Mahdi Mohammad Abdullah Al Momani[1], Dr. P. S. Puttaswamy[2]

Secure IoT Middleware Protocol with Multi-Key Authentication and Adler32-Based Session Key Generation

JES Journal of Electrical Systems

Abstract: - The proliferation of Internet of Things (IoT) devices across several industries has heightened the requirement for communication protocols that assure consumer data integrity and privacy. This study presented a novel authentication framework named Secure IoT Middleware Protocol, with multi-key authentication and session key generation utilizing Adler32 for secure communication in IoT settings. The protocol utilizes Blockchain technology for tamper-proof data validation and decentralized authentication, eliminating single points of failure to reduce zero-trust issues. Generate secret keys dynamically for each communication session to safeguard device-to-device interactions between validated user identities using multi-key techniques. It utilizes the MQTT protocol, facilitating far more efficient data transport than HTTP. The protocol utilizes advanced cryptography technology and Blockchain smart contracts to establish a scalable, trustworthy, and transparent infrastructure for IoT communication. Experimental results indicate that despite increasing users, the selective broadcast secret key and reaction times are minimized. At the same time, the computational overhead produced by the proposed system remains relatively constant, yielding satisfactory security and performance from our system. This research can advance authentication, data protection, and secure key management solutions in significant IoT security concerns.

Keywords: Adler32 Session Key Generation, Blockchain Technology, Decentralized Authentication, Multi-Key Authentication, Message Queuing Telemetry Transport (MQTT), IoT Security and Secure IoT Middleware**.**

INTRODUCTION

The Internet of Things (IoT) is a network enabling devices such as sensors and actuators to transmit data to the Internet via a processor (Sethi & Sarangi, 2017). It can transmit data from one physical thing to another across the Internet. Recently, the Internet of Things (IoT) has become integral to our daily existence. The Internet of Things (IoT) is ubiquitous in vehicles, home automation devices, intelligent office systems, and fitness trackers. The IoT network paradigm posits that the digital and physical realms can be interconnected to facilitate collaboration among individuals or physical entities in achieving their objectives through this integration. Rose et al. (2015) assert that the Internet of Things (IoT) expands computational and networking capabilities to non-computer items, facilitating data generation, sharing, and exchange autonomously, without human intervention. The Internet of Things (IoT) denotes a versatile worldwide network framework characterized by self-configuring capabilities established on open communication standards, wherein intelligent network-integrated interfaces are embedded in physical and digital objects.

The IoT project articulated by Minerva et al. (2015) amalgamated numerous definitions from diverse sources and environmental contexts to build an appropriate IoT definition encompassing various things. Research on small-scale architecture states that an IoT network links uniquely identifiable things to the Internet. Entities can perceive, respond, and exhibit a certain level of programmability. Data regarding tangible objects and the capacity to modify their condition can be acquired using distinct identification and sensing technologies. Connected devices are programmable and encompass sensors, actuators, and unique digital identities. The depiction details item identity, status, location, and any additional information requisite for business, personal, or social interactions due to their distinctive methods of self-identification, data acquisition, communication, and autonomous action based on that data. The service is accessible at all times and for any purpose, with security considerations integrated through

[1]Research Scholar, PET Research Center, Mandya
[2]Prof. and Head, Adichunchagiri University, B G Nagar, Bellur

intelligent interfaces. The foundational elements of IoT encompass computing, communication, sensing, and action (Voas, 2016).

IoT Security

The growth of the IoT is transforming human lives and how organizations function. Due to various issues, the fast deployment of IoT technology carries severe cybersecurity risks. This section discusses how IoT security vulnerabilities affect users.

IoT Threats

As the IoT becomes more widespread, things and IoT systems deployed become increasingly vulnerable to cyberattacks. With the IoT, you inheritthe security weaknesses of your connected devices and networks, such as wireless and sensor networks (Zanella et al., 2014). Attacks on physical devices and assaults on communication protocols and IoT tools are thus all possibilities in terms of IoT system security. IoT threats also arise due to the numerous difficulties in implementing strong security measures. Implementing IoT security solutions has its share of challenges.

Security Threats in IoT

Cyber security is the concern while protecting data in software and its systems. Security for IoT devices has been left to fend for itself. The application layer attack is more vulnerable than the other attack presentations in the IoT layer (Ashraf et al., 2020).

The security threats in different layers of IoT protocols have three tiers of security and take into consideration the following:

(i) Design Security: To build a secure framework, there is a need to begin at thefoundational level. Ex: IP protection

(ii) Hardware Security: Tough environments necessitate tamper-resistant devices.

(iii) Data Security: Cryptography can protect and prevent IoT data.

