

ABSTRACT The network-enabled smart objects are evolving everywhere in the modern era to solve numerous problems like real-world data collection from the environment, communication, analysis, and security. However, these objects (Internet-of-Things), in combination with low latency networks, are still not qualified for complex tasks and do not deliver efficient services due to the restriction of access and lack of secure authentication protocol. Because the data is collected by the embedded sensors inside the smart object in a real-time manner from the environment and communicated to the destination Centre (server) for intelligent decisions, are vulnerable to numerous threats, attention is required for the security and needs to be secure, as it transmits via an open network channel. This security issue can only be handled by designing a flawless, lightweight, and robust mutual authentication scheme. To do so, we have proposed a mutual authentication scheme using a simple hash cryptographic function, Elliptic Curve Cryptographic (ECC) technique, and XOR operations. The proposed scheme is lightweight, efficient, and effective in performance while offering secure transmission sessions among all the participants. The security of the proposed mechanism has been formally tested using GNY (Gong-Needham-Yahalon) logic, ProVerif2.03, and informally using propositions and realistic discussions. By comparing it with many of the existing authentication protocols, it has been demonstrated that our scheme is lightweight in terms of computation and communication metrics.

INDEX TERMS Authentication, cryptography, security, encryption, curve, gateway, verification, GNY.

I. INTRODUCTION

The smart object is placed in a building, bridge, at home etc. and then connected to the internet for providing services to users at any time and from any location. This technology benefits various application sectors, including healthcare systems [1], transportation surveillance, infrastructure inspection, and home monitoring. In 2020 the estimated IoT device industry was round about 25 billion USD, while the prediction for 2025 is approximately 6 trillion USD [2]. In the smart home scenario, the user can enjoy a high level of convenience using IoT devices, such as if the user desires to turn on/off lights, open/close doors, increase/decrease temperature, check surveillance, etc., user can easily control these smart objects (IoT) from any comfort zone using a portable device. However, smart objects are inefficient in terms of computing storage and battery consumption.

A robust authentication system is difficult to establish since embedded devices have limited storage. In order to deploy smart objects for smart home monitoring, vigorous authentication and lightweight security system are required, and acceptable carefullness for user ease in managing the system [3] is needed. So that malicious entities cannot gain illegal access to the data sent towards the smart object (IoT) [3]. As many homes are now-a-days internet-connected and visible to the public and malicious actions could easily compromise a user’s privacy. An intruder, for example, could eavesdrop on data exchanged between a user’s mobile and a smart object(s), and by collecting data repeatedly, they could then estimate when the homeowner wakes up, leaves for work, sleeps, and even travels. An attacker may prepare more serious attacks based on the eavesdropped data, such as burglary, kidnapping, and theft [4]. Majority of smart homes’ authentication techniques described in the literature are insecure against various attacks, including insider attacks, gateway node and smart object impersonation.
attacks, smart card theft attacks, and denial-of-service (DoS) attacks.

The lifestyle of a layman has changed due to the recent development of high-speed internet and the increased use of IoT in homes, cities, healthcare, industries, and security and rescue operations. IoT devices communicate with each other and with a centralized control system for different functions remotely. However, the security and privacy of IoT devices are still a big issue for researchers because of their heterogeneous nature. In this regard, different researchers have designed numerous authentication mechanisms using diverse cryptographic techniques. So far, Zhang et al. [5] proposed a simple hash and XOR-based authentication and key agreement scheme (AKA) for Internet-of-Drones (IoD) in which Unmanned Aerial Vehicle (UAV) has been used for search and rescue operations. However, their scheme is vulnerable as it suffers from physical capture, side-channel, and time synchronization attacks and has design flaws. Jan et al. [6], [7] proposed Hash Message Authentication Code Secure Hash Algorithm (HMACSHA1) and Public Key Infrastructure (PKI)-based authentication schemes for securing IoD in which drones can be used for different activities like wildlife surveillance, pipe-line inspection, sidewalk monitoring and agricultural spraying. However, after analysis, it has been demonstrated that their protocols’ computation and communication metrics are still not up to the mark, requiring more effort to make them lightweight.

Won et al. [8] presented three certificateless cryptographic schemes for real-world environmental monitoring using smart objects placed at different buildings, bridges and stations. They claimed that their first efficient certificateless signcryption tag key encapsulation mechanism (eCLSC-TKEM) is operational when data transmission is performed between smart objects and drones. Their [8] second certificateless multi-recipient encryption scheme (CL-MRES) is applicable when services are delivered from a drone to many smart objects. Their [8] last certificateless data aggregation (CLDA) protocol’s functioning when broadcasting is performed among many small objects towards a drone. However, these schemes are difficult to implement due to aggregate data verification instead of one-to-one. Wazid et al. [9] designed an Elliptic Curve Cryptographic (ECC) based authentication scheme for IoD deployment in the civilian domain, but their plan could also not be implemented practically due to less power in the smart objects and increased processing of data.

Furthermore, some available security protocols are weak against desynchronization and stolen verifier attacks. For example, in 2020, Hong et al. [10] demonstrated a security mechanism for reconnaissance and attacking drones equipped with many smart objects and are deployed in clusters for real-world environment monitoring. The [10] claimed that the airborne control and command platform (AC2P) establishes information broadcasting for reconnaissance and attacking drones, communicating real-time information collected from the environment to the centralized control system for an intelligent decision. After an extensive analysis of their [10] protocol, it has shown that their security mechanism is suffering from desynchronization, stolen-verifier, and privileged insider attacks. An improved protocol suite presented by Jan et al. [11] for IoD deployment military drones using the concept of pairing cryptography and identity authentication. They [11] have significantly tackled the weaknesses of [10] and claimed that a drone alone could not perform a complex tactical task; drones must be operationalized in many clusters subject to collaboration and coordination among them. However, the performance analysis of [11] still needs more effort for modification. Using the notion of symmetric-key cryptosystems, [12] introduced a lightweight authentication protocol for vehicle-to-infrastructure communication (V2I) in which successful authentication with the trusted third party (TA), the system distributed a secret key among the car and the roadside unit (RSU) or many smart objects to allow them to communicate with the server for real-time traffic monitoring. Although their [12] strategy used lightweight operations like hash and XOR function but is still insufficient for the restricted resources to make the system credible. Their scheme [12] also wastes a significant amount of storage space when storing the secret internal parameters among all the participants. Another disadvantage of their [12] method is that it does not accomplish reachability security features.

