Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2024.056741

ARTICLE

Improving Smart Home Security via MQTT: Maximizing Data Privacy and Device Authentication Using Elliptic Curve Cryptography

Zainatul Yushaniza Mohamed Yusoff[1], Mohamad Khairi Ishak[2,*], Lukman A. B. Rahim[3] and Mohd Shahrimie Mohd Asaari[1]

[1]School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, Nibong Tebal, Penang, 14300, Malaysia

[2]Department of Electrical and Computer Engineering, College of Engineering and IT, Ajman University, Ajman, 346, United Arab Emirates

[3]Faculty of Science and IT, Universiti Teknologi Petronas, Seri Iskandar, Perak, 32610, Malaysia

*Corresponding Author: Mohamad Khairi Ishak. Email: m.ishak@ajman.ac.ae

Received: 29 July 2024 Accepted: 24 September 2024 Published: 22 November 2024

ABSTRACT

The rapid adoption of Internet of Things (IoT) technologies has introduced significant security challenges across the physical, network, and application layers, particularly with the widespread use of the Message Queue Telemetry Transport (MQTT) protocol, which, while efficient in bandwidth consumption, lacks inherent security features, making it vulnerable to various cyber threats. This research addresses these challenges by presenting a secure, lightweight communication proxy that enhances the scalability and security of MQTT-based Internet of Things (IoT) networks. The proposed solution builds upon the Dang-Scheme, a mutual authentication protocol designed explicitly for resource-constrained environments and enhances it using Elliptic Curve Cryptography (ECC). This integration significantly improves device authentication, data confidentiality, and energy efficiency, achieving an 87.68% increase in data confidentiality and up to 77.04% energy savings during publish/subscribe communications in smart homes. The Middleware Broker System dynamically manages transaction keys and session IDs, offering robust defences against common cyber threats like impersonation and brute-force attacks. Penetration testing with tools such as Hydra and Nmap further validated the system's security, demonstrating its potential to significantly improve the security and efficiency of IoT networks while underscoring the need for ongoing research to combat emerging threats.

KEYWORDS

Smart home; confidentiality; ECC; security; lightweight cryptography; authentication; integrity; efficiency

1 Introduction

The Internet of Things (IoT) represents a complex and expansive network ecosystem, integrating diverse software and hardware components to revolutionize various domains, including healthcare, transportation, and smart homes. Smart homes have gained significant traction among these applications, where intelligent devices are interconnected via networks to facilitate data sharing and interaction. This setup typically involves a central home router connecting smart devices like locks, TVs, lights, and appliances to a service provider's cloud network, safeguarded by a firewall [1]. The IoT architecture is structured into layers—perception, middleware, network, and application—each with a crucial role in data processing and communication. This meticulous organization ensures efficient and secure interactions within smart home environments. As shown in Fig. 1, a visual representation of a smart home network architecture, the home router takes center stage. It connects smart locks, TVs, lights, alarm systems, and smart appliances, serving as the primary conduit for data flow between various smart devices and a service provider. Data from these devices is transmitted through the router to the service provider's cloud network, protected by a firewall to ensure security [2]. However, the convenience of smart homes also brings significant security and privacy challenges that must be addressed to protect user data and device integrity. From the end user's perspective, smart home security involves several key considerations. Users play a crucial role in maintaining the physical security of their smart homes, requiring constant vigilance and alertness to protect interconnected devices from unauthorized access. Being aware of potential risks and taking responsibility for securing the environment is essential. Additionally, web interfaces are vulnerable to exploitation if not adequately secured, highlighting the need for robust security measures. Ensuring software and firmware security through continuous updates and patches is another critical responsibility, as proactive maintenance can significantly enhance overall security and empower users as guardians of their digital domains. Mobile communication interfaces also pose security risks if not adequately protected, necessitating user vigilance to prevent potential threats. Insecure network services can further expose the system to various risks, while transportation vulnerabilities demand secure data transport to avoid interception and manipulation. Inefficient authorization and authentication mechanisms can compromise the entire system's security, making it imperative to implement adequate controls. Protecting user data from unauthorized access is crucial for maintaining privacy and confidentiality while ensuring data integrity, which is vital for maintaining the accuracy and consistency of information. Finally, smart homes are particularly vulnerable to Distributed Denial of Service (DDoS) attacks, which can disrupt their operations, underscoring the need for comprehensive security measures [3].