REVIEW OF LITERATURE

The Internet of Things has emerged as a transformative technology applied across various sectors, including smart cities, healthcare, transportation, and industrial automation. The increasing popularity of quick adoption has resulted in considerable security and data privacy problems, prompting extensive study into viable mitigating techniques. IoT devices are essential to human existence, facilitating user data collection and interconnectivity between platforms with unprecedented ease. Nikooghadam and Amintoosi (2023) examined integrating IoT devices with cloud servers to mitigate constraints related to storage and processing capabilities. This integration enhances the use of IoT but necessitates sophisticated authentication systems for secure communication. Kumari et al. (2018) identified susceptibilities to replay and stolen-verifier attacks within the ECC-based system. The authors proposed a lightweight protocol to ensure mutual authentication, user anonymity, and forward secrecy, validated by BAN logic and Scyther tools. This protocol demonstrated a minimal computational cost and is suitable for the IoT environment with constrained capacity.

Uppuluri and Lakshmeeswari (2023) focused on smart home device communication, an essential aspect of IoT. They proposed a Modified Honey Encryption (MHE-IS-CPMT) among other peer-to-peer devices adopting Inverse Sampling - Conditional Probability Model Transform and ECC for improved key negotiation and provide authentication. Our model supports secure initialization, registration, login, session key agreement, and dynamic vital updates. By ensuring that the link between the user and device is secured and difficult to attack, the protocol showed improved security attributes compared to current methods countering most vulnerabilities in smart home IoT networks.

Chi Ho Lau (2023) discussed the issues of authenticity and privacy in different types of devices in IoT networks. The second aspect model secure networks, and to this end, the study proposed the Authenticated Device Configuration Protocol (ADCP) and an Authenticated Device Transmission Protocol, which can block zero-day attacks and allow for zero round trip time key exchange. The protocols use blockchain to record tests

(authentications) and data integrity without reprograming software. Formal analysis verified its resistance to attacks, and a stochastic threat model showed that network security was dramatically improved.

Al Ahmed et al. (2023) examined the limitations of centralized key exchange servers in IoT networks and proposed the Authentication-Chains protocol based on Blockchain technology. It is a lightweight protocol that addresses identity spoofing and data tampering while optimizing resource use for restricted IoT smart sensors. This solution facilitates device authentication efficiently while enhancing the confidence and transparency of the methods employed. It has numerous characteristics for augmenting IoT security and facilitating effective applications across various domains. Manikandan et al. (2023) indicated that to address these challenges, machine learning (ML) techniques, including federated and differential learning, are partially utilized to protect digital information. Nonetheless, issues about data safety, particularly for federated learning-oriented algorithms, continue to constitute a significant study domain in local and worldwide datasets.

Chandrakar et al. (2022) argued that embracing blockchain technology is one promising solution to IoT security. The decentralized nature of blockchain allows IoT devices to communicate with each other securely, providing solutions for challenges such as authentication or data integrity in open environments. The novel secure communication methods have shown improvement over prior protocols, especially in reducing security threats from autonomous IoT networks. Authentication is one of the key aspects of IoT security. Although the loss of information mentioned above is a temporary sabotage, Zhu et al. (2022) emphasized that existing identity-based authentication protocols are vulnerable to physical attacks and temporary loss of information. Our previous design proposed a strong protocol based on real-or-random (ROR) modeling and BAN logic to overcome these limitations, ensuring secure communication. However, efficiency is somewhat low in IoT environments.

Conventional security algorithms cannot be accommodated since many IoT devices are resource-constrained regarding memory and processing capability. Hajian et al. (2021) and Kanini (2022) enforced some lightweight authentication mechanisms and suggested novel device-to-device (D2D) mutual authentication protocols, respectively, to improve the security of IoT. They reduce communication expenses and handle significant attacks like key compromise impersonation (KCI) and clogging assaults. Xu et al. (2021) point out the absolute need for encryption for data transfer to edge servers in IoT. The study proposed an identity-protected symmetric encryption model between security and CSI complexity, in which its secure storage and transmission activities can be conducted based on the general processes. Saqib et al. (2021) used identity, passwords, and digital signatures to prevent cryptographic attacks while preserving the computation time efficiency and contributed by providing a three-factor authentication framework for IoT applications.

OBJECTIVES OF THE STUDY

The objectives chosen for the research work are as follows:

1. To design and develop a user authentication and authorization protocol with IoT gateway security as middleware using a multi-key mechanism for authentication.

2. To provide additional security by designing and developing a protocol that uses the Adler32 hashing to generate session keys based on the request ID and timestamp.

Schematic of Secured Authentication System

The proposed secured authentication communication network for the IoT environment is shown below in Figure 1. The implementation involves the following methodology to develop a secure platform for the IoT Environment: (i) Authentication, (ii) Contract, and (iii) Communication (Subathra et al., 2022).

