Secure and Fast Remote Application–Based Authentication Dragonfly using an LED Algorithm in Smart Buildings

1st Batool Mohammed Radhi
Department of Computer Science
College of Education for Pure Sciences
University of Basrah, Basrah 61004, Iraq
batool.mohammed@uobasrah.edu.iq

2nd Mohammed Abdulridha Hussain
Department of Computer Science
College of Education for Pure Sciences
University of Basrah, Basrah 61004, Iraq
mohammed.abdulridha@uobasrah.edu.iq

3rd Zaid Ameen Abduljabbar
Department of Computer Science
College of Education for Pure Sciences
University of Basrah, Basrah 61004, Iraq
zaid.ameen@uobasrah.edu.iq

4th Vincent Omollo Nyangaresi
Department of Computer Science and Software Engineering, Jaramogi Oginga Odinga University of Science and Technology, Bondo 40601, Kenya
vnyangaresi@jooust.ac.ke

***Abstract*— The proliferation of the internet of things (IoT) has led to the emergence of a wide range of intelligent devices, creating a broad domain with significant security concerns. These concerns impose a high level of security; unfortunately, IoT devices usually have limited resources in terms of little memory, low computing power, and a short battery life. Therefore, IoT application developers must use lightweight cryptographic tools to achieve a trade-off between performance and security. The storage and high computation capacity of cloud computing is often exploited to manage the vast amount of data produced by such gadgets. Some methods still suffer from attacks, and others cannot achieve low complexity. We propose a secure and low-complexity system for smart buildings in transferring data between the local server, the cloud, and users authorized by the owner. The LED encryption algorithm, which is lightweight and requires limited resources and less energy, was used to create a mobile application system characterized by confidentiality, authentication, and privacy. For further security, the owner's biometrics were used and derived as the key to decrypt data from the cloud. We have leveraged Dragonfly authentication technology to transfer data from the local server to the users. The owner can add authorized persons in the cloud database and local server to enjoy using the application. Moreover, we successfully balance security complexity and performance in our work. As a result, we achieve good results with a computation cost of 0.281 s and a communication cost of 1472 *bit*.**

***Keywords*— *authentication, confidentiality, lightweight LED, dragonfly technology, mobile application*.**

I. INTRODUCTION

The internet of things (IoT) grew quickly and is expected to continue growing over the next years. There is projected to be approximately 25 billion connected IoT devices and sensors by 2025 [1]. These devices can move and produce data via a network without the assistance of a human, which is in line with the expectation that IoT devices and their applications will reach and connect every element of our daily lives. These days, a wide range of fields and applications, including smart homes, smart cities, water, electricity, green energy, traffic congestion, waste management, disaster alerting, recycling, agriculture, breeding, and healthcare, have adopted connectivity, cloud computing, and big data analytics due to significant advancements in IoT-enabling technologies [2],[3],[4]. These sensors and devices generate data that may include private or sensitive patient information, such as medical records, photos of people, and license plate numbers from IoT surveillance cameras in checking zones. All of this has made the security and privacy of such data increasingly important. Furthermore, IoT devices must be secured, and their data must be shielded from unwanted access because attackers might leverage any weakness [5],[6]. Thus, robust and enhanced encryption techniques should be considered to safeguard the transfer of sensitive information. In the context of IoT, lightweight cryptographic primitives are advised while considering the trade-off between security assurance and performance [7],[8]. The data many smart devices produce is one of the biggest concerns with IoT. With so many various types of IoT devices producing such a large database store, cloud computing (CC) has emerged as a critical technology for managing it [9],[10].

WPA3, released in June 2018, is the most recent security system, created to improve security on existing Wi-Fi networks and address issues with earlier iterations. It uses WPA3 password-based simultaneous authentication equality (SAE) technology for client authentication. Originally designed for usage in mesh WLANs, the SAE protocol has been modified and shown to deliver the promised protection [11]. This resistance is achieved using a Dragonfly handshake to take advantage of logarithms and a discrete ellipse curved encoder. One of the most basic tasks in the IoT is to protect data transmitted over the local or global network and check who can access the data [12],[13]. In this work, an LED is used to protect data in motion, while Dragonfly protocols are used as a switch for a lightweight LED algorithm. One of its weaknesses is its static key, so by using Dragonfly's technology to authenticate users, we could use it for a single session key [14]. The proposed scheme also has powerful features, such as secure authentication and a lightweight, confidential, and secure algorithm for changing key sessions every time. User privacy is also important. Therefore, we use the owner's biometric key to decrypt the data received from the cloud

and send it encrypted to the user once during registration. Contributions include the following:

- Our proposed system achieves the confidentiality of data stored and encrypted in the cloud through a lightweight LED algorithm.
- By reducing the amount of data transferred (abnormal or every half hour), we were able to reduce the cost of cloud storage.
- Our proposed system uses Dragonfly technology to authenticate between the user and the local server.
- Data encryption and authentication were implemented between the cloud, the local server, and the user.
- Our system contributed to developing a distinguished mobile application with confidentiality and authentication of data transfer in local and global networks.
- Owner biometrics are used as the key to decrypt cloud data to enhance privacy for users.

This is how the remainder of the paper is structured. Related work is reviewed in Section II. Primitive tools are shown in Section III. The proposed method is described in Section IV. Results and discussion are presented in Section V and security analyses in Section VI. The paper concludes in Section VII.

II. RELATED WORK

Khoa Tran et al. (2020) [15] describe a system that may be used on a website or app with security and lighting monitoring functions. The mobile application lets users control when to turn their devices on and off. SHA-256 was utilized by the finder for authentication. There are two modes for this application: manual and automatic. When IoT devices notice a shift in sensor coverage, they go into automatic mode. Additionally, customers can utilize the smartphone app to operate the device and switch the system on or off. Users can add more sensors to the system without installing or configuring the device, and the system serves as the home's security manager. Nevertheless, the suggested system has certain drawbacks. Every lightbulb it uses requires an ESP8266 chip, which can be costly. It also only uses the most basic forms of authentication (SHA-256), and the transmitted data is not encrypted, which puts it at risk for security issues.

