Received February 27, 2020, accepted April 1, 2020, date of publication April 8, 2020, date of current version April 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986480

Robust and Lightweight Mutual Authentication Scheme in Distributed Smart Environments

GURJOT SINGH GABA[1], (Member, IEEE), GULSHAN KUMAR[2], (Member, IEEE), HIMANSHU MONGA[3], (Member, IEEE), TAI-HOON KIM[4], (Member, IEEE), AND PARDEEP KUMAR[5], (Member, IEEE)

[1]Department of Electronics and Electrical Engineering, Lovely Professional University, Phagwara 144411, India
[2]Department of Computer Science and Engineering, Lovely Professional University, Phagwara 144411, India
[3]Department of Electronics and Communication Engineering, Jawahar Lal Nehru Government Engineering College, Mandi 175018, India
[4]School of Economics and Management, Beijing Jiaotong University, Beijing 100044, China
[5]Department of Computer Science, Swansea University, Swansea SA1 8EN, U.K.

Corresponding authors: Gulshan Kumar (gulshan3971@gmail.com) and Tai-Hoon Kim (taihoonn@daum.net)

ABSTRACT In the smart environments several smart devices are continuously working together to make individuals' lives more comfortable. Few of the examples are smart homes, smart buildings, smart airports, etc. These environments consist of many resource constrained heterogeneous entities which are interconnected, controlled, monitored and analyzed through the Internet. One of the most challenging tasks in a distributed smart environment is how to provide robust security to the resource constraint Internet-enabled devices. However, an authentication can play a major role ensuring that only authorized devices are being connected to the distributed smart environment applications. In this paper, we present a robust and lightweight mutual-authentication scheme (RLMA) for protecting distributed smart environments from unauthorized abuses. The proposed scheme uses implicit certificates and enables mutual authentication and key agreement between the smart devices in a smart environment. The RLMA not only resists to various attacks but it also achieves efficiency by reducing the computation and communication complexities. Moreover, both security analysis and performance evaluation prove the effectiveness of RLMA as compared to the state of the art schemes.

INDEX TERMS Authentication, elliptic curve Qu-Vanstone (ECQV), Internet of Things (IoT), implicit certificate, security.

I. INTRODUCTION

A smart environment is one of the emerging trends that allow people and objects to stay connected via the information and communication technologies. Smart environments (also knows as Internet of Things (IoT)) include smart homes [1], smart healthcare [2], smart car and cities [3] and many more. Note that smart environments/objects and IoT applications/objects are interchanged throughout the paper. In a recent research report, it is estimated that the "things" in connected smart environments to grow tremendously and is anticipated to reach up to billions of devices by 2025 [4].

In smart environment, IoT objects are computationally constraint devices, such as sensors, that can sense, compute, and extend connectivity between the last miles systems and users via the Internet in a ubiquitous manner. Fig. 1 shows a typical network of distributed smart environments, where several heterogeneous objects/nodes are installed to control and monitor the applications through the IoT cloud. All the sensors, objects or nodes collect data within their respective environments and send it to the cloud via the networking technologies, e.g., Zwave, ZigBee, and other IoT protocols. The collected data can be used for many purposes which depend on an application of interest e.g., health monitoring, data analytics for smart homes and cities [5], faults reporting in a flight system, leakage alarm of chemical in a factory etc.

As data from IoT objects is precious, inadequate security measures in IoT devices may invite various security threats to the applications. An unauthorized data access may cause harm to an application where the end-users are directly involved. An attacker may exploit vulnerabilities in IoT devices to collect data through eavesdropping, and may gain financial profit by selling collected data. Moreover, recently security researchers have pointed out several vulnerabilities

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

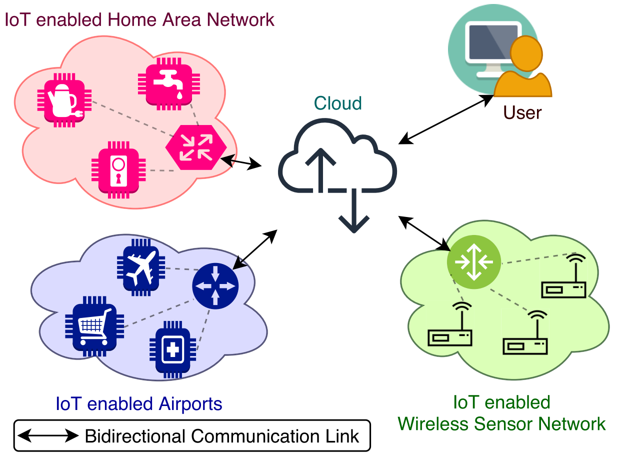


FIGURE 1. Distributed smart environments: IoT home area network; IoT-airport; IoT wireless sensor networks.

in smart cities technologies, few of them are attributed to *authentication flaws*, thus leaving IoT applications unsecured [6]. Ali-Awad have pointed out various vulnerabilities including *lack of sufficient authentication* in the smart home technologies, and have claimed that these vulnerabilities may pose many risks to the individuals [7].

In [8], security researchers have claimed that an attacker can access several home routers (i.e., 1,700 IoT devices) by exploiting a list of default login credentials on the IoT devices. Stellios et al. have shown verified cyberattacks on various IoT enabled domains, e.g., smart grid, intelligent transport network, industrial control system, medical IoT, and smart homes, etc. [9]. The authors have also claimed that the vulnerabilities (e.g., *design flaws in authentication mechanism*) in a smart light may lead to many threats in a smart home. Moreover, a Dyn Attack is carried out by the IoT Botnet named 'Mirai' which has seriously affected many of IoT devices as claimed in [10]. Nevertheless, such *lack of sufficient authentication* and/or *design flaws in authentication* mechanisms in IoT devices leads to sensitive information or data breach which may be misused. Resultant, security has been one of the main challenges in the success of distributed smart environments and applications.

A. RELATED WORK

In order to provide security in smart environments, several schemes have been proposed for smart environments or IoT networks. Each scheme has own merits and demerits. We have divided the related work into two parts namely: (1) asymmetric and (2) symmetric key based schemes.

1) ASYMMETRIC KEY BASED SCHEMES

In [11], Sciancalepore *et al.* have proposed a key management protocol for IoT networks (IoTnet). The scheme is based on the concept of Elliptic Curve Diffie Hellman (ECDH) and Elliptic Curve Qu-Vanstone (ECQV) implicit certificates. The authors have claimed that their proposed scheme is lightweight and secure against security attacks. However, the threat model designed in this paper does not include many popular attacks, such as impersonation, man-in-the-middle (MITM), etc. As a consequence, their scheme may be incompetent to protect against impersonation and MITM attacks. In addition, to execute the scheme (e.g., mutual authentication phase), the system incurs high time complexities. Therefore, this scheme may not be practical for resource-constrained devices.

In [12], Porambage *et al.* have introduced introduced a pair-wise key establishment scheme for wireless sensor networks (WSNs). The scheme uses ECC based implicit certificates for pair-wise key establishment. The authors first performed the bootstrapping followed by establishment of pair-wise key between nodes. However, physical capturing of a node may lead to disclosure of authentication key, which may emanate high security risks to other non-compromised IoT applications.

Sciancalepore *et al.* [13] have implemented a public key based authentication and key agreement protocol for IoTnet. The authors have employed ECDH with ECQV implicit certificates for achieving authentication. However, in their scheme, 2 different keys are needed to operate the protocol and the efficiency of key generation depends upon the key derivation function. Hence, malfunctioning of KDF may lead to connection abortion between entities. In their scheme, future keys are generated with the use of master key and any disclosure of related information may lead to loss of forward secrecy.

Patel *et al.* have described an authentication and access control protocol for IoT [14]. The scheme has used ECC based mutual authentication (EMA) and capability based access control (CBAC) for operation. Elliptic Curve Discrete Logarithm Problem (ECDLP) and ECDH are used for generating and sharing the common secret keys for authentication. In order to do this, the protocol utilized a plethora of cryptosystem operations which make it compute expensive.

Hossain *et al.* [15] have proposed an authentication technique, which is based on hardware and software co-verification for IoT. The authors have pointed out that since inception of IoT, targeting devices through cloning of hardware has become easy. To address cloning issue, they have proposed a physical unclonable function (PUF) based security protocol. The proof-of-concept is implemented on Contiki operating system. This method is claimed to be very first attempt to prevent the IoT devices from cloning and reprogramming attacks.

In [16], the authors proposed an authentication model for IoT enabled smart home. The authors claimed that their scheme is lightweight and secured against vulnerabilities. The basic idea of this scheme is to utilize the concept of temporary identity, keyed-hash chain mechanism and fog computing to achieve mutual authentication and identity assurance. Nevertheless, the scheme may fail to provide complete confidentiality and protection against DoS, known key attacks etc. Moreover, the communication and computation cost in [16] is a hindrance to its acceptance as an authentication model for resource limited devices of smart homes.