Similarly, the smart objects are also used to collect the most relevant environmental and climatic data for agricultural land to determine the amount of water pumped to the crop. Humidity, rain, soil moisture, soil pH, light, and temperature sensors are among the sensors used [13] for operating smart objects in these activities. Then, the system and analyzer servers transmit and analyze the said environmental and climatic data and forward the processed results to the responsible agriculture expert, who can then issue instantaneous commands to the water pumping actuators. Also, by using various machine learning methods, the system analyzer server can assess the arriving environmental and meteorological data to estimate the required water level. Again, most existing smart objects have used the Internet of Things (IoT) and Wireless Sensor Network (WSN) technologies in agriculture development to establish communication channels amongst stakeholders for onward decisions regarding productive crop output [14]. The said sensitive activities require a fast CPU and large memory space in the smart object/sensors, which is impossible for such a tiny device. A security mechanism with less computation/communication costs and robust security is the only solution for achieving the earlier goals related to agricultural land.

Finally, as stated above, most of the security protocols available in the literature are either unprotected in maintaining traceability and user anonymity or having high costs due to modular exponentiation. An insider threat is noted in these schemes in which an attacker uses the power analysis technique to steal the users’ confidential credentials, such as identification, private certificates, and password. While in an impersonation attack, the intruder may produce genuine
messages and send them to the appropriate smart object, causing the messages to be treated as legitimate but later on show severe damage to the whole system. An attacker could also utilize the extracted information from a stolen smart object to deduce the user’s secret credentials and later use it for malicious deeds. Therefore, LMAS-SHS has been presented here in this article to address these drawbacks. The key contributions are as under:

1. LMAS-SHS is an ECC-based lightweight security mechanism, which requires less power consumption due to reduced computation costs due to compressed message size.
2. LMAS-SHS is secure; the security of LMAS-SHS has been scrutinized on two methods:
   i. GNY Logic is used for checking the hash values and the security of random numbers exchanged among participants.
   ii. A programming verification toolkit, ProVerif 2.03, has been used for checking the session key secrecy, confidentiality, and reachability.
3. LMAS-SHS is lightweight, efficient, and effective in performance while offering secure transmission sessions among all the participants.
4. A pragmatic illustration for different attacks shows that LMAS-SHS resists all known attacks.

A. NETWORK MODEL
The proposed model for the network consisted of three main entities, i.e., mobile user (M), gateway (trusted entity), and smart object (having low power capability and less memory). The smart object sends real-time information to the gateway node. The mobile user is connected with the gateway node to disseminate real-time information fusion to the gateway node, as shown in Fig. 1. It is worth mentioning that the smart home must be equipped with smart objects for measuring different conditions like proximity, humidity, temperature, and the entire IoT installed devices like air conditions, fans, doors, locks, refrigerators, etc. Secondly, the service of the gateway node must be in control of home appliances on system architecture. The mobile user can use the managing service to regulate the operations of smart actuators connected to home equipment like lamps and fans. Finally, there must be a robust mutual authentication and lightweight cross-verification protocol that can access the entire entities having authorized people’s identification characteristics.

B. ADVERSARY MODEL
Modelling the role of attackers is essential in cyber defense since it helps to guarantee that security assessments are scientifically sound, especially for conceptual contributions that are difficult to test or where comprehensive testing is impossible. In a computer or networked system, an adversary model is a formalization of an attacker. Depending on how extensive this formalization is, the opponent might be an algorithm or a collection of affirmations about skills and discretions. This umbrella confines a variety of techniques in many domains of computer security. Therefore, keeping in view the adversary model, an adversary interacts with our smart home security architecture by representing themselves as a malicious user through gateway node in the following manner.

1. An adversary may extract stored data from gateway memory and use it to verify the secret credentials of a legitimate user for malicious deeds.
2. An adversary may alter, erase, upgrade, corrupt, or insert false information in the transmitted data over a public network channel.
3. An Adversary may replay, alter, or erase beneficial information exchanged between participants over a private channel.
4. An adversary may achieve the goal of entering the internal sensitive credential from a stolen smart object or mobile device.
5. An adversary might shape the memory of a stolen or misplaced smart object using reverse engineering approaches or key tags in offline mode, but not both simultaneously.

Therefore, to make the system efficient and effective, a cryptographic-based protocol is mandatory for adequately achieving integrity, confidentiality, authentication, and non-repudiation, ensuring perfect forward secrecy in unfavourable channels between participants. Regardless of who is involved, all legal parties in a session must trust one another to meet specific information security-related goals, which are the aim of this research.

C. ELLIPTIC CURVE CRYPTOGRAPHY (ECC)
Mathematicians and cryptographers introduced ECC for cryptosystem (a public key cryptographic method). It is lightweight based on an algebraic pattern of a curve over a finite field. It can deliver better security, faster computation, and network broadcasting. ECC is used for key controlling and authentication and can be defined in the following equation:

\[ y^2 = x^3 + ax + b \]  

i. Suppose P and Q are two points on eq: (1), then its addition can be represented as \( P + Q = R \) whereas \( P \neq Q \). In the curve, the lines via \( P \) and \( Q \) intersect at \( R \).
ii. Suppose \( Q = -P \), then \( P + Q = P + (-P) = P - P = 0 \), which means \( P \) and \( -P \) interest the curve at \( Q \), called the point of infinity.
iii. Suppose \( P \) is itself taken like \( P + Q = 2P = Q \), which means \( P \) intersects \( -Q \) and is reflected over the x-axis at \( Q \).
iv. Suppose a point is added to the curve is \( k \) means \( k \). \( P = P + P + P \ldots (P \times k) = k \in Z_p^*, \) in the cyclic group \( G \).
v. ECC is smaller, as RSA occupies 1024 bits of memory, while ECC is just 160 bits of space.
vi. An ECC-based key solution is almost impossible. It requires steps to be solved, whereas P means cardinality of the curve (a large prime number).

vii. Require less power consumption due to smaller computation and compressed message size; therefore, ECC is recommended for the resource-constrained environment.

II. PROPOSED SCHEME
We propose an ECC-based, lightweight, and secure mutual authentication scheme for smart home surveillance called LMAS-SHS. The LMAS-SHS consists of i) setup phase, ii) registration phase, and iii) mutual authentication phase. These phases are described under the following headings, while the different notations used for designing LMAS-SHS are shown in Table 1.