Shuai et al. (2019) [16] proposed an efficient and anonymous authentication scheme for smart home environments using elliptic curve cryptography (ECC). It uses the public key, the private key, and the hash key. The proposed scheme avoids keeping the verification table for authentication purposes. In addition, the random number method is also adopted to resist replay attacks and prevent the clock synchronization problem. The proposed scheme achieves a delicate balance between security and efficiency and is more suitable for realistic environments. However, there are some disadvantages to the proposed system. Data transmitted over the network should be encrypted. In addition to using the XOR method, it is considered an easy method to break.

Fadhil et al. (2021) [17] The current approach uses the lightweight AES algorithm to secure IoT networks. AES algorithm layers such as S-Box, keys, and shifting values are modified based on various chaotic systems as lightweight AES (LAES). The Raspberry Pi device and sensors are the IoT hardware components used in this work. This work aims to secure data from IoT sensors, including temperature, humidity, and flame fire sensors, by encrypting the data using the LAES algorithm before sending it over the network. The following phases distinguish the LAES algorithm layers from the original AES: Shift Rows are replaced with the initial permutation (IP), and Mix Columns are replaced with dynamic Shift Rows. Furthermore, the chaotic logistic map system provides the basis for generating the S-Box used in subbyte operations. The drawback of the research is that in light of the recent change to the AES algorithm—which is vulnerable to hacking—it would have been better to use lightweight algorithms like LED instead of LAES, especially in the context of IoT. In addition, it is better to use a session key (Dragonfly).

Al-Mashhadani and Shujaa (2022) [18] suggested that a group of sensors were linked to the ESP32. The ESP32 chip receives and encrypts the data using AES encryption, then transmits it to a secure internet page requiring access to a username and password. However, there are some drawbacks to this approach. The encryption key is fixed, which makes it vulnerable to piracy, and it would have been better to use a lightweight algorithm like the LED. Additionally, the internet page that requires a username and password can also be hacked (using the owner's biometric for authentication).

Chowdhury et al. (2019) [19] the Launchpad TI-CC3200 is a model with an integrated microcontroller and an internal Wi-Fi shield, providing WPA or WEP security that allows control and management of electrical appliances within the home. In addition, the home security system can be operated without requiring the user to enable a data connection on their phone. The Launchpad connects to a Wi-Fi network within the home or office, and the microcontroller can make decisions or send videos or photos to the owner or guest for further action. However, the drawback of the proposed system is that the board has WEP or WPA security, both of which have been compromised [11]. It would have been better to use WPA3 and Dragonfly technology, which made the TI-CC3200 launcher unsafe.

Savaştürk et al. (2021) [20]used ESP32-CAM in their study. The ESP32-CAM can capture video and images and connect to wireless networks for communication. It encrypts video or images using the AES algorithm; after encryption, the data is sent to the web server and from there to the desired device, such as a mobile phone, tablet, or computer. However, the problem is that the AES algorithm can be broken by brute force attack [21]. Our work has a solution to this problem. The algorithm can be made with a one-session key instead of a static (Dragonfly), and a lightweight algorithm LED can be used. Additionally, the researcher can use a local server instead of a web server to store data in a

database and filter out unwanted data, further enhancing the system's security.

Mohammed (2021) [22] proposes a new secure mutual authentication protocol for remote users based on ephemeral identities and multi-factor authentication for IoT smart building environments. The protocol ensures that only legitimate users can authenticate with smart building controllers anonymously and untraceably. One of the disadvantages of working is that it relies on hash functions only in its work and stores the previous user's location in the database and thus is vulnerable to being hacked.

Yan et al. (2015) [23] created a smart home system and installed the necessary gear and software. Data is transmitted to a home proxy—a mobile phone or tablet—by a group of sensors known as the smart unit. It is a console that is connected to a distant server. A new remote-control method that uses XMPP combines a remote server and home proxy. The system allows several users; a home proxy can help with synchronization issues. A single phone can also register many home proxies, allowing it to manage multiple smart offices or home automation systems. Thus, services for various residences and workplaces can be delivered via the remote server. One disadvantage of the recommended approach is that it is not encrypted, so the shared data is not safe. There was no use of system authentication.

Zaid Ameen et al. (2022) [24] provide symmetric key authentication approaches for smart home networks. The suggested approach uses bitwise exclusive-or operations and one-way hashing, two cryptographic primitives. According to the findings, this plan has the lowest computing, storage, and transmission costs compared to other comparable cutting-edge methods. It offers strong mutual authentication, anonymity, forward key secrecy, backward key secrecy, and unlink ability. Disadvantages are related to security and privacy. This work faces the challenge of impersonation and acquired personality patterns, as the behavioral habits of energy consumers can be inferred by analyzing energy consumption data.

Previous research found many negatives and defects related to security, authentication, and privacy. The goals of smart homes are to reduce computing and communication costs and increase power and speed. Therefore, we found our proposed system to provide security features by using a lightweight encryption algorithm in smart buildings, to reduce cost and an authentication system (Dragonfly) as a one-time session key. Adding biometrics for the owner will increase security between the user and the cloud. Our system can resist many attacks, as discussed in the security analysis and the Scyther tool.

III. PRIMITIVE TOOLS

A. *Dragonfly protocol*

The Dragonfly protocol utilizes the SAE method. This method enables devices to mutually authenticate using a password or passphrase. The SAE protocol only uses the shared password for authentication, avoiding sending the actual passphrase over the wireless medium. This is a countermeasure against offline attacks [25]. In the Dragonfly protocol, the password element (PE) is used instead of the password for computing keys. The PE is determined at session time, using an agreed set of elliptic curve parameters p, which is a large prime number used to define the initial field of the elliptic curve, and q, which is another large prime number in the order of the group g, agreed between the client and AP using discrete logarithmic arithmetic. The catch-and-click technique, with a password as the initial value, is described in Figure 1 [11].