Dey and Hossain [17] developed a model of authentication for smart homes. The authors emphasized the need of a new security model for smart homes as distinct devices with different computational abilities work altogether. Their scheme exploited the Diffie Hellman Key Exchange (DHKE) protocol for achieving the mutual authentication and sharing of key. The security strength of their scheme is evaluated on protocol security analyzer tool *AVISPA* (automatic verification of internet protocols and applications). However, in spite of emphasizing on the computation and communication cost, the scheme still incurred high complexities, fails to ensure message freshness and may not withstand with known-key attack.

2) SYMMETRIC KEY BASED SCHEMES

Kumar *et al.* have suggested a lightweight session key establishment protocol for smart home environments [18]. A session key is produced using a short authentication token, which uses the silicon chip-identity. The authors claimed their scheme is efficient in terms of computation and communication costs and capable of protecting against attacks e.g., DoS, eavesdropping, masquerade, message forgery. In addition, their scheme satisfies the property of mutual authentication, session key establishment, confidentiality, integrity, and freshness. However, the scheme may not resist time synchronisation attacks. For instance, if clock loses synchronisation, then the scheme is vulnerable to replay attack. Moreover, anonymity and unlinkability issues are not addressed in the scheme [19].

In another research [20], the authors have elaborated that IoT networks have become a honeypot for attackers, thereby turning the privacy of the individuals under threat. The session key in their protocol is continuously renewed to prevent replay attacks. However, the authors have introduced several cryptosystem operations which made it bulky e.g. eight times hashing operations.

Gope and Sikdar [21] have not only emphasized on vulnerability of IoT devices at public places but also realized a need of robust IoT device authentication strategy. The authors proposed an authentication model using PUF to make IoT devices invulnerable to physical and cloning attacks. Authors claimed that their scheme is resilient to impersonation, achieves untraceability and also exhibit security properties e.g., mutual authentication, protection against physical attacks etc. However, their scheme may incur high computation requirements due to massive use of hash operations and high communication complexities. Thus, the scheme may not pertinent for the resource constrained and sensitive applications of IoT.

In summary, most of the schemes are insufficient as they are either not considering reasonable threat model that might cause some security issues or incurring high complexities at resource constrained nodes. Generally, the smart environments the objects triggers sensitive events where a packet being sent over the network; while the content of the packet is encrypted, knowing which device sends the packet reveals the device identity (i.e., privacy issue). In the related work except the scheme proposed in [20], a device therefore can easily be traced because most of the schemes do not use the concept of anonymity and/or untraceability to hide identities of nodes.

Thus, there is a necessity of an authentication scheme (while providing privacy) which can protect the distributed smart environments from unauthorized abuses with less complexity and more robustness against attacks.

B. OUR CONTRIBUTION

- In this paper, we propose a robust and lightweight mutual-authentication scheme (RLMA) for the distributed smart environments.
- To achieve the efficiency and lightweightness at resource constrained nodes, elliptic curve cryptography (ECC), implicit certificates, and symmetric encryption are used.
- The proposed scheme exhibits several security properties, such as mutual authentication, session key agreement, message freshness and anonymity and/or untraceability. Besides security properties, security analysis also shows that the proposed scheme is secure against many security attacks, e.g., replay, message modification, node compromise, key compromise, impersonation, known key, denial of service (DoS), and man-in-the-middle (MITM).
- Performance evaluation (including energy efficiency) and comparison demonstrates its high computational and communicational efficiency as compared to the state-of-the-art schemes.

II. SYSTEM MODEL, ADVERSARY MODEL AND SECURITY GOALS

A. SYSTEM MODEL

Fig. 2 depicts a high level system model in distributed smart environment. The system model mainly consists of following entities, such as IoT nodes, bi-directional communication channel, certification authority, etc.

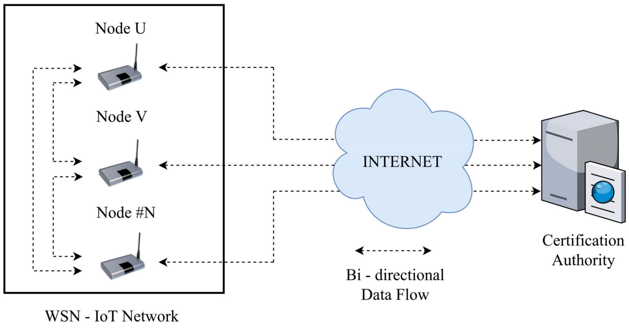


FIGURE 2. System model for authentication of IoT devices in smart homes.

1) *WSN-IoT Network*: In a smart network, the resource-constrained sensor nodes collect the data

(e.g., humidity, light, etc.) from their respective environments and send the data wirelessly to the sink node via utilizing low-powered technologies, e.g., ZigBee. More precisely, sensors data is easily available from anywhere in an ad-hoc manner. From the security perspective, the IoT nodes request security credentials from the certificate authority. These security credentials are later utilized to perform the mutual authentication.

2) *Certificate Authority (CA)*: The CA is a trusted entity, and is responsible for generating and distributing implicit certificates to the entities. Moreover, it is considered to be a tamper proof entity.

3) *Communication Link*: In the distributed IoT applications, IoT-nodes communicate with each other through bi-directional wireless technologies, such as Zigbee, Bluetooth, etc. In addition, the IoT nodes can communicate to CA either directly through GPRS/WiFi functionality or via gateway and cloud.

B. ADVERSARY MODEL

Following [22], consider a smart living environment where an attacker can have full control of the IoT network, and can modify, alter, drop and replay the wireless messages to mount different attacks. More precisely, an adversary can replay old messages with an intention to get unauthorized access between two smart devices. An attacker can perform the impersonation attack by creating the fake legitimate identity to steal critical information from entities. An attacker can disrupt the operations of the CA/IoT node through DoS and MITM attacks.

C. SECURITY GOALS

The proposed scheme provides following security goals. Note that the security goals are adopted from [20], [23].

1) *Mutual authentication and session key establishment*: In IoT networks, each node should perform the mutual authentication and verify the genuineness of the requesting node. After performing the mutual authentication, both the nodes should establish a session key to secure the further communication.

2) *Message integrity and freshness*: Message integrity ensures that no alteration has taken place during transit of messages. The received data should be fresh to avoid misinterpretation due to replaying of old messages.

3) *Lightweightness*: The devices in IoT networks are resource constrained, so overhead must be reduced during authentication and key establishment phase.

4) *Safeguard to popular attacks*: The proposed scheme must be resistant to popular attacks like impersonation, replay, node compromise, man-in-the-middle attack.

III. PROPOSED SCHEME

Assume a distributed smart environment, for instance a smart home (also known as a home area network (HAN)), which consists of several WSN-IoT nodes. These nodes collect data within a smart home and forward it to the IoT cloud and to the user. In order to provide security in such application, this section proposes a robust and lightweight authentication scheme. Note that in order to run the proposed scheme (i) all the entities are assumed to have identical cryptographic systems including encryption and hashing algorithms, (ii) each certificate has its lifetime, e.g., a year. The proposed scheme consists of three phases: system set-up phase, registration phase, and authentication and key exchange phase.

TABLE 1. Symbols and descriptions.

Notations	Descriptions
r_U, R_U	Random integer and E.C. point generated by node U
G, n	Base point generator and its order
r_{CA}	A random integer value generated by CA
$Cert_N$	Implicit certificate of Nth node
e, s	Hash value of Implicit Certificate and signature
d_{CA}, d_U, d_V	Private key of CA, Node U and V
LT	Lifetime of certificate
U, V, ID_{CA}	Identity of Node U, V and CA
Q_{CA}, Q_U, Q_V	Public key of CA, Node U and V
K_{UV}	Shared Secret key between Node U and V
n_U, n_V	A random positive integer generated by Node U & V
H, H_K, E_K	Hash, keyed Hash and Encryption with Kth key
Key(N), KS*n*	Symmetric keys used for encryption and decryption

A. SYSTEM SET-UP PHASE

In this phase, the CA off-line initializes the cryptographic mechanisms (such as, EC, n, point generator, hash function, symmetric encryption algorithm). Table 1 shows the notations and descriptions which are used throughout the paper. Note that the background on ECC is omitted intentionally due to the space limit. However, the interested may refer to [24] for ECC details. The CA generates own public key (Q_{CA}) and private key (d_{CA}). In addition, it generates a key pool of secret keys (e.g., $KS1, KS2, \ldots KSn$) for the HANs ($HAN1, HAN2, \ldots HANn$). It then publishes EC, n, point generator, Q_{CA}.