A. SETUP PHASE
The gateway node selects a curve \( E(F_q) \) of equation \( y^2 = x^3 + ax + b \) whereas \( a, b \in F_q \). Suppose all the three participants, i.e., mobile device (M), gateway node (GW), and smart object (SO), select their secret keys \( s \), compute public key \( K_M = s.P \), \( K_G = s.P \), and \( K_{SO} = s.P \), whereas P means a point in the curve.

B. REGISTRATION PHASE
This phase of LMAS-SHS is completed in the following two sub-phases:

1) MOBILE (M) REGISTRATION PHASE
This sub-phase is completed in the following set of computations:

\[
\begin{align*}
M & \text{generates unique identity ID}_M, \text{selects nonce } N_M, \text{computes PID}_M = ID_M || N_M, M = PID_M || K_M, \text{and transmits (ID}_M, ID_M) \text{over a secure channel.}
\end{align*}
\]

\[
\begin{align*}
G & \text{receives (ID}_M, ID_M) \text{and computes S1 = N_G \oplus N_M, S2 = S1 \oplus R_G, S3 = h(S1||ID_M), S4 = h(h(S2||ID_M)||N_M), S5 = h(ID_M||K_M||N_M), S6 = h(ID_G||K_G||N_G), S7 = R_G \oplus S5 \oplus S6, S8 = N_G \oplus S5 \oplus S6 \text{stores (ID}_M, h(S1||ID_M), S2) \text{and transmits (S1, S2, S3, S4, S5, S6, S7, S8, N_G) message back towards mobile (M) over a secure channel.}
\end{align*}
\]

MR3: The mobile (M) confirms N_G and computes S1′ = N_G \oplus N_M, S2′ = S1′ \oplus R_G, S5′ = h(ID_M||K_M||N_M), S6′ = h(ID_G||K_G||N_G), S7′ = N_G \oplus N_M and stores (ID_M, ID_G, h(S1′||ID_G), S2), as shown in module 1(a).

2) SMART OBJECT (SO) REGISTRATION PHASE
This sub-phase of LMAS-SHS competed in these steps:
S. U. Jan et al.: LMAS-SHS: Lightweight Mutual Authentication Scheme for Smart Home Surveillance

MOBILE (M)  |  GATEWAY NODE (GW)
---|---
Selects identity ID_M, nonce N_M
Compute: PID_M=ID_M∥N_M
I_M=PID_M∥K_M

|  | I_M, ID_M |
---|---|
Confirms: N_M
Generate Identity ID_G, Nonce N_G
Selects random number R_G
Computes: S_1=N_G⊕N_M

SOR1: The smart object (SO) first selects identity ID_SO, random nonce N_SO, and computes: PID_SO = ID_SO∥N_SO.
I_SO=PID_SO∥K_SO

|  | S_1, S_2, S_4, S_5, S_6, R_G |
---|---|
< |
Computes: S_1′=N_G⊕N_M
S_2′=S_1⊕R_G
S_4′=h(S_3∥ID_M)∥N_M
Verify S_4′?=S_4
S_5′=h(ID_M∥K_M∥N_M)
S_6′=h(ID_G∥K_G∥N_G) and stores: {ID_M, ID_G, h(S_1∥ID_M), S_2}

GATEWAY NODE (GW)

|  | I_SO, ID_SO |
---|---|
Confirms: N_SO
Generate Identity ID_G, Nonce N_G
Selects random number R_G
Computes: Z_1=N_G⊕N_SO
Z_2=Z_1⊕R_G and Z_3=h(Z_1∥ID_SO)∥N_SO
Z_4=h(Z_3∥ID_SO∥N_SO)
Z_5=h(ID_SO∥K_SO∥N_SO)
Z_6=h(ID_G∥K_G∥N_G) and Stores: {ID_SO, h(Z_1∥ID_SO), Z_2}

SMART OBJECT (SO)  |  GATEWAY NODE (GW)
---|---
Selects identity ID_SO, nonce N_SO
Compute: PID_SO=ID_SO∥N_SO
I_SO=PID_SO∥K_SO

|  | S_1, S_2, S_4, S_5, S_6, R_G |
---|---|
< |
Confirms: R_G
Computes: Z_1′=N_G⊕N_SO
Z_2′=Z_1′⊕R_G and Z_3′=h(Z_2′∥ID_SO)∥N_SO
Verify Z_3′?=Z_4
Z_5′=h(ID_SO∥K_SO∥N_SO)
Z_6′=h(ID_G∥K_G∥N_G) and Stores: {ID_SO, h(Z_1∥ID_SO), Z_2}

MODULE 1. (A) Mobile registration phase. (B) Smart Object (SO) registration phase.

SOR2: Upon receiving (I_M, ID_M) message, the gateway node (GW) confirms R_G, random nonce N_G, selects random number R_G, and computes: Z_1 = N_G ⊕ N_SO, Z_2 = Z_1 ⊕ R_G, Z_3 = h(Z_1∥ID_SO), Z_4 = h(h(Z_1∥ID_SO)∥N_SO), Z_5 = h(ID_SO∥K_SO∥N_SO), Z_6 = h(ID_G∥K_G∥N_G), stores (ID_SO, h(Z_1∥ID_SO), Z_2) in its memory and sends (S_1, S_2, S_4, S_5, S_6, N_G) towards the smart object (SO) over a secure path.