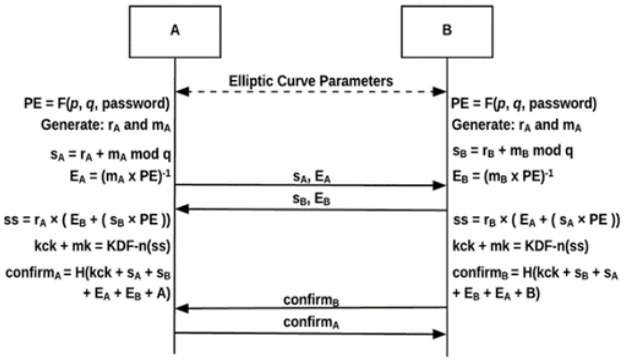


Fig. 1. Dragonfly handshake diagram.

Remark 1: The role of the Dragonfly in our work is to generate a one-session key for authentication. The key generation is the same for E2E. This key is used to encrypt data using the LED algorithm.

B. *The LED block cipher algorithm*

A lightweight encryption algorithm is a symmetric cryptographic method ideally suited for contexts with limited resources, such as low-power devices and IoT devices. It offers effective encryption with a small footprint. It was created to solve the requirement for encryption on gadgets with constrained memory and processing capabilities [26]. LED is an encryption algorithm from the S-PN family [27]. Depending on the key length, it comes in two primary variations. First, there is the LED 64, which has a 64-bit key length, and then there is the LED 128 with a 128-bit key length [28]. Data encryption and decryption were accomplished in this study using the LED 64. The LED refers to the LED 64 version for future usage. Add Constants, S-Box, Shift Rows, and Mix Columns are the four primary functions of the LED algorithm. The Add Constants function seeks to execute XOR operations between a constant matrix and plain text[29].

Remark 2: The role of the LED algorithm in our work is to encrypt the data transmitted within the local network or the internet to maintain data confidentiality. The key is used with an LED algorithm for encryption and decryption.

C. *Scyther*

A tool for analysing, falsifying, and verifying security protocols is called Scyther. It is a cutting-edge, freely accessible tool with unique capabilities not found in other tools[30]. As a tool for formal security analysis protocols, Scyther can quickly investigate protocol properties. Scyther protocols are written using the programming language Security Protocol Description Language (SPDL). Many important cryptographic operations e.g. are sending and

receiving messages between components. The functions that each component plays are supported by SPDL[31]. The users of the GUI are those interested in comprehending or confirming a protocol. The interfaces for scripting and command lines make Scyther easier to use. conduct extensive protocol verification testing[32].

IV. THE PROPOSED METHOD

A robust protocol for confidentiality and authentication was presented in our study. As shown in Figure 2, the proposed secure technology for mobile applications consists of the following components: users (Ui), local databases (LDB), cloud computing (CC), and owners (own)—steps to configure, register, log in, and read data. In the registration phase, a new user interface (Ui) is registered in LDB. The explanation of these stages is as follows:

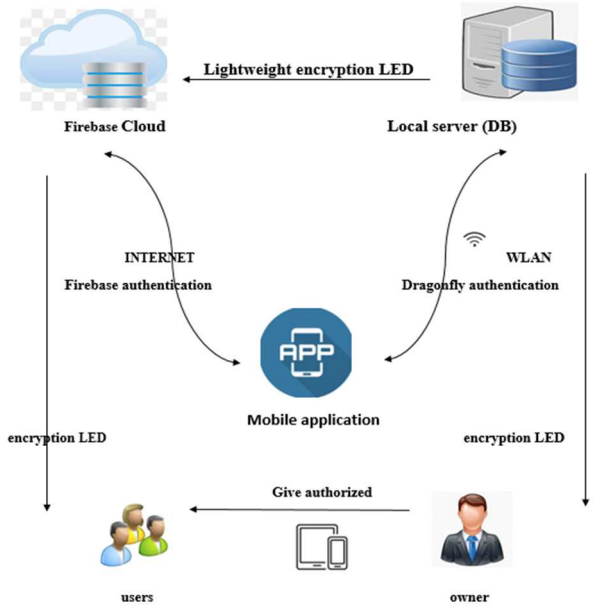


Fig. 2. System design basic components for the authenticated and secure mobile application.

A. Initialization Phase

The local database, which consists of three tables (author-user, test, and abnormal), is configured on the local server (LS). The first table allows authorized users to see the data; the second table holds all the sensor's data, and the third table has the filtered data. The authorized user's table and the filtered data table are the two tables that comprise the CC database. Improved authentication is achieved by using the owner's biometrics as a key for decryption processes. The server program configures the steps.

- Receives the sensor readings.
- Encrypts and decrypts the data using the owner's biometrics key.
- Stores all the data in the database. As shown in the Figure 3.
- Filters the data that is converted to the CC database.

	id	username	password
Edit Copy Delete	1	maher@yahoo.com	123456@@
Edit Copy Delete	2	ahmed@yahoo.com	123456@@

Fig. 3. Figure shows the database of people who are authorized to access the data by entering both ID and PW.

All this data transmitted over WLAN, or the internet, is encrypted using a lightweight LED algorithm. The authentication method is via Dragonfly on the local server.

B. Registration Phase

The owner of the smart building is the one who identifies the people authorized to access the system and read the data stored in the local database LDB or the global database CC and takes the following steps:

Step 1. The owner registers the authorized user using the mobile application based on the user's email (ID) and password (PW) in the table LDB and CC.

Step 2. After the authorized person registers, the owner's biometric data must be transferred to the authorized person's device as a key to decrypt the CC data. To maintain data confidentiality, we transferred the encrypted data with the Dragonfly algorithm, as shown in Figure 4.

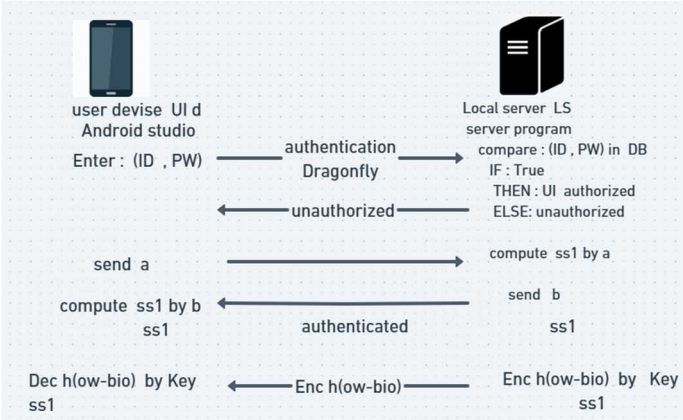


Fig 4. Procedures for sending the owner's biometric data to the user designated upon first registration.

