Secure Lightweight Authentication for Multi User IoT Environment

Chintan Patel, Member, IEEE

Abstract— The Internet of Things (IoT) is giving a boost to a plethora of new opportunities for the robust and sustainable deployment of cyber-physical systems. The cornerstone of any IoT system is the sensing devices. These sensing devices have considerable resource constraints, including insufficient battery capacity, CPU capability, and physical security. Because of such resource constraints, designing lightweight cryptographic protocols is an opportunity. Remote User Authentication ensures that two parties establish a secure and durable session key. This study presents a lightweight and safe authentication strategy for the user-gateway (U-GW) IoT network model. The proposed system is designed leveraging Elliptic Curve Cryptography (ECC). We undertake a formal security analysis with both the Automated Validation of Internet Security Protocols (AVISPA) and Burrows–Abadi–Needham (BAN) logic tools and an informal security assessment with the Deley-Yao channel. We use publish/subscribe based Message Queuing Telemetry Transport (MQTT) protocol for communication. Additionally, the performance analysis and comparison of security features show that the proposed scheme is resilient to well-known cryptographic threats.

Index Terms—Generic IoT, Home Area Network, ECC, Authentication, MQTT

I. INTRODUCTION

Expansion of internet-based services significantly impacts routine life of the people. Due to its ambient and easy-to-use support services, the adoption of Internet of Things (IoT) based services (i.e., smart electricity supply, smart agricultural, smart amenity distribution, industry 4.0, smart healthcare, smart home) has soared in recent years.

According to recent Gartner forecasts, more than 25.44 billion IoT gadgets will be incorporated into the cyber world’s working environment by 2030. IoT-based smart health care connects healthcare entities such as patients, doctors, hospital administrators, drug suppliers, ambulances, and pharmacists using a network of smart healthcare devices. Doctors can obtain real-time health data from patients via wearable health devices using an intelligent patient monitoring system.

IoT based smart grid provides a intelligent system for electricity generation, electricity distribution and customer payments [1]. The smart grid aims to convert current client-server type electricity distribution to peer-to-peer type energy distribution. Using a two-way electricity line, consumers can also sell their self-generated energy to power distribution companies. The smart meter provides real-time monitoring for the energy distribution system [2].

IoT-based intelligent and smart agriculture helps farmers in environment monitoring, soil quality analysis, fertilizer requirement analysis, water distribution control, and so on. Farmers can save water, money, and time by using smart agriculture applications. Farmers can get inputs from agriculture scientists and the government about their fertilizer needs, and new crop decreases using the real-time data generated from the ambient deployment of nano-sensors on farms [3].

An IoT-based smart home provides automation and smart control for their ambient home devices. Smart home users can control home devices from anywhere globally using internet-connected intelligent home appliances. The smart home user can do the security checkup, thermal monitoring, light controlling, washing machine controlling, and many more smart activities using in-hand mobile devices.

The vast data generated from these IoT-based services is stored in the cloud and is used for generating knowledge through data processing. Thus, IoT provides an opportunity for the real-time monitoring of sensor data for quick decision-making and storage of data for data analysis and futuristic planning. Every IoT based data service pass through the following four basic layers:

- The ground layer provides the physical deployment of sensing devices and actuator devices on the ground. This layer works as a data generator.
- The second layer consists of gateway devices that collect data from ground layer sensing devices and deliver this data to the end-user as well as to the cloud for any further processing.
- The third layer provides data analysis using machine learning techniques and artificial intelligence-based intelligent decision making.
- The fourth layer focuses on end-users who collect the data from the third layer as well as the ground layer (based on application) via second layer devices.

The IoT deployment comes with many challenges and opportunities. Starting from deploying thousands of tiny and heterogeneous devices on the ground level to data collection, data analysis, and intelligent decision-making forms significant challenges for researchers. The recent past surveys by [4] show that standardization, communication protocol designing, data analysis, security, and privacy are the highly notable challenges in the IoT setup. These surveys also highlight that security is a big concern, among others. Some of the significant security issues are data confidentiality, user privacy, device authentication, physical security, and so on [5]. Followings are two crucial reasons behind the need for a full-proof security mechanism for the IoT system.

- Numerous heterogeneous tiny resource constraint devices on the ground level.
The elliptic curve cryptography (ECC) is a branch of mathematics that deals with a range of threats, involving node bypassing, a lack of reciprocal authentication, and the likelihood of an insider attack. Between 2010 to 2013, many authors proposed an authentication scheme, but the continuous fix-brake channel of the authentication brightens up new vulnerabilities and challenges for the two-factor authentication.

In 2013, Xue et al. [13] put forward a temporal credential-based mutual authentication scheme using a password for the WSN. In 2015, Jiang et al. [14] highlighted vulnerabilities like identity guessing, privilege insider attack, and stolen SC attack inside [13] and also proposed a new RUA scheme. In 2016, Amin et al. [15] came up with a lightweight mutual authentication scheme for the WSN architecture. However, in 2017, Wu et al. [16] derived vulnerabilities like sensor capture attack, user forgery attack, gateway forgery attack, and user tracing attack security loopholes in [15].

**Contributions**: For the U-GW-based network model presented in section [A] we provide a secured and coherent two-factor authentication protocol employing a password and a smart card (SC). In the U-GW-based network model, we consider that the IoT application user communicates with the gateway device for receiving the sensor data. Over here, we believe that the sensing device deployed in the local network are tiny devices, and they do not perform security mechanisms.

In this model, the IoT user receives sensor data from the gateway device through a precarious channel. We provide a rigorous security analysis of the proposed scheme using globally recognized tools such as BAN Logic and AVISPA. An informal security analysis using widely adopted Dolev-Yao attack model is also provided. We also compare the provided solution to other approaches and analyze performance of proposed work. This examination demonstrates the proposed work’s uniqueness, efficiency, and reliability.