B. IoT-NODE REGISTRATION PHASE

In each home area network (HAN_i), an IoT node (e.g., node U) needs to be registered to the CA and obtains security credentials including a certificate and a key. The flow of registration phase is depicted in Fig. 3 and described as follows:

1) Initially, the node U generates a random number r_U and computes $R_U = r_U G$. It then computes $H1 = H(R_U || U)$ and $M1 = E_{Q_{CA}}[r_U, U]$. Finally, the node U sends a *cert-request* message $\{M1, H1\}$ to the CA.

2) Upon receiving *cert-request*, the CA decrypts $M1$ using d_{CA} and obtains r_U, U, and computes $R_U = r_U G$ and $H1'$ and verifies $H1' == H1$. It then generates a random number r_{CA} and implicit certificate $Cert_U = R_U + r_{CA} G$, computes $e = H(Cert_U)$, $s = er_{CA} + d_{CA}$ (mod n), $H2 = H(Cert_U, s, LT, KS1, R_U, U, ID_{CA})$, $Key = (r_U \oplus U \oplus ID_{CA})$ and $M2 = E_{Key}[Cert_U,$

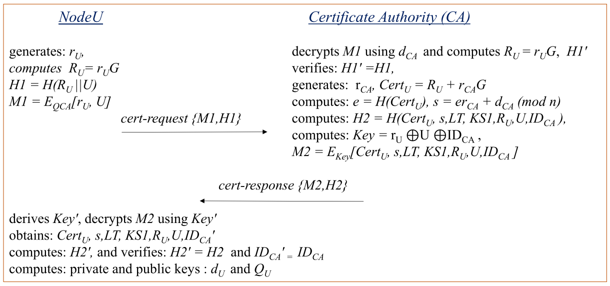

NodeU
Certificate Authority (CA)
generates: r_U,
computes $R_U = r_U G$
$H1 = H(R_U || U)$
$M1 = E_{QCA}[r_U, U]$
cert-request {M1,H1}
decrypts $M1$ using d_{CA} and computes $R_U = r_U G$, $H1'$
verifies: $H1' = H1$,
generates: r_{CA}, $Cert_U = R_U + r_{CA} G$
computes: $e = H(Cert_U)$, $s = er_{CA} + d_{CA}$ (mod n)
computes: $H2 = H(Cert_U, s, LT, KS1, R_U, U, ID_{CA})$,
computes: $Key = r_U \oplus U \oplus ID_{CA}$,
$M2 = E_{Key}[Cert_U, s, LT, KS1, R_U, U, ID_{CA}]$
cert-response {M2,H2}
derives Key', decrypts $M2$ using Key'
obtains: $Cert_U, s, LT, KS1, R_U, U, ID_{CA}'$
computes: $H2'$, and verifies: $H2' = H2$ and $ID_{CA}' = ID_{CA}$
computes: private and public keys : d_U and Q_U

FIGURE 3. Generation of implicit certificate.

$s, LT, KS1, R_U, U, ID_{CA}]$. Here, LT is the certificate lifetime of node U. Finally, the CA sends *cert-response* message $\{M2, H2\}$ to the node U.

3) The Node U derives $Key' = (r_U \oplus U \oplus ID_{CA})$ decrypts $M2$ using Key' and obtains $Cert_U$, s, LT, $KS1$, R_U, U, ID_{CA} and stores them. Now it computes $H2'$ and verifies $H2' == H2$. Upon successful verification, the node U computes own public and privacy keys from the received implicit certificate, as follows:
$d_U = er_U + s(\text{mod n})$
$Q_U = d_U G$
$= (er_U + s(\text{mod n}))\text{G}$
$= (er_U + er_{CA} \text{ (mod n)} + d_{CA} \text{ (mod n) (mod n)})\text{G}$
$= (er_U + er_{CA} \text{ (mod n)} + d_{CA} \text{ (mod n)})\text{G}$
$= e(r_U + r_{CA})G + d_{CA}G$
$= e(r_U G + r_{CA} G) + Q_{CA}$
$= e(R_U + r_{CA} G) + Q_{CA}$
$Q_U = eCert_U + Q_{CA}$

C. MUTUAL AUTHENTICATION AND KEY EXCHANGE PHASE

The flow of mutual authentication and pair-wise key establishment is shown in Fig. 4. This phase invokes when two nodes (node U and node V) want to negotiate a secret key within a HAN.

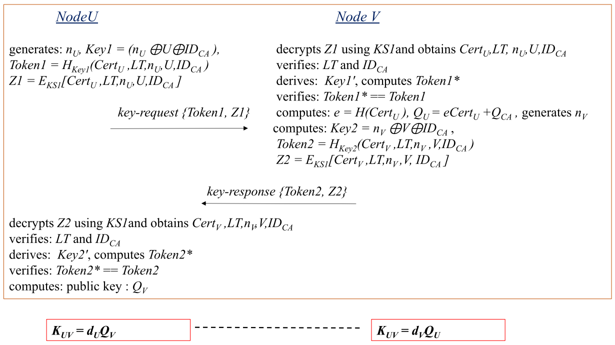

NodeU
Node V
generates: n_U, $Key1 = (n_U \oplus U \oplus ID_{CA})$,
$Token1 = H_{Key1}(Cert_U, LT, n_U, U, ID_{CA})$
$Z1 = E_{KS1}[Cert_U, LT, n_U, U, ID_{CA}]$
key-request {Token1, Z1}
decrypts $Z1$ using $KS1$and obtains $Cert_U, LT, n_U, U, ID_{CA}$
verifies: LT and ID_{CA}
derives: $Key1'$, computes $Token1*$
verifies: $Token1* == Token1$
computes: $e = H(Cert_U)$, $Q_U = eCert_U + Q_{CA}$, generates n_V
computes: $Key2 = n_V \oplus V \oplus ID_{CA}$,
$Token2 = H_{Key2}(Cert_V, LT, n_V, V, ID_{CA})$
$Z2 = E_{KS1}[Cert_V, LT, n_V, V, ID_{CA}]$
key-response {Token2, Z2}
decrypts $Z2$ using $KS1$and obtains $Cert_V, LT, n_V, V, ID_{CA}$
verifies: LT and ID_{CA}
derives: $Key2'$, computes $Token2*$
verifies: $Token2* == Token2$
computes: public key : Q_V
$K_{UV} = d_U Q_V$
$K_{UV} = d_V Q_U$

FIGURE 4. Mutual authentication and pair wise key establishment.

1) In the proposed scheme, the node U initiates the communication and it generates a random number n_U, $Key1 = (n_U \oplus U \oplus ID_{CA})$ and $Token1 = H_{Key1}(Cert_U, LT, n_U, U, ID_{CA})$. Here, H_{Key1} is a keyed-hash. Now, it computes $Z1 = E_{KS1}[Cert_U, LT, n_U, U, ID_{CA}]$, and sends *key-request* $\{Token1, Z1\}$ to the node V.

2) Upon receiving *key-request* message, the node V decrypts $Z1$ using $KS1$, obtains $Cert_U, LT, n_U, U, ID_{CA}$. It first verifies the lifetime LT of the certificate and ID_{CA} of the CA. If these conditions are true then it goes to the next step. In order to verify the authenticity of node U, now the node V derives $Key1'$ and computes $Token1^*$ and verifies $Token1^* == Token1$. If this condition fails then it aborts the session. Otherwise, the node V computes node U's public key as follows: $e = H(Cert_U)$ and $Q_U = eCert_U + Q_{CA}$. The proof is as same as shown in the IoT node registration phase (refer to step 3).

3) Now, the node V generates n_V, $Key2 = (n_V \oplus V \oplus ID_{CA})$ and $Token2 = H_{Key2}(Cert_V, LT, n_V, V, ID_{CA})$. Here, H_{Key2} is a keyed-hash. It computes $Z2 = E_{KS1}[Cert_V, LT, n_V, V, ID_{CA}]$. It sends *key-response* message $\{Token2, Z2\}$ to the node U. Finally, the node V computes a pair-wise key ($\mathbb{K}_{UV} = d_V Q_U$) using own private key d_V and U's public key Q_U.

4) Upon receiving *key-response* message, the node U decrypts $Z2$ using $KS1$, obtains $Cert_V, LT, n_V, V, ID_{CA}$. It first verifies the lifetime LT of the certificate and ID_{CA} of the CA. If these conditions are true then it goes to the next step. In order to verify the authenticity of node V, now the node U derives $Key2'$ and computes $Token2^*$ and verifies $Token2^* == Token2$. If fails then it aborts the session. Otherwise, the node U computes node V's public key (Q_V) and pair-wise key (K_{UV}) as follows:
$e = H(Cert_V)$ and $Q_V = d_V\text{G}$
$= (er_V + \text{s(mod n)})\text{G}$
$= (er_V + er_{CA} \text{ (mod n)} + d_{CA}\text{(mod n)(mod n)})\text{G}$
$= (er_V + er_{CA} \text{ (mod n)} + d_{CA} \text{ (mod n)})\text{G}$
$= er_V + r_{CA} \text{ (mod n))G} + d_{CA}G$
$= e(r_V G + r_{CA} G) + Q_{CA}$
$= e(r_V + r_{CA})G + Q_{CA}$
$Q_V = e(Cert_V) + Q_{CA}$
$\mathbb{K}_{UV} = d_U Q_V$
$Alternatively, \mathbb{K}_{UV} = d_U d_V G$

The pair-wise key is established successfully.

