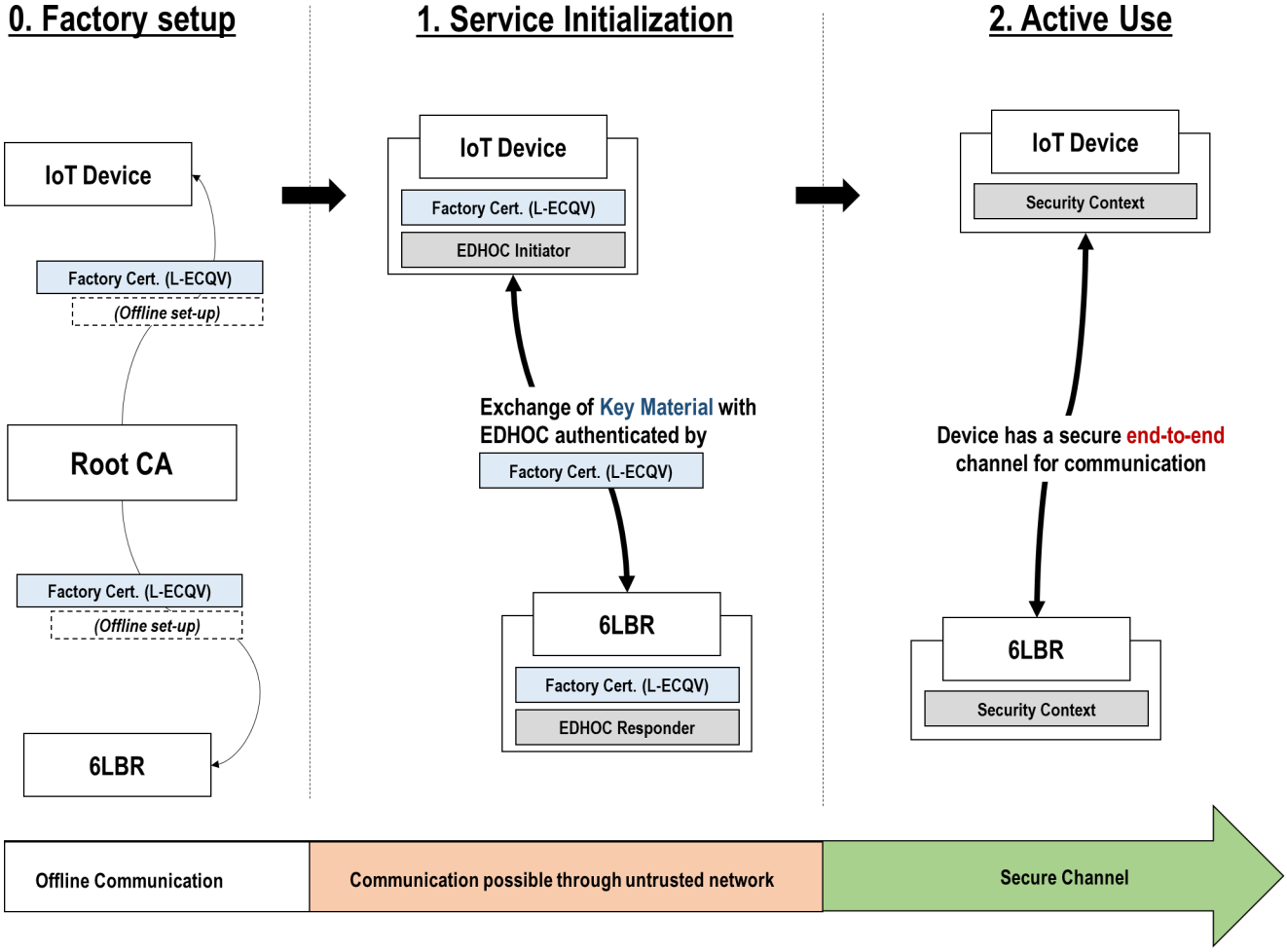


**FIGURE 10.** Framework of secure communication with proposed factory installed L-ECQV certificates.

authentication and exchange of keying material with EDHOC.

3) In the last phase, the active phase, the devices derive OSCORE security context and use it to communicate over a secure and authenticated channel.

Next, a description of EDHOC specification [6] with certificate-based authentication is detailed.

### C. EDHOC WITH CERTIFICATE-BASED AUTHENTICATION

As already discussed, EDHOC is regarded as a potential LAKE protocol to be implemented in constrained IoT scenarios along with the OSCORE standard. In this section, we describe the EDHOC message exchange for key establishment in IoT. It is important to note that these details are based on the usage for CoAP protocol, which has been used to transport EDHOC messages. To support our discussion, we have defined a set of notations (refer to Table 4) which will be used at all times in this article. Additionally, henceforth, we use the symbol *I* and *R* for representing parameters generated by Initiator and Responder respectively. Figure 11 shows the EDHOC message exchanges with certificate-based authentication mode. It is important to note that PSK and RPK authentication modes are not included, since they are not scalable and do not provide authenticated credentials respectively. The process of communication between EDHOC initiator and responder for certificate-based authentication is depicted in Figure 11 and explained as follows:

1) **EDHOC Message 1 →**
In the certificate-based authentication mode of EDHOC, the underlying assumption is that the initiator and responder have received their certificates from the trusted CA (e.g. during manufacturing or while bootstrapping). The protocol is commenced by the initiator which further generates its own ephemeral public key, G_X and connection identifier, C_I. In addition to this, the initiator adds supported cipher suites and a data item METHOD_CORR specifying the authentication method (here, certificates having signature keys) and correlation properties of the handshake. Finally, the message 1 is composed and encoded as a CBOR sequence and sent to the responder. Figure 11 shows

**TABLE 4.** Notations used by EDHOC protocol.

| Notation | Meaning |
|---|---|
| G_X | ECDH Ephemeral Public Key of the Initiator |
| G_Y | ECDH Ephemeral Public Key of the Responder |
| C_I | Connection Identifier of the Initiator |
| C_R | Connection Identifier of the Responder |
| CRED_I | Authentication Credential containing Public authentication key of Initiator *(here, L-ECQV certificate)* |
| CRED_R | Authentication Credential containing Public authentication key of Responder *(here, L-ECQV certificate)* |
| I | Private Authentication Key of Initiator |
| R | Private Authentication Key of Responder |
| ID_CRED_I | Identification Information of authentication credential of Initiator |
| ID_CRED_R | Identification Information of authentication credential of Responder |
| Enc | Encryption using key derived from shared secret |
| AEAD | Authenticated Encryption with additional data |
| MAC | Message Authentication Code |
| HKDF | HMAC-based key derivation function |
| G_XY | ECDH shared secret |
| K_2e | Pseudo random Key extracted from shared secret G_XY in message 2 for encryption |
| K_3ae | Pseudo random Key extracted from shared secret G_XY in message 3 for authenticated encryption |

the EDHOC message exchanges with certificate-based authentication mode.

2) **EDHOC Message 2** ←
After receiving message 1, the responder verifies its support for at least one algorithm from set of cipher suites received from the initiator and generates its ephemeral key pair (G_Y), connection identifier C_Y and the shared secret G_XY. In EDHOC, the authentication can be provided either through static DH keys or signature keys. In this mode, the consideration is on signature keys or public authentication keys which are included in the certificate owned by the entity and are issued by the CA. CRED_I and CRED_R are the authentication credentials which contain the public authentication key of the initiator and responder. The EDHOC depends on COSE for providing these credentials which could be either CWTs, C509 or X.509 certificates. It is assumed that both parties have agreed to a common encoding method for the credentials. The identification information ID_CRED_I and ID_CRED_R help to retrieve the authentication credentials, CRED_I and CRED_R and respective authentication keys. This information does not have any cryptographic purpose in EDHOC as CRED_I and CRED_R are already intergrity protected. These could either encapsulate the authentication credential or reference them if they have been pre-provisioned or acquired through some out-of-band mechanism.