SOR3: The smart object (SO), when receiving (S_1, S_2, S_4, S_5, S_6, N_G) message from the gateway (GW), first confirms N_G and computes: S_1′ = N_G ⊕ N_SO, Z_2′ = Z_1 ⊕ R_G, Z_4′ = h(h(Z_2∥ID_SO)∥N_SO), verify Z_4′? = Z_4, Z_5′ = h(ID_SO∥K_SO∥N_SO), Z_6′ = h(ID_G∥K_G∥N_G), and finally stores (ID_SO, ID_G, h(Z_1∥ID_G), Z_2) values in the memory of smart object (SO) which will be used later in the authentication, as shown in module 1(b).
### MOBILE (M)  |  GATEWAY NODE (GW)  |  SMART OBJECT (SO)
---|---|---
| Provides Identity $\text{ID}_M$, nonce $\text{N}_M$, and Timestamp $\text{T}_1$ | |  
| Computes: $J_1 = h(\text{ID}_M || \text{N}_M) \oplus S_4$  
$J_2 = E_{\text{SM}}(J_1 || \text{N}_M || \text{T}_1 || S_3) \oplus \text{ID}_M$  
$J_1, J_2, \text{T}_1$ | Verify $\text{N}_M$, by decrypting $D_{\text{EG}}(J_2) = J_1 || \text{N}_M || \text{T}_1 || S_3 \oplus \text{ID}_M$  
Checks: $\text{T}_1 - \text{T}_2 \leq \Delta T$  
Select nonce $\text{N}_G$ and Timestamp $\text{T}_2$  
Computes:  
$M_1 = S_3 \oplus Z_3 \oplus \text{N}_G$  
$M_2 = \text{ID}_G \oplus \text{N}_G \oplus S_1$  
$M_3 = \text{ID}_M \oplus \text{N}_G \oplus Z_1$  
$M_4 = h(M_1 || \text{N}_G || \text{R}_G || \text{T}_2) \oplus S_1$  
$M_5 = h(M_1 || M_2 || \text{ID}_G || \text{T}_2) \oplus Z_1$  
$M_1, M_2, M_3, M_4, M_5$ | Verify $\text{N}_G$ from $h(M_1 || M_2 || \text{R}_G || \text{ID}_G)$  
Checks: $\text{T}_3 - \text{T}_2 \leq \Delta T$  
Extracts share secrets $h(S_1 || \text{ID}_G)$ and $h(Z_1 || \text{ID}_G)$ and take $\text{T}_4$  
Confirms it with the stored records  
Computes: $V_1 = h(S_1 || \text{ID}_G || \text{T}_4)$  
Extracts: $V_2 = h(S_1 || \text{ID}_G)$ from $M_2$  
$K_S = h(V_1 || V_2 || \text{N}_G)$  
$M_1, M_2, T_3, M_4, M_5, R_G$  
\[<\]
| Confirm $R_G$  
Extracts share secrets $h(S_1 || \text{ID}_G)$ and $h(Z_1 || \text{ID}_G)$  
$\text{T}_3 - \text{T}_2 \leq \Delta T$  
Confirms it with the stored values  
Computes: $V_1 = h(S_1 || \text{ID}_G || \text{T}_6)$  
Extracts: $V_2 = h(Z_1 || \text{ID}_G)$ from $M_2$  
$K_S = h(V_1 || V_2 || \text{N}_G)$  
$M_1, M_2, T_3, M_4, R_G$  
\[<\]
| Verify $R_G$  
Extracts secrets from $h(M_1 || M_2 || R_G || \text{ID}_G)$  
$\text{T}_7 - \text{T}_6 \leq \Delta T$  
Confirms it with the stored?  
Computes: $V_1 = h(S_1 || \text{ID}_M)$  
Extracts: $V_2 = h(S_1 || \text{ID}_M)$ from $M_2$  
$K_S = h(V_1 || V_2 || \text{N}_G)$

### MODULE 2. Mutual authentication phase.

### C. MUTUAL AUTHENTICATION PHASE

This phase of the scheme is completed in the following steps:

**A1:** The user, through a mobile device, provides identity $\text{ID}_M$, computes: $J_1 = \text{ID}_M || \text{N}_M$, $J_2 = J_1 \oplus \text{K}_M$ and transmits  
$\{S_4, S_5, Z_4, Z_5, J_1, J_2\}$, whereas $S_4, S_5, Z_4, Z_5$ are the stored values in registration phase.

**A2:** The gateway node first verifies $\text{N}_M$ and $\text{N}_G$ from $S_3$ and $Z_3$, computes:  
$M_1 = S_3 \oplus Z_3 \oplus \text{N}_G$  
$M_2 = \text{ID}_G \oplus \text{N}_G \oplus S_1$  
$M_3 = \text{ID}_M \oplus \text{N}_G \oplus Z_1$  
$M_4 = h(M_1 || \text{N}_G || \text{R}_G || \text{T}_2) \oplus S_1$  
$M_5 = h(M_1 || \text{ID}_G || \text{T}_2) \oplus Z_1$  
and transmits $\{M_1, M_2, M_3, M_4\}$ towards smart object over wireless channel.

**A3:** Here, first verify $\text{N}_G$ from $h(M_1 || \text{ID}_G)$, and extracts share secrets $h(S_1 || \text{ID}_G)$, confirms it with the stored values, computes: $V_1 = h(Z_1 || \text{ID}_SO)$ and extracts: $V_2 = h(S_1 || \text{ID}_G)$ from $M_2$ and compute session key $K_S = h(V_1 || V_2 || \text{N}_G)$. Transmits the message except $M_3$ back towards gateway node $\{M_1, M_2, M_4\}$.

**A4:** There first verify $\text{N}_G$ from $h(M_1 || \text{ID}_G || \text{ID}_G)$, extracts share secrets $h(S_1 || \text{ID}_G)$, confirms it with the stored, computes: $V_1 = h(S_1 || \text{ID}_G)$, again extracts: $V_2 = h(Z_1 || \text{ID}_SO)$ from $M_2$ and computes session key $K_S = h(V_1 || V_2 || \text{N}_G)$. Also, transmits the message except $M_3$ back towards gateway node $\{M_1, M_2, M_4\}$.

**A5:** The user upon receiving $\{M_1, M_2, M_4\}$ message, first verify $\text{N}_G$, extracts secrets from $h(M_1 || \text{ID}_G || \text{ID}_G)$, computes: $h(S_1 || \text{ID}_G)$, confirms it with the stored record, computes: $V_1 = h(S_1 || \text{ID}_M)$, extracts: $V_2 = h(S_1 || \text{ID}_G)$ from $M_2$ and calculates session key $K_S = h(V_1 || V_2 || \text{N}_G)$ as shown in module 2.
III. SECURITY ANALYSIS

This section analyzes the cryptographic protocol’s trust, freshness, and robustness and designates a protocol’s correctness. Also, it tells the readers why widespread authentication protocol attacks occur, and then it addresses the robustness based on trustworthiness and freshness. Therefore, keeping in view the goals mentioned above, we will scrutinize the security of LMAS-SHS formally using:

a) Gong-Needham-Yahalon called GNY logic [15], b) programming verification software toolkit ProVerif2.03; and informally using sensible explanation and arguments.

A. GNY LOGIC ANALYSIS

This section analyses our proposed scheme using GNY logic [15]. First, we introduce statements and formulae used in GNY logic; after that, we define the goals and assumptions of our proposed protocol. In the end, we verify the security of our proposed scheme using GNY logic [15].