C. Login and data transfer Phase

- In local server (WLAN)

After the authorized users are logged into LDB, the user's device ($Ui\ d$) starts to connect by their log in ID and PW. The Ui is authenticated by Dragonfly protocol. Algorithm 1 and Figure 5 show Dragonfly working for authentication.

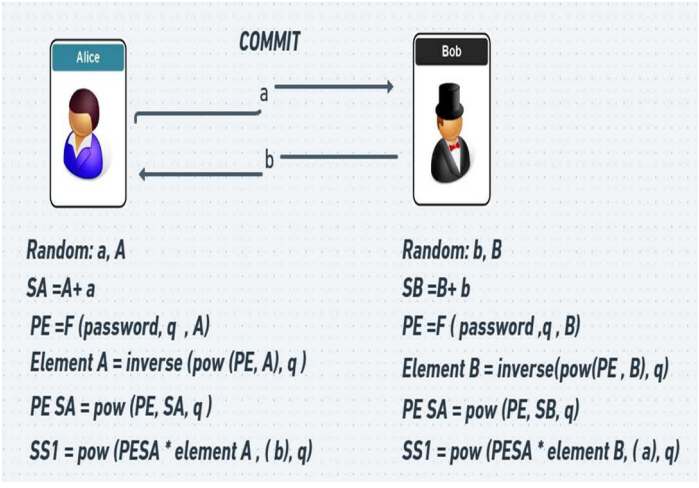


Fig. 5. Figure shows the action of a dragonfly.

Algorithm 1: Dragonfly handshakes (One Session Key)

constant: q, text
Output: Key (SS1)
1: Random: a, A
2: SA =A+ a
3: PE =F (text, q, A)
4: Element A = inverse (pow (PE, A), q)
5: PE SA = pow (PE, SA, q)
6: SS1 = pow (PESA * element A, (b), q)
7: End SS1
8: Random: b, B
9: SB =B+ b
10: PE =F (text, q, B)
11: Element B = inverse (pow (PE, B), q)
12: PE SA = pow (PE, SB, q)
13: SS1 = pow (PESA * element B, (a), q)
14: End SS1

Algorithm 1. This algorithm explains how the Dragonfly protocol works. We impose two random numbers and perform low complexity and robust mathematical operations through them while using one of the random numbers on the other side (a, b). Thus, we can access the session key (SS1), which is either the same key for both parties or a variable key for each connection used if the key is not equal for both parties and the person is unauthenticated.

Step 1. Choose a prime number q and text (constant).
Step 2. Choose the number of ransoms a, A.
Step 3. Use extended Euclidean algorithm gcd = 1.
Step 4. Create the same secure key for both parties (UI and LS).
Step 5. SS1 (one session key).
Step 6. Use SS1 to encrypt data using the LED algorithm and send it to Ui d. Follow the steps in Figure 6.

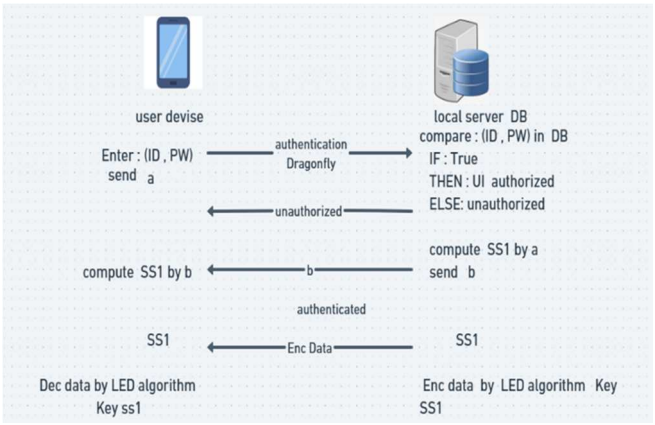


Fig. 6. shows the session key (SS1) and encrypt data using LED.

- In Cloud (Internet)

After the authorized users are logged into the CC, data is stored encrypted within the CC database. Ui d starts to connect by log in ID and PW. Firebase relies on Google authentication.

Step 1. Ui d sends ID and PW to CC.
Step 2. CC compare (ID, PW) in authentication CC database. If is authorized or unauthorized in CC Authentication.
Step 3. Data is sent encrypted to Ui d without a key.
Step 4. To open the encrypted data, the Ui d needs a decryption key and the owner's biometric to decode the data encrypted with the LED algorithm. This is written with detailed pictures in Figure 7.
The encryption algorithm used in the proposed system is LED

Algorithm 2: LED Algorithm (64-bit Key)

Input: Key (SS1), Plaintext (State)
Output: Cipher text
1: For i =1 to 8 ▷ LED *enc*
2: State ← State ⊕ K
3: for m=0 to 3
4: substitute (State)
5: SBOX(State)
6: permute (State)
7: add key (State, rk)
8: end for
9: End For
10: Cipher text ← State ⊕ K
11: For i =8 to 1 ▷ LED *dec*
12: State ← Cipher text (State) ⊕ K
13: for m=3 to 0
14: add key (State, rk)
15: inverse permute (State)
16: SBOX (State)
17: inverse substitute (State)
18: end for
19: End for
20: Plaintext ← State ⊕ K

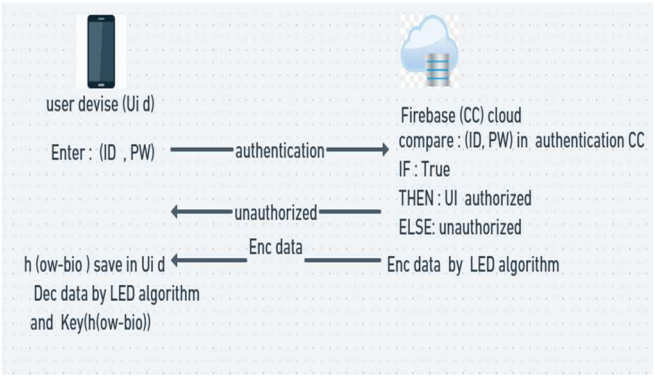


Fig. 7. Mechanism for transferring encrypted data from the cloud to the user.