IV. SECURITY AND COMPARATIVE ANALYSIS

A. FORMAL ANALYSIS

Following [18], [20], we utilize AVISPA (automatic verification of internet protocols and applications) tool to evaluate the security strength of the proposed RLMA against the Dolev-Yao attack model. The AVISPA tool uses High Level Protocol Specification Language (HLPSL). The HLPSL script is further translated to Intermediate format (IF) using a HLPSL2IF translator [25]. The IF is feed to the backend, e.g., on-the fly model-checker (OFMC). For more details on the backends, the reader may refer to [25]. Finally the back-end generates the Output file (OF) concluding the protocol as safe or unsafe.

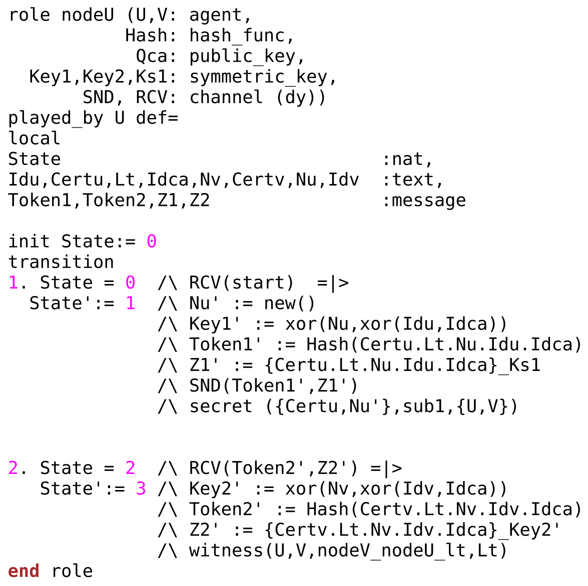
The HLPSL script of a protocol always begins with the basic roles. These roles are played by agents and contain local declarations. It defines the transitions when certain events are met and the corresponding changes in the states of the node. On the other hand, composition role have no transition section and executes various sessions in parallel. The last role i.e. environment role is very significant as it declares global constants and may composed of one or more sessions. The knowledge of the intruder (*i*) is also declared in this role and he may play some roles to camouflage profile of legitimate users. The channel (*dy*) uses the Dolev-Yao (DY) attack model for communication between the nodes.

To assess the strength of RLMA, the mutual authentication and pairwise key establishment phase is scripted in HLPSL and tested on AVISPA. Initially, basic roles of node U and V are defined which comprise of agent details (*U, V*), crypto-operations, local declarations (*Key1* etc.), channel (*dy*), initial state and transitions. Due to HLPSL keyword reservations, some of the parameters are represented with different acronyms in AVISPA as compared to acronyms used in algorithm. Those acronyms are *i* (intruder), ID_U (Identity of Node U), and ID_V (Identity of Node V). Node U acts as an initiator. After initialization at $State = 0$ [RCV(*start*)], it transitions to $State = 1$, where fresh nonce is prepared, $N_u' := new()$ followed by generation of $Token1' = \{Hash(Certu.Lt.Nu.Idu.Idca)\}$, and $Z1' = \{Certu.Lt.Nu.Idu.Idca\}\_Ks1$. Node U sends $Token1'$ & $Z1'$, $(SND(Token1', Z1'))$ to Node V for accomplishing mutual authentication and pair wise key establishment considering same channel (*dy*) properties. The goal predicates set by Node U is privacy of *Certu* & Nu' as shown in Fig. 5.

Node V receives the $Token1'$ and $Z1'$ in its initial state, $State = 1$ [RCV($Token1', Z1'$)] and extracts information during 2^{nd} *State*. Similarly, Node V sends $Token2', Z2'$ to Node U for successful accomplishment of mutual authentication and key establishment as shown in Fig. 6. The message confidentiality of $Z2'$ and authentication of $Token2'$ is modelled in terms of goals predicate, secrecy $\{Certv, Nv\}$ and witness $\{nodeU\_nodeV\_lt\}$ respectively. Witness ensures that the lifetime (LT) of the certificate (*Certu*) is verified before use.

Likewise, Node U recovers the information from the received message [RCV($Token2', Z2'$)]. Further, Node U at $State = 3$, verifies (witness(U, V, nodeV_nodeU_lt, Lt)) the validity of *Certv* before processing the request of pairwise key establishment.

Fig. 7 shows the composition of arguments used by agents, $nodeU(U, V, Hash, Qca, Key1, Key2, Ks1, SU, RU)$ $/\backslash nodeV(U, V, Hash, Qca, Key1, Key2, Ks1, SV, RV)$ These arguments are either sent or used by agents during the session. The most important is environment role because it constitutes of global constants declarations, defines intruder knowledge, elucify composition of sessions and set up goals of interest. As per the Dolev-Yao attack model, intruder is able to eavesdrop, intercept and analyze the information for e.g., $nodeU$, $nodeV$, h, $key1i$, $key2i$, $ks1i$, qca. The intruder knowledge is specified in environment and is used by security protocol analyzer tool (OFMC, CL-AtSe)

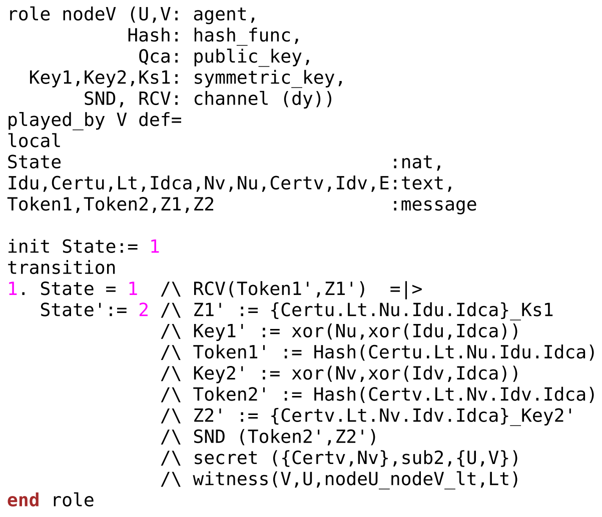


```
role nodeU (U,V: agent,
            Hash: hash_func,
             Qca: public_key,
  Key1,Key2,Ks1: symmetric_key,
        SND, RCV: channel (dy))
played_by U def=
local
State                                :nat,
Idu,Certu,Lt,Idca,Nv,Certv,Nu,Idv   :text,
Token1,Token2,Z1,Z2                  :message

init State:= 0
transition
1. State = 0  /\ RCV(start)  =|>
  State':= 1  /\ Nu' := new()
              /\ Key1' := xor(Nu,xor(Idu,Idca))
              /\ Token1' := Hash(Certu.Lt.Nu.Idu.Idca)
              /\ Z1' := {Certu.Lt.Nu.Idu.Idca}_Ks1
              /\ SND(Token1',Z1')
              /\ secret ({Certu,Nu'},sub1,{U,V})


2. State = 2  /\ RCV(Token2',Z2') =|>
   State':= 3 /\ Key2' := xor(Nv,xor(Idv,Idca))
              /\ Token2' := Hash(Certv.Lt.Nv.Idv.Idca)
              /\ Z2' := {Certv.Lt.Nv.Idv.Idca}_Key2'
              /\ witness(U,V,nodeV_nodeU_lt,Lt)
end role
```

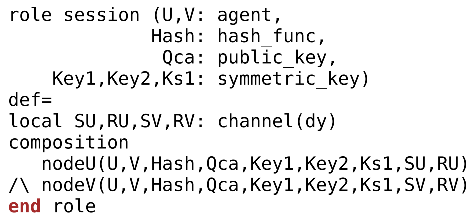
FIGURE 5. Specification of the node U role.



```
role nodeV (U,V: agent,
           Hash: hash_func,
            Qca: public_key,
  Key1,Key2,Ks1: symmetric_key,
       SND, RCV: channel (dy))
played_by V def=
local
State                                :nat,
Idu,Certu,Lt,Idca,Nv,Nu,Certv,Idv,E:text,
Token1,Token2,Z1,Z2                  :message

init State:= 1
transition
1. State = 1  /\ RCV(Token1',Z1')  =|>
   State':= 2 /\ Z1' := {Certu.Lt.Nu.Idu.Idca}_Ks1
              /\ Key1' := xor(Nu,xor(Idu,Idca))
              /\ Token1' := Hash(Certu.Lt.Nu.Idu.Idca)
              /\ Key2' := xor(Nv,xor(Idv,Idca))
              /\ Token2' := Hash(Certv.Lt.Nv.Idv.Idca)
              /\ Z2' := {Certv.Lt.Nv.Idv.Idca}_Key2'
              /\ SND (Token2',Z2')
              /\ secret ({Certv,Nv},sub2,{U,V})
              /\ witness(V,U,nodeU_nodeV_lt,Lt)
end role
```