**Motivations**: The strong motivation for this paper is the existence of numerous vulnerabilities in the recently proposed schemes. After a thorough literature study, we critically observed that it is challenging to design a lightweight authentication security scheme for sensing-based applications. These devices require an authentication scheme that uses less energy, less time, and lower space for the secure key exchange; hence IoT users can timely receive heterogeneous IoT sensor data from the gateway device. In the proposed scheme, the secure key exchange between the gateway and the IoT user assures secure data transmission of sensing data among IoT user and gateway device over the anxious public internet. Another strong motivation for proposing this work is a synchronous implementation of the presented work. We implemented the proposed scheme using the real-time deployment of sensors and microprocessors (Raspberry-Pi).

**Related work**

In 1981, Lamport introduced the first RUA scheme based on the hash chain and with the password table at the server-side [9]. By considering the limitations of password table-based schemes, in 1993, Chang et al. [10] introduced the first Smart Card (SC) based authentication scheme. In the SC-based authentication schemes, the user keeps an SC generated by the service provider as another security feature. Following this work, many other researchers proposed an RUA scheme for the client-server model used on the internet.

In 2009, Das et al. set forth the first two-factor authentication scheme for the wireless sensor network [11]. In 2010, Khan et al. [12] performed cryptanalysis on Das et al.’s scheme and successfully highlighted several vulnerabilities in their system. They underlined that Das et al. technique is vulnerable to a range of threats, involving node bypassing, a lack of reciprocal authentication, and the likelihood of an insider attack. Between 2010 to 2013, many authors proposed an authentication scheme, but the continuous fix-brake channel of the authentication brightens up new vulnerabilities and challenges for the two-factor authentication.

Thus, the communication between sensors, gateway devices, and end-users must be secure enough to fulfill all the critical security goals. The secure authentication mechanism between these devices provides secure key generation for communication and mutual trust among each communicating party. A secure authentication system fulfills most major security goals except some like access control [6], [7].

Cryptography is a branch of mathematics that deals with enumerations and executions of cryptosystems. With the run-up towards smart technology, the need for lightweight cryptography came into the picture. The reason behind this need is the use of numerous resource constraint devices in the deployment of sensor-based IoT application deployments. These resource constraint devices are short of computation memory and storage capability. It is nearly impossible for these devices to perform sixteen rounds of Data Encryption Standard (DES) and exponential computations of the RSA promptly without higher energy utilization. In 2003, Hankerson et al. highlighted Elliptic Curve Cryptography (ECC), which is much lighter than traditional crypto methods in computations and storage requirements [8]. Due to its appealing features such as reduced key sizes, relatively short time requirements, and limited resource utilisation, the ECC became a well-known cryptographic approach for resource constraint devices. The Elliptic Curve Diffie-Hellman encryption (ECDHE) provides a lighter version of Diffie-Hellman with Elliptic Curve Discrete Logarithm Problem (ECDLP).

We can divide the IoT devices among three major parts:

- **User devices** in the IoT are a combination of resource constraints and resource-capable devices. Resource constraint devices like wearable devices to resource-capable devices like laptops and mobiles.
- **Gateway devices** in the IoT are considered resource-capable devices and can work as intermediary devices between sensing devices and user devices. They provide setup support to the other devices in the IoT system. The IoT user communicates with the gateway device to receive runtime sensing data.
- **Sensing devices** in the IoT involve tiny sensors and actuator devices. They are not capable of performing any traditional and non-traditional security mechanisms. Many security papers highlight that sensing devices do cryptographic operations, but it is non-practical in real time due to energy issues and on-time service requirements. e.g., A thermal sensor deployed in the house can not run for the cryptographic operation when you ask for temperature. They transmit data to the home gateway device, and the user receives data from it.

In 2013, Xue et al. [13] put forward a temporal credential-based mutual authentication scheme using a password for the WSN. In 2015, Jiang et al. [14] highlighted vulnerabilities like identity guessing, privilege insider attack, and stolen SC attack inside [13] and also proposed a new RUA scheme. In 2016, Amin et al. [15] came up with a lightweight mutual authentication scheme for the WSN architecture. However, in 2017, Wu et al. [16] derived vulnerabilities like sensor capture attack, user forgery attack, gateway forgery attack, and user tracing attack security loopholes in [15].

**Contributions**: For the U-GW-based network model presented in section [A] we provide a secured and coherent two-factor authentication protocol employing a password and a smart card (SC). In the U-GW-based network model, we consider that the IoT application user communicates with the gateway device for receiving the sensor data. Over here, we believe that the sensing device deployed in the local network are tiny devices, and they do not perform security mechanisms.
In 2017, Jiang et al. [14] proposed a new three-factor scheme after analysis of the scheme proposed by Amin et al. They identified several security loopholes like known session-specific temporary attacks and tracking attacks, side-channel attacks, and impersonation attacks in [15]. In 2017, Chen et al. [17] proposed an authentication scheme for the IoT environment and claimed that it is secured from all the well-known attacks, but recently in 2019, Patel et al. [18] provided cryptanalysis for Chen et al.’s scheme and proved that their scheme is not secure enough against attacks such as sensor device anonymity and gateway device bypassing attack. Recently, in 2020, Patel et al. also proposed a lightweight authentication scheme for the same network model [19]. They also highlighted that their proposed scheme is secure and lightweight for the U-GW network model. The proposed network model can be applied for UAV communications [20], [21] also where users securely communicate with UAV devices.

We limit the related work discussion due to the restricted manuscript size. We suggest readers of this manuscript to refer other ECC based RUA schemes for further references [22], [23], [24], [19].

B. Road map of the paper

The remaining portion of the paper is organized as follows: The section II contains the essential preliminaries that are employed throughout this work. We propose an authentication system in section III. The section IV encompasses formal and informal security analysis which is followed by comparative analysis of the proposed system with existing schemes. The section VII provides implementation details. Finally, section VIII summarizes this work by addressing some future aspects of the proposed work.

II. PRELIMINARIES

This section discusses preliminaries such as the Network model, Elliptic curve cryptography (ECC) encryption/decryption, and the threat model. We request readers to follow [19] for the basics of hash function and ECC.