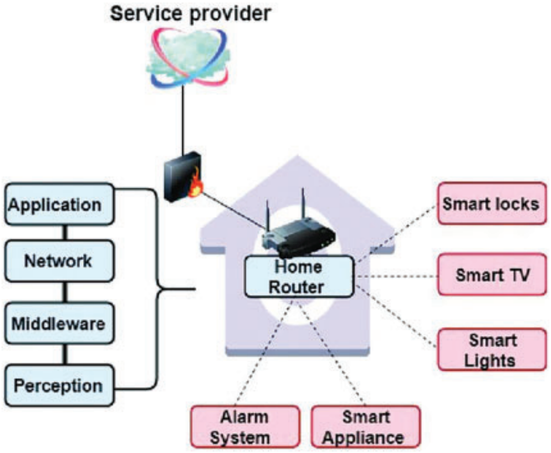


Figure 1: Smart home architecture [4]

The application layer of IoT systems, particularly in smart homes, introduces critical security challenges, including data access, authentication, privacy, and identity protection [5]. Securing smart home devices is particularly challenging due to the numerous potential attack vectors [6]. Without robust security mechanisms, attackers can exploit vulnerabilities to intercept and manipulate data, create routing loops, increase power consumption, and even deny access to authorized users, ultimately compromising the entire IoT network's security [7]. This study addresses these challenges by developing an MQTT-based security and privacy protocol designed explicitly for resource-constrained IoT devices in smart homes. The protocol integrates Elliptic Curve Cryptography (ECC) to facilitate efficient and secure key pair generation within the MQTT framework. Critical parameters are securely transmitted using a hash value during the authentication and communication phases, ensuring that all received parameters are reevaluated for stability and security. The proposed ECC-based security communication protocol enhances the security posture of smart home devices, reinforcing the overall IoT ecosystem's integrity.

The structure of this article is as follows: Section 2 provides an overview of recent related studies, and Section 3 discusses the authentication by Dang-Scheme. Our proposed protocol is introduced in Section 4. Section 5 covers the security analysis, and a performance analysis in Section 6. Scalability and complexity analysis are discussed in Section 7. Section 8 examines the applications and benefits of the middleware broker system across different domains. Section 9 presents a comparative evaluation, while Section 10 highlights the novel contributions of this work. Finally, Section 11 concludes the article.

2 Related Works

The rapid evolution of IoT technologies has brought about numerous advancements in various domains, such as smart homes, healthcare, and transportation. However, these developments have also introduced significant challenges, particularly concerning the security and privacy of data transmission in resource-constrained environments. Researchers have substantially addressed these challenges by developing cryptographic techniques, authentication protocols, and secure communication frameworks. In this section, we critically analyze the existing literature, identifying the strengths and limitations of these approaches and setting the stage for the proposed methodology. By understanding the current landscape of IoT security, this paper aims to build upon these foundations and propose innovative solutions to enhance the security and efficiency of IoT networks.

Researchers in [8] have noted that even modern primary care physicians are vulnerable to various security threats, including shoulder surfing, brute force attacks, and smear attacks. In particular, the study by Huszti et al. [8] on scalable, password-based, and threshold authentication for smart homes highlights several significant drawbacks. These include the system's complexity and reliance on resource-constrained devices, which pose challenges in implementation and performance, particularly in larger smart home environments. Additionally, the dependence on passwords could undermine security, mainly if weak passwords are used. Although the protocol is designed with scalability in mind, the increasing number of devices adds complexity, potentially leading to latency and maintenance issues. Furthermore, the centralization of the device manager's role creates a single point of failure, making the entire system vulnerable if an attacker compromises that component. A template-based authentication system was developed to address these issues, as outlined in [9]. The primary disadvantages of the paper revolve around its narrow focus and limited real-world applicability. The research heavily centers on specific scenarios and environments, which may only partially represent broader or more diverse contexts. Additionally, implementing the proposed approaches may face challenges when scaled or applied in different or more complex IoT systems. These limitations could hinder the generalization and adoption of the findings across various use cases, making the paper's contributions less impactful in practice.