FIGURE 6. Specification of the node V role.



```
role session (U,V: agent,
             Hash: hash_func,
              Qca: public_key,
    Key1,Key2,Ks1: symmetric_key)
def=
local SU,RU,SV,RV: channel(dy)
composition
   nodeU(U,V,Hash,Qca,Key1,Key2,Ks1,SU,RU)
/\ nodeV(U,V,Hash,Qca,Key1,Key2,Ks1,SV,RV)
end role
```

FIGURE 7. Specification of the session role.

during vulnerability evaluation of protocol against attacks. The next part of the environment role specifies the various sessions of message exchanges among nodes. Though it is expected to have sessions amongst legitimate agents only $(nodeU, nodeV, h, qca, key1, key2, ks1)$, but the possibility of intruder intervening in the session of legitimate nodes also prevails $(nodeU, i, h, qca, key1i, key2i, ks1i)$, $(i, nodeV, h, qca, key1i, key2i, ks1i)$. A total of four goals are specified out of which two are associated to secrecy and rest two corresponds to authentication as shown in Fig. 8. The description of the goals are:

```
role environment ()
def=
const nodeU,nodeV: agent,
qca: public_key,
key1,key2,ks1,key1i,key2i,ks1i: symmetric_key,
idu,certu,lt,idca,e,nv,nu,certv,idv: text,
h: hash_func,
nodeU_nodeV_lt,nodeV_nodeU_lt,sub1,sub2: protocol_id

intruder_knowledge={nodeU,nodeV,h,key1i,key2i,ks1i,qca}

composition
session(nodeU,nodeV,h,qca,key1,key2,ks1)
/\session(nodeU,i,h,qca,key1i,key2i,ks1i)
/\session(i,nodeV,h,qca,key1i,key2i,ks1i)
end role

goal
secrecy_of sub1
secrecy_of sub2
authentication_on nodeU_nodeV_lt
authentication_on nodeV_nodeU_lt
end goal

environment ()
```

FIGURE 8. Specification of the goal and environment for the proposed RLMA.

- Secrecy_of sub1 represents that $\{Cert_U, N_U\}$ are kept secret between node U and node V.
- Secrecy_of sub2 represents that $\{Cert_V, N_V\}$ are kept secret between node V and node U.
- Authentication_on nodeU_nodeV_lt states that the lifetime (i.e., LT) of certificate $\{Cert_U\}$ will be verified at the Node V.
- Authentication_on nodeV_nodeU_lt states that the lifetime (i.e., LT) of certificate $\{Cert_V\}$ will be verified at the Node U.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/IoT-HAN.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.04s
  visitedNodes: 19 nodes
  depth: 4 plies
```