D. Reading data Phase from local server and from cloud

After completing the registration and accessing the data showing interfaces, we will go through how to log in to the security application in this phase. Currently, Ui d is ready to receive and view the decrypted data of LDB as follows:

- Local server

Step 1. Data is transmitted, authenticated, and encrypted across a local network.
Step 2. Local server (Enc $_{LED}$ (SS1, data) = = Ui d (Dec $_{LED}$ (SS1, data).

Step 3. The data appears in the mobile application (application security) on the decoded local client page.

- Cloud (CC)

Step 1. Data is transmitted, authenticated, and encrypted to Ui d across a global network.
Step 2. To reduce the data stored in CC (normal or abnormal).
Step 3. Normal = every 30 minutes or abnormal = 23c –30c.
Step 4. The key is not transmitted through the network– Key_{ss1} (H (Bio owner)).
Step 5. To decrypt the data, the user takes the key encrypted with SS1 from the local server.

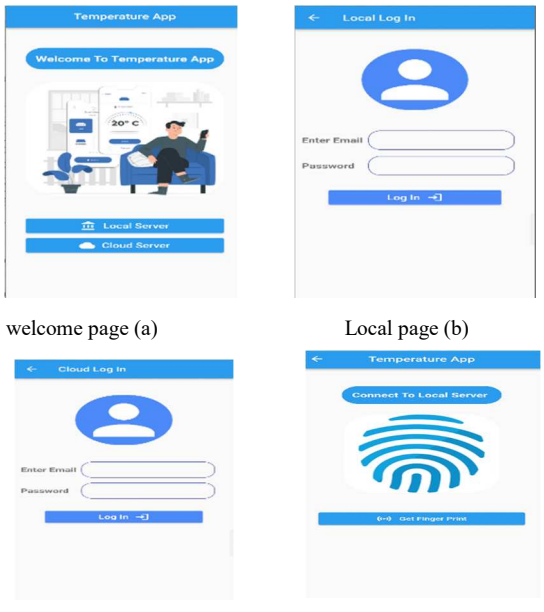
V. RESULTS AND DISCUSSION

In this part, the data has been transferred from the database (XAMPP) into the local server for both users and the cloud. Our work used Android Studio to create the app. It is based on the Flutter platform and the Dart language.

The local server (server program) is in Python language, as mentioned in the initialization phase. The data is transmitted encrypted to authenticate authorized users to decrypt the data, after which the data readings are visible within the application.

To start building a mobile application, we installed the Android Studio program, wrote the code as pages within the program, and then connected it to an actual device. The order of the pages is in the following format:

After the basic steps we have mentioned, the welcome page will appear (a), which consists of two options: permitting to read data from the cloud (c) or from a local server (b). After choosing the cloud, we need the owner's biometrics only once so that the biometrics page appears (d), as shown in the following Figure 8.



welcome page (a) Local page (b)

Cloud page (c) Bio owner (d)

Fig. 8. The mobile app pages are displayed in (a), (b), (c), and (d).

Our scheme enjoys a key merit system. The fast authentication of the system in the Dragonfly process can be shown through the results in Figure 9, in addition to the Dec speed shown in Figure 10. The speed of encryption and sending data to CC is shown in Figure 11.

```
I/flutter ( 5811): Dragon Fly Authintecated in :
I/flutter ( 5811): 233908
I/flutter ( 5811):  Microseconds
```

Fig.9. Time Dragonfly authentication

```
I/flutter ( 5811):  Decryption in  :
I/flutter ( 5811): 3271
I/flutter ( 5811):  Microseconds
```

Fig. 10: Time Decryption.

```
Send Data to Firebase in :
0:00:02.188948 MicroSecound
192.168.0.121 - - [18/Sep/2023 18:45:25
```

Fig.11: Time encryption and Send

A. scyther tool to detect system security

This section demonstrates how the suggested protocol can achieve high levels of privacy and security compared to alternatives by analyzing its security using the Scyther tool, an essential tool for formal security analysis[33]. Scyther has several advantages:

- It is regarded as an infinite means of validation for many security systems, including access control, authentication, and verification.
- It enables you to verify the soundness of the suggested plan for any potential action, including assaults. The proposed schemes must be written in SPDL to be u used [32]. SPDL defines systems and protocols and allows expressions, authentication, encryption, and decryption.

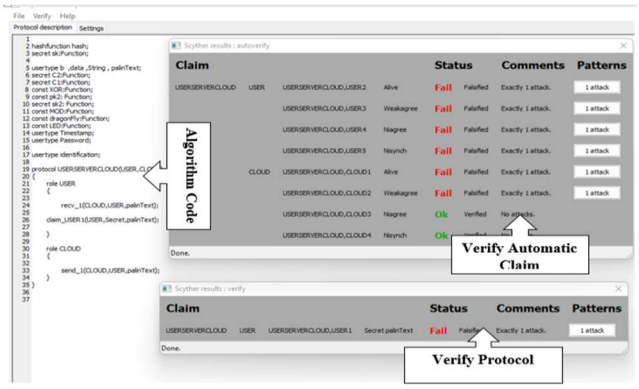


Fig. 12: The system before adding authentication and confidentiality.

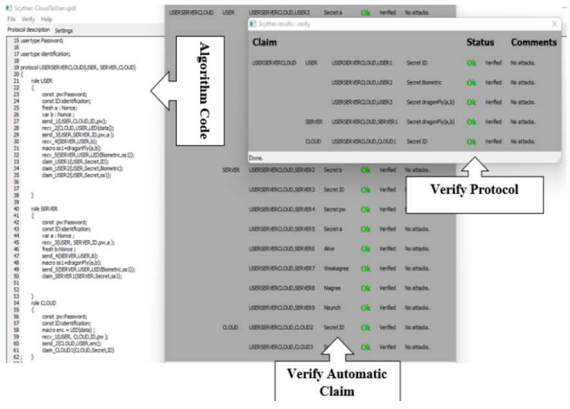


Fig. 13: Verify Protocol and Automatic Climes.