1) GNY LOGIC FORMULAS AND STATEMENTS

The formulae used in GNY logic are as follows

a) \((W, N): the conjunction of two formulae W and N\)

b) \(\{X\}^{K}_K\) encryption/decryption with key K

c) \(h(W): A one-way function of W\)

d) \(*W*: W is not originated here

Here the W and N are the range over formulae while X and Y are the principals. The statements used in LMAS-SHS are as follow.

a) X \(\triangleleft W\): X is told formula W

b) P \(\triangleright W\): W possesses formula W

c) X|~ W: W once conveyed formula W

d) X \(\equiv \#(W): W\) believes that W is fresh

e) P | \(\equiv \phi(W): W\) believes that W is recognizable

f) X \(\equiv W \xrightarrow{S_K} Y\): P believes that S_K is a suitable secret key for X and Y

g) X \(\Rightarrow W\): P has jurisdiction over X

h) X \(\triangleleft W\): X is told that formula X which did not convey previously in the current run.

To apt the GNY logic with the LMAS-SHS, we make several notations’ changes as under:

(1) M \(\rightarrow GW\): (\{J_1, J_2, T_1\})

(2) GW \(\rightarrow SO\): (\{M_1, T_3, M_2, M_3, M_4, M_5\})

(3) SO \(\rightarrow GW\): (\{M_1, M_2, T_4, M_3, M_5, R_G\})

(4) GW \(\rightarrow M\): (\{M_1, M_2, T_4, M_5, R_G\})

2) GNY LOGIC ASSUMPTION FOR LMAS-SHS

A1: \(M \ni \#(N_M)\)  A10: \(GW \ni \#(N_M)\)  A19: \(SO \ni \#(N_M)\)

A2: \(M \ni \#(t_1)\)  A11: \(GW \ni \#(t_1)\)  A20: \(SO \ni \#(t_1)\)

A3: \(M \ni \#(N_G)\)  A12: \(GW \ni \#(N_G)\)  A21: \(SO \ni \#(N_G)\)

A4: \(M \ni \#(T_2)\)  A13: \(GW \ni \#(T_2)\)  A22: \(SO \ni \#(T_2)\)

A5: \(M \ni \#(N_SO)\)  A14: \(GW \ni \#(N_SO)\)  A23: \(SO \ni \#(N_SO)\)

A6: \(M \ni \#(R_G)\)  A15: \(GW \ni \#(R_G)\)  A24: \(SO \ni \#(R_G)\)

A7: \(M \ni ID_M\)  A16: \(GW \ni ID_M\)  A25: \(SO \ni ID_M\)

A8: \(M \ni ID_G\)  A17: \(GW \ni ID_G\)  A26: \(SO \ni ID_G\)

A9: \(M \ni ID_{SO}\)  A18: \(GW \ni ID_{SO}\)  A27: \(SO \ni ID_{S}\)

It means the nonce generated by M, said that N_M is fresh (#N_M) and possesses K_{MN}, S_4, T_1, and ID_M. Similarly, the N_G and R_G are fresh (#R_G), (#N_G) and possesses S_1, T_2, ID_M and ID_G. Also, the K_S generated on smart object (SO) side is fresh (#K_S) and possesses ID_{SO}, T_4, Z_1, ID_G, and M_2.

\[ M \ni N_M, M \ni (\#N_M), M \ni K_{KM}, M \ni S_4, \]
\[ M \ni T_1, M \ni ID_M \]
\[ GW \ni T_2, GW \ni S_1, GW \ni ID_M, GW \ni N_G, \]
\[ GW \ni R_G, GW \ni ID_G \]
\[ SO \ni ID_{SO}, SO \ni Z_1, SO \ni T_4, SO \ni ID_G, SO \ni M_2 \]

3) GNY LOGIC ANALYSIS FOR LMAS-SHS

We changed the notation of LMAS-SHS to fill in GNY logic.

a) \(M \ni GW\) \(\ni \{ID_M|\#N_G\ni S_4\}\)

\[ M \ni \{T_1\} \]

b) \(GW \ni SO\) \(\ni \{(M_1, T_3, M_2, M_3, M_4, M_5)\}\)

c) \(SO \ni GW\) \(\ni \{S_2 \oplus Z_3 \ni N_G\}\)

\[ SO \ni \{ID_G \ni N_G \ni S_1\} \]

\[ SO \ni \{h(M_1|\#N_G|ID_G|T_2) \ni S_1\} \]

\[ SO \ni \{h(M_1|\#N_G|ID_G|T_2) \ni Z_1\} \]

\[ SO \ni \{N_G\} \]

d) \(GW \ni M\) \(\ni \{(S_1 \ni Z_3 \ni N_G)\}\)

\[ GW \ni \{ID_G \ni N_G \ni S_1\} \]

\[ GW \ni \{h(M_1|\#N_G|ID_G|T_2) \ni S_1\} \]

\[ GW \ni \{N_G\} \]

The exchange among the participants in the protocol has completely performed secretly like M believes that M \(\ni GW\) exchanges ID_M \((\ni N_S)\ni S_4 \ni J_1, J_2, T_1\) secretly. In the same way the exchange of credentials between GW \(\ni SO\) are exchanged much secretly like (M_1|\#M_2|\#N_G|\#R_G) \ni S_1 (M_1|\#M_2|\#N_G|\#R_G) \ni Z_1, (M_1|\#M_2|\#N_G|\#R_G) \ni T_2) \ni ID_M, and M_1, M_2, M_3, M_4, M_5, T_2. Among SO \(\ni GW\) and GW \(\ni M\), the credentials exchanged are M_1, M_2, M_3, M_4, M_5, T_5, R_G and M_1, M_2, M_4, M_7, T_7, R_G performed in a secure manner.

\[ M \ni GW \ni \{ID_M|\#N_M\ni S_4\}\]

\[ M \ni GW \ni \{T_1\}\]

\[ GW \ni GW \ni \{(M_1|\#M_2|\#N_G|\#R_G)\ni Z_1\}\]

\[ GW \ni GW \ni \{h(M_1|\#M_2|\#N_G|ID_G|T_2)\ni S_1\}\]

\[ GW \ni GW \ni \{h(M_1|\#M_2|\#N_G|ID_G|T_2)\ni Z_1\}\]

\[ GW \ni GW \ni \{N_G\}\]

4) GNY LOGIC GOALS FOR LMAS-SHS

1) Goal 1: \(GW \ni N_M\)