The responder, then computes MAC on CRED_R, G_X (as received in message 1) and G_Y. This object is later signed using the private authentication key of responder, R to form the COSE_Sign1 object which along with the identification information of authentication credentials, ID_CRED_R is encrypted using the derived keys. For deriving keys, EDHOC uses HMAC based Extract and Expand key derivation (HKDF) method to compute pseudorandom keys (PRK) from the shared secret, G_XY. The responder computes such a PRK (here, K_2e) for encrypting the complete security context of message 2 which also includes ID_CRED_R that enables the identification and transport of public authentication keys of responder. The responder computes the COSE_Encrypt0 object with the EDHOC AEAD algorithm from the selected cipher suite. The COSE_Encrypt0 object along with C_R, C_I and G_Y is sent to the Initiator in the form of CBOR encoded message.

3) **EDHOC Message 3** →
When the initiator receives message 2, it decodes the same and retrieves the protocol state using the connection identifier C_I. It decrypts the COSE_Encrypt0 object by using AEAD and verifies the identity and the signature of the responder by using the algorithm in the selected cipher suite. If this process succeeds, the initiator starts formatting and processing of message 3. The generation of message 3 follows the same procedure as message 2. The only difference is related to the COSE_Encrypt0 object. In this case, it is encrypted using the secret key K_3ae, computed by applying HKDF on the shared secret, G_XY and the authentication is provided through ID_CRED_I and COSE_Sign1 object. It is important to note that while Sign_or_MAC2 is signed using private authentication key of responder, the Sign_or_MAC3 object is signed using private authentication key of initiator, I. Finally, the connection identifier (C_R) and the COSE_Encrypt0 object is sent to the responder as a CBOR encoded message.

When message 3 is sent, the initiator is guaranteed that only the responder may calculate the symmetric pseudo random key (this is called implicit key authentication). The underlying application can now derive symmetric keys and may even send protected data alongside or along with message 3 using the EDHOC interface. However, the initiator is still unsure that the responder has actually computed the key and it defers to store its keying material till the responder computes the key. This criteria is akin to transport protocol's waiting for an acknowledgement (ACK). For this explicit key confirmation, EDHOC optionally sends a message (message 4) from the responder to the initiator.

### D. L-ECQV AS AUTHENTICATION CREDENTIAL IN EDHOC

Preliminary EDHOC proposals like [11], [60], [79], and [80] include the capabilities for PSK and RPK based

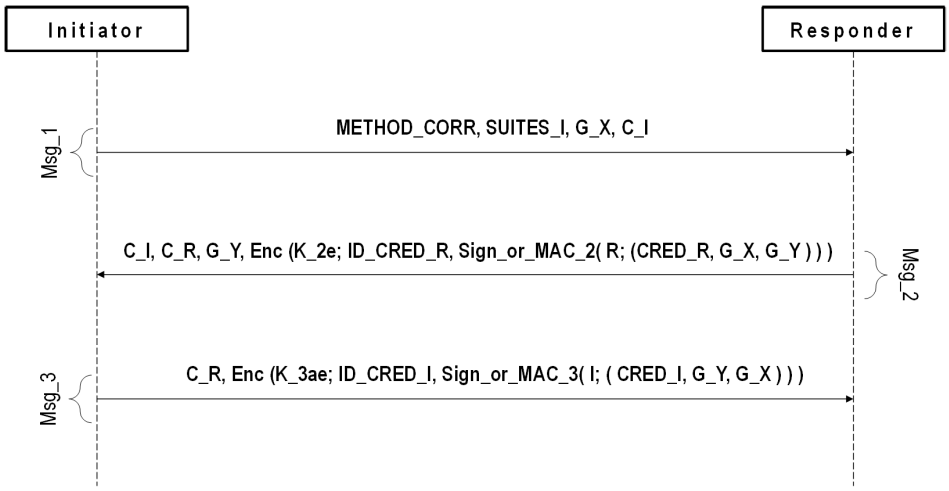


**FIGURE 11.** EDHOC message exchange with certificate based authentication.

authentication. However, PSK method is not scalable and are vulnerable to key compromise attacks. Therefore, we design L-ECQV for using EDHOC with certificates. The initial secure session is established by factory installed certificates. Since the process of verification depends on the authentication method used, herein certificates are deployed and thus, the Sign_or_MAC fields in message 2 and message 3 are used to carry signatures.

Additional design choices include sending full certificates as part of the EDHOC handshake or merely references, which are conveyed in the CRED field in Figure 11. The reference format, if chosen in EDHOC, aims to include the hash of the certificate in the CBOR encoded form. Though this option can reduce the communication overhead but this process requires the communicating parties to have the referenced certificates in their memory locally. For dynamically changing IoT network with new devices joining and exiting, this assumption is not reasonable and therefore, this work proposes the usage of full certificates. The proposed work transports (and not references) the identification credentials (ID_CRED_I and ID_CRED_R) that contain the actual value of authentication credential (CRED_R and CRED_I) which are effectively certificates encoded as a CBOR messages. The certificates can be explicit like X.509 encoded in ASN.1 or CBOR format (C509) or even implicit certificates like ECQV.

In this work, the proposed L-ECQV implicit certificate have been used as authentication credential (CRED_I and CRED_R) in EDHOC and ID_CRED_I and ID_CRED_R hold the value of these credentials. Since L-ECQV certificates are already CBOR encoded, they can be directly used with EDHOC protocol. A cipher suite is essentially a group of ordered algorithms to secure communications in EDHOC including the AEAD, hash, MAC, key exchange and signature algorithms. The EDHOC specification [6] allows to define new private cipher suites with values -24, -23, -22, -21. Thus, a new cipher suite for EDHOC with L-ECQV certificates with value -24 has been presented. The novel cipher suite of EDHOC with L-ECQV certificate implemented in this thesis is **EDHOC_ECDHE_ECQV_WITH_AES_128_CCM_SHA256**. This novel cipher suite initializes the different protocol elements of EDHOC as follows:

- METHOD_CORR: 0 (Indicating both entities use signature keys)
- SUITES_I: -24
- G_X: Ephemeral Public Key of IoT device
- C_I: 37 (unique connection identifier chosen by IoT device)
- C_R: 27 (unique connection identifier chosen by 6LBR)
- G_Y: Ephemeral Public Key of 6LBR
- CRED_I: *L-ECQV* Certificate of IoT device
- CRED_R: *L-ECQV* Certificate of 6LBR
- ID_CRED_I: Encapsulated *L-ECQV* for transportation
- ID_CRED_R: Encapsulated *L-ECQV* for transportation
- Encryption algorithm: AES in CCM mode
- Hash: SHA256
- Key exchange: ECDHE

## VI. L-ECQV: SECURITY ANALYSIS AND CONSIDERATIONS

The security of proposed approach relies heavily on security strength of well proven algorithms ECQV and EDHOC. The formal analysis of EDHOC has been performed in [79] and that of ECQV in [23]. In this section, we provide the security analysis of L_ECQV through various security and functionality features listed in Table 5, which proves its robustness against the possible attacks of the DY and CK adversarial models discussed in Section V-A.