```
% ATSE
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/results/IoT-HAN.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.02 seconds
  Computation: 0.00 seconds
```

FIGURE 9. RLMA results using OFMC and CL-AtSe backend.

The robustness of proposed protocol against attacks is verified using OFMC backend. Fig. 9 illustrates that RLMA can withstand against severe attacks and is reported safe to use in Internet based applications. Likewise OFMC, the CL-AtSe backend also reported safe. Hence, the attacks considered in the DY attack model cannot harm the RLMA security protocol.

B. INFORMAL PROOF

Following the attack model (as shown in Subsection II.B), this subsection deals with the understanding of how the designed protocol withstand against the attacks, such as modification of messages, known key attack, impersonation attack, replay, node compromise attack, etc.

Proposition: Secure against message modification.

Proof: Consider a communication between the node U and V, where an attacker intercepts $key - request$ $\{Token1, Z1\}$ and tries to fabricates $Z1$ to $Z1'$ using own key. Then it sends $Token1, Z1'$ to the node V. Since, $Z1'$ is computed via a wrong key (i.e., adversary key), it cannot be decrypted at the legal node V. In addition, as $Z1'$ cannot be decrypted, resultant $Token1$ cannot be verified. Note that here $Token1$ is a keyed-hash $(H_{Key1}(Cert_U, LT, n_U, U, ID_{CA}))$ and the key is only generated by the legitimate nodes using n_U, U, ID_{CA}. Hence, message modification cannot work from the node U to V communication. Likewise, the $key - response$ message is secure from the node V to U communication.

Proposition: RLMA is safe against impersonation attack.

Proof: Impersonation attacks can be prevented by properly authenticating the nodes e.g., Node U, computes the public key of Node V (Q_V), using $Cert_V$ and public key of CA, Q_{CA}. Likewise, the node V computes the Q_U. Both interested entities use their private keys (d_U, d_V) and opposite entity public Key (Q_U, Q_V) for generating the session key $\{Node\ U, K_{UV} = d_U Q_V;\ Node\ V, K_{UV} = d_V Q_U\}$. The process guarantees the execution of same secret keys at both entities when the certificates are issued by the valid CA. Node U, can trust the received certificate if it is encrypted by secret key $(Key = r_U \oplus U \oplus ID_{CA})$, which is exchanged between CA and node U. Therefore, impersonation attacks are difficult to conduct in RLMA as nodes among themselves use keyed-hash based Token approach for mutual authentication.

Proposition: RLMA is resistant to node compromise attack.

Proof: It is widely accepted that smart devices are difficult to prevent if they are not tamper proof [20]. Assume if the attacker captures the node and tries to collect the information. The information may constitute of a Certificate $(Cert_N)$. As, each certificate has its lifetime and unique nonce, the misuse of a compromised node can be prevented. In a smart home (i.e., HAN), as every node is embedded with unique id, certificate and KS1, thus compromising these parameter cannot compromise the security of non-compromised HANs. Therefore, the proposed scheme addresses security against node compromise attack to some level.

Proposition: Secure against Known key attack.

Proof: Known-key attack means that if a session is compromised then it should not compromise other session keys. In our scheme, suppose an attacker tries to generate a pair-wise session key (K_{UV}). However, this key does not help to deduce the key of other sessions since the pair-wise key is being computed over a nonce (N) and a high entropy random number (r). Note that these parameters are independent and different for each session. More precisely, a fresh random number guarantees that the certificate is unique for each node $\{(R_U = r_U G), (Cert_U = R_U + r_{CA} G)\}$ which further certify that generation of session key, $\{K_{UV} = d_U(e(Cert_V + Q_{CA})\}$ is independent and distinct for every session, thereby protecting the protocol against known key and ephemeral secret leakage attacks.

Proposition: Resilient to MITM attack.

Proof: The attacker node may have eavesdropped the messages exchanged between nodes and CA or between nodes. The attacker may have intentions to disrupt the system by retrieving the information as a middle agent and relay it after modifying. The attacker needs d_{CA} and E_{Key} to compute $\{E_{Key}, [Cert_U, s, LT, R_U, U]\}$, which he would never be able to get as d_{CA} is the private key of CA and never shared over the medium, thus attacker would not be able to modify the authenticator messages of RLMA. Moreover, the legitimate devices are mutually authenticated, $K_{UV} = d_V d_U G$ with the secret key (K_{UV}, never shared over medium), hence it would not be possible for an attacker to launch MITM.

Proposition: RLMA is resistant to Denial-of-service (DoS) and to replay attack.

Proof: Protecting a network from denial-of-service attacks is very hard as it can be mounted at every layer in a smart environment. However, a replay attack is one of them that can degrade the smart environment performance severely [26]. For instance, in the proposed scheme – suppose an adversary (A) eavesdrops and intercepts the valid messages $\{Token1, Z1\}$ and $\{Token2, Z2\}$ between the node U and V. Later adversary tries to replay $\{Token1, Z1\}$ to node V to keep the node V busy. However, this attempt fails as $Token1 = H_{Key1}\ (Cert_U, LT, n_U, U, ID_{CA})$ and $Z1 = E_{KS1}[Cert_U, LT, n_U, U, ID_{CA}]$ contains a fresh nonce (n_U), which is utilized to protect against replay attack. Similarly, the attacker intercepts the valid message $\{Token2, Z2\}$ and later replay's to node U. This attempt fails as $Token2 = H_{Key2}(Cert_V, LT, n_V, V,\ ID_{CA})$ and $Z2 = E_{KS1}[Cert_V, LT, n_V, V,\ ID_{CA}]$ utilize nonce (n_V). Hence, a replay attempt is detected very easily at node U. Moreover, nonce cannot be modified as it is shielded with *KS1* and keyed-hash with SHA-1. Therefore, the proposed scheme is safeguard to a replay attack, and to a DoS attack to some extend (i.e., a partial protection against DoS).

Proposition: RLMA attained Mutual Authentication.

Proof: The main purpose of mutual authentication is to cease the unauthorized access of intruders into the network. In our approach, mutual authentication is carried out between two nodes as follows:

- $\boldsymbol{U \rightarrow V}$: $Token1, Z1;$
  $Z1 = E_{KS1}[Cert_U, LT, n_U, U, ID_{CA}]$
  $Key1 = (n_U \oplus U \oplus ID_{CA})$
  $Token1 = H_{Key1}\ (Cert_U, LT, n_U, U, ID_{CA})$
- $\boldsymbol{U \leftarrow V}$: $Token2, Z2;$
  $Z2 = E_{KS1}[Cert_V, LT, n_V, V,\ ID_{CA}]$
  $Key2 = (n_V \oplus V \oplus ID_{CA})$
  $Token2 = H_{Key2}(Cert_V, LT, n_V, V, ID_{CA})$

Upon receiving $key - request$ from node U, the node V decrypts $Z1'$ and computes $Key1'$, $Token1^*$ and verifies $Token1^* == Token1$, for mutual authentication. Successful verification clearly indicate the legitimacy of node U. Similarly, node U verifies the authenticity of node V by evaluating $Token2$. As keyed-hash is a one way function, so $Token2$ cannot be reversed. Therefore, unauthorized nodes can never read the content of $H_{Key2}(Cert_V, LT, n_V, V, ID_{CA})$.

Proposition: Message freshness.

Proof: The proposed protocol ensures the presence of freshness component in messages through nonces (N) and ephemeral random numbers (r). The freshness not only protects against the replay and DoS attacks but also restricts the entities to prevent wastage of the resources in processing the old requests e.g., one or more components (n_U, n_V, r_U, R_U) of the freshness is added in every single exchange of message, e.g., $U \leftrightarrow CA$, $U \leftrightarrow V$,

- $\{E_{Q_{CA}}[r_U, U]\}||H(R_U||U)\ (\because R_U = r_U G)$
- $Z1 = E_{KS1}[Cert_U, LT, n_U, U, ID_{CA}]$
- $Token2 = H_{Key2}(Cert_V, LT, n_V, V, ID_{CA})$

The expressions prove the attainment of freshness property.

Proposition: Secure session key agreement.

Proof: The key agreement can be observed in the expression:

- Node U: $K_{UV} = d_U Q_V$; $Q_V = eCert_V + Q_{CA}$;
  $Z2 = E_{KS1}[Cert_V, LT, n_V, V,\ ID_{CA}]$
  $Cert_V = R_V + r_{CA} G$
- Node V: $K_{UV} = d_V Q_U$; $Q_U = eCert_U + Q_{CA}$;
  $Z1 = E_{KS1}[Cert_U, LT, n_U, U, ID_{CA}]$
  $Cert_U = R_U + r_{CA} G$

where LT ensures the expiry of the certificate and in turn session after a certain time period. Hence, new key will be formed for each session. Moreover it can be observed that the certificates are not sent in plaintext, thereby obtaining security of the parameters used for key establishment. In addition R_U and R_V, will be different for each session which guarantees a different K_{UV} for every session. In this way, a secure session key agreement is provided between the node U and V.

Proposition: RLMA procured the property of anonymity and/or untraceability.

Proof: Untraceability can be achieved by keeping the identity of the device hidden [27]. Attacker usually tries to track the device by eavesdropping of messages. In the RLMA, the ID's of the devices are not sent in plaintext, thereby it will be hard to trace the communicating parties, e.g., $Z1 = E_{KS1}[Cert_U, LT, n_U, U, ID_{CA}]$, and $Token1 = H_{Key1}(Cert_U, LT, n_U, U, ID_{CA})$, are shared by node U with node V. The information contains the ID's, which is encrypted to ensure that adversary could not find a way to decode the identities of the devices in communication. Therefore, the proposed protocol ensures the untraceability of entities.