However, it is important to recognize that these technologies are vulnerable to corruption attacks. A lightweight and secure system was developed to counter such threats. Nonetheless, this approach's threat model did not consider prevalent attacks such as Man-in-the-Middle (MitM) and phishing, and it was also found to have significant time complexity. These issues are like those discussed in the scheme [10]. The paper explores cryptographic techniques for securing data transmission in IoT systems, but they have several drawbacks. While it stresses the need for lightweight cryptographic solutions due to IoT devices' resource limitations, it must sufficiently address the trade-offs between security and performance. The study is heavily theoretical and needs more practical implementation or real-world case studies, reducing the applicability of its conclusions. Furthermore, although it identifies key security threats like unauthorized access and data tampering, it needs to delve into emerging threats or the evolving landscape of IoT security challenges, making it less comprehensive in preparing for future risks. The author of [11] introduced a paired-key structure, but this approach remains vulnerable to physical hacking, potentially resulting in the leakage of authentication keys. Meanwhile, the protocol described in [12] relies on two different keys, and a failure in the key distribution function could break the relationship between entities. Additionally, the method outlined in [13] incurs high computational costs due to its reliance on numerous cryptographic operations. In contrast, the protocol in [14] falls short of guaranteeing complete confidentiality and fails to protect against well-known severe attacks or denial-of-service (DoS) threats. As a result, the authors of [15] developed a highly complex authentication system that does not effectively prevent attacks using available keys, needs more assurance of message relevance, and introduces a high level of complexity. Similarly, the session key generation method described in [16] remains susceptible to replay and time synchronization attacks, failing to sufficiently address concerns related to anonymity and relevance. The paper outlines several disadvantages of the proposed lightweight mutual authentication and key exchange protocol for IoT smart home environments. A key issue is the dependence on cumulative keyed-hash chains and temporary identities, which, while providing anonymity, could complicate session management and increase computational overhead. Additionally, enforcing security policies through virtual domain segregation, though effective in countering insider threats, may reduce the flexibility of device interactions and require careful configuration to prevent operational inefficiencies. Integrating fog computing for identity assurance while enhancing security might also introduce additional complexity and latency, particularly in environments with limited resources. Lastly, the reliance on dynamic identities and symmetric keys may need to offer more robustness against more sophisticated attacks, potentially leading to vulnerabilities in long-term use. Furthermore, employing the PUF-based authentication approach, established in [17], leads to substantial computational and connection overhead. Additionally, the methods proposed in [15,16] and [18] need more anonymization or non-traceability measures, diminishing their effectiveness in advanced monitoring scenarios.

The self-signing and access control mechanisms detailed in [19] protect against disclosing confidential data, modifying codes, and creating new ones. However, their implementation necessitates certification authority. Similarly, the simplified authentication technique presented in [20] relies on a trusted authority (TA) to provide credentials to IoT sensors and gateway nodes, creating a single point of failure. The user and password authentication system proposed in [21] is susceptible to user identity attacks, offline password attacks, and timing attacks through session information disclosure. In contrast, the user authentication method presented in [22,23] ensures user anonymity and non-traceability but necessitates additional accounts and communication. The paper identifies several potential disadvantages of the LAM-CIoT mechanism. A key issue is the computational load on resource-constrained IoT devices, which may reduce the scheme's effectiveness in environments with limited processing power. Although the paper highlights the lightweight design of the authentication mechanism, using cryptographic hash functions and bitwise XOR (Exclusive OR) operations may still introduce overhead that could affect performance in resource-limited scenarios. XOR is a logical operation that compares corresponding bits from two binary numbers and returns one (1) if they are different and zero (0) if they are the same. Additionally, the scheme's dependence on a trusted authority (TA) for credential generation and management presents a potential single point of failure, raising concerns about security and scalability in large-scale IoT systems. Finally, while LAM-CIoT strives to balance security and performance, the complexity of managing secure communications and session keys in real time may present challenges in highly dynamic IoT environments. Pham et al. [24] proposed a three-step machine-to-machine and device-to-server authentication process; however, including two-factor authentication increases data storage and transaction processing costs and communication expenses. The paper introduces a hybrid ECC-based authentication scheme for resource-constrained IoT environments but notes several limitations. A significant drawback is the dependence on centralized servers, which, despite being somewhat mitigated by device-to-device (D2D) communication, still poses a risk of a single point of failure. The entire network could be at risk if the server goes down, even with D2D functionality. Moreover, while the scheme is intended to be lightweight, the computational demands of ECC operations, though minimized, may still be too demanding for highly resource-constrained devices, limiting the scheme's effectiveness in some IoT scenarios. Additionally, requiring each device to maintain a list of trusted devices may overburden the limited memory resources of tiny IoT devices, leading to scalability challenges in larger networks. In [7], authors presented a fingerprint-based password generator for IoT-enabled smart homes, rendering the system immune to online dictionaries and man-in-the-middle attacks. Nevertheless, the system's versatility contributes to high installation costs. Chowdhury et al. [25] developed a cost-effective system with minimal home security and automation requirements. However, it does not address online hacker assaults. Singh et al. [26] recommended installing a home security system to protect vulnerable populations but primarily focused on security vulnerabilities in physical hardware rather than the online environment. Vasudev et al. [27] devised a V2V authentication method using ECC, enabling vehicles to request real-time information from their agents. This system employed identification numbers, smart cards, secure hashing, and unreadable characters. Meanwhile, various safety regulations have led to increased transportation and storage costs. Garg et al. [28] devised a mutual authentication-based key agreement protocol, demonstrating resilience to various assaults through rigorous security analysis. However, these schemes require complex interactions and at least two rounds of computing. Kumar et al. [29] proposed a security mechanism for IoT devices based on facial recognition, leveraging Raspberry Pi 3 to enhance performance while reducing power and energy usage. Nevertheless, installation and storage costs remain significant, and the system does not consider security measures against cyberattacks. Finally, Raju et al. [30] recommended a protection system utilizing Node Microcontroller Unit (Node MCU), a low-cost open-source IoT platform, for devices involved in home automation. However, their system is open to cyber assaults due to a lack of consideration for potential threats. Table 1 compares various approaches to smart home security and authentication protocols, focusing on their underlying technologies, strengths, and weaknesses. For instance, solutions like Huszti et al. [8] offer scalability but introduce complexity and centralization issues. On the other hand, hybrid schemes [24,31] enhance security through ECC but face challenges related to computational demands and scalability in resource-constrained environments. This comparison highlights the trade-offs that must be balanced between security, complexity, and resource efficiency in designing secure IoT systems for smart homes. The work [32] applied various tasks and their efficiencies at different production stages. It examined how Sub-assembly A, which involved assembling several components, generally operated at 75%–85% efficiency but occasionally experienced delays due to supply chain issues. Sub-assembly B, which focused on more intricate and time-consuming components, typically ran at 70%–80% efficiency. This section encountered challenges due to the complexity of the components. The final assembly line, where Sub-assemblies A and B were combined, was identified as the most challenging stage, with an efficiency rate of 60%–70%. This part of the process had the highest potential for bottlenecks. To ensure the final products met standards, the quality inspection stage maintained an efficiency of 85%–90%, though occasional slowdowns occurred when defects were detected. Overall, the analysis highlighted how supply chain delays and the complexity of components impacted the efficiency at various stages of production. References [33,34] examined the challenges of protecting information systems from cyber threats such as data breaches, malware, and insider attacks. It emphasized a risk-based approach, integrating cybersecurity into system architecture through layered defence and secure design. Key concepts like confidentiality, integrity, and resilience were discussed, focusing on building security controls early in development. The paper also recommended a defence-in-depth strategy and Zero-Trust Architecture, ensuring no system components were trusted without verification. It stressed the importance of continuous monitoring, regular updates, and employee training. Drawing from defence and government sectors, the principles were applied to commercial settings, guiding engineers to incorporate cybersecurity into complex systems.