Figure 12. When the system works without authentication and encryption while sending data, it is vulnerable. Figure 13. Authentication and encryption are added to our proposed system, so the verification process is safe and resists hackers.

B. Performance analysis

In this section, the protocol is compared to those in [34], [35], [36], [37], [5] to see how it performs in terms of computational and communication overheads. We used an environment (Windows 11, 8GB RAM, 64-bit, Intel(R) Core(TM) i7-8550U CPU 1.80 GHz 1.99 GHz). We used the language Python 3.11.4 and the Flutter platform with Dart language.

- Computation cost

The time complexity of the suggested scheme is ascertained by calculating the computational cost. The suggested protocol consists of three stages: authentication, encryption, and display data. We specify the computing needs for the one-way hash function T_H, the mathematical operation T_{DRG}, the encryption and decryption with the symmetric key T_{LED}, and the authentication process T_{AUT}. Based on the benchmarks, where the computation cost of cryptographic functions is given in Table I, we compare our work with other relevant efforts.

TABLE I . COMPUTATION COST OF OPERATION COMPARISON RESULT

Operation	General Meaning	Time (second)
T_{DRG}	Mathematical operation (Dragonfly)	0.233
T_h	One-way hash function	0.109
T_{LED}	Symmetric key encryption LED	0.005
T_{LED}	Symmetric key decryption LED	0.003
T_{AUT}	Authentication	0.043
T_{LS}	$T_{DRG} + T_{AUT} + T_{LED}$	0.281
T_{cc}	$T_h + T_{DRG} + T_{LED}$	0.347

TABLE II. COMPUTATION COST COMPARISON RESULT

Protocol	Time Complexity	Result
[34]	$9\,T_h + 2T_{bp} + 2T_{sym} + 5T_{ecc}$	51.51
[35]	$14T_h + 2T_{sym} + 4T_{ecc}$	18.64
[36]	$3h + 2T$	0.644
[37]	$10T_h$	0.8
[5]	$10T_h$	0.8
Proposed LS	$T_{DRG} + T_{AUT} + T_{LED}$	0.281
Proposed CC	$T_h + T_{DRG} + T_{LED}$	0.347

The data transferred to the user via the local server and the cloud needs confidentiality and efficiency, so we use a lightweight algorithm, and the proposed protocol is less time-consuming than related work.

Using the cost of all operations as stated in Table I, we can compute the total cost for authentication, encryption, and display data, explaining the mechanism of total computation cost.) $T_{DRG} + T_{AUT} + T_{LED} <== 0.281$, based on Table II.

- Communication cost

Based on the size in bits for each operation, we compute the communication cost for the proposed scheme. Random numbers (256 bits), block size (128 bits), Dragonfly (SS1 256 bits), hash function (720 bits), and symmetric key encryption (64 bits) are supported. Table III shows a comparison of the communication costs of the relevant protocols.

TABLE III. COMPUTATION COMMUNICATION COST WITH OTHER RELATED WORKS

Protocol	Message length	Number of messages
[34]	1472 *bits*	2 *messages*
[35]	2528 *bits*	2 *messages*
[36]	2688*bits*	4 *messages*
[37]	3136 *bit*	4 *message*
[5]	3136 *bit*	4 *messages*
Proposed LS	1472 *bit*	3 *messages*
Proposed CC	1536 *bits*	2 *messages*

The proposed system (Local server LS) has three exchange messages. The first message represents the Ui registration stage and sends the ID and password. The second and third messages represent the random numbers (a, b) used to obtain SS1, and the lengths of the 256-bit limits are used for this.

Message 1. Ui to LS: The user sends the ID and PW to the local server for registration (1024 bits).

Message 2. Ui a to LS: To extract SS1, send the random number a (224 bits).

Message 3. LS b to Ui : To extract SS1, send the random number b (224 bits).

Total of messages length: 1472 bit, as shown in Table III

In the proposed system (cloud CC), we send two messages: The first represents the user sending the ID and PW to register, while the second represents the owner's biometrics so that they can use them to decode the data.

Message 1. Ui to cloud: The user sends the ID and PW to the CC for registration (1024 bits).

Message 2. Ui to cloud: To extract Bio owner, send the random number a (224 bits).

Total of messages length: 1536 bit, as shown in Table III.

VI. SECURITY ANALYSIS

This section discusses known attacks and explain that our protocol successfully resists such attacks.

Theorem 1. Our proposed method can resist a man-in-the-middle attack

Proof: An attacker could eavesdrop on the communication, tamper with transmitted data, or even impersonate one or both parties involved. Our system can resist eavesdropping attacks because the channel is encrypted with a one-session key and cannot be processed by the Dragonfly protocol to generate a single session key. As shown in Algorithm (1) and Figure 5, impersonation cannot be possible because the attacker does not generate a key because they do not have the initial (static) parameters, as shown in Algorithm 1.

Additionally, in the cloud, the data is encrypted using the owner's biometrics, so the attacker cannot obtain the owner's biometrics, as shown in Figure 4. Preventing unauthorized access to data is done by external and internal attacks.

Theorem 2. Our proposed method can resist a replay attack

Proof: This attack occurs when an attacker copies data packets sent between two parties during a communication session and retransmits them late later[38]. In our system, even if it stores old packets and retransmits them, it will be useless because the key changes in every session.

Theorem 3. Our proposed scheme supports strong verification properties

Proof: Secure verification enables each component to verify the other securely. Licensing by registering authorized users on the local server and in the cloud is specified throughout the registration process. The Dragonfly protocol supports both sides in generating a shared key. This is shown in Figure 5 and Algorithm 1 in steps 6 and 13.

Theorem 4. Our proposed method can resist a dictionary attack and a brute force attack

Proof: Dictionary and brute force attacks are widely used to crack users' passwords online[39],[40]. Our system can resist these attacks by adopting system-based authentication in which the key is fresh for each session, as shown in Algorithm 1 and Figure 5.

Theorem 5. The proposed system is designed to resist attacks by a one-time key and owner biometrics

Proof: In this system, the LED encryption algorithm shown in Algorithm 2 is used, and the encryption key is generated as a single session key using Dragonfly technology in the local network, as shown in Algorithm 1. When using the cloud in the global network, we used the owner's biometrics as the encrypted key to transmit over the network. The user uses Dragonfly authentication technology to further enhance security, as shown in Figure 4.