TABLE 2. Analysis and comparison of protocols based on protection against attacks and security properties.

$\mathcal{A}$ & $\mathcal{SF}$	[13]	[14]	[15]	[16]	[17]	RLMA
$\mathcal{A}_1$			✓	✓	✓	✓
$\mathcal{A}_2$	✓		✓	✓	✓	✓
$\mathcal{A}_3$		✓	✓	✓	✓	✓
$\mathcal{A}_4$						✓
$\mathcal{A}_5$	✓	✓		✓	✓	✓
$\mathcal{A}_6$		$\mathcal{P}$			$\mathcal{P}$	$\mathcal{P}$
$\mathcal{A}_7$	✓	✓	✓	✓	✓	✓
$\mathcal{SF}_1$	✓	✓	✓	✓	✓	✓
$\mathcal{SF}_2$	✓	✓	✓	✓		✓
$\mathcal{SF}_3$	✓	✓	✓	✓	✓	✓
$\mathcal{SF}_4$				✓		✓
$\mathcal{SF}_5$		✓				✓

Acronyms: $\mathcal{A}$: Attacks, $\mathcal{SF}$: Security features, $\mathcal{A}_1$: Modification of messages, $\mathcal{A}_2$: Impersonation, $\mathcal{A}_3$: Node compromise, $\mathcal{A}_4$: Known key, $\mathcal{A}_5$: MITMA, $\mathcal{A}_6$: Denial of service, $\mathcal{A}_7$: Replay, $\mathcal{SF}_1$: Mutual authentication, $\mathcal{SF}_2$: Message freshness, $\mathcal{SF}_3$: Session key agreement, $\mathcal{SF}_4$: Anonymity and/or Untraceability, $\mathcal{SF}_5$: Confidentiality, $\mathcal{P}$: Partially protected

Finally, we summarize the security features of RLMA and compare its security with the state-of-the-art schemes. Table 2 shows that protocol proposed in [13], is prone to node compromise attack. More precisely, a single node compromises may lead to several attacks to the whole network. Other protocols (e.g., [14]–[17]) are subjected to known key attack. The schemes presented in [14] is vulnerable to impersonation attacks. In addition, it can be noticed that most of the state-of-the-art schemes do not consider the property of anonymity and/or untraceability, which is paramount requirement in many of smart environments use-cases where privacy is equally important, such as smart healthcare monitoring. In summary, it can be observed from Table 2 that the proposed scheme can provide more security features than the existing schemes.

V. PERFORMANCE ANALYSIS

A. EXPERIMENTAL SETTING

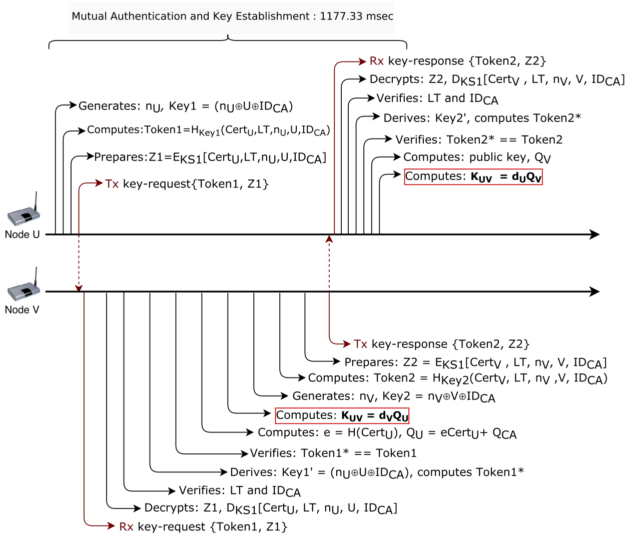
We experimented a prototype of RLMA scheme on a TelosB mote/device powered by TinyOS. Here, a TelosB mote equipped with a 16 bit processor (i.e., Texas Instruments MSP430 processor) that runs at a clock frequency of 8 MHz having 48 KB and 10 KB of ROM and RAM respectively [28]. We built a network of two TelosB nodes, i.e., node U and node V and a laptop (*Configuration*: Intel core i3-2310M processor with clock frequency and RAM of 2.10 GHz and 4 GB respectively). For the experimental purpose, we utilized a rich set of cryptographic libraries including AES (Advanced Encryption Standard), one-way hash function (i.e., SHA-1) and TinyECC [29]. In our experiment, we use the following message sizes, for instance IDs = 1 byte, hashing = 20 bytes, pseudo random number = 4 bytes, lifetime = 4 bytes, certificate = 16 bytes, s = 20 bytes, nonce = 4 bytes, and symmetric key size = 16 bytes. Therefore, the total length of messages in RLMA, i.e., *key-request* and *key-response* are 46 bytes each.

B. EVALUATION OF RLMA

We evaluated the performance of RLMA considering computation, communication and energy prices for the authentication and key establishment phase.

1) COMPUTATIONAL COSTS

As shown in Fig. 4, the Node U initiates the communication and sends a *Key-request* packet to the node V, which is further connected to the server, i.e., the laptop. Moreover, the node U receives (i.e., a *Key-response*) from the node V. Nevertheless, the total execution time taken by Node U is 1177.33 ms, for performing mutual authentication and key establishment with Node V, as shown in Fig. 10. We further evaluate the execution time for individual cryptographic operations. As as shown in Table 3, SHA-1, AES-encryption, AES-decryption, and multiplication take 112.32 ms, 16.38 ms, 178.10 ms, and 870.53 ms, respectively. This computation time can be reduced by using more high class smart devices, e.g., raspberry pi. However, in terms of the key establishment time, it is a well-suited time for the resource-constrained devices in smart environments.



Mutual Authentication and Key Establishment : 1177.33 msec
Rx key-response {Token2, Z2}
Decrypts: Z2, $D_{KS1}[Cert_V, LT, n_V, V, ID_{CA}]$
Verifies: LT and ID_{CA}
Derives: Key2', computes Token2*
Verifies: Token2* == Token2
Computes: public key, Q_V
Computes: $\mathbf{K_{UV} = d_U Q_V}$
Generates: n_U, Key1 = $(n_U \oplus U \oplus ID_{CA})$
Computes:Token1=$H_{Key1}(Cert_U,LT,n_U,U,ID_{CA})$
Prepares:Z1=$E_{KS1}[Cert_U,LT,n_U,U,ID_{CA}]$
Tx key-request{Token1, Z1}
Node U
Node V
Tx key-response {Token2, Z2}
Prepares: Z2 = $E_{KS1}[Cert_V, LT, n_V, V, ID_{CA}]$
Computes: Token2 = $H_{Key2}(Cert_V, LT, n_V, V, ID_{CA})$
Generates: n_V, Key2 = $n_V \oplus V \oplus ID_{CA}$
Computes: $\mathbf{K_{UV} = d_V Q_U}$
Computes: e = $H(Cert_U)$, $Q_U = eCert_U + Q_{CA}$
Verifies: Token1* == Token1
Derives: Key1' = $(n_U \oplus U \oplus ID_{CA})$, computes Token1*
Verifies: LT and ID_{CA}
Decrypts: Z1, $D_{KS1}[Cert_U, LT, n_U, U, ID_{CA}]$
Rx key-request {Token1, Z1}

FIGURE 10. Time elapsed for mutual authentication and key establishment.

We further evaluated energy-efficiency for the cryptographic operations as the smart objects are battery powered devices in many of use-cases. Following the formula (i.e., $\{E = V \times I \times t\}$) used in [14], we calculated the energy prices for our cryptographic operations. Here, V, I and t are the voltage, current and execution time, respectively. We have adopted the values of $V = 3V$ and $I = 1.8mA$ from [30]. The value of 't' for *RLMA* is measured from the experiment, as shown in Table 3. On a battery-powered smart

TABLE 3. Execution time and energy costs.

	O_1	O_2	O_3	O_4	O_T
Execution Time (ms)	112.32	16.38	178.10	870.53	1177.33
Energy Costs (mJ)	0.606	0.088	0.961	4.701	6.356

Acronyms: O_1: hash, O_2: encryption, O_3: decryption, O_4: multiplication, O_T: $O_1 + O_2 + O_3 + O_4$

device, the total energy required for the proposed RLMA is 6.356 *mJ*. More precisely, Table 3 also demonstrates the total energy incurred by *RLMA* for executing individual cryptographic operations, e.g., hash, encryption, decryption, and multiplication are 0.606 *mJ*, 0.088 *mJ*, 0.961 *mJ*, and 4.701 *mJ*, respectively.

TABLE 4. Computation cost comparisons.

T_{op}	[13]	[14]	[15]	[16]	[17]	RLMA
O_1	2H	1H	1H	8H	1H	2H
O_2	-	2MAC	6MAC	-	1MAC	-
O_3	2HMAC	-	-	-	-	-
O_4	-	-	-	-	1E + 2D	1E + 1D
O_5	-	-	2S	-	-	-
O_6	2M	-	1M	-	2M	1M

Acronyms: T_{op}: type of operation, O_1: hash (H), O_2: message authentication code (MAC), O_3: hash based message authentication code (HMAC), O_4: encryption (E) /decryption (D), O_5: signatures (S), O_6: multiplication (M)

In addition, a comparison of the computation cost among state-of-the-art schemes is presented in Table 4. Note that we simply chose asymmetric key based schemes for the comparison purposes. For the convenience of evaluation, following notations are being used:

- H: the time for performing a hash operation.
- MAC: the time for performing a MAC operation.
- HMAC: the time for performing a HMAC operation.
- E: the time for performing an encryption operation.
- D: the time for performing an decryption operation.
- S: the time for performing a signature operation.
- M: the time for performing a multiplication operation.

It can be seen from Table 4, the proposed RLMA makes use of 2 hash operations, 1 time encryption (E) & decryption (D), and 1 time multiplication operation for executing the mutual authentication and key establishment between the node U & V. Whereas the schemes proposed in [13], [15]–[17] makes use of excessive hash, MAC, HMAC, signatures, encryption, decryption and multiplication operations, which may not be efficient for the resource-hungry nodes. In addition, the scheme proposed in [14] incurred less computations than the proposed RLMA but does not provide adequate security services as shown in Table 2.

TABLE 5. Communication energy costs.

Cost		[13]	[14]	[15]	[16]	[17]	RLMA
Node U	S	624	-	768	1008	-	368
↕	R	624	-		368	-	368
Node V	T	1248	-	768	1376	-	736
$E.C.\ (mJ)$		0.898	-	0.587	1.023	-	0.562

Acronyms: Symbol (-): unspecified, S: bits sent by node U to node V, R: bits received by node U from node V, T: total exchange of bits, $E.C.$: energy consumed

2) COMMUNICATION COST

To investigate the communication cost, we have evaluated the energy required to transmit/receive the *key-request* and *key-response* messages between the node *U* and *V*. Following the scheme proposed in [18], transmitting and receiving a bit on TelosB consumes $0.72 \times 10^{-3} mJ$ and $0.81 \times 10^{-3} mJ$ of energy, respectively. Therefore, to send a *key-request* (i.e., 368 bits) to node V, the node U requires 0.264 *mJ*. Likewise, to receive a *key-response* (i.e., 368 bits) from node V, the node U needs 0.298 *mJ* energy. The total energy required for communication by RLMA is 0.562 *mJ* as shown in Table 5 and it can also be noticed that the proposed scheme incurred less communication energy than the other schemes.

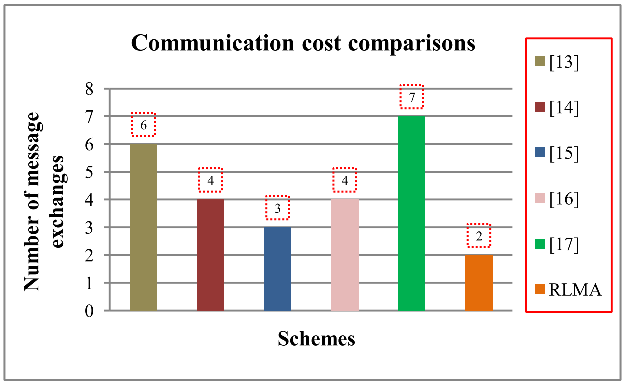


FIGURE 11. Communication cost comparisons in terms of the number of message exchanges.