The following section shows the steps followed in each one of these phase implementations, and these phases are pictorially represented.

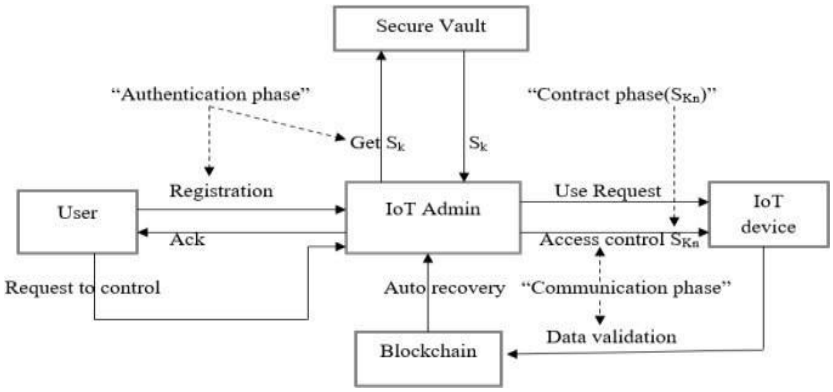

Figure 1: Schematic of Secured Authentication Mechanism

1. **The Authentication Phase involves the following.**

   (i). To configure the user device and manage the user profile

   (ii). Data security key generation

   (iii. Distribution and mapping of the user profile with keys

   (iv). Accept user request

   (v). Get secret key (Sk) from Secure Vault for the User (U), represented as U_{Sk}

   Secure Vault: The secure vault contains n keys, each m bits long. The value of m is the key size. We denote all the keys as K[0], K[1], K[2],…..K[n-1]. During the deployment of the IoT device, the secure vault is shared between the device and the server. The vault should be stored in an encrypted format on the device. On the server, secure vaults are stored in a secure database. The secure vault generates the secret key for the User (U)=$U_{sk.}$

   (vi). Encrypt using a key to generate Secure key EU_{sk}

   (vii). Share Secure key to User EU_{sk}

   (viii). The User should Decrypt EU_{sk} to identify the key and return back DU_{sk}

   (ix). Validate EU_{sk}= DU_{sk}, and identify the User as a valid user to create a contract.

2. **Contract Phase** deals with the following

   (i). The contract phase generates a session key (SKn) for the valid user after the authentication phase.

   (ii). The Server should distribute S_{Kn} to the User and IoT environment for any further transactions.

   (ii). Each session key (S_{Kn}) has been assigned with timeout for the contract and expires automatically based on timeout (Sakhare et al., 2023).

   Authentication process: Before user registration, the admin has to log in to allow the user to communicate further. The steps are explained below.

1. **Admin login**: Admins are responsible for managing the user access control list and permissions for IoT devices. Their main task is to manage the registration and de-registration of IoT devices and nodes in the system. Furthermore, the admin permits the end users to access IoT devices.

2. **Cloud**: The cloud hosts compute and storage servers that aggregate and store IoT data. The data can then be subjected to heavy processing and analytics by the cloud servers.

3. **Registration Phase**: The steps involved in this phase are explained below.

   To register with the cloud server S, the embedded device Di sends a unique IDi to the server. On receiving this request, the cloud server generates a unique password Pi for every device Di, as given below.

   Embedded Device Di◊ Server S: IDi 268, Server S generates Pi.

Authentication Phase: In this phase, the embedded device and cloud server mutually authenticate each other using ECC parameters.

The server selects a unique random number Ri for every device and generates a cookie CK = H(Ri | X | EXP_TIME | IDi) where X is the private key of the server and stores the cookie on the embedded device 2 as ECC point CK" = CK × G. The server also calculates the security parameters Ti = Ri⊕H(X), Ai = H(Ri⊕ H(X) ⊕ Pi⊕ CK") and stores Ai "272 = H(Ri⊕ H(X) ⊕Pi⊕ CK") × G, Ti corresponding to the identity IDi of the device Di in its database. The server stores the expiration time of the cookie EXP_TIME corresponding to a particular embedded device's identity. When the cookie expires, the expiration time is updated to EXP_TIME," and the cookie is updated as CK = H(Ri | X | EXP_TIME" | IDi).

Server -> Embedded Device Di: Before every login, the device selects a random number N1, calculates an ECC point P1= N1 × G, and stores it in its memory. Embedded Device Calculates ECC point P1 To login with the cloud server, the device calculates the ECC point P2= H(N1×CK") and sends the P1, P2, and its IDi to the server. Embedded Device ◊ Server: IDi, P1, P2 The code developed for this purpose presented below is the user authentication function for IoT devices.