According to the standard specification [6], the EDHOC protocol in its default form provides mutual authentication, perfect forward secrecy and identity protection. Since the EDHOC protocol is based on SIGMA [8] protocol and therefore, EDHOC authenticated with proposed LECQV

**TABLE 5. Comparative study on security and functionality features.**

| | [31] | [19] | [42] | [44] | Proposed |
|---|---|---|---|---|---|
| SFF1 | ✓ | ✓ | ✓ | ✓ | ✓ |
| SFF2 | | | | | ✓ |
| SFF3 | ✓ | ✓ | | | ✓ |
| SFF4 | ✓ | ✓ | | | ✓ |
| SFF5 | ✓ | ✓ | | | ✓ |
| SFF6 | | | | | ✓ |
| SFF7 | | | ✓ | ✓ | ✓ |
| SFF8 | | | ✓ | ✓ | ✓ |
| SFF9 | | | | | ✓ |
| SFF10 | | | ✓ | | ✓ |

Note: SFF1-Mutual Authentication; SFF2-Perfect Forward Secrecy; SFF3-Confidentiality; SFF4- Integrity; SFF5-Non-repudiation; SFF6-Anonymity and untraceability; SFF7: Resilience to Impersonation Attack; SFF8: Resilience to Replay Attacks; SFF9: Resilience to Modification Attack; SFF10 -Resilience to Man-in-the-middle Attack

certificate-based signature keys ensure identity protection of the initiator against active attackers [6]. It also provides key compromise impersonation protection against an attacker having long access to long-term key or the ephemeral secret key. The implementation of authenticated encryption in EDHOC and message authentication code using keys derived from shared secret ensures confidentiality, integrity and non-repudiation in the communication. Suppose, $\mathcal{A}$ eavesdrops and monitors the messages Msg_1 - Msg_3. Msg_1 is exchanged in plaintext format but it does not contain any identifying information while Msg_2 and Msg_3 are encrypted with keys derived from ephemeral-ephemeral ECDH shared secret G_XY. Moreover, these messages are constructed using random connection identifiers and public keys which are dynamic in nature from one session to another which makes tracing a user difficult for $\mathcal{A}$. Thus, the proposed scheme preserves the anonymity property. The expansion of message authentication coverage to additional elements like previous plain text messages, external authorization data prevents from replay and modification attacks from $\mathcal{A}$. The L-ECQV certificates are presumed to be installed at the manufacturer domain and the CA is considered to be the trusted entity. Even if $\mathcal{A}$ physically captures the IoT device, it can extract certificate issued by the factory CA while it cannot compute the public key from the certificate as the private key k_A was set at the time of initial bootstrapping phase in the secure environment of manufacturer. Hence, the compromised information does not help in computing the public key and thus, the session key is also secured. Thus, the proposed scheme is resilient against this attack. Table 5 provides a comparative study of security and functionality features of proposed work with relevant literature.

The major components of proposed lightweight certificate based key exchange is reliant on existing standards, which have already been scrutinized for security in detail. In this section, we highlight the additional findings and evaluate if the proposed solution creates new vulnerabilities either in the design or in the implementation. We do not assert to cater to all vulnerabilities associated with IoT operating systems and their parts, since those are being executed in parallel by many security researchers and developers. Here, we only target the vulnerabilities of underlying protocols and risks instilled by new proposed components.

*Protocol Vulnerabilities:* The three main components: EDHOC, DTLS and proposed certificate format, L-ECQV are all based on ECC. The selection of ECC is motivated by the lower cost compared to RSA. Also, we deploy the compressed format to represent ECC keys to further reduce the size of the key. However, the advancement in quantum computing may lead to the security of ECC and other cryptographic methods as obsolete but any such attack on the recommended ECC key lengths cannot be forseen in the near future [81]. Besides, the foundations of L-ECQV lies in the security of ECQV certificates, which has been thoroughly analyzed in [82]. Even the current version of DTLS 1.3 prohibits the use of insecure hashing algorithms MD5 and SHA-1 which the previous versions did not.

It is worth highlighting a significant point in SIGMA [8] protocol used; all things considered if the constituents keep their guarantees, the overall protocol implementation will also provide the expected security services. If any vulnerabilities are found in the primitives of a cipher suite or implementation, the respective constituent must be replaced or updated. This emphasizes the need for secure software updates, another security requirement which is enabled by PKI for IoT solutions.

*New Component Functionality:* Though this work does not address software security of IoT devices, the reduced complexity of CBOR format is intended for easy parsing on limited devices and the attack surface is even reduced by using a basic parser instead of the complex ASN.1 parser. An attacker may provide incorrectly compressed certificate but the lost effort is less than whole energy of verifying a certificate with right format but invalid signature. As a result, this new assault is less likely than current possibilities for attacks on the security service of availability.

*Lifetime of Security Context:* While many IoT devices are becoming resource rich, the asymmetric operations are still resource intensive. However, energy is still a major concern, and thus it should be used thoughtfully. Once the key has been established, the IoT devices may keep it for as long as they deem safe, requiring only symmetric operations for further communication. It is also important to ensure that lifetime of security context is less than the validity period of the certificates and therefore it becomes necessary to ensure timely termination of security contexts.

*IoT Devices as Bots:* The Proliferation of IoT devices creates new avenues for novel botnet attacks with IoT devices used as bots for launching DDoS attacks. L-ECQV ensures robust and lightweight security to IoT devices and prevents

them from being jeopardized. However, they are still susceptible to physical cloning and hacking. Newer improved IDS and firewalls compatible with IoT may still be required to defend against such attacks to IoT devices from the Internet and vice-versa. The preliminary efforts in this direction have been proposed in some of the recent works as in [83]. Additionally, prevention mechanisms like backing-off from a malicious request [84] to mitigate the effects of attack and restrain its impact may also be focused upon.

*Certificate Enrollment and Re-Enrollment:* This work is focused on initial authentication of IoT device with pre-installed certificates. Preferably, this pre-loading of certificates is usually performed in the secure manufacturer environment during the installation of initial firmware on the IoT device. Our work solves the obstacles of initial bootstrapping of an IoT device but when the device is deployed in a dynamic network, there must be mechanisms for certificate enrollment and re-enrollment to achieve a fully automated PKI. Thus, a scalable certificate enrollment protocol for IoT needs to be developed for a fully automated and lightweight certificate management in IoT.

### A. APPLICABILITY TO REAL IoT SCENARIOS AND CHALLENGES IN IMPLEMENTATION

The proposed L-ECQV certificates are lightweight versions of ECQV certificates, a type of digital certificate that uses ECC to provide secure key exchange and authentication. Similar to ECQV certificates, the proposed L-ECQV certificates can be used for a variety of purposes in practical IoT scenarios, such as device authentication, data encryption, and secure communication between devices.

- **Device Authentication:** In an IoT ecosystem, L-ECQV certificates can be used to authenticate devices, ensuring that only authorized devices are allowed to access the network. For example, a smart home security system may use L-ECQV certificates to authenticate devices such as smart locks, cameras, and sensors.
- **Secure Data Transfer:** L-ECQV certificates can be used to encrypt data that's transmitted between IoT devices, ensuring that sensitive information is protected from prying eyes. For instance, a healthcare IoT system may use L-ECQV certificates to secure the transmission of medical data from a patient's wearable device to a healthcare provider's server.
- **Secure Firmware Updates:** Updating firmware is a critical part of IoT device maintenance. However, if the firmware update process is not secure, it can be vulnerable to malicious attacks. L-ECQV certificates can be used to authenticate firmware updates, ensuring that only authorized updates are installed on the device. For example, a smart home thermostat may use L-ECQV certificates to verify the authenticity of firmware updates.
- **Remote Access Control:** In some IoT scenarios, it may be necessary to control devices remotely. L-ECQV certificates can be used to ensure that only authorized individuals can access the device remotely. For instance, a smart irrigation system may use L-ECQV certificates to authenticate a farmer's mobile app, allowing them to control the system remotely.