VII. CONCLUSION

The main barriers to the widespread use of smart home applications are security and privacy concerns. In earlier studies, there are very few thorough authentication techniques appropriate for smart building environments. In this research, we demonstrate a secure and lightweight authentication system based on the Dragonfly handshake, which is a step in the right direction. The suggested technique enables a legitimate user to authenticate with the local server and the cloud by leveraging the database. A symmetric session key (SS1) is produced for future secure connections between the mobile device and the local server upon success. The owner's biometrics allows authenticated users to access each other securely. We use a lightweight LED algorithm to encrypt the transmitted data to increase its security, and the algorithm encryption key is the session key generated by Dragonfly. The proposed scheme has been proven to be safe. Through rigorous formal proof by using Scyther, the results demonstrate the power of our system. Moreover, the official security verification is done using Dragonfly technology. It shows that the proposed scheme can achieve session confidentiality by successfully achieving a mutual authentication key. This has been proven. The proposed method is robust enough to resist many well-known attacks. The results indicate that the proposed system is safe and suitable for the smart home environment.

REFERENCES

[1] U. Farooq, N. Ul Hasan, I. Baig, and N. Shehzad, "Efficient adaptive framework for securing the Internet of Things devices," *Eurasip J. Wirel. Commun. Netw.*, vol. 2019, no. 1, 2019, doi: 10.1186/s13638-019-1531-0.

[2] L. H. Al-Farhani, Y. Alqahtani, H. A. Alshehri, R. J. Martin, S. Lalar, and R. Jain, "IOT and Blockchain-Based Cloud Model for Secure Data Transmission for Smart City," *Secur. Commun. Networks*, vol. 2023, 2023, doi: 10.1155/2023/3171334.

[3] T. Jabeen, I. Jabeen, H. Ashraf, N. Z. Jhanjhi, A. Yassine, and M. S. Hossain, "An Intelligent Healthcare System Using IoT in Wireless Sensor Network," *Sensors*, vol. 23, no. 11, 2023, doi: 10.3390/s23115055.

[4] Z. A. Abduljabbar *et al.*, "Session-Dependent Token-Based Payload Enciphering Scheme for Integrity Enhancements in Wireless Networks," *J. Sens. Actuator Networks*, vol. 11, no. 3, 2022, doi: 10.3390/jsan11030055.

[5] B. A. Alzahrani, A. Irshad, A. Albeshri, and K. Alsubhi, "A Provably Secure and Lightweight Patient-Healthcare Authentication Protocol in Wireless Body Area Networks," *Wirel. Pers. Commun.*, vol. 117, no. 1, pp. 47–69, 2021, doi: 10.1007/s11277-020-07237-x.

[6] M. A. Al Sibahee *et al.*, "Lightweight Secure Message Delivery for E2E S2S Communication in the IoT-Cloud System," *IEEE Access*, vol. 8, pp. 218331–218347, 2020, doi: 10.1109/ACCESS.2020.3041809.

[7] Y. Yao, M. Yang, P. Kiaei, and P. Schaumont, "Dimming Down LED: An Open-source Threshold Implementation on Light Encryption Device (LED) Block Cipher," 2011, [Online]. Available: https://github.com/Secure-Embedded-Systems/Open-Source-Threshold-

[8] M. A. Al Sibahee, A. I. Abdulsada, Z. A. Abduljabbar, J. Ma, V. O. Nyangaresi, and S. M. Umran, "Lightweight, secure, similar-document retrieval over encrypted data," *Appl. Sci.*, vol. 11, no. 24, 2021, doi: 10.3390/app112412040.

[9] F. Thabit, A. P. S. Alhomdy, A. H. A. Al-Ahdal, and P. D. S. Jagtap, "A new lightweight cryptographic algorithm for enhancing data security in cloud computing," *Glob. Transitions Proc.*, vol. 2, no. 1, pp. 91–99, 2021, doi: 10.1016/j.gltp.2021.01.013.

[10] S. M. Umran, S. F. Lu, Z. A. Abduljabbar, and V. O. Nyangaresi, "Multi-chain blockchain based secure data-sharing framework for industrial IoTs smart devices in petroleum industry," *Internet of Things (Netherlands)*, vol. 24, no. September, 2023, doi: 10.1016/j.iot.2023.100969.

[11] C. P. Kohlios and T. Hayajneh, "A comprehensive attack flow model and security analysis for Wi-Fi and WPA3," *Electron.*, vol. 7, no. 11, 2018, doi: 10.3390/electronics7110284.

[12] M. Fareed and A. A. Yassin, "A lightweight and secure multilayer authentication scheme for wireless body area networks in healthcare system," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 2, pp. 1782–1794, 2023, doi: 10.11591/ijece.v13i2.pp1782-1794.

[13] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 1005–1019, 2018, doi:

10.1016/j.future.2016.12.028.

[14] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6917 LNCS, pp. 326–341, 2011, doi: 10.1007/978-3-642-23951-9_22.

[15] T. A. Khoa *et al.*, "Designing Efficient Smart Home Management with IoT Smart Lighting: A Case Study," *Wirel. Commun. Mob. Comput.*, vol. 2020, 2020, doi: 10.1155/2020/8896637.

[16] M. Shuai, N. Yu, H. Wang, and L. Xiong, "Anonymous authentication scheme for smart home environment with provable security," *Comput. Secur.*, vol. 86, pp. 132–146, 2019, doi: 10.1016/j.cose.2019.06.002.

[17] M. S. Fadhil, A. K. Farhan, and M. N. Fadhil, "A lightweight aes algorithm implementation for secure iot environment," *Iraqi J. Sci.*, vol. 62, no. 8, pp. 2759–2770, 2021, doi: 10.24996/ijs.2021.62.8.29.

[18] M. Al-Mashhadani and M. Shujaa, "IoT Security Using AES Encryption Technology based ESP32 Platform," *Int. Arab J. Inf. Technol.*, vol. 19, no. 2, pp. 214–223, 2022, doi: 10.34028/iajit/19/2/8.