Implementation of Authentication and Authorization

Working Architecture consists of three levels, and they are

(i) Client application

(ii) IoT operations and

(iii) Blockchain server.

Client Application:

The working of the Client application is as follows:

- The MainMenu class is a subclass of Form, representing the application's main window.
- In the constructor MainMenu(), the InitializeComponent() method is called. This method is typically auto-generated by the Visual Studio designer and initializes the components of the Form.
- There are event handlers for the menu items:
- registerToolStripMenuItem_Click: This method is called when clicking the "Register" menu item. It creates an instance of the Registration form (Registration reg = new Registration ();), sets the main Form (this) as its MDI parent (assuming it's an MDI application), and then shows the registration form.
- checkStatusToolStripMenuItem_Click: Similar to the previous method, but for checking status. It creates an instance of the CheckStatus form (CheckStatus chk = new CheckStatus();), sets the main Form (this) as its MDI parent, and then shows the check status form.
- exitToolStripMenuItem_Click: This method is called when clicking the "Exit" menu item. It iterates through all open forms in the application, closes them, and then exits the application.
- requestToolStripMenuItem1_Click: This method is called when clicking the "Request" menu item. It creates an instance of the Form1 form (Form1 frm = new Form1();), sets the main Form (this) as its MDI parent, and then shows the Form1 form.

This code sets up a basic client application with a main menu and functionality to register, check status, and make requests.

Flowchart: Client Application

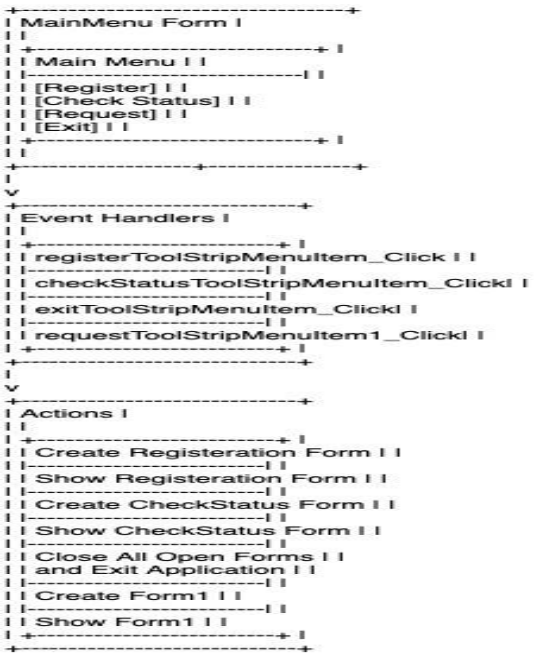

Figure 2: Flow chart for Client Application

This flowchart shown above represents the structure of the Main Menu form, the event handlers for each menu item click event, and the corresponding actions taken in each event handler.

Secure Authentication: Effective authentication mechanisms ensure that only authorized individuals or devices can access the network. This includes strong password policies, multi-factor authentication, and other authentication methods like biometrics or digital certificates (Sakhare, et al., 2023).

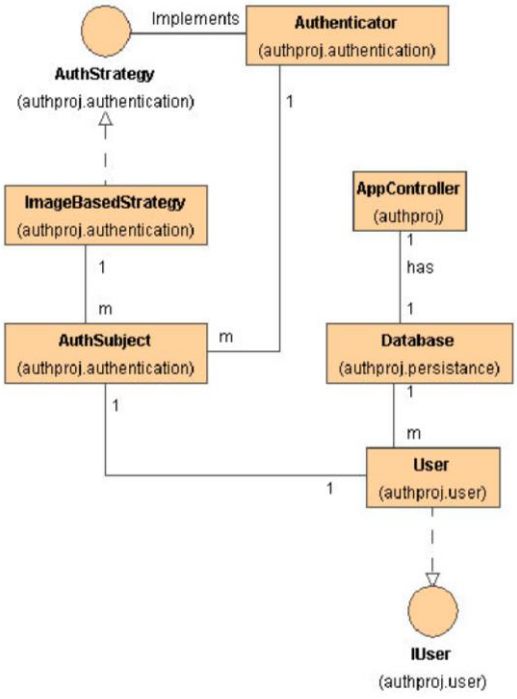

Figure 3: Secure Authentication Mechanism

Source: Singh et al. (2021).

Implementation of the Proposed Methodology

There are two folders: **AdminConsole** and **UserApp**. Microsoft Visual Studio 2010 or above needs to be installed on the computer to run this. This has to run on two computers but can even run on one computer. Now, go to the UserApp folder. In this, there is a file called Microsoft Visual Studio Solution. Select that file with a right-click, and then it opens with Microsoft Visual Studio code above 2010 versions (Singh et al., 2021). Once the file is open in the project, then go to view and select Solution Explorer; when all the files connected to the project are displayed, Go to Build and select Build Solution so that after compiling everything, it will show that Build Succeeded, meaning all the things are compiled and on running this a new window is displayed.