A. Network model

IoT has started its journey with RFID (Radio-frequency identification) technology. RFID gets attraction due to its properties like being free from the line of sight and acceptable range. Thus, RFID tags on things can provide necessary information about a thing, and internet-connected RFID readers can help monitor and collect data. The IoT-based network models vary from application to application. In this paper, we consider the IoT-based U-GW network model [19]. We can use a smart home network as an example of the U-GW network model, where the user accesses data of all inner deployed sensing devices using the gateway device. This model is also applicable to other IoT applications (such as smart hospital management), where every registered user receives data from the underlying sensing devices through the gateway device.

B. Notations and symbols

Table I presents notations used for articulation of the proposed scheme.

| Symbols | Description |
|---------|-------------|
| UID     | User Identity |
| || | String concatenation |
| UPW     | User Password |
| ⊕       | Ex-OR Operation |
| GW      | Gateway     |
| Sk      | Session key  |
| U       | User        |
| UR_i    | User random number |
| GWR_j & n_gw | Gateway random number |
| Hash(·) | One way Hash function |
| T_k     | Time stamp  |
| K_gw    | Gateway computed key |
| K_u     | User computed key |
| A_d     | Adversary   |
| →       | Insecure communication |
| ⇒       | Secured communication |

Figure 1. Generic IoT network model
C. ECC encryption/decryption

Device A and Device B uses key derivation function (KBKDF) that provides every time same output if user provides same input every time. The encryption/decryption operation over elliptic curve occurs as follow:

**Device A performs**
1. Device A randomly generate \( r_a \) where \( r_a \in \{1,2,...,n-1\} \) and is a private key for device A.
2. Device A computes \( Pub_A = r_a * G \). Here \( Pub_A \) serves as a public key. For \( EF_p \), \( G \) is known as a public generator point shared between devices.
3. Device A publishes \( Pub_A \).

**Device B performs**
1. Device B selects \( r_b \) where \( r_b \in \{1,2,...,n-1\} \) and is a private key for device B.
2. Device B computes public key \( Pub_B = r_b * G \)
3. Device B computes \( r_b * PUBA = r_b * (r_a * G) = r_a * (r_b * G) \).
4. Device B inputs \( r_b * PUBA \) and generates symmetric key \( K \) using the KBKDF. Device B encrypts message \( M \) as \( M_E = E_{ncK}(M) \). Device B sends \( M_E \) message to Device A along with the value \( Pub_B \) through public channel.

**Device A performs**
1. Device A inputs \( r_a * (Pub_B) \) into the same KBKDF and generates symmetric key for the decryption as \( K \). Device A decrypts \( M_E \) as \( M = DecK(M_E) \).

D. Threat Model

We follow the following threat model considered from [25] which derives certain capabilities for the adversary \( A_d \). The polynomial time \( A_d \) can:
1. compute valid pair of identity * password offline in the polynomial time.
2. extract information from the user’s smart card [26].
3. fully access communication channel between U-GW.
4. obtain the previously calculated session key.
5. get secrets of gateway node during system failure situations [27].
6. get physical access to user device and can extract saved information through power analysis.
7. Adversary \( A_d \) is unable to access the user password or SC simultaneously. The simultaneous access of both the password and SC leads to impersonation attack [19].

III. PROPOSED SCHEME

This section discusses the proposed ECC-based two-factor authentication scheme for the U-GW paradigm. The proposed scheme is divided into four steps: (1) Setup phase, (2) User registration phase, (3) Login and key Generation phase, and (4) Password update phase. We assume that there are \( n \) users and a single gateway device for the proposed work. Table II bright ups tabular presentation for the login and key generation phase.

A. Setup phase

During the setup phase, the user device (\( U_i \)) and the gateway device (\( GW \)) setup computation environment. For the proposed scheme, the setup phase performs as follow:

**UF1:**
Step 1: Both \( U_i \) and \( GW \) devices agree on \( EF_p \) (represents curve defined over finite field \( F \) using prime number \( p \)), the generator point \( G(X,Y) \) and the KBKDF.
Step 2: \( U_i \) produces random number \( UR_i \) in such way \( UR_i \in Z_p^* \). Here \( p \) is a large prime number and the range of \( Z_p^* \) is \((0,1,2,...,p-1)\).
Step 3: \( U_i \) calculates \( D_u = UR_i * G \).
Step 4: \( U_i \) sends \( D_u \) to the \( GW \) in offline manner.

**GWF1:**
Step 1: \( GW \) generates random number \( GWR_j \) where \( GWR_j \in Z_p^* \).
Step 2: \( GW \) calculates \( S_i = GWR_j * G \).
Step 3: \( GW \) transmits \( S_i \) to the \( U_i \).

**UF2:**
Step 1: \( U_i \) generates symmetric key \( K_u \) using \( UR_i, S_i \) by KBKDF. This is not the session key but this is a key that is used for encryption/decryption by both \( U_i \) and \( GW \) whenever they establish session.

**GWF2:**
Step 1: \( GW \) generates symmetric key \( K_{gw} \) using \( GWR_j, D \) by KBKDF.
Step 2: \( GW \) discards \( D \).
Thus, after completion of setup phase, \( U_i \) device have \{\( UR_i, D_u, S_i \}\} in its confidential memory while \( K_{gw} \) in the secret memory and the \( GW \) device have \{\( GWR_j, S_i \}\} in its confidential memory while \( K_{gw} \) in its secret memory. The data stored in confidential memory is read-only and can be read by anyone using power analysis or reverse engineering. But, the data stored in secret memory can be read by the device itself only.

B. User Registration Phase

In this phase, the \( U_i \) registers to the \( GW \) by using identity \( UID \) and password \( UPW \). The gateway device generates an SC (SC) with numerous parameters. These parameters are used by \( U_i \) during the login and key generation phase.