Overall, L-ECQV certificates offer a robust and secure solution for authentication and data protection in IoT scenarios. As shown in Figure 10 and discussed above, the framework of proposed EDHOC authentication with factory-installed L-ECQV certificates is based on the offline setup of IoT devices in the secure manufacturer domain. Thus, the framework introduces zero additional network overhead for configuring the IoT devices with the proposed L-ECQV certificates. The proposed certificate was also proven to be standard ECQV compatible and processable by any device that supports ECQV certificates. However, the Contiki code size was slightly increased compared to conventional EDHOC due to the addition of the compression mechanism of ECQV certificates on the top of the EDHOC library. Additionally, in the current implementation, there is a need for external libraries (discussed in the next section) which can be removed if further optimizations are made to the compressed certificate rendering it an independent entity.

## VII. IMPLEMENTATION AND EVALUATION

We develop L-ECQV implementation as modules in C for easy adaptation to IoT devices, particularly the classes of devices defined in the RFC 7228 [21] which provides a terminology on power, memory, and processing resources of these devices that have been useful in the standardization work for constrained-node networks. Such devices typically run embedded operating systems like Contiki, RIOT OS, Mbed OS, TinyOS, etc. In this work, the L-ECQV implementation has been evaluated in Contiki [85] and contains EDHOC and CBOR encoding and decoding. For testing and research, we have developed our own C library for the encoding of ECQV certificates. The encodings conform to the Standards for Efficient Cryptography (SECG) by Certicom Corp. [4]. Additionally, the point decompression support is provided by the microECC library [1] and CBOR functionality from CN-CBOR.[2]

In this section, we evaluate the size of different messages of EDHOC based on different certificates. As a baseline, the already existing EDHOC library[3] has been extended. Currently, this library supports X.509 certificates with DER encoding. We integrated our proposed profile with this library to evaluate the performance of proposed certificate formats.

### A. EXPERIMENTAL SETUP AND METHOD

We follow an experimental research methodology where we measure the effect of one variable in setup while keeping the others as static. This helps in ensuring proper attribution

[1] https://github.com/kmackay/micro-ecc
[2] https://github.com/cabo/cn-cbor
[3] https://github.com/alexkrontiris/Enrollment-over-EDHOC

**TABLE 6. Certificate sizes.**

| Certificate | Encoding | Size |
|---|---|---|
| RSA | DER | 0.8kB-3kB |
| ECDSA | DER | 296-2000 B |
| ECDSA [19] | CBOR | 140-160 B |
| ECQV | DER | 92 B |
| L-ECQV (Profile and Compressed) | CBOR | 73 B |

of results to specific change in the system. In addition to the basic system tests, we also present the microbenchmarks showing the discrete aspects. Both the ASN.1 and CBOR versions are used in comparisons to show CBOR advantages and its compatibility with conventional deployments. We develop a client-server module in Contiki OS which is further extended to EDHOC for security of COAP messages by keeping other parameters as constant for fair comparisons. The proposed module tests the performance of L-ECQV with EDHOC against unmodified EDHOC.

### B. MICRO BENCHMARKS

#### 1) CERTIFICATE SIZES

Table 6 lists the expected certificate sizes in bytes for the recommended CBOR and conventional ASN.1 encoding of X.509 and ECQV certificates. In DER encoding, the RSA certificates with 2048-bit keys and minimal extensions have a lower size limit of 800 bytes whereas a profiled ECQV certificate is approximately 10% of the RSA certificates. In comparison to the existing ECC profiled certificates in [19], the proposed L-ECQV certificates are 50% smaller in size.

It is evident that such reductions have been obtained on the basis of specific profile for constrained IoT networks whereby some fields were dropped or recreated. Other compression mechanisms like gzip/deflate [28] could further reduce the size of certificates but no reduction in sizes of already profiled implicit certificates is possible.

#### 2) EDHOC HANDSHAKE MESSAGE OVERHEAD

In IoT-constrained environments, an important aspect to be considered is the message overhead due to the constraints of network bandwidth. In this regard, EDHOC utilizes CBOR encoded COSE objects to minimize the size of messages. However, this overhead largely depends on the type of authentication mechanism chosen during the EDHOC key exchange. For instance, the Pre-shared Key (PSK) and Raw Public Key (RPK) mechanisms incur less overhead as they implement static DH keys which are identified by key identifiers. However, these schemes are not scalable and thus, kept out of this comparative evaluation. Therefore, we only evaluate the message overhead with certificate based authentication credentials in this analysis. For the comparison of message overhead over different certificates, Table 7 presents a detailed view of overhead incurred by each message. Since message 1 does not carry any authentication credential, its size is same for all types of certificate-based authentication credentials.

Based on the different type of certificate and encoding method, message 2 and message 3 require variable message size in an EDHOC handshake. The explicit certificate X.509 in ASN.1 encoding format and the CBOR encoded X.509 (C509) incur 419 and 234 bytes in message 2 and 394 and 218 bytes in message 3, respectively. The CBOR encoding of X.509 certificates represents a 42% decrease in the total overhead than an ASN.1 encoded X.509 certificate. The overhead is further reduced by replacing the X.509 certificates with implicit ECQV certificates by 52%. Finally, our proposed profiled and compressed L-ECQV certificates incur the minimum overhead of 368 bytes which is 56% less than the conventional approach. Based on these results, it is evident that notable overhead reduction in handshake messages is achieved by the proposed certificate profile which has a key impact in IoT networks where the Maximum Transmission Unit (MTU) is limited and fragmentation of packets is required.

#### 3) CERTIFICATE VALIDATION TIME

The experimental evaluation is based on the different ECC curves to analyze the suitability of compressed ECQV certificates in IoT devices for authenticating the communication. To this aim, we assess the certificate size and the total time lapse in validating the certificates which includes the public key extraction time and the validation time of the key pair. Table 8 shows the certificate sizes and timings required for certificate validation for the uncompressed (ASN.1) and the compressed (CBOR-encoded) certificate profiles. The experiments are repeated (i.e. 10,000 times) and the mean is computed with a confidence interval of 95%.

As discussed the performance evaluation of implicit certificates is carried out with Contiki OS. ECQV certificates have smaller key sizes as compared to the explicit X.509 certificates developed by ECC and RSA. Additionally, the validation and verification of implicit certificates is extremely fast compared with standard explicit certificates. This is due to fewer elliptic curve operations and implicit signature verification.

It is clearly evident from Table 8 that ECC curves with smaller key sizes are smaller in size and consequently, result in lesser validation time. Furthermore, it was noted that ECQV curves 'secp256r1' and 'secp256k1' provide optimum security with same level of security and nearly same validation times, we choose the comparatively stronger NIST approved curve 'secp256r1' to be the most optimum for implementing certificate-based authentication in IoT.

The proposed profile and compression has been implemented as an application for the Contiki OS. The application supports the operations of compression, decompression and verification of compressed ECQV certificates and generation of new certificates. The developed app *lecqv* (from Lightweight CBOR encoded ECQV) is placed in the *contiki-ng/apps/lecqv* directory.