The proposed method has been implemented for validity, and the implementation flow chart is shown in Figure 4.

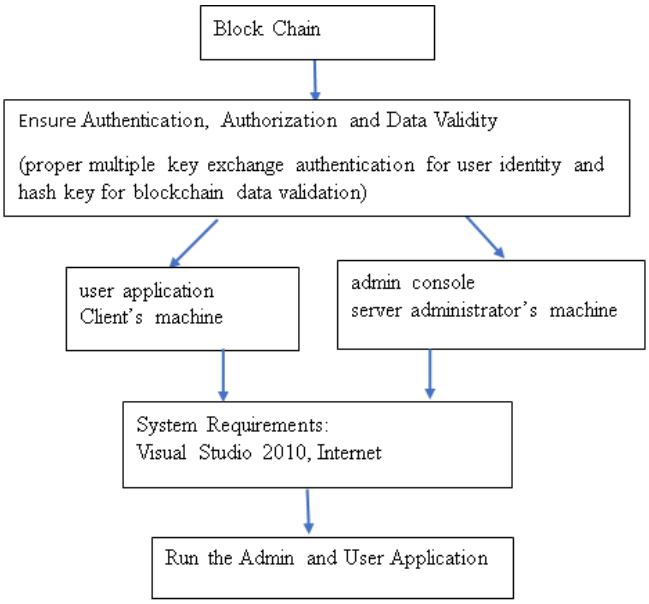

Figure 4: Implementation Flow chart

Role of the Consoles: The Admin Console is used to start the connection service, approve/disapprove connection requests from a user, and moderate the connecting devices. The client will use the User Application to mainly control the IOT device by connecting to the network and making a connection request.

Running the Admin and User applications

In the present system, open the 'AdminConsole' folder first to run the Admin console. Next, open the 'AdminConsole.sln' file using Visual Studio (Xie, 2016).

The following screen is opened and displayed when this is executed, as shown in Figure 5.

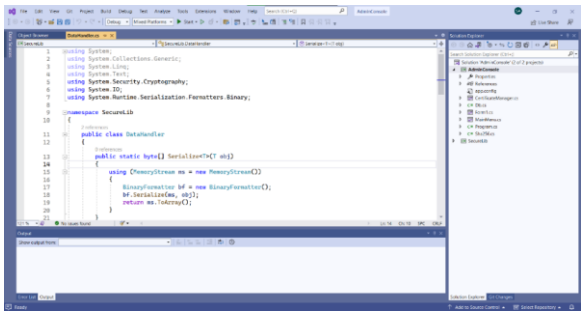

Figure 5: Admin Console

When the screen is opened, click 'Build' on the Menubar, then select 'Rebuild solution' from the drop-down, as shown in Figure 6.

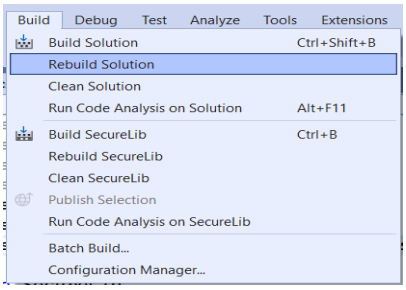

Figure 6: Rebuild Admin.sln file

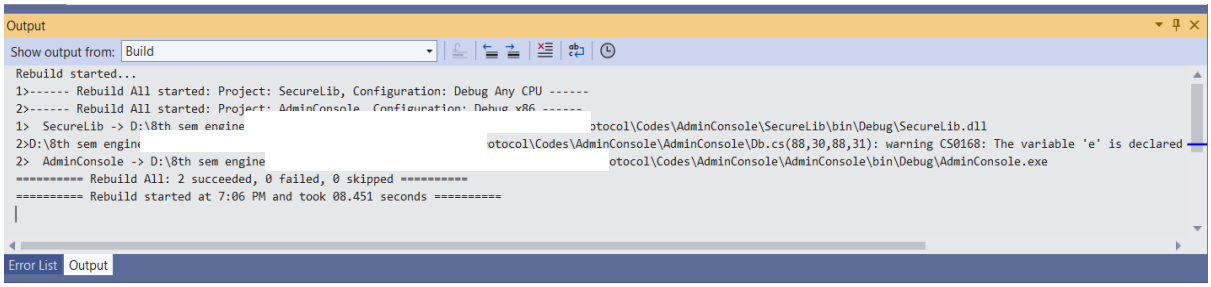

Figure 7: Admin Console code rebuild

Once the rebuild process is completed, click the 'Start' button on the toolbar or press the F5 button to run the program and start the console. After this, a new admin console window will be created. This is the window/console that distinguishes normal users from an administrator since only the administrator console can control the user base of the IoT server and start/stop the IoT server on the go. Admin Console window is displayed (Hou, et al., 2017).