2) Goal 2: \(SO \ni N_G\)
3) Goal 3: GW| ≡ NG
4) Goal 4: M| ≡ NG
According to assumptions A1 and A3, we get
\[
M \ni NG, M \ni NM \\
M \ni S1
\]
According to assumptions A7 and A1, we get
\[
M \ni S1, M \ni IDM, M \ni NM \\
M \ni S4
\]
According to assumptions A7 and A1, we get
\[
M \ni IDM, M \ni NM, M \ni S4 \\
M \ni J1
\]
According to assumptions A16, A10, and GNY postulates, T1 and P1
\[
GW \ni J1 \\
GW \ni J1
\]
According to assumptions A7, A1, A10, we obtain
\[
M \ni J1, M \ni NM, M \ni S5, M \ni t \\
M \ni J2
\]
According to assumptions A10, A11, and GNY postulates, T1 and P1
\[
GW \ni J2 \\
GW \ni J2
\]
According to assumptions A11, A11, A16, and GNY postulates, T1 and P1
\[
GW \ni J1, GW \ni NM, GW \ni t, GW \ni S5, GW \ni IDM \\
GW \ni J2
\]
The user sees the identity of gateway along with the message communicated with key KM, and then gateway believes its identity and message received.
\[
M \ni IDM, M \ni [J1, J2, T1]KM \\
GW \ni IDM, GW \ni [J1, J2, T1]KM
\]
The GW now possesses IDM and [J1, J2, T1]KM, so, we can represent it by the following form
\[
GW \ni IDM, GW \ni RG \\
GW \ni IDM, GW \ni (IDM||RG)
\]
In order to achieve Goal 1, the GW recognizes J2 and applies R1
The SO sees part of the message and computes. According to assumptions A19 and A21, we obtain
\[
SO \ni NG, SO \ni NM \\
SO \ni S1
\]
According to assumptions, A25 and GNY postulate T1 and P1
\[
SO \ni S1, SO \ni IDM \\
SO \ni S3
\]
According to assumptions A21 and A23, we obtain
\[
SO \ni NG, SO \ni NSO \\
SO \ni Z1
\]
According to A25 and GNY postulates, T1 and P1
\[
SO \ni Z1, SO \ni IDSO \\
SO \ni Z2
\]
According to assumption A25 and GNY T1 and P1, we get
\[
SO \ni S3, SO \ni Z1, SO \ni NG \\
SO \ni M1
\]
According to assumptions A26, A21, and GNY T1 and P1, we obtain
\[
SO \ni IDG, SO \ni NG, SO \ni S1 \\
SO \ni M2 \\
SO \ni M3
\]
If the smart object (SO) sees the identity of mobile along with the message communicated, then smart object (SO) can definitely believe its own identity and the message broadcasted.
\[
SO \ni IDM, SO \ni [M1, M2, M4, M7, T7, RG]RG \\
SO \ni IDSO, SO \ni [M1, M2, M4, M7, T7, RG]RG
\]
The smart object possesses IDSO and M1, M2, M4, M7, T7, RG, so, we can represent it by the following form:
\[
GW \ni IDSO, GW \ni RG \\
SO \ni IDSO, SO \ni (IDSO||RG)
\]
In order to achieve Goal 2, the SO recognizes M2 and applies R1.
The SO sees some part of the message and computes
\[
SO \ni Z1, SO \ni IDSO \\
SO \ni V1
\]
\[
SO \ni S1, SO \ni IDSO, SO \ni S1 \\
SO \ni V2
\]
According to assumption A21, and GNY T1, P1, we obtain
\[
SO \ni V1, SO \ni V2, SO \ni NG \\
SO \ni KS
\]
According to assumption A12, and GNY T1, P1, we obtain
\[
GW \ni KS \\
GW \ni KS
\]
If the gateway node GW sees its own identity IDG, along with the message communicated, then GW believes IDG and transmitted message.
\[
GW \ni IDG, SO \ni [(M1||M2||NG||RG)\oplus Z1]R5 \\
GW \ni IDG, SO \ni [(M1||M2||NG||RG)\oplus Z1]R5
\]
The mobile GW possesses the identity of smart object IDSO then we can represent it by the following way on holding message (M1||M2||NG||RG)\oplus Z1.
\[
GW \ni IDSO \\
GW \ni IDSO, SO \ni [M1||M2||NG||RG)\oplus Z1]R5
\]
In order to achieve Goal 3, the GW recognizes $K_S$ and applies $R1$

$$GW\equiv \phi(V_1), \ GW\equiv \phi(V_2), \ GW\equiv \phi(N_G) , \ GW\equiv \phi(K_S)$$

Goal 3 Achieved

According to assumption A9, and GNY T1, P1, we obtain

$$GW \equiv V_1, \ GW \equiv V_2, \ GW \equiv N_G$$

$$GW \equiv K_S$$

According to assumption A3, and GNY T1, P1, we obtain

$$M \leftarrow K_S$$

Finally, if the mobile M sees the identity of smart object $ID_{SO}$, along with the message communicated, then mobile believes $ID_{SO}$ and transmitted message.

$$M \leftarrow ID_{SO}, \ M \leftarrow (M_1 \parallel M_2 \parallel N_G \parallel R_G \parallel T_2) \oplus ID_M$$

The mobile M possesses the identity of smart object $ID_{SO}$ then we can represent it by the following way on holding message $(M_1 \parallel M_2 \parallel N_G \parallel R_G \parallel T_2) \oplus ID_M$.

$$M \equiv ID_{SO}$$

$$M \equiv ID_{SO}, \ M \equiv (M_1 \parallel M_2 \parallel N_G \parallel R_G \parallel T_2) \oplus ID_M$$

In order to achieve Goal 4, the M recognizes $K_S$ and applies $R1$

$$M \equiv \phi(V_1), \ M \equiv \phi(V_2), \ M \equiv \phi(N_G) , \ M \equiv \phi(K_S)$$

Goal 4 Achieved

Therefore, all the three entities securely transmit the different credentials with each other and its honesty is confirmed by applying GNY logic.

B. PROVERIF.03 SIMULATION

The issues of confidentiality, reachability, integrity, and the secrecy of all the credentials (secret keys, identity, random numbers, parameters, and timestamp) have been tested by using well-known software verification toolkit ProVerif2.03 [20]. The code and result are shown in appendix of the article.