Finally, from Fig.11, it is easy to visualize that a practical authentication and key establishment in the proposed scheme requires 2 message exchanges, whereas the schemes proposed in [14], [16] require 4 message exchanges and the scheme proposed in [13], [15], [17] needs 6, 3, and 7 message exchanges respectively. It should be noted that in real-world applications the actual number of message exchanges may vary if the packet transmission required multi-hop communications.

Considering computational, communication and node energy costs, it is clear that the proposed RLMA is efficient compared to other related schemes.

VI. CONCLUSION

In this paper, we proposed a robust and lightweight mutual-authentication (RLMA) scheme for the distributed smart environments. RLMA utilized implicit-certificate to achieve its simplicity and efficiency. The accomplishment of the security goals (i.e., secrecy, authentication, and message freshness) of the proposed scheme has been proven through formal (*AVISPA*) and informal analysis. We have demonstrated, through the performance evaluation, that RLMA is robust against attacks, consumes less computation and communication energy costs. All these properties make the RLMA suitable for the next generation smart home area networks. As future work, the authors plan to extend the proposed model to support authentication between users and devices in Internet of Things environment.

ACKNOWLEDGMENT

The authors would like to thank Editor-in-Chief, Editor and Reviewers for their valuable reviews.

REFERENCES

[1] W. M. Khan and I. A. Zualkernan, "SensePods: A ZigBee-based tangible smart home interface," *IEEE Trans. Consum. Electron.*, vol. 64, no. 2, pp. 145–152, May 2018.

[2] P. Sundaravadivel, K. Kesavan, L. Kesavan, S. P. Mohanty, and E. Kougianos, "Smart-log: A deep-learning based automated nutrition monitoring system in the IoT," *IEEE Trans. Consum. Electron.*, vol. 64, no. 3, pp. 390–398, Aug. 2018.

[3] D. Díaz-Sánchez, R. S. Sherratt, F. Almenarez, P. Arias, and A. Marín, "Secure store and forward proxy for dynamic IoT applications over M2M networks," *IEEE Trans. Consum. Electron.*, vol. 62, no. 4, pp. 389–397, Nov. 2016.

[4] Y.-T. Lee, W.-H. Hsiao, Y.-S. Lin, and S.-C.-T. Chou, "Privacy-preserving data analytics in cloud-based smart home with community hierarchy," *IEEE Trans. Consum. Electron.*, vol. 63, no. 2, pp. 200–207, May 2017.

[5] J. An, F. Le Gall, J. Kim, J. Yun, J. Hwang, M. Bauer, M. Zhao, and J. Song, "Toward global IoT-enabled smart cities interworking using adaptive semantic adapter," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5753–5765, Jun. 2019.

[6] (2018). *Smart Cities Around the World Were Exposed to Simple Hacks*. Accessed: Dec. 13, 2018. [Online]. Available: https://www.cnet.com/news/smart-cities-around-the-world-were-exposed-to-simple-hacks/

[7] B. Ali and A. Awad, "Cyber and physical security vulnerability assessment for IoT-based smart homes," *Sensors*, vol. 18, no. 3, p. 817, 2018.

[8] (2018). *Leak of 1, 700 Valid Passwords Could Make the IoT Mess Much Worse*. Accessed: Oct. 28, 2018. [Online]. Available: https://arstechnica.com/information-technology/2017/08/leak-of-1700-valid-passwords-could-make-the-iot-mess-much-worse

[9] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, and J. Lopez, "A survey of IoT-enabled cyberattacks: Assessing attack paths to critical infrastructures and services," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3453–3495, 4th Quart., 2018.

[10] (2018). *The 5 Worst Examples of IoT Hacking and Vulnerabilities in Recorded History*. Accessed: Aug. 4, 2018. [Online]. Available: https://www.iotforall.com/5-worst-iot-hacking-vulnerabilities

[11] 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.*, 2015, pp. 37–42.

[12] P. Porambage, P. Kumar, C. Schmitt, A. Gurtov, and M. Ylianttila, "Certificate-based pairwise key establishment protocol for wireless sensor networks," in *Proc. IEEE 16th Int. Conf. Comput. Sci. Eng.*, Dec. 2013, pp. 667–674.

[13] 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.

[14] S. Patel, D. R. Patel, and A. P. Navik, "Energy efficient integrated authentication and access control mechanisms for Internet of Things," in *Proc. Int. Conf. Internet Things Appl. (IOTA)*, Jan. 2016, pp. 304–309.

[15] M. Hossain, S. Noor, and R. Hasan, "HSC-IoT: A hardware and software co-verification based authentication scheme for Internet of Things," in *Proc. 5th IEEE Int. Conf. Mobile Cloud Comput., Services, Eng. (MobileCloud)*, Apr. 2017, pp. 109–116.

[16] M. Alshahrani and I. Traore, "Secure mutual authentication and automated access control for IoT smart home using cumulative keyed-hash chain," *J. Inf. Secur. Appl.*, vol. 45, pp. 156–175, Apr. 2019.

[17] S. Dey and A. Hossain, "Session-key establishment and authentication in a smart home network using public key cryptography," *IEEE Sensors Lett.*, vol. 3, no. 4, pp. 1–4, Apr. 2019.

[18] 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.

[19] C. T. R. Hager, S. F. Midkiff, J.-M. Park, and T. L. Martin, "Performance and energy efficiency of block ciphers in personal digital assistants," in *Proc. 3rd IEEE Int. Conf. Pervas. Comput. Commun.*, Mar. 2005, pp. 127–136.

[20] P. Kumar, A. Braeken, A. Gurtov, J. Iinatti, and P. H. Ha, "Anonymous secure framework in connected smart home environments," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 968–979, Apr. 2017.

[21] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 580–589, Feb. 2019.

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

[23] J. Kim, J. Baek, and T. Shon, "An efficient and scalable re-authentication protocol over wireless sensor network," *IEEE Trans. Consum. Electron.*, vol. 57, no. 2, pp. 516–522, May 2011.

[24] M. Campagna. (2013). *SEC4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)*. Accessed: Aug. 4, 2018. [Online]. Available: http://www.secg.org/sec4-1.0.pdf

[25] L. Viganò, "Automated security protocol analysis with the AVISPA tool," *Electron. Notes Theor. Comput. Sci.*, vol. 155, pp. 61–86, May 2006.

[26] N. Enneya, A. Baayer, and M. El Koutbi, "A dynamic timestamp discrepancy against replay attacks in MANET," in *Informatics Engineering and Information Science*, A. A. Manaf, S. Sahibuddin, R. Ahmad, S. M. Daud, and E. El-Qawasmeh, Eds. Berlin, Germany: Springer, 2011, pp. 479–489.

[27] D. Samfat, R. Molva, and N. Asokan, "Untraceability in mobile networks," in *Proc. 1st Annu. Int. Conf. Mobile Comput. Netw.*, 1995, pp. 26–36.

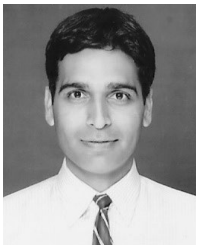
[28] (2018). *Telos Ultra Low Power IEEE 802.15.4 Compliant Wireless Sensor Module*. Accessed: Aug. 4, 2018. [Online]. Available: http://www2.ece.ohio-state.edu/ bibyk/ee582/telosMote.pdfl

[29] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2008, pp. 245–256.

[30] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, 2005, pp. 364–369.

GURJOT SINGH GABA (Member, IEEE) is currently pursuing the Ph.D. degree in electronics and electrical engineering with specialization in security of the Internet of Things (IoT) with Lovely Professional University (L.P.U.). He is currently working as an Assistant Professor with the School of Electronics and Electrical Engineering, L.P.U. His current research interests include security in cyber-physical systems, sensor networks, and the IoT.

GULSHAN KUMAR (Member, IEEE) received the Ph.D. degree in computer science from Lovely Professional University (L.P.U.), Punjab, India. He is currently working as an Assistant Dean and an Associate Professor with the Division of Research and Development, L.P.U. He has authored and coauthored more than 35 research articles including international journals (IEEE Internet of Things Journal, IEEE Access, the IEEE Sensors Journal, and IJDSN) and conferences. His current research interests include cyber physical systems, blockchain, edge, and cloud computing. He is a member of various technical organizations, such as ISCA and so on.



HIMANSHU MONGA (Member, IEEE) received the Ph.D. degree in optical and wireless networks from the Thapar Institute of Engineering and Technology, Punjab, India. He is currently the Dean Academics and a Professor with the Jawahar Lal Nehru Government Engineering College (Directorate of Technical Education, Government of Himachal Pradesh) and prior to that as the Director/Principal with Jan Nayak Chaudhary Devi Lal Lal Vidyapeeth, Sirsa. He has successfully completed six research grant projects along with consultancy projects. He has authored and coauthored more than 150 research articles in reputed conferences and journals. His current research interests include free space optics, 5G, and the Internet of Things. He is a member of ISTE.

TAI-HOON KIM (Member, IEEE) received the B.E. and M.E. degrees from Sungkyunkwan University, South Korea, and the joint Ph.D. degree from the University of Bristol, U.K., and the University of Tasmania, Australia. He is currently with Beijing Jiotong University, Beijing, China. His main research interests include security engineering for IT products, IT systems, development processes, and operational environments.

PARDEEP KUMAR (Member, IEEE) received the Ph.D. degree in computer science from Dongseo University, Busan, South Korea, in 2012. He is currently working with the Department of Computer Science, Swansea University, U.K. He worked with the Department of Computer Science, Oxford University, from 2016 to 2018. His current research interests include security in sensor networks, smart environments, cyber physical systems, and the Internet of Things.