**UF1:**
Step 1: \( U_i \) chooses unique identity \( UID \) and secure password \( UPW \).
Step 2: \( U_i \) calculates \( h_i = Hash(D_u || UPW || UID) \).
Step 3: \( U_i \) generates message \( M_1 = \{UID, h_i\} \) and sends \( M_1 \) to \( GW \) over secure channel. We created a Transport Layer Security (TLS) channel over the MQTT protocol for data transmission across a secure means in our implementation.

**GWF1:**
Step 1: \( GW \) receives message \( M_1 \) and retrieves user identity \( UID \). \( GW \) verifies \( UID \) in the database and checks its availability. If similar \( UID \) is used by another user then \( GW \) transmits message such as "Identity
C. Login and key generation phase

The user will supply their identity, password, and SC to the card reader during this phase. The card reader will authenticate the user’s identity and password before communicating with the Gateway to establish the session key.

**UF1:**

Step 1: $U_i$ inserts $UID$, $UPW$ and $SC$ in to SC reader (SCR). SC computes $Dec_{C_{PW*P}}(Z)$ and extracts $D_a$ from its secret memory.

Step 2: calculates $h_i = Hash(D_a||UPW||UID)$, $X_i^* = Hash(UID||h_i)$, and verify $X_i^* \oplus X_i = T = O_i \oplus h_i$, $L_i = Hash(D_a||UID)$, $PID = T \oplus Hash(UID||L_i||T_k)$, $Z = Enc_{K_{gw}}(D_a)$.

Step 3: $U_i$ frames $M = \{MID, Z, T_k, PID\}$ and sends it to the gateway device. Over here $T_{ki}$ is the current time stamp of User.

**GWF1:**

Step 1: GW receives message $M$ and computes own timestamp $T_k^*$. Verify timestamp $\Delta T \leq T_k^* - T_k$.

Step 2: GW calculates $Dec_{K_{gw}}(MID)$ to receive $UID$.

Step 3: GW extracts $D_a$ by calculating $Dec_{C_{gw}}(Z)$.

Step 4: GW computes $T = Hash(GWR_j||K_{gw})$, $N_i = Hash(D_a||UID)$, $PID^* = T \oplus Hash(UID||N_i||T_{ki})$.

Step 5: GW verifies $PID = PID^*$. After successful verification only GW continues.

Step 6: GW generates $n_{gw}$ in order to $n_{gw} \in Z_p^*$ and computes $NS = Enc_{K_{gw}}(n_{gw})$.

Step 7: GW computes $S_k = Hash(UID||T||n_{gw} \ast D_a)$.

Step 8: GW computes key verifier $S_Q_i = Hash(S_k||n_{gw}||T||T_{k_{new}})$ and forwards $M_{new} = \{S_Q_i, NS, T_{k_{new}}\}$ to user.

**UF2:**

Step 1: $U_i$ recovers $n_{gw} = Dec_{K_{w}}(NS)$.

Step 2: $U_i$ calculates $S_k^* = Hash(UID||T||n_{gw} \ast D_a)$.

Step 3: $U_i$ computes $S_Q_i^* = Hash(S_k^*||n_{gw}||T||T_{k_{new}})$.

Step 4: $U_i$ verifies $S_Q_i^* = S_Q_i^*$. If verification is successful then he/she considers $S_k^*$ as a new current session key.

D. Password update phase

Using this phase, $U_i$ updates his/her identity $UID$ and password $UPW$ through SCR device.

**UF1:**

Step 1: $U_i$ provides $SC$ to SCR and selects password update option.

Step 2: $U_i$ provides $UID$, $UPW$, $UPW_{new}$

**SCRF1:**

Step 1: SCR validates $\{UID, UPW\}$. SCR computes $h_i = Hash(UID||UPW||D)$, $X_i^* = Hash(UID \oplus h_i)$ and checks $X_i^* \neq X_i$.

Step 2: Computes $h_{i_{new}} = Hash(UID||UPW_{new}||D_a)$.

Step 3: Computes $O_i = h_i \oplus h_{i_{new}} \oplus O_i$, $X_{i_{new}} = Hash(UID \oplus h_{i_{new}})$, $Z_{new} = Enc_{UPW_{new}}(D_a)$.

Step 4: Replaces $\{X_i, O_i, Z\}$ by $\{X_{i_{new}}, O_{i_{new}}, Z_{new}\}$.

IV. Security Analysis

In this section, we put forward an informal security analysis using the Dolev-Yao channel \[25\] and a formal security analysis using the AVISPA and the BAN Logic.

A. Informal security analysis

We present an informal security analysis for the proposed authentication mechanism employing the Dolev-Yao threat model in this subsection. \[25\]. Table \[III\] provides a security comparison for the proposed scheme with the other existing schemes.

F1. Password guessing attack:

In this attack, adversary $A_d$ performs offline and online password assumptions and verifies numerous passwords. $A_d$ uses a famous brute force dictionary method to get success in this attack. In our scheme, an $A_d$ cannot achieve success in password guessing attacks due to hashing of $Hash(UID||UPW||D_a)$. Even though adversary may guess the correct $UID$ and $UPW$; then also, he/she will not receive value of $D_a$. After guessing the correct $\{UID, UPW\}$ pair, an adversary cannot compute the final session key because of the unavailability of the random parameter $n_s$. As a result, the proposed scheme is resistant to the password guessing attack.

F2. Message Replay attack:

In this attack, the polynomial adversary $A_d$ captures communication between $U_i$ and GW. These messages are used for spoofing and impersonation type activities. We use three random parameters that provide immunity from the replay activities of the attacker. The first parameter is a timestamp ($T_{ki}$), the second one is a random variable $UR_i$, and the last parameter is random nonce $N_i$. With the help of these parameters, the receiver can quickly validate the freshness of received messages. As a result, the proposed scheme is protected against the message replay attack.

F3. User anonymity:

If adversary $A_d$ acquires the user’s identity, we can state that the user anonymity attack is successful. In our scheme, we use hash function to protect identity and password at user side ($h_i = Hash(UID||UPW||Dec_{C_{PW}})$)
as well as at gateway side, we validate inside hash computations. Thus, neither entity shares UID over an insecure channel in the proposed scheme.