C. INFORMAL SECURITY ANALYSIS

The security analysis of LMAS-SHS will informally be demonstrated as under:

$$GW\equiv \phi(J_1), \ GW \equiv \phi(N_M), \ GW \equiv \phi(t), \ GW \equiv \phi(S_\parallel), \ GW \equiv \phi(ID_M)$$

Goal 1 Achieved

$$SO\equiv \phi(M_1), \ SO \equiv \phi(M_2), \ SO \equiv \phi(N_G), \ SO \equiv \phi(ID_G), \ SO \equiv \phi(T), \ SO \equiv \phi(Z_1)$$

Goal 2 Achieved

1) RESISTS INSIDER ATTACK

As we do not prefer a storage table inside the gateway for secret credentials storage, an attacker cannot access the internal secrets. Similarly, the identity is secretly transmitted over the public network channel if, for example, an adversary copies a message from the open line due to random numbers different for each session, collision-free hash function, XOR operations, and nonce; they cannot theft any credentials to reach internally and hijack the system. Therefore, LMAS-SHS is free of insider threats.

2) WITHSTANDS TRACEABILITY ATTACK

The mobile, smart object and gateway node have first extracted nonce randomly and concatenated it with other credentials to make it secure. Due to this, an attacker cannot trace different sessions of the same system at different times. Therefore, LMAS-SHS is safe against traceability attacks.

3) RESISTS DOS ATTACK

Confirmation steps have been introduced in each round trip of the proposed protocol, i.e., confirms $h(Z_1 \parallel ID_M), h(Z_1 \parallel ID_G)$, and $h(Z_1 \parallel ID_{SO})$. The checks can, in turn, mitigate denial of device attacks on the system. Similarly, after receiving the message by any participant, it first verifies the nonce received in, if successful, onward processing start, else, considered DoS attack from a potential attacker. Therefore, LMAS-SHS resists the DoS attack.

4) RESISTS REPLAY ATTACK

After receiving a message $\{J_1, J_2, T_1\}$ by a gateway, it first checks the timestamp $(T_1)$ with the current timestamp $(T_c)$ $T_1 - T_c \leq \Delta T$, and if it is out of the pre-defined time threshold, the gateway considers it a potential replay attack, discard the message and does not proceed for further computation. Furthermore, if an attacker diverts $(M_1, M_2, T_3, M_4, M_5, N_G)$ message from the open network channel, the smart object also checks the timestamp $(T_5)$ with its current time $(T_c)$ to withstand with replay attack and vice versa, therefore, LMAS-SHS is safe against replay attacks.

5) WITHSTANDS MAN-IN-THE-MIDDLE ATTACK

Due to randomness in each transmitted message, the nonce is different for different sessions. Also, in the random extraction of large prime numbers, the adversary, if, for example, injects something new into the public network channel, they cannot do so due to no knowledge of $N_M, N_G$, and $N_{SO}$. Therefore, LMAS-SHS is robust against a man-in-the-middle attack.
6) FREE FROM DE-SYNCHRONIZATION ATTACK
There is no need to update parameters on the SO or M sides in LMAS-SHS. In contrast, in the case of some changes, each participant validates it correspondingly. Therefore, the SO, GW and M do not require synchronization properties in LMAS-SHS.

7) SUPPORT ANONYMITY
In LMAS-SHS, the GW uses anonymous identities for both SO, M, and itself, which means that the identity is untraceable. Also, two sessions are not stated with the same credentials due to random selection of the nonce (NG, NM, NSO) and timestamps. Therefore, LMAS-SHS supports the anonymity feature.

8) RESISTS STOLEN VERIFIER ATTACK
LMAS-SHS does not store any random number. The verification and validation of every credential do not require any database or tables on the M/SO side. Thus, if an adversary tries to reach internally to access the necessary certificates, they cannot masquerade as M or SO to mislead the GW in the authentication process. Therefore, LMAS-SHS resists stolen verifier attacks.

9) FREE FROM MASQUERADE ATTACKS
If an attacker uses a fake identity of any participant (IDM, IDSO, or IDW) and tries to gain authorized access of the public channel, due to nonce, random numbers, and timestamp, any illegal attempt of an adversary will be denied by the system because of multiple checks in different round trip of the protocol. Therefore, LMAS-SHS is free from masquerade attacks.

IV. PERFORMANCE ANALYSIS
This paper section can be examined by considering computation, communication, and comparison analysis. These different performance metrics are as under:

A. COMPUTATION COSTS
In this section, we calculate the LMAS-SHS computation cost and compare it with state of the art scheme. We refer to the work of [16], [22] and [23] for a detailed comparison. The execution time of ECC point multiplication $T_M$ is ($\sim$7.3529), hash function $T_h$ is ($\sim$0.0004), fuzzy extractor $T_F$ is ($\sim$7.3529), and encryption/decryption $T_s$ is ($\sim$0.1303). The calculation of LMAS-SHS computation cost and detailed comparison with other protocols are shown in Table 2 and graphically in Fig. 2.

B. COMMUNICATION COST
We calculated the LMAS-SHS communication cost in this section and compared it with other schemes. We consider the work done in [16], [22] and [23] that defined encryption/decryption, ECC point; random number, hash function, timestamp and identity are {256, 320, 160, 160, 32, and 128}.

TABLE 2. Computation cost analysis.

| Schemes | Total cost | Cost in ms |
|---------|------------|------------|
| [16]    | 12$T_T$+14$T_M$+TR | 90.2983 |
| [17]    | 15$T_h$    | 4.0168ms |
| [18]    | 16$T_T$+3$T_m$ | 22.0651ms |
| [19]    | 25$T_T$+4$T_r$+1$T_k$ | 7.8841ms |
| [20]    | 21$T_T$+8$T_r$+1$T_k$ | 2.281ms |
| LMAS-SHS | 19$T_T$+4$T_r$ | 0.5288ms |

TABLE 3. Communication cost analysis.

| Schemes | Number of Messages Communicated | Communication Costs in bits |
|---------|---------------------------------|----------------------------|
| [16]    | 3                               | 2528                      |
| [17]    | 3                               | 1484                      |
| [18]    | 4                               | 1920                      |
| [19]    | 4                               | 3360                      |
| [20]    | 4                               | 2592                      |
| LMAS-SHS | 4                               | 1424                      |
TABLE 4. Security and functionalities comparison.

| Security features                  | Schemes | [16] | [17] | [18] | [19] | [20] | LMAS-SHS |
|------------------------------------|---------|------|------|------|------|------|---------|
| Impersonation attack               | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| Replay attack                      | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| Session key disclosure attack      | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| MITM attack                        | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| Password guessing attack           | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| Insider attack                     | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| Stolen device attack               | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| Session temporary attack           | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| Desynchronization attack           | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| Mutual authentication              | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| untraceability                     | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| Anonymity                          | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |
| Perfect forward secrecy            | ✓       | ✓    | ✓    | ✓    | ✓    | ✓    | ✓       |

and \( M_5 = \{ M_1, M_2, M_4, N_G \} \) as shown in Table 3 and Fig. 3.