**TABLE 7.** Total overhead of EDHOC handshake.

| Authentication Credential | Message_1 | Message_2 | Message_3 | Total Overhead (Bytes) |
|---|---|---|---|---|
| X.509 | 37 | 419 | 394 | 850 |
| C509 | 37 | 234 | 218 | 489 |
| ECQV | 37 | 197 | 172 | 406 |
| **L-ECQV (Profile and Compressed)** | **37** | **178** | **153** | **368** |

**TABLE 8.** Certificate size and validation times of standard and profiled ECQV implicit certificate for authentication with different curves and encodings.

| Profile | Encoding | Curves | Private Key (Bytes) | Uncompressed Public Key (Bytes) | Compressed Public key (Bytes) | Certificate Size (Bytes) | Certificate Generation Time (in ms) | Public Key Extraction Time (in ms) | Certificate Validation Time (in ms) |
|---|---|---|---|---|---|---|---|---|---|
| Standard ECQV | ASN.1 DER | secp160r1 | 20 | 40 | 21 | 80 | 0.765 | 0.388 | 0.719 |
| | | secp192r1 | 24 | 48 | 25 | 84 | 0.883 | 0.454 | 0.851 |
| | | secp224r1 | 28 | 56 | 29 | 88 | 1.502 | 0.811 | 1.447 |
| | | secp256k1 | 32 | 64 | 33 | 92 | 1.834 | 0.950 | 1.781 |
| | | secp256r1 | 32 | 64 | 33 | 92 | 2.234 | 1.144 | 2.190 |
| | CBOR | secp160r1 | 20 | 40 | 21 | 69 | 0.718 | 0.364 | 0.680 |
| | | secp192r1 | 24 | 48 | 25 | 73 | 0.877 | 0.450 | 0.842 |
| | | secp224r1 | 28 | 56 | 29 | 77 | 1.493 | 0.805 | 1.418 |
| | | secp256k1 | 32 | 64 | 33 | 81 | 1.677 | 0.879 | 1.635 |
| | | secp256r1 | 32 | 64 | 33 | 81 | 2.220 | 1.142 | 2.179 |
| Proposed LECQV | ASN.1 DER | secp160r1 | 20 | 40 | 21 | 68 | 0.748 | 0.382 | 0.709 |
| | | secp192r1 | 24 | 48 | 25 | 72 | 0.828 | 0.422 | 0.794 |
| | | secp224r1 | 28 | 56 | 29 | 76 | 1.458 | 0.786 | 1.380 |
| | | secp256k1 | 32 | 64 | 33 | 80 | 1.648 | 0.861 | 1.597 |
| | | secp256r1 | 32 | 64 | 33 | 80 | 2.135 | 1.093 | 2.093 |
| | CBOR | secp160r1 | 20 | 40 | 21 | 61 | 0.604 | 0.259 | 0.563 |
| | | secp192r1 | 24 | 48 | 25 | 65 | 0.714 | 0.319 | 0.685 |
| | | secp224r1 | 28 | 56 | 29 | 69 | 1.283 | 0.647 | 1.245 |
| | | secp256k1 | 32 | 64 | 33 | 73 | 1.544 | 0.756 | 1.479 |
| | | **secp256r1** | **32** | **64** | **33** | **73** | **1.931** | **0.980** | **1.968** |

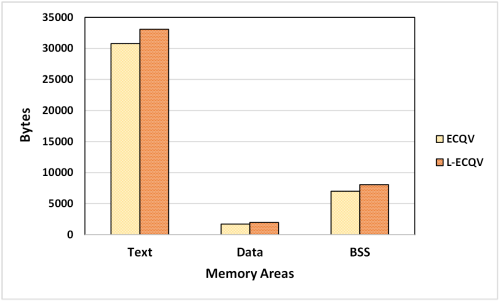


**FIGURE 12.** Memory usage in compressed and uncompressed certificates.

#### 4) MEMORY USAGE

In this section, we compare the size of compiled program and memory areas in the implementation of the proposed compressed (*lecqv*) certificate application. Our application is written in C and therefore, we compare the size of compiled program with respect to different memory areas: text segment, initialized data segment and uninitialized data segment (bss). As shown in Figure 12, it is clear that the addition of compression mechanism of ECQV certificates on the top of EDHOC library adds a little memory overhead. Since the compression mechanism and both protocols OSCORE and EDHOC are based on the same set of technologies (CBOR and COSE), the integration of proposed compression mechanism incurs less overhead on IoT devices.

#### 5) ENERGY CONSUMPTION

Energy consumption is one of the key constraints for battery-powered IoT devices. For deployment in realistic scenarios, knowing, comprehending and controlling the energy usage for any new functionality is essential. In this section, we show how reduced communication translates to lesser energy requirements. We measure the time spent on each operation of EDHOC handshake using the Energest [86] timer mechanism in Contiki [85]. This enables us to compute the consumption based on voltage and current levels using the CC2538 hardware datasheet [87].

Based on the discussion of EDHOC handhshake, we compare two scenarios: one with uncompressed ECQV certificate and other with proposed L-ECQV certificates. In this evaluation, we compute the energy consumption for the three messages of the handshake process corresponding to certificate exchanges. Each experiment is repeated 10,000 times. The results of energy consumption are shown in Figure 13. As expected, the energy consumption for message 1 of EDHOC handshake is the same as equal sized messages are sent in both scenarios. For message 2 and message 3, the proposed L-ECQV certificate incurs communication cost of 3.2 mJ and 2.8 mJ respectively. It is clear that just by compression the overall energy consumption is reduced by 27%.

### C. COMPARISON WITH PREVIOUS STATE-OF-THE-ART

In this section, we compare the performance of proposed work with the existing state-of-the-art proposals. As already emphasized in Section II, only those works have been included for comparison which are based on the standardized IoT protocol stack [24] and those which implement compressed PKI certificates (both implicit and explicit) in some form in the IoT.

**TABLE 9. Comparison of proposed work with state-of-the-art proposals.**

| Metric | Forsby et al. [31] | XIOT [19] | LightCert [34] | Indraj [32] | BLE [38] | **Proposed L-ECQV** |
|---|---|---|---|---|---|---|
| Certificate Size (Bytes) | 146 | 140 | 742-1170 | 306 | 103 | **73** |
| Energy Consumption (mJ) | 19.8 | 9.0 | 25.0-30.0 | >100 | 34.13 | **6.0** |
| RAM (in KB) | 1.15 | 1.1 | 32 | 2.5 | - | **1.01** |
| ROM (in KB) | 4.4 | 3.7 | 512 | 11 | - | **3.3** |

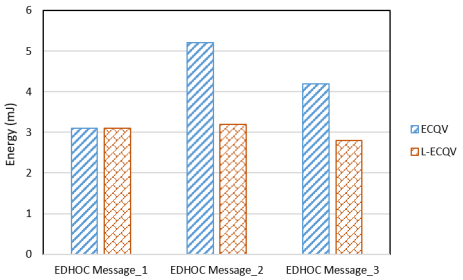


**FIGURE 13. Comparison of energy consumption for compressed and uncompressed certificates.**

The previous work on compressed certificates is represented by the work in [19], [31], [32], and [34]. All these works correspond to the compressed X.509 certificates for authenticating IoT communication protected with DTLS. In [34], the authors implemented their proposed lightweight certificate on ARM Cortex-M3 MCU with 512kb ROM and 32 kb RAM and removed the certificate contents which were common across many certificates within IoT sub-network to achieve smaller certificate sizes ranging from 742-1170 bytes. Their proposed compressed certificate led to a reduction in energy consumption by up to 25-30 mJ. In [19] and [31] measures are taken to encode X.509 certificates in CBOR format. The focus of the work in [32] was on the enrollment of compressed certificates over DTLS proposed by the authors in [31]. This work reused the semantics of the Enrollment over Secure Transport (EST) protocol designed for conventional Internet devices. In [19], CBOR encoding is performed during DTLS handshake while our implementation is using EDHOC handshake, making it hard to directly compare, but from viewpoint of certificate sizes, memory, and energy overhead on an IoT device, our work is more efficient than previous works on compressed certificates as discussed below.