F4. Perfect forward secrecy attack:
In this threat, the availability of a gateway secret key $K_{gw}$ does not lead an adversary towards successful session key computations. The proposed scheme uses a random number (n$_{gw}$) at the gateway side for the session key computation; thus, even though the adversary gets the gateway secret key $K_{gw}$, they do not succeed in the previous session key as well as session key for future communications.

F5. Stolen SC attack:
In the stolen SC attack, $A_d$ receives SC and performs power analysis to extract data. After extracting data, $A_d$ require password to receive $Z$. Thus, the extraction of SC data does not provide the right direction to an adversary for the session key computations. As a result, the proposed scheme is resistant to stolen SC threats.

F6. Privileged insider attack:
In this attack, $A_d$ is an insider to the gateway device and can see the received messages. We do not relieve UID anywhere in the plaintext during computation in the proposed scheme. After receiving PID from the user, the gateway performs verification for the PID$^*$ = PID, and it is secured using hash. Thus, $A_d$ at the gateway device neither receive identity nor password. For key computation, $A_d$ needs {UID, T, D$_u$} and it is near to impossible for an $A_d$ to re-frame the same pair. As a result, the proposed scheme is impervious to privileged insider attacks.

F7. Mutual authentication:
The proposed scheme achieves mutual authentication. The gateway GW calculates PID$^*$ and validate it with the PID$^*$. The GW also validates ID and PW of the $U_i$.

The $U_i$ authenticates GW by verifying PID$^*$ = Q$^*$. The Q$^*$ is calculated with the help of received key. As a result, we may assert that the suggested system meets the mutual authentication property.

F8. Man-In-The-Middle (MITM) attack:
In this attack, $A_d$ receives messages communicated between the $U_i$ and the GW. In proposed scheme, even $A_d$ receives {PID, MID, Z} then also he/she will not get success in order to read inside data due to hashing and ECC encryption. Using {NS (Computed parameter using n$_{gw}$), SQ$^*$}, $A_d$ could not calculate the S$_k$ because of inadequate information about {n$_{gw}$, UID, D}. As a result, the suggested technique is protected against an MITM attack.

F9. User impersonation attack:
To impersonate as a legitimate user, an adversary $A_d$ captures SC parameters such as {O$_i$, S$_i$, A$_i$, Z}. Using these parameters, an $A_d$ can not generate correct login requests {PID, T$_{ki}$, MID, Z}. As a result, an adversary’s imitation attempts are unsuccessful. Hence, the proposed scheme is secured against the user impersonation attacks.

F10. Gateway impersonation attack:
To impersonate as a legitimate gateway, an adversary $A_d$ captures the public message and public parameters such as the gateway device. Using these parameters, an $A_d$ can not generate correct reply $M_{new} = \{SQ^*, NS, T_{k_{new}}\}$ for the user due to inadequacy of parameter $K_{gw}$. Thus, an adversary does not get success in user impersonations. Similarly, an adversary $A_d$ can not decrypt the message MID and Z due to the nonavailability of gateway master secrets. As a result, the proposed scheme is unlikely to be affected by a gateway impersonation attack.

F11. Denial of service attack:
Using this attack, the polynomial adversary $A_d$ tries to...
Figure 2 shows basic postulates used by the BAN Logic. Over

C. Postulates

Figure 2 shows basic postulates used by the BAN Logic. Over here P and Q are the communicating principals, M1 and M2 are the communicated messages. The key KS is used for the encryption/decryption operations and $k_s$ is the shared secret.

D. Inference rules

Inference rules are derivations that are derived from postulates by the BAN logic. The inference rules prove that the proposed authentication scheme satisfies the mutual authentication following. Figure 3 shows basic inference rules generated for mutual authentication proof.

E. Assumptions

BAN Logic works based on the following assumptions:

- There are some shared secrets.
- Principals can compute fresh nonces.
- Each principal believes on each other.
- Each principal can recognize his/her messages.
- If principal P believes that KS is his public key, then P must know corresponding private key $KS^{-1}$.

stop $U_i$ and $GW$ from key generation through either flooding or any other means. An adversary $A_d$ gives rise to redundant requests for either device and generates too much delay in the key generation process. In the real-time scenario, it is nearly impossible to achieve full proof protection against DoS-based attacks; still, we tried to protect our scheme using the time-stamp and random numbers. As a result, it prevents an attacker $A_d$ from gaining complete control of the system.

B. Mutual authentication using BAN logic

In this subsection, we show that our scheme achieves mutual authentication property using this BAN logic tool. The BAN logic generates trust between the principles (communicating parties). It focuses on the proposed scheme’s coherence and feasibility. It works on proper formulations of postulates, inference rules, and assumptions with realistic goals.

TABLE III

| Security Parameter | 12 | 28 | 29 | 30 | 31 | Proposed |
|--------------------|----|----|----|----|----|----------|
| F1 Offline Password guessing | | | | | | ✓ |
| F2 Replay | ✓ | ✓ | ✓ | | | ✓ |
| F3 User anonymity | ✓ | ✓ | ✓ | ✓ | | ✓ |
| F4 Perfect forward secrecy | ✓ | ✓ | ✓ | ✓ | | ✓ |
| F5 Stolen smart card | | | | | | ✓ |
| F6 Privilege insider | | | | | | ✓ |
| F7 Mutual authentication | ✓ | ✓ | ✓ | | | ✓ |
| F8 Man-in-the-Middle | | | | | | ✓ |
| F9 User impersonation | ✓ | ✓ | ✓ | | | ✓ |
| F10 Gateway impersonation | ✓ | ✓ | ✓ | | | ✓ |
| F11 Denial of Service | ✓ | ✓ | ✓ | | | ✓ |

F. Goals

For any U-GW model, the followings are the primary goals or conclusions those must be achieved during the authentication. Hence, using above said inference rules, assumptions and postulates, both user ($U_i$) and the gateway ($GW$) have to achieve following goals:

**Theorem 1.** The proposed satisfies mutual authentication property among $U_i$ and $GW$.

**Proof.** We rewrite the messages communicated in login and authentication phase of the proposed scheme in the generic form as below:

**MS 1:** $U_i \rightarrow GW$: ($\langle ID \rangle, (S_i||Kgw)\oplus Hash(D)||UPW||UID) \oplus Hash(D)||UID||UPW) \oplus Hash(UID) || Hash(D) || UID) || T_ki, Enc(D), Ts_i)

**MS 2:** $GW \rightarrow U_i$: ($\langle Enc(n_{gw})$, Hash$(UID) || Hash(GWRj||Kgw) || ns * D) || Hash(GWRj||Kgw) || T_knew, T_knew)$

**Idealized form:** We can rephrase MS1 and MS2 in idealized form as follows:

**MS 1:** $U_i \rightarrow GW$: ($\langle ID \rangle, (S_i||Kgw)\oplus (D||UPW||UID), (D||UID||UPW) \oplus (UID || Hash(D)||UID) || T_ki, D), Ts_i)$

**MS 2:** $GW \rightarrow U_i$: ($\langle n_{gw} \rangle, ((UID) || (GWRj||Kgw) || ns * D) || (GWRj||Kgw) || T_knew, T_knew)$

**Goal:** We define goals of the proposed scheme in idealized form as follow:

**GL1:** $GW \equiv U_i \xrightarrow{S_k} GW$

**GL2:** $U_i \equiv U_i \xrightarrow{S_k} GW$

Following assumptions are used to prove the mutual authentication:

**Y1:** $U_i \equiv \#(T_ki)$

**Y2:** $GW \equiv \#(T_ki)$

**Y3:** $U_i \equiv \#(T_knew)$

**Y4:** $GW \equiv \#(T_knew)$

**Y5:** $U_i \equiv \langle U_i \xrightarrow{S_k} GW \rangle$
Figure 2. Postulates

| Construct | Meaning of construct |
|-----------|----------------------|
| $P \models M_1$ | Principal P believes that statement $M_1$ is true. |
| $P \models M_1$ | Message $M_1$ is received by P and P can see, perform operation on it and can send it to other principal. |
| $P \models \sim M_1$ | Principal P once said message $M_1$, either now or sometime in past but it is true that P believed message $M_1$. |
| $P \models M_1$ | Principal P has jurisdiction over statement $M_1$ and all principals can trust P for statement $M_1$. |
| $(\#(M_1))$ | Message $M_1$ is recent message and same message not sent in past at any time of communication. |
| $(\langle M_1 \rangle_{M_2})$ | Statement $M_1$ is combined with equation $M_2$ so if any principal say $M_2$ then it will provide identity of whoever say $(\langle M_1 \rangle_{M_2})$. So $M_2$ can be used as a proof of origin of $M_1$. |

$\{M_1\}_{k}$ | Statement $M_1$ is encrypted using key $k$. |
| $\triangleright P \iff Q$ | Key $k$ is shared between principals P and Q. |
| $k \triangleright P$ | key KS is public key of principal P. |
| $K_2$ is secret which is only known by principals P and Q. Both principals use $k_2$ to prove their identity to each other. |

Figure 3. Inference Rules

| Rule No. | Rule | Meaning |
|----------|------|---------|
| Z1 | $P \models P \overset{KS}{\triangleleft} Q, P \models (M_1) \iff P \models (\#(n_{gw}))$ | Message Meaning Rule that says, if P receives message $M_1$, that is encrypted using $K_2$ and $K_2$ is shared only with Q then, message is sent by Q. |
| Z2 | $P \models (\#(M_1)), P \models (Q \sim M_1)$ | None verification rule is used to prove that the message $M_1$ is most recent message and sender still believes on $M_1$. |
| Z3 | $P \models Q \sim M_1 \iff P \models (\#(M_1))$ | Jurisdiction rule says that if P believes that Q has jurisdiction over $M_1$ then P believes on $Q$ over the correctness of message $M_1$. |
| Z4 | $P \models Q \sim M_1 \iff P \models (\#(M_1))$ | Seeing rule says that if message $M_1$ is encrypted by the shared key $k_2$ then, both P and Q can see the message $M_1$. |
| Z5 | $P \models (\#(M_1)) \iff P \models (\#(n_{gw}))$ | Seeing rule say that if P sees (M1,M2) then P sees M1 or P sees M2. |
| Z6 | $P \models (\#(M_1)) \iff P \models (\#(n_{gw}))$ | Fresh rule says that if one part of the formula M1 is fresh, it means that entire formula is fresh. |
| Z7 | $P \models (\#(M_1)) \iff P \models (\#(n_{gw}))$ | Belief rule says that if P believes complete formula then it also believe each part of formula. |

$Y_6$. $GW \models (U_i \overset{S_i,D}{\triangleleft} GW)$ |

$Y_7$. $U_i \models GW \models (U_i \overset{S_i,D}{\triangleleft} GW)$ |

$Y_8$. $GW \models \#(n_{gw})$ |

$Y_9$. $U_i \models \#(n_{gw})$ |

Mutual authentication among $U_i$ and $GW$ is achieved as given below:

**W1:** From MS 1, $GW \models (U_i \overset{S_i,D}{\triangleleft} GW)$

**W2:** Using W1, $Z1$ and $Y6$, we obtain, $GW \models (U_i \overset{S_i,D}{\triangleleft} GW)$

**W3:** Using W2, $Y2$, $Z2$, we gain, $GW \models (U_i \overset{S_i,D}{\triangleleft} GW)$

**W4:** Using W3, W4, Z8, we achieve, $GW \models \#(n_{gw})$ |

**W5:** Using W3, W4, Z8, we achieve, $GW \models \#(n_{gw})$ |

**W6:** Using W1, W4, Z2, Y8, we obtain, $GW \models \#(n_{gw})$ |

**W7:** From message 2, we receive, $U_i \overset{S_i,D}{\triangleleft} GW$ |

**W8:** Using W7, Z1 and $Y6$, we receive, $U_i \overset{S_i,D}{\triangleleft} GW$ |

**W9:** Using W8, Y2, Z2, we obtain,
The HLPSL protocol is translated into Intermediate Format (IF) using the HLPSL2IF translator. We are using Delev-Yao medium that works based on two fundamental operations called as SND and RCV. This is the only medium supported by AVISPA, for communication simulation. We ran a simulation for the proposed protocol’s three phases (registration phase, login phase, and key generation phase).