C. FUNCTIONALITIES COMPARISON

Suppose we compare LMAS-SHS with different protocols like [16]–[20] in terms of varying security functionalities/attacks. In that case, the proposed scheme resists all known attacks and is better than these schemes, as shown in Table 4. Whereas ✓ means Secure × means insecure.

V. CONCLUSION

In this paper, we have presented a lightweight mutual authentication scheme for smart home surveillance called LMAS-SHS. In LMAS-SHS, different smart objects are fixed in various places for real-time information exchange towards the nearest gateway node regarding the health of the infrastructure and monitoring of the home. The ECC technique is used to design LMAS-SHS, which is lightweight and provides robust security. The security of LMAS-SHS has formally been verified using GNY logic and ProVerif2.03 and informally using pragmatic explanation. The performance analysis of LMAS-SHS is measured by considering computation and communication metrics. Consequently, the researchers have proved that LMAS-SHS is robust, lightweight, free from insider, stolen verifier attacks, and has no design flaw. So this is a more efficient protocol and provides security for smart home surveillance and can also be utilized for infrastructure inspection of a big city. Also, if this protocol is implemented for disaster purposes, it can quickly communicate the health of the whole infrastructure with the centralized server. Also, it can install in drone technology equipped with different smart objects near workers and employees for effective monitoring them to increase their work output and as an effective tool in the real estate business.

We plan to use Elliptic Curve Digital Signature Algorithm (ECDSA) to design a security mechanism for proving the transitional authentication of users in the teleworking environment. The security analysis of the said ECDSA-based transitional authentication scheme shall be manipulated via AVISPA (Automatic Validation of Internet Security Protocol Authentication). It is to mention that its performance will distress due to exponentiation in the discrete logarithmic function without affecting security.

APPENDIX

To check whether the session shared key is confidentiality communicated and whether it is reachable to each peer in an authentic manner, we used a verification toolkit ProVerif2.03. The result shows that an attacker at any stage could not crack the secrecy, confidentiality, and reachability of the session key.

```plaintext
(*=======CONSTANT AND VARIABLES=======*)
free NM:bitstring.
free MG:bitstring.
free NSO:bitstring.
free RG:bitstring.
free IDM:bitstring.
free IDG:bitstring.
free IDSO:bitstring.
free KM:bitstring.
free KG:bitstring.
free KSO:bitstring.

(*=======FUNCTIONS=======*)
fun h(bitstring):bitstring.
fun con(bitstring,bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.

(*=======EQUATIONS=======*)
equation forall a:bitstring, b:bitstring; XOR(XOR(a, b), b) = a.

(*=======EVENTS=======*)
event start_M(bitstring).
event end_M(bitstring).
event start_GW(bitstring).
event end_GW(bitstring).
event start_SO(bitstring).
event end_SO(bitstring).

(*=======QUERYES=======*)
query KS:bitstring [private].
query attacker(KS).
query IDM:bitstring; inj-event(end_M(IDM)) ==> inj-event(start_M(IDM)).
query IDG:bitstring; inj-event(end_GW(IDG)) ==> inj-event(start_GW(IDG)).
```
inj-event(start_M(IDM)).
query ISO:bitstring; inj-event(end_SO(IDSO)) ==> inj-event(start_SO(IDSO)).

(*=======RESULT GENERATED=======*)
secure against any attack upon running the code. Its
The result indicated that the session key is much more
0.

(********RESULT GENERATED*********)

REFERENCES
[1] S. U. Jan, S. Ali, I. A. Abbasi, M. A. A. Mosleh, A. Alsanad, and
H. Khattak, “Secure patient authentication framework in the healthcare
system using wireless medical sensor networks,” J. Healthcare Eng.,
vol. 2021, pp. 1–20, Jul. 2021.

[2] Samsung Smarthings Developers Documentation. Accessed: Jan. 17, 2019. [Online]. Available: https://smarthings.developer.samsung.
com/blog/en-us/2019/01/17/Shape-the-Future-of-IoT-with-SmartThings
SAFE ULLAH JAN received the Ph.D. degree in network security from the University of Malakand, in 2021. He is currently working as a Lecturer in computer science at the Higher Education, Achieves & Libraries Department, Government of Khyber Pakhtunkhwa, Pakistan. He is also working as a Controller of Examinations for nine B.S. Disciplines with the Government College Wari (Dir Upper), a far-flung remote area of the province where most of the youngsters have no access to universities/institutions for higher education. Furthermore, he has conducted research in many areas, including green computing, distributed computing, privacy-preserving parallel computation, and drone security and authentication. He has published over 25 research papers in prestigious conferences and journals and written an introductory book in computer science for beginners. The Government of Khyber Pakhtunkhwa, awarded “Best Teacher Award” for the year 2019–2020 out of 11000 College Teachers in 309 public sector colleges in the province.

IRSHAD AHMED ABBASI (Member, IEEE) received the M.S. degree in computer science from COMSATS University Islamabad, Pakistan, and the Ph.D. degree in computer science from Universiti Malaysia Sarawak, Malaysia. He worked as a Senior Lecturer at King Khalid University, Saudi Arabia, from 2011 to 2015. He is currently working as an Assistant Professor with the Computer Science Department, University of Bisha, Saudi Arabia. He was declared as the Best Teacher at the Faculty of Science and Arts Belgarn, University of Bisha, in 2016. He has over 12 years of research and teaching experience. He is the author of many articles published in top quality journals. His research interests include VANETS, MANETs, FANETs, mobile computing, the IoT, cloud computing, cybersecurity, soft computing, and drone security and authentication. He has received multiple awards, scholarships, and research grants. He is serving as an editor. He is also acting as a reviewer for many well reputed peer-reviewed international journals and conferences.

MOHAMMED A. ALQARNI received the bachelor’s degree in computers from King Khalid University, Saudi Arabia, in 2008, the M.Sc. degree in computational sciences from Laurentian University, Sudbury, ON, Canada, in 2012, and the Ph.D. degree in computer science from McMaster University, Ontario, Canada, in 2016. He is currently an Associate Professor at the College of Computer Science and Engineering, University of Jeddah, Saudi Arabia.