Perform the same steps for the 'UserApp' code in the client machine, which will open a small window, as shown in Figure. Once the process is completed, the app will be available for use by the user.

Figure 8: User Application Preface screen

The window contains information regarding the Internet of Things as a whole for the user's basic knowledge. It also details the risks and security concerns involving the IoT as a whole to the user.

Click on the green-colored right arrow button to open the main User application window, and then the application is as shown in Figure 8.

Now click on the green arrow, and it will navigate to the next screen, which is the welcome screen for the user. This forms the user and the admin part and runs on different computers or the same computer.

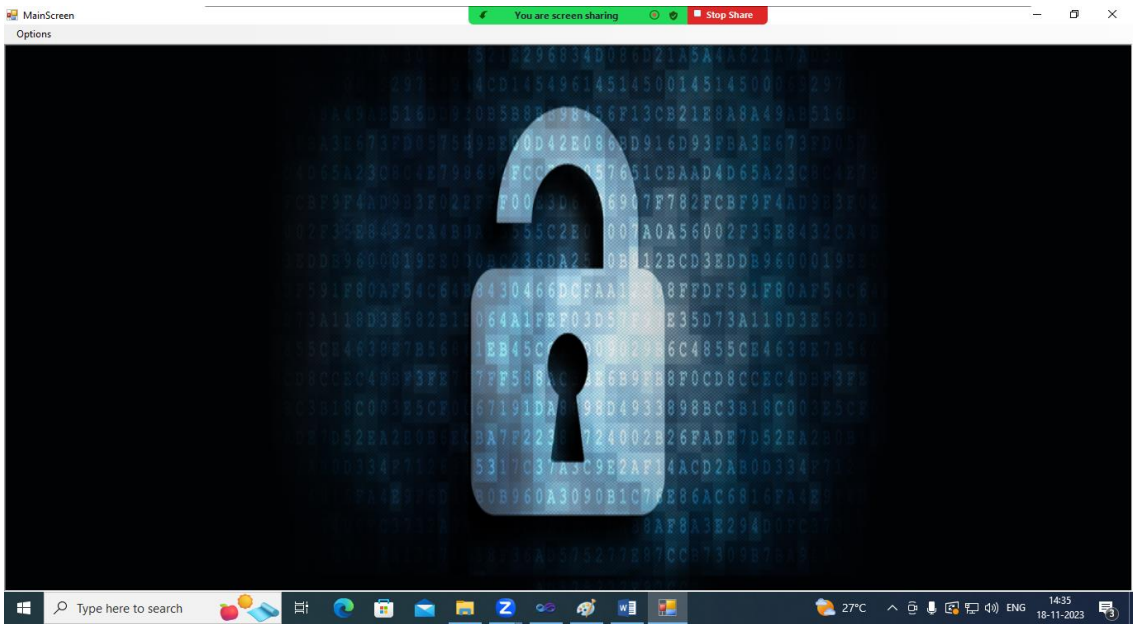

Figure 9: User application window

Now go to the AdminConsole folder, and there will be a Microsoft Visual Studio Solution file. Open the file with Visual Studio 2010 or above versions. Next, using the same procedure, we will get all the files, go to build select, build the solution, and run it for access to the administration part.

Adler-32 Algorithm

Hash is created for each entry in the Blockchain ledger. In order to create hash, authors use Adler-32 algorithm. It is a popular checksum algorithm designed to detect corruption in the data. Since it is faster than the other checksum algorithms, Adler-32 is chosen for the current work. If the uncompressed information does not match with the Adler32 checksum, the application can notify its protocol or the handler that the information is corrupted (Yu, et al., 2017).

In order to get an Adler-32 checksum, we have to calculate the two 16-bit checksums P and Q and adding their bits into a 32-bit integer. P is the aggregate of all bytes in the stream in addition to one and Q is the aggregate of the individual values of P from each phase.

At the start of an Adler-32 run, P is set to 1, Q to 0. The sums are done modulo Pn (the highest prime number). The bytes are stored in network order, Q occupying the two most significant bytes.

An Adler-32 function can be explained as

$$P = 1 + Z1 + Z2 + \ldots + Zn \ (\text{mod } Pn)$$

$$Q = (1 + Z1) + (1 + Z1 + Z2) + \ldots + (1 + Z1 + Z2 + \ldots + Zn) \ (\text{mod } Pn)$$

$$Q = n^* Z1 + (n-1)^* Z2 + (n-2)^* Z3 + \ldots + Zn + n \ (\text{mod } Pn)$$

$$\text{Adler-32}(Z) = Q^*(Pn + \text{Checksum}) + P$$

where Z is the string of bytes for which the checksum is to be calculated, Pn is the highest prime number and n is the length of Z (Yu, et al., 2017).