A comparison of the proposed implementation with the previous state-of-the-art is given in Table 9. The works in [31] and [19] have been used for comparison as they are directly based on the compression of certificates and integrated with DTLS security protocol, which provides security to CoAP protocol like EDHOC. The results in Table 9 clearly depict that our implementation adds fewer kilobytes in comparison to memory footprint of other works in [19] and [31] as well as the certificate size is also smaller compared to existing works. In terms of energy consumption, the proposed work requires only 6.0 mJ of energy compared to 19.8 mJ in [31], 25-30 mJ in [34], more than 100 mJ in [32] and 9.0 mJ in [19]. A recent work in [38] has also been included for comparing the certificate size and energy consumption with the proposed work. The reason to include this work was that the certificate modeled in this work was based on the DTLS IoT Profile Certificate [2]. Overall, the reasons for the enhanced performance of the proposed work in terms of the reduction in the certificate size and subsequently in energy and memory consumption are: (i) the usage of smaller implicit certificates compared to explicit certificates; (ii) the profiling of implicit certificate to include only essential fields in the constrained IoT network; (iii) the implementation of CBOR encoding mechanism to reduce the certificate size further; and (iv) the manual provisioning of proposed L-ECQV certificates in IoT devices and their integration as authentication credentials in EDHOC results in lower communication and energy overhead during the handshake process.

### D. EVALUATION SUMMARY

The analytical evaluations of the proposed and conventional ECQV certificates with EDHOC protocol saved 27% of energy overhead. In comparison to X.509 certificates, the savings were even bigger i.e. approximately 56% lesser bytes in EDHOC handshake process. The microbenchmarks and system tests indicate that it is practical to implement lightweight ECQV implicit certificate for EDHOC key establishment with acceptable memory overhead. L-ECQV offers improved performance compared to standard ASN.1 encoding with EDHOC in terms of certificate validation times and energy consumption.

## VIII. CONCLUSION

In the deployment of a secure and authenticated communication in IoT using PKI, the size and parsing of certificates is one of the major challenges. While explicit certificates are relatively larger and incur higher operational costs, the implicit certificates with superimposed public key and digital signature are smaller and faster. Additionally, the conventional X.509 explicit DER encoded certificates are not optimized for constrained IoT environments, a more condensed and compact encoding like CBOR reduces the size of certificates significantly.

To provide a lightweight certificate-based authentication scheme for IoT devices and networks, this study has proposed L-ECQV, a lightweight certificate profile of ECQV implicit certificates for IoT devices and networks and analyzed experimentally its performance in securing the IoT

communications. The proposed profile and a novel encoding of implicit certificates significantly reduces the certificate size from 92 bytes to 73 bytes. The proposed certificate format has been implemented as an open-source library in the Contiki operating system and has also been validated through experimentation with EDHOC key exchange protocol on the application layer in standardized IoT communication stack. Similar to the conventional EDHOC with explicit certificates, the EDHOC key establishment with proposed L-ECQV is also secure against replay, key compromise impersonation, and man-in-the-middle attacks and provides perfect forward secrecy among other security features. The analytical evaluations show that overall handshake EDHOC message overhead is reduced by 52% and the energy consumption by 27% by implementing the proposed L-ECQV certificate.

In the future, this study can be extended to multi-factor authentication by including Physical Unclonable Functions (PUF) like the ones in [88] for added security and authentication in IoT devices and networks. Additionally, the proposed work will be implemented on a real-time IoT network to provide a prototype level of the practical approach to the proposed work. The functionalities can also be further extended by integration with a certificate enrollment protocol for the automatic enrollment of proposed L-ECQV certificates in IoT.

In this study, the implementation has only been tested for IoT systems. To prove that proposed L-ECQV certificates are a generic solution that can be adapted by many different technology domains like VANETs, Vehicle-to-Grid (V2G), smart grid, 5G, etc., further tests need to be done. Thus, it would be worthwhile to extend the proposed L-ECQV certificates to these technology domains as future research work. Besides, they can also be used for secure firmware updates, secure data transfer, remote access control, etc. in IoT systems and beyond.

## REFERENCES

[1] I. Markit. (2017). *Number of Connected IoT Devices Will Surge to 125 Billion by 2030, IHS Markit Says*. Accessed: Feb. 19, 2022. [Online]. Available: https://news.ihsmarkit.com/prviewer/release-only/slug/number-connected-iot-devices-will-surge-125-billion-2030-ihs-markit-says

[2] E. Rescorla and N. Modadugu, *Datagram Transport Layer Security Version 1.2*, document RFC 6347, RFC Editor, Jan. 2012.

[3] Certicom. (2016). *Explaining Implicit Certificates*. Certicom. [Online]. Available: https://www.certicom.com/content/certicom/en/code-and-cipher/explaining-implicit-certificate.html

[4] C. Research, "Standards for efficient cryptography, SEC 4: Elliptic curve QU–Vanstone implicit certificate scheme (ECQV)," Version 1.0, Certicom Corp., Mississauga, ON, Canada, Tech. Rep., 2013.

[5] G. Selander, J. Mattsson, F. Palombini, and L. Seitz, *Object Security for Constrained RESTful Environments (OSCORE)*, document RFC 8613, RFC Editor, Jul. 2019.

[6] G. Selander, J. P. Mattsson, and F. Palombini. (Jan. 11, 2023). *Ephemeral Diffie-Hellman Over COSE (EDHOC)*. Work in Progress, Internet-Draft, Draft-IETF-Lake-Edhoc-15. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-lake-edhoc-15

[7] C. Bormann and P. Hoffman, *Concise Binary Object Representation (CBOR)*, document RFC 8949, RFC Editor, Dec. 2020.

[8] H. Krawczyk, "SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie–Hellman and its use in the IKE protocols," in *Proc. Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA: Springer, 2003, pp. 400–425.

[9] Z. Shelby, K. Hartke, and C. Bormann, *The Constrained Application Protocol (CoAP)*, document RFC 7252, RFC Editor, Jun. 2014.

[10] J. Postel, *User Datagram Protocol*, document RFC 768, RFC Editor, Aug. 1980.

[11] S. Pérez, J. L. Hernández-Ramos, S. Raza, and A. Skarmeta, "Application layer key establishment for end-to-end security in IoT," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2117–2128, Mar. 2020.

[12] J. Schaad, *CBOR Object Signing and Encryption (COSE)*, document RFC 8152, RFC Editor, Jul. 2017.

[13] C. Gundogan, C. Amsuss, T. C. Schmidt, and M. Wahlisch, "Content object security in the Internet of Things: Challenges, prospects, and emerging solutions," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 1, pp. 538–553, Mar. 2022.

[14] S. Josefsson and I. Liusvaara, *Edwards-Curve Digital Signature Algorithm (EdDSA)*, document RFC 8032, RFC Editor, Jan. 2017.

[15] M. Malik, M. Dutta, and J. Granjal, "A survey of key bootstrapping protocols based on public key cryptography in the Internet of Things," *IEEE Access*, vol. 7, pp. 27443–27464, 2019.

[16] K. T. Nguyen, M. Laurent, and N. Oualha, "Survey on secure communication protocols for the Internet of Things," *Ad Hoc Netw.*, vol. 32, pp. 17–31, Sep. 2015.

[17] 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, RFC Editor, May 2008.

[18] ITU-T. *Introduction to ASN.1*. Accessed: Oct. 28, 2022. [Online]. Available: https://www.itu.int/en/ITU-T/asn1/Pages/introduction.aspx

[19] 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.

[20] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Cham, Switzerland: Springer, 2006.

[21] C. Bormann, M. Ersue, and A. Keranen, *Terminology for Constrained-Node Networks*, document RFC 7228, IETF, May 2014.

[22] D. S. Vanstone, "Certicom's bulletin of security and cryptography code and cipher," *Code Cipher*, vol. 2, no. 1, pp. 1–6, 2004.