[19] S. S. Chowdhury, S. Sarkar, S. Syamal, S. Sengupta, and P. Nag, "IoT Based Smart Security and Home Automation System," *2019 IEEE 10th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2019*, pp. 1158–1161, 2019, doi: 10.1109/UEMCON47517.2019.8992994.

[20] P. Savaştürk, Ö. Aydın, and G. Dalkılıç, "AES Encrypted Real-Time Video Stream and Image Transmission from ESP32-CAM," *SSRN Electron. J.*, vol. 17, no. 4, pp. 447–452, 2022, doi: 10.2139/ssrn.4171323.

[21] L. I. B. Mahendra, Y. K. Santoso, and G. F. Shidik, "Enhanced AES using MAC address for cloud services," *Proc. - 2017 Int. Semin. Appl. Technol. Inf. Commun. Empower. Technol. a Better Hum. Life, iSemantic 2017*, vol. 2018-Janua, pp. 66–71, 2017, doi: 10.1109/ISEMANTIC.2017.8251845.

[22] M. M. Alshahrani, "Secure multifactor remote access user authentication framework for iot networks," *Comput. Mater. Contin.*, vol. 68, no. 3, pp. 3235–3254, 2021, doi: 10.32604/cmc.2021.015310.

[23] Soukaena Hassan and M. Abd Zaid, "Modification Advanced Encryption Standard for Design Lightweight Algorithms," *J. Kufa Math. Comput.*, vol. 6, no. 1, pp. 21–27, 2019, doi: 10.31642/jokmc/2018/060104.

[24] V. O. Nyangaresi *et al.*, "Energy Efficient Dynamic Symmetric Key Based Protocol for Secure Traffic Exchanges in Smart Homes," *Appl. Sci.*, vol. 12, no. 24, 2022, doi: 10.3390/app122412688.

[25] N. Tschacher, "Why a new handshake ?," 2018.

[26] R. Ayachi, A. Mhaouch, and A. Ben Abdelali, "Lightweight Cryptography for Network-on-Chip Data Encryption," *Secur. Commun. Networks*, vol. 2021, 2021, doi: 10.1155/2021/9943713.

[27] P. Panahi, C. Bayılmış, U. Çavuşoğlu, and S. Kaçar, "Performance Evaluation of Lightweight Encryption Algorithms for IoT-Based Applications," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 4015–4037, 2021, doi: 10.1007/s13369-021-05358-4.

[28] W. El Hadj Youssef, A. Abdelli, F. Dridi, and M. Machhout, "Hardware implementation of secure lightweight cryptographic designs for IoT applications," *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/8860598.

[29] H. M. Al-Saadi and I. S. Alshawi, "Provably-Secure LED Block Cipher Diffusion and Confusion Based on Chaotic Maps," *Inform.*, vol. 47, no. 6, pp. 105–114, 2023, doi: 10.31449/inf.v47i6.4596.

[30] N. Z. Almuzaini and I. Ahmad, "Formal Analysis of the Signal Protocol Using the Scyther Tool," *2nd Int. Conf. Comput. Appl. Inf. Secur. ICCAIS 2019*, pp. 1–6, 2019, doi: 10.1109/CAIS.2019.8769532.

[31] M. K. Kabier, A. A. Yassin, Z. A. Abduljabbar, and S. Lu, "Role Based Access Control Using Biometric the in Educational System," *Basrah Res. Sci.*, vol. 49, no. 1, pp. 85–101, 2023, doi: 10.56714/bjrs.49.1.8.

[32] M. H. Alzuwaini and A. A. Yassin, "An Efficient Mechanism to Prevent the Phishing Attacks," *Iraqi J. Electr. Electron. Eng.*, vol. 17, no. 1, pp. 125–135, 2021, doi: 10.37917/ijeee.17.1.15.

[33] N. Hamed and A. Yassin, "Secure Patient Authentication Scheme in the Healthcare System Using Symmetric Encryption," *Iraqi J. Electr. Electron. Eng.*, vol. 18, no. 1, pp. 71–81, 2022, doi: 10.37917/ijeee.18.1.9.

[34] S. Khatoon, S. M. M. Rahman, M. Alrubaian, and A. Alamri, "Privacy-Preserved, Provable Secure, Mutually Authenticated Key Agreement Protocol for Healthcare in a Smart City Environment," *IEEE Access*, vol. 7, pp. 47962–47971, 2019, doi: 10.1109/ACCESS.2019.2909556.

[35] A. Ostad-Sharif, D. Abbasinezhad-Mood, and M. Nikooghadam, "An enhanced anonymous and unlinkable user authentication and key agreement protocol for TMIS by utilization of ECC," *Int. J. Commun. Syst.*, vol. 32, no. 5, pp. 1–23, 2019, doi: 10.1002/dac.3913.

[36] X. Liu, R. Zhang, and M. Zhao, "A robust authentication scheme with dynamic password for wireless body area networks," *Comput. Networks*, vol. 161, pp. 220–234, 2019, doi: 10.1016/j.comnet.2019.07.003.

[37] Z. Xu, C. Xu, H. Chen, and F. Yang, "A lightweight anonymous mutual authentication and key agreement scheme for WBAN," *Concurr. Comput. Pract. Exp.*, vol. 31, no. 14, pp. 1–12, 2019, doi: 10.1002/cpe.5295.

[38] G. 2021. 3059648. pdfja. Suthokumar, V. Sethu, C. Wijenayake, and E. Ambikairajah, "Modulation dynamic features for the detection of replay attacks," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2018-Septe, no. September, pp. 691–695, 2018, doi: 10.21437/Interspeech.2018-1846.

[39] A. B. Puthuparambil and J. J. Thomas, "Freestyle, a randomized version of ChaCha for resisting offline brute-force and dictionary attacks," *J. Inf. Secur. Appl.*, vol. 49, p. 102396, 2019, doi: 10.1016/j.jisa.2019.102396.

[40] Z. A. Abduljabbar *et al.*, "SEPIM: Secure and efficient private image matching," *Appl. Sci.*, vol. 6, no. 8, pp. 1–21, 2016, doi: 10.3390/app6080213.