CONCLUSION

This paper describes a secure middle-ware protocol in IoT environment with multi-key authentication and session key generation based on adler32. The ZKP-based authentication mechanism protects IoT data in a secure and robust way. Blockchain in IoT devices makes it impossible and prevents tampering by ensuring tamper-proof identity management coupled with data validation. Blockchain is used to track any change in data modification and theft to resolve forgery issues over RFID and the alteration of data. This study highlights the importance of

authentication for protecting digital systems by preventing illegal access and making users accountable. This effectively applies strong user identity verification like multi-key authentication, and it can register user machines with auto-fetch systems to provide access only when approved by an admin. The secret key is per session, and instead of it being static as is normal with a cookie-based implementation, it is effectively dynamic (per request) with Adler32 hashing done on the request-id along with a timestamp. MQTT protocol provides the best communication conditions between IoT devices and users, surpassing HTTP in terms of message delivery reliability, response time, and throughput.

Blockchain enables decentralized Authentication, that removes simple failure Wiki and reduces unauthorized access risks. Smart contracts monitor and identify the authentication process, reduces human error while increasing the efficiency of transactions. The ledger for each authentication transaction is immutable, generating a transparent audit trail and facilitating the identification of any unauthorised actions.

With asymmetric encryption, you can exchange keys securely while keeping them secure as well. It provides data encryption, digital signatures and identity proofreading through public-private key infrastructure along with hashing algorithms. The homomorphic encryption allows the carrying of computations on the encrypted data, thus securing sensitive information in IoT environments. MQTT outperformed HTTP in all respects throughout the experiment (lower overhead per message, higher average response time for increasing users and message counts). The multi-ledger concepts in Blockchain allow even manipulation transactions to be recoverable events in the system's life cycle.

Suggestions

For a more robust design and development of a secure authentication contract communication network for an IoT environment, the following suggestions were listed:

- **Multi-factor authentication**: The implementation considered in the present work is single-factor authentication, with admin approval. Implement a robust authentication mechanism that requires multiple factors such as passwords, biometrics, or hardware tokens to verify the identity of IoT devices.
- **Certificate-based authentication**: Employ certificate-based authentication to ensure the authenticity of IoT devices. Each device should have a unique digital certificate signed by a trusted certificate authority.
- **Secure communication protocols**: Use secure protocols like Transport Layer Security (TLS) or Datagram Transport Layer Security (DTLS) to establish encrypted communication channels between IoT devices and the network.
- **Secure firmware updates**: Develop a secure mechanism for updating the firmware of IoT devices. Ensure that only authenticated and authorized updates are applied to maintain the integrity and security of the devices.
- **Anomaly detection and monitoring**: Deploy anomaly detection techniques to identify any abnormal behavior or potential security threats within the IoT environment. Continuously monitor and analyze network traffic for signs of intrusion or malicious activity.
- **Security auditing and compliance**: Security audits are regularly performed to assess the effectiveness of the authentication and communication network. Ensure compliance with industry standards and regulations related to IoT security.

REFERENCES

[1] Ahmed, M., Hashim, F., Hashim, S., & Abdullah, A. (2023). Authentication-chains: Blockchain-inspired lightweight authentication protocol for IoT networks. *Electronics, 12*, 867. https://doi.org/10.3390/electronics12040867

[2] Ashraf, J., et al. (2020). A review of intrusion detection systems using machine and deep learning in the internet of things: Challenges, solutions, and future directions. *Electronics*, *9*(7), 1177.

[3] Chandrakar, P., Bagga, R., Kumar, Y., Dwivedi, S. K., & Amin, R. (2022). Blockchain-based security protocol for device-to-device secure communication in the Internet of things networks. *Wiley Online Library*.

https://doi.org/10.1002/spy2.267

[4] El-Gendy, S., & Azer, M. (2020). Security framework for Internet of Things (IoT). In *2020 International Conference on Communication, Computing and Electronics Systems (ICCES)*. https://doi.org/10.1109/ICCES51560.2020.9334589

[5] Gagana, M. C., Chandana, K. J., Divyashree, N., Affan, M., Kumar, R. V., & Kayarga, T. (2021). Secure authentication and security system for IoT environment. *International Journal of Engineering Research & Technology (IJERT), 10*(7).