[23] M. Campagna, "Standards for efficient cryptography sec 4: Elliptic curve QU–Vanstone implicit certificate scheme (ECQV)," Version 1.0, Certicom, Mississauga, ON, Canada, Tech. Rep., Jan. 2013.

[24] M. R. Palattella, "Standardized protocol stack for the Internet of (Important) Things," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1389–1406, 3rd Quart., 2013.

[25] H. Birkhol, C. Vigano, and C. Bormann, *Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures*, document RFC 8610, RFC Editor, Jun. 2019.

[26] J. Hoglund, S. Raza, and M. Furuhed, "Towards automated PKI trust transfer for IoT," in *Proc. IEEE Int. Conf. Public Key Infrastruct. Appl. (PKIA)*, Sep. 2022, pp. 1–8.

[27] D. McGrew and M. Pritikin, *The Compressed X.509 Certificate Format*. Fremont, CA, USA: Internet-Draft, Draft-Pritikin-Comp-X509-00, IETF Secretariat, May 2010.

[28] L. P. Deutsch, *DEFLATE Compressed Data Format Specification Version 1.3*, document RFC 1951, RFC Editor, May 1996.

[29] G. Edgecombe. (Dec. 2016). *Compressing X.509 Certificates*. Accessed: Sep. 30, 2018. [Online]. Available: https://www.grahamedgecombe.com/blog/2016/12/22/compressing-x509-certificates

[30] M. Schukat and P. Cortijo, "Public key infrastructures and digital certificates for the Internet of Things," in *Proc. 26th Irish Signals Syst. Conf. (ISSC)*, Jun. 2015, pp. 1–5.

[31] 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*. Cham, Switzerland: Springer, 2017, pp. 123–133.

[32] Z. He, M. Furuhed, and S. Raza, "Indraj: Digital certificate enrollment for battery-powered wireless devices," in *Proc. 12th Conf. Secur. Privacy Wireless Mobile Netw.*, May 2019, pp. 117–127.

[33] H. Kwon, S. Raza, and J. Ko, "POSTER: On compressing PKI certificates for resource limited Internet of Things devices," in *Proc. Asia Conf. Comput. Commun. Secur. (ASIACCS)*, 2018, pp. 837–839.

[34] H. Kwon, J. Ahn, and J. Ko, "LightCert: On designing a lighter certificate for resource-limited Internet-of-Things devices," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 10, pp. 1–15, Oct. 2019.

[35] F. Marino, C. Moiso, and M. Petracca, "PKIoT: A public key infrastructure for the Internet of Things," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 10, pp. 1–15, Oct. 2019.

[36] C.-S. Park and W.-S. Park, "A group-oriented DTLS handshake for secure IoT applications," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1920–1929, Oct. 2018.

[37] M. Goworko and J. Wytrębowicz, "A secure communication system for constrained IoT devices-experiences and recommendations," *Sensors*, vol. 21, no. 20, p. 6906, 2021.

[38] C. Gupta and G. Varshney, "An improved authentication scheme for BLE devices with no I/O capabilities," *Comput. Commun.*, vol. 200, pp. 42–53, Feb. 2023.

[39] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "PAuthKey: A pervasive authentication protocol and key establishment scheme for wireless sensor networks in distributed IoT applications," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 7, Jul. 2014, Art. no. 357430.

[40] D. A. Ha, K. T. Nguyen, and J. K. Zao, "Efficient authentication of resource-constrained IoT devices based on ECQV implicit certificates and datagram transport layer security protocol," in *Proc. 7th Symp. Inf. Commun. Technol.*, Dec. 2016, pp. 173–179.

[41] S. Sciancalepore, A. Capossele, G. Piro, G. Boggia, and G. Bianchi, "Key management protocol with implicit certificates for IoT systems," in *Proc. Workshop IoT challenges Mobile Ind. Syst.*, May 2015, pp. 37–42.

[42] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public key authentication and key agreement in IoT devices with minimal airtime consumption," *IEEE Embedded Syst. Lett.*, vol. 9, no. 1, pp. 1–4, Mar. 2017.

[43] 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.

[44] D.-H. Lee and I.-Y. Lee, "A lightweight authentication and key agreement schemes for IoT environments," *Sensors*, vol. 20, no. 18, p. 5350, Sep. 2020.

[45] S. M. Farooq, S. M. Hussain, S. Kiran, and T. S. Ustun, "Certificate based security mechanisms in vehicular ad-hoc networks based on IEC 61850 and IEEE WAVE standards," *Electronics*, vol. 8, no. 1, p. 96, 2016.

[46] H.-Y. Kwon and M.-K. Lee, "Fast verification of signatures with shared ECQV implicit certificates," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4680–4694, May 2019.

[47] P. Porambage, P. Kumar, A. Gurtov, M. Ylianttila, and E. Harjula, *Certificate Based Keying Scheme for DTLS Secured IoT*, Fremont, CA, USA: Internet-Draft, Draft-Pporamba-Dtls-Certkey-00, IETF Secretariat, Jun. 2013.

[48] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed IoT applications," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2014, pp. 2728–2733.

[49] M. A. Simplicio Jr., M. V. M. Silva, R. C. A. Alves, and T. K. C. Shibata, "Lightweight and escrow-less authenticated key agreement for the Internet of Things," *Comput. Commun.*, vol. 98, pp. 43–51, Jan. 2017.

[50] A. P. Sarr, P. Elbaz-Vincent, and J.-C. Bajard, "A new security model for authenticated key agreement," in *Proc. Int. Conf. Secur. Cryptogr. Netw.* Amalfi, Italy: Springer, 2010, pp. 219–234.

[51] Y. Al-Nidawi and M. Z. Abdullah, "Incorporating IEEE 1609.2–2016 standard with Internet of Things-based low power WAVE devices," *J. Southwest Jiaotong Univ.*, vol. 55, no. 1, pp. 1–12, 2020.

[52] P. S. L. M. Barreto, M. A. Simplicio, J. E. Ricardini, and H. K. Patil, "Schnorr-based implicit certification: Improving the security and efficiency of vehicular communications," *IEEE Trans. Comput.*, vol. 70, no. 3, pp. 393–399, Mar. 2021.

[53] M. Qi and J. Chen, "Two-pass privacy preserving authenticated key agreement scheme for smart grid," *IEEE Syst. J.*, vol. 15, no. 3, pp. 3201–3207, Sep. 2021.

[54] S. Garg, K. Kaur, G. Kaddoum, F. Gagnon, S. H. Ahmed, and D. N. K. Jayakody, "LiSA: A lightweight and secure authentication mechanism for smart metering infrastructure," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[55] Z. Chen, S. Chen, H. Xu, and B. Hu, "A security scheme of 5G ultradense network based on the implicit certificate," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–11, May 2018.

[56] J. Shen, S. Chang, Q. Liu, J. Shen, and Y. Ren, "Implicit authentication protocol and self-healing key management for WBANs," *Multimedia Tools Appl.*, vol. 77, no. 9, pp. 11381–11401, May 2018.

[57] G. S. Gaba, G. Kumar, H. Monga, T.-H. Kim, and P. Kumar, "Robust and lightweight mutual authentication scheme in distributed smart environments," *IEEE Access*, vol. 8, pp. 69722–69733, 2020.

[58] G. S. Gaba, G. Kumar, H. Monga, T.-H. Kim, M. Liyanage, and P. Kumar, "Robust and lightweight key exchange (LKE) protocol for industry 4.0," *IEEE Access*, vol. 8, pp. 132808–132824, 2020.

[59] G. Selander, F. Palombini, and K. Hartke, *Requirements for Coap End-to-End Security*. Fremont, CA, USA: Internet-Draft, Draft-Hartke-Core-E2E-Security-Reqs-03, Jul. 2017.