The AVISPA tool comprises four sub-tools that function as the back-end to IF, as indicated in Figure 4. To obtain Output Format (OF), the Protocol transformed in IF is injected into these tools, the Tree Automata based on Automatic Approximation for the Analysis of Security Protocol tool, the SAT-Based Model Tracker tool, The On-the-Fly Model (OFM) Checker tool, and the Constraint Logic-based Model Checker tool are four major back end tools utilized for computing OF, which states whether or not the proposed protocol is secure against said attacks.

The simulation of the proposed protocol using the famed OFMC tool can be seen in the Figure 5. We utilized two back-end tools (OFMC and CL-AtSe) to simulate the proposed protocol. Due to their inability to simulate bitwise XOR operations, SATMC and TA4SP cannot simulate the proposed protocol, hence whatever results these tools provide are inconclusive.

The protocol tested 25 states during simulation using the CL-AtSe tool, of which 21 were attainable, as shown in Figure 6. While an OFMC simulation revealed that the simulation visited 476 nodes up to a depth of 27 heaps in 0.21 seconds. The protocol is secure against a replay attack and a Man-in-the-Middle attack, according to simulations employing both tools.

VI. Performance Analysis

In this section, we evaluate the developed protocol’s performance in terms of computing and communication overhead. The computation cost analyzes the proposed protocol in terms of the utilization of cryptographic operation. The commu-
TABLE V
Comparison of Computation Cost

| Scheme | Computation Cost |
|--------|-----------------|
| 12     | $6T_{Pa} + 10T_{Ah} \approx 13.4078$ |
| 28     | $6T_{Pa} + 2T_{Sym} + 11T_{Ah} \approx 13.3905$ |
| 29     | $5T_{Pa} + 3T_{Sym} + 13T_{Ah} \approx 13.4767$ |
| 30     | $6T_{Pa} + 4T_{Sym} + 11T_{Ah} \approx 13.3997$ |
| 31     | $13T_{Ah} + 4T_{Sym} \approx 8.9339$ |
| Proposed | $2T_{Pa} + 5T_{Sym} + 11T_{Ah} \approx 4.5003$ |

TABLE VI
Comparison of Communication Cost

| Scheme | No of Message | Comm. cost |
|--------|---------------|------------|
| 12     | 2             | 1184 bits  |
| 28     | 2             | 1184 bits  |
| 29     | 2             | 1600 bits  |
| 30     | 2             | 1952 bits  |
| 31     | 3             | 1280 bits  |
| Proposed | 2             | 992 bits   |

cation cost analyzes protocol regarding the number of "bits" transmitted by each participating party.

A. Computation cost analysis

The computation cost defines the total time consumed by the scheme for session key generation. In the proposed scheme, we use different functions/methods for session key generation. Let us define these functions as $T_{Ah}$, $T_{Pa}$, $T_{pm}$, and $T_{Sym}$. That is time complexity for the hash computation, elliptic curve point addition operation, elliptic curve scalar multiplication operation and symmetric encryption and decryption operation respectively. We extracted the individual time required for each these functions. That was 0.0024 ms, 0.029 ms, 2.227 ms, 0.0047 ms for $T_{Ah}$, $T_{Pa}$, $T_{pm}$, $T_{Sym}$. Table V shows comparative analysis for the computation cost between the proposed scheme and other existing schemes and proves computational efficiency of the proposed scheme.

B. Communication cost analysis

The communication cost shows the total number of bits transmitted before establishing the key over a channel for authentication. We computed the size of individual parameters (in bits) for communication cost computations. The size of the user identity is 160 bits; the output of the hash operation is 320 bits. The public key size is 320 bits, and the private key $\text{UPW} \times P$ size is 160 bits as per ECC computations. Table VI presents a analogous analysis of the communication costs for our protocol with other available protocols for the similar environment.

VII. Implementation

For implementation, we used two types of user devices and two types of gateway devices. We used five nodeMCU and five Raspberry-Pis as light-weight user devices and two laptops as resource-capable user devices. As a resource-capable gateway device, we used a laptop with a configuration of 8 GB RAM, Intel (R) Core (TM) i7-5500U CPU, 2.40 GHz, 64 bit, Ubuntu 16.04 operating system. As a lightweight gateway device, we used Raspberry Pi 3 Model B with Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM with BCM43438 wireless LAN, and Bluetooth Low Energy (BLE) on board. As a programming language, we used python 3. We used publish-subscribe-based MQTT Protocol as a communication protocol at the application layer, which uses TCP at the transport layer protocol and 6LoWPan at the network layer.

We implemented all basic elliptic curve operations in python language by connecting Raspberry Pi as a gateway device and the desktop system as a user device.

A. Computed session key

Following Figure 7 shows computed session key using proposed work:

B. Networking parameters

We computed individual networking parameters, such as round trip delay, packet loss, and throughput. Implementing the proposed scheme using ECC operations and the MQTT protocol reduces the round trip delay, increases throughput, and reduces packet loss for the proposed scheme. We used the widely adopted packet sniffing and packet analyzing tool "Wireshark" to compute the aforementioned parameters. The Wireshark captures MQTT packets and provides all essential networking parameters in an unstructured format; thus, we used python programming to retrieve data more structured way. We interpret these parameters using an automated python script that reads Wireshark data and evokes the required information.