[6] Hajian, R., Haghighat, A., & Erfani, S. H. (2022). A secure anonymous D2D mutual authentication and key agreement protocol for IoT. *Internet of Things, 18*, 100493. https://doi.org/10.1016/j.iot.2021.100493

[7] Haque, S., Kumar, K., Haque, M. A., Faizanuddin, M., Shakeb, E., & Singh, A. K. (2021). Blockchain technology for IoT security.

[8] Hou, S., Ye, Y., Song, Y., et al. (2017). HinDroid: An intelligent android malware detection system based on structured heterogeneous information network. *SIGKDD Explorations*.

[9] Kanini, C. (2022). A critical review of Internet of Things communication environment: Privacy and security constraints. *GSC Advanced Engineering and Technology, 4*(2), 42–57.

[10] Kumari, S., Karuppiah, M., Das, A. K., Li, X., Wu, F., & Kumar, N. (2018). A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers. *The Journal of Supercomputing, 74*. https://doi.org/10.1007/s11227-017-2048-0

[11] Lata, M., & Kumar, V. (2021). Standards and regulatory compliances for IoT security. *International Journal of Service Science Management Engineering and Technology, 12*, 133-147. https://doi.org/10.4018/IJSSMET.2021090109

[12] Lau, C. H., Yeung, K. H., Yan, F., & Chan, S. (2023). Blockchain-based authentication and secure communication in IoT networks. *Wiley Online Library*. https://doi.org/10.1002/spy2.319

[13] Manikandan, K. P., Anusha, P., Jaya, A., Pundir, S., Rammohan, K., & Adhikary, P. (2023). A multi-criteria intelligence aid methodology and IoT based data protection using machine learning. In *2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*.

[14] Minerva, R., Biru, A., & Rotondi, D. (2015). Towards a definition of the Internet of Things (IoT). *IEEE Internet Initiative, 1*, 1–86.

[15] Nikooghadam, M., & Amintoosi, H. (2023). Secure communication in CloudIoT through the design of a lightweight authentication and session key agreement scheme. *International Journal of Communication Systems, 36*(1), e4332. https://doi.org/10.1002/dac.4332

[16] Rose, K., Eldridge, S., & Chapin, L. (2015). The Internet of Things: An overview. *The Internet Society (ISOC)*.

[17] Sakhare, A., Kshirsagar, A., & Pachghare, V. (2023). Survey on Data Privacy Preserving Techniques in Blockchain Applications. *2023 9th International Conference on Smart Computing and Communications (ICSCC)*, 321-326.

[18] Saqib, M., Jasra, B., Moon, A. H., & Hassan, M. (2022). A lightweight three-factor authentication framework for IoT-based critical applications. *Journal of King Saud University - Computer and Information Sciences, 34*(9), 6925-6937. https://doi.org/10.1016/j.jksuci.2021.07.023

[19] Sethi, P., & Sarangi, S. R. (2017). Internet of Things: Architectures, protocols, and applications. *Journal of Electrical and Computer Engineering, 2017*, 1–25.

[20] Singh, S., Sanwar Hosen, A.S., & Yoon, B. (2021). Blockchain Security Attacks, Challenges, and Solutions for the Future Distributed IoT Network. *IEEE Access, 9*, 13938-13959.

[21] Sousa, P. R., Magalhães, L., Resende, J. S., Martins, R., & Antunes, L. (2021). Provisioning, authentication

and secure communications for IoT devices on FIWARE. *Sensors (Basel), 21*(17), 5898. https://doi.org/10.3390/s21175898

[22] Subathra, G., Antonidoss, A., & Singh, B.K. (2022). Decentralized Consensus Blockchain and IPFS-Based Data Aggregation for Efficient Data Storage Scheme. *Security and Communication Networks*.

[23] Uppuluri, S., & Lakshmeeswari, G. (2023). Secure user authentication and key agreement scheme for IoT device access control based smart home communications. *Wireless Networks, 29*, 1333–1354. https://doi.org/10.1007/s11276-022-03197-1

[24] Voas, J. (2016). Demystifying the Internet of Things. *Computer, 49*(6), 80–83.

[25] Xie, J. (2016). Design and implementation of marine massive heterogeneous data integration system based on cloud platform. *Ship Science and Technology*.

[26] Xu, T., Fu, Z., Yu, M., Wang, J., Liu, H., & Qiu, T. (2021). Blockchain-based data protection framework for IoT in untrusted storage. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*.

[27] Yu, Y., Xin, L., Dan, L., et al. (2017). An innovative library decision-making support service system based on multi-source heterogeneous database integration. *Research Library Science*.

[28] Zanella, N., Bui, A., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things Journal, 1*(1), 22–32.

[29] Zhu, Y., Chen, C.-M., Li, X., Liu, S., Wu, M.-E., & Kumari, S. (2022). Enhanced authentication protocol for the Internet of Things environment. *Security and Communication Networks*.