[60] M. Gunnarsson, J. Brorsson, F. Palombini, L. Seitz, and M. Tiloca, "Evaluating the performance of the OSCORE security protocol in constrained IoT environments," *Internet Things*, vol. 13, Mar. 2021, Art. no. 100333.

[61] K. Norrman, V. Sundararajan, and A. Bruni, "Formal analysis of EDHOC key establishment for constrained IoT devices," 2020, *arXiv:2007.11427*.

[62] A. Bruni, T. Sahl Jörgensen, T. Grønbech Petersen, and C. Schürmann, "Formal verification of ephemeral Diffie-Hellman over COSE (EDHOC)," in *Proc. Int. Conf. Res. Secur. Standardisation*, Darmstadt, Germany: Springer, 2018, pp. 21–36.

[63] M. Vucinic, G. Selander, J. P. Mattsson, and T. Watteyne, "Lightweight authenticated key exchange with EDHOC," *Computer*, vol. 55, no. 4, pp. 94–100, Apr. 2022.

[64] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, "Securing IIoT using defence-in-depth: Towards an end-to-end secure industry 4.0," *J. Manuf. Syst.*, vol. 57, pp. 367–378, Oct. 2020.

[65] A. Krontiris, "Evaluation of certificate enrollment over application layer security," M.S. thesis, School Elect. Eng. Comput. Sci., KTH Royal Inst. Technol., Stockholm, Sweden, 2018. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1255401/FULLTEXT01.pdf

[66] R. Sanchez-Iborra, J. Sánchez-Gómez, S. Pérez, P. Fernández, J. Santa, J. L. Hernández-Ramos, and A. F. Skarmeta, "Enhancing LoRaWAN security through a lightweight and authenticated key management approach," *Sensors*, vol. 18, no. 6, p. 1833, Jun. 2018.

[67] J. Hoglund and S. Raza, "LICE: Lightweight certificate enrollment for IoT using application layer security," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2021, pp. 19–28.

[68] L. P. Fraile, A. Fournaris, and C. Koulamas, "Design and performance evaluation of an embedded EDHOC module," in *Proc. 10th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2021, pp. 1–6.

[69] A. P. Haripriya and K. Kulothungan, "ECC based self-certified key management scheme for mutual authentication in Internet of Things," in *Proc. Int. Conf. Emerg. Technol. Trends (ICETT)*, Kollam, India, Oct. 2016, pp. 1–6.

[70] M. Ma, D. He, N. Kumar, K.-K. R. Choo, and J. Chen, "Certificate-less searchable public key encryption scheme for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 759–767, Feb. 2018.

[71] K. T. Nguyen, N. Oualha, and M. Laurent, "Novel lightweight signcryption-based key distribution mechanisms for MIKEY," in *Proc. Inf. Secur. Theory Pract.* Crete, Greece: Heraklion, 2016, pp. 19–34.

[72] A. Ghedini and V. Vasiliev. (Oct. 2018). *TLS Certificate Compression*. Internet-Draft, draft-IETF-TLS-Certificate-Compression-04. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-tls-certificate-compression-04

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

[74] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.

[75] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Innsbruck, Austria: Springer, 2001, pp. 453–474.

[76] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[77] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and secure session-key establishment scheme in smart home environments," *IEEE Sensors J.*, vol. 16, no. 1, pp. 254–264, Jan. 2016.

[78] M. Wazid, A. K. Das, S. Shetty, J. J. P. C. Rodrigues, and Y. Park, "LDAKM-EIoT: Lightweight device authentication and key management mechanism for edge-based IoT deployment," *Sensors*, vol. 19, no. 24, p. 5539, Dec. 2019.

[79] J. Kim, D. G. Duguma, S. Lee, B. Kim, J. Lim, and I. You, "Scrutinizing the vulnerability of ephemeral Diffie–Hellman over COSE (EDHOC) for IoT environment using formal approaches," *Mobile Inf. Syst.*, vol. 2021, pp. 1–18, Sep. 2021.
[80] A. Durand, P. Gremaud, J. Pasquier, and U. Gerber, "Trusted lightweight communication for IoT systems using hardware security," in *Proc. 9th Int. Conf. Internet Things*, Oct. 2019, pp. 1–4.
[81] M. Roetteler, M. Naehrig, K. M. Svore, and K. Lauter, "Quantum resource estimates for computing elliptic curve discrete logarithms," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Hong Kong: Springer, 2017, pp. 241–270.
[82] D. R. Brown, R. Gallant, and S. A. Vanstone, "Provably secure implicit certificate schemes," in *Proc. Int. Conf. Financial Cryptogr.* Cham, Switzerland: Springer, 2001, pp. 156–165.
[83] M. Malik, M. Dutta, and J. Granjal, "IoT-sentry: A cross-layer-based intrusion detection system in standardized Internet of Things," *IEEE Sensors J.*, vol. 21, no. 24, pp. 28066–28076, Dec. 2021.
[84] Kamaldeep, M. Malik, and M. Dutta, "Contiki-based mitigation of UDP flooding attacks in the Internet of Things," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 1296–1300.
[85] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Nov. 2004, pp. 455–462.
[86] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *Proc. 4th Workshop Embedded Networked Sensors*, Jun. 2007, pp. 28–32.
[87] T. Instruments. (2015). *CC2538 Powerful Wireless Microcontroller System-on-Chip for 2.4-GHz IEEE 802.15.4. 6LoWPAN, and ZigBee Applications*. [Online]. Available: http://www.ti.com/product/CC2538#
[88] Z. Siddiqui, J. Gao, and M. K. Khan, "An improved lightweight PUF–PKI digital certificate authentication scheme for the Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19744–19756, Oct. 2022.

**MANISHA MALIK** was born in Chandigarh, India, in 1990. She received the B.E. degree in computer science and engineering from Chitkara University, in 2012, and the M.E. degree in computer science and engineering from Panjab University, in 2015, where she is currently pursuing the Ph.D. degree in computer science engineering with the National Institute of Technical Teachers' Training and Research (NITTTR), Chandigarh.

During her postgraduation, she developed a keen interest in the area of cyber security and forensics and has delivered a number of lectures for teachers of polytechnic and engineering colleges. She is currently working on the security of the Internet of Things. Her research interests include communication and network security in the IoT. Her current research interests include the design and implementation of lightweight cryptographic primitives for next-generation IoT networks.

**KAMALDEEP** is currently pursuing the Ph.D. degree in computer science engineering with the National Institute of Technical Teachers' Training and Research (NITTTR), Panjab University, Chandigarh, India. He is currently working on the security of the Internet of Things. His research interests include communication and network security in the IoT. His current research interests include the design and implementation of lightweight security solutions for next-generation IoT networks.

**MAITREYEE DUTTA** is currently a Professor with the Computer Science and Engineering Department, NITTTR, Chandigarh, India. She has over 18 years of teaching experience. She has more than 100 research publications in reputed journals and conferences. She completed one sponsored research project, "Establishment of Cyber Security Lab" funded by the Ministry of IT, Government of India, New Delhi, amounting to Rs. 45.65 lac. Her research interests include digital signal processing, advanced computer architecture, data warehousing and mining, image processing, and the Internet of Things.

**JORGE GRANJAL** (Member, IEEE) received the Ph.D. degree, in 2014. He is currently an Assistant Professor with the Department of Informatics Engineering, Faculty of Science and Technology, University of Coimbra, Portugal, where he is also a Researcher with the Laboratory of Communication and Telematics, Centre for Informatics and Systems. His current research interests include computer networks, network security, and wireless sensor networks. He is also a member of ACM communications groups.