VIII. Conclusion and Future Work

For the U-GW-based generic IoT paradigm, this article developed a unique, reliable, and lightweight SC-based remote user authentication technique. For the discrete logarithm operations, we employed ECC. We performed an informal security study utilising the Dolev-Yao channel and a formal security analysis using the widely used BAN Logic and AVISPA tools for the proposed method. We show that the suggested scheme is very efficient and reliable in terms of communication cost, computation cost, end-to-end delay, packet loss, and throughput for the U-GW system model by comparing it to other current methods. We utilised the Raspberry Pi and the MQTT protocol for implementation. We’re also working on a multi-factor model to overcome the restrictions of two-factor authentications and also on U-GW-sensor model with...
lighter implementation using protocols like light-MQTT and Bluetooth low energy as a future work for the proposed scheme.

REFERENCES

[1] H. Farhangi, “The path of the smart grid,” *IEEE power and energy magazine*, vol. 8, no. 1, pp. 18–28, 2009.

[2] R. Zafar, A. Mahmood, S. Razzaq, W. Ali, U. Naeem, and K. Shehzad, “Prosumer based energy management and sharing in smart grid,” *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 1675–1684, 2018.

[3] A. Antonacce, F. Arduini, D. Moscone, G. Palleschi, and V. Scognamiglio, “Nanostructured (bio) sensors for smart agriculture,” *TrAC Trends in Analytical Chemistry*, vol. 98, pp. 95–103, 2018.

[4] A. Whitmore, A. Agarwal, and L. Da Xu, “The internet of things—a survey of topics and trends,” *Information systems frontiers*, vol. 17, no. 2, pp. 261–274, 2015.

[5] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, “Internet of things security: A survey,” *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.

[6] M. Ammar, G. Russello, and B. Crispo, “Internet of things: A survey on the security of iot frameworks,” *Journal of Information Security and Applications*, vol. 38, pp. 8–27, 2018.

[7] R. Ch, G. Srivastava, T. R. Gadekallu, P. K. R. Maddikunta, and S. Bhattacharya, “Security and privacy of uav data using blockchain technology,” *Journal of Information Security and Applications*, vol. 55, p. 102670, 2020.

[8] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[9] L. Lamport, “Password authentication with insecure communication,” *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[10] C.-C. Chang and S.-J. Hwang, “Using smart cards to authenticate remote passwords,” *Computers & Mathematics with Applications*, vol. 26, no. 7, pp. 19–27, 1993.

[11] M. L. Das, “Two-factor user authentication in wireless sensor networks,” *IEEE transactions on wireless communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[12] S. H. Islam and M. K. Khan, “Cryptanalysis and improvement of authentication and key agreement protocols for telecare medicine information systems,” *Journal of medical systems*, vol. 38, no. 10, pp. 1–16, 2014.

[13] K. Xue, C. Ma, P. Hong, and R. Ding, “A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 316–323, 2013.

[14] Q. Jiang, J. Ma, X. Lu, and Y. Tian, “An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks,” *Peer-to-peer Networking and Applications*, vol. 8, no. 6, pp. 1070–1081, 2015.

[15] R. Amin and G. Biswas, “A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks,” *Ad Hoc Networks*, vol. 36, pp. 58–80, 2016.

[16] F. Wu, L. Xu, S. Kumari, X. Li, J. Shen, K.-K. R. Choo, M. Wazid, and A. K. Das, “An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in iot deployment,” *Journal of Network and Computer Applications*, vol. 89, pp. 72–85, 2017.

[17] Y. Chen, J.-F. Martinez, P. Casillejo, and L. Lopez, “A privacy protection user authentication and key agreement scheme tailored for the internet of things environment: Priauth,” *Wireless Communications and Mobile Computing*, vol. 2017, 2017.

[18] C. Patel and N. Doshi, “Cryptanalysis of ecc-based key agreement scheme for generic iot network model,” in 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2019, pp. 1–7.

[19] C. Patel and N. prafulchandra Doshi, “Secure lightweight key exchange using ecc for user-gateway paradigm,” *IEEE Transactions on Computers*, 2020.

[20] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, “Blockchain-based reliable and efficient certificateless signature for iot devices,” *IEEE Transactions on Industrial Informatics*, 2021.

[21] F. Iwendi, Z. Jilil, A. R. Javed, T. Reddy, R. Kaluri, G. Srivastava, and O. Jo, “Keysplitwatermark: Zero watermarking algorithm for software protection against cyber-attacks,” *IEEE Access*, vol. 8, pp. 72650–72660, 2020.

[22] H. Xiong, J. Tao, and C. Yuan, “Enabling telecare medical information systems with strong authentication and anonymity,” *IEEE Access*, vol. 5, pp. 5648–5661, 2017.

[23] A. Chaturvedi, D. Mishra, and S. Mukhopadhyay, “An enhanced dynamic id-based authentication scheme for telecare medical information systems,” *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 1, pp. 54–62, 2017.

[24] W. Li, L. Xuelian, J. Gao, and H. Y. Wang, “Design of secure authenticated key management protocol for cloud computing environments,” *IEEE Transactions on Dependable and Secure Computing*, 2019.

[25] D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[26] T. Messeres and E. Dabbish, “& sloan, rh (2002), examining smart card security under the threat of power analysis attacks,” *IEEE Transactions on Computers*, vol. 51, no. 5.

[27] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *Annual international cryptology conference*. Springer, 1999, pp. 388–397.

[28] L. Zhang and S. Zhu, “Robust ecc-based authenticated key agreement scheme with privacy protection for telecare medicine information systems,” *Journal of medical systems*, vol. 39, no. 5, pp. 1–11, 2015.

[29] V. Odelu, A. K. Das, and A. Goswami, “A secure biometrics-based multi-server authentication protocol using smart cards,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[30] W. Liu, Q. Xie, S. Wang, and B. Hu, “An improved authenticated key agreement protocol for telecare medicine information system,” *SpringerPlus*, vol. 5, no. 1, pp. 1–16, 2016.

[31] S. Qiu, G. Xu, H. Ahmad, and L. Wang, “A robust mutual authentication scheme based on elliptic curve cryptography for telecare medical information systems,” *IEEE access*, vol. 6, pp. 7452–7463, 2017.