Table 1: Comparison of related works in smart home security and authentication

Authors	Proposed solution	Underlying technology	Pros	Cons
Huszti et al. [8]	Scalable, password-based, and threshold authentication	Password-based authentication, threshold schemes	Scalable, suited for smart homes	Complex implementation relies on resource-limited devices, the central point of failure
Muhammad et al. [9]	Template-based authentication system	Cryptographic templates	Focused on specific scenarios, effective within its context	Limited applicability, challenges in scaling to broader IoT environments
Xiao et al. [10]	Credential-less authentication framework	Cryptographic techniques	Eliminates need for credentials, enhances security	High-time complexity overlooks common attacks such as MitM and phishing
Pham et al. [11]	Hybrid ECC-based authentication scheme	Elliptic Curve Cryptography (ECC)	Improved security for resource-constrained devices	Centralized server dependency, scalability issues, and ECC's computational demands on tiny devices

(Continued)

Table 1 (continued)

Authors	Proposed solution	Underlying technology	Pros	Cons
Garg et al. [28]	Mutual authentication-based key agreement protocol	Secure hashing, ECC	Robust against multiple attacks, secure	High computational costs require multiple complex interaction.
Singh et al. [26]	Enhanced home security system	Hardware-based security	Focused on physical security vulnerabilities	Limited focus on online security threats, the potential for cyber-attacks
Raju et al. [30]	Home automation and security system with Node MCU	IoT and microcontroller unit (MCU) technology	Simple, cost-effective	Susceptible to cyber-attacks due to lack of advanced security measures
Chowdhury et al. [25]	IoT-based smart security and home automation system	Basic IoT technologies	Cost-effective, minimal requirements	It does not address online hacker assaults
Vasudev et al. [27]	V2V authentication method	ECC, secure hashing	Real-time vehicle-to-vehicle communication	High transportation and storage costs
Alshahrani et al. [16]	Cumulative keyed-hash chain-based mutual authentication	Keyed-hash chain	Anonymity and security within smart homes	Complicated session management, increased computational overhead
Kumar et al. [29]	IoT device security based on facial recognition	Facial recognition, Raspberry Pi technology	Reduces power usage, efficient	High installation costs, limited consideration of cyberattacks

3 Preview of Dang-Scheme

In our protocol, we have adapted an existing scheme known as the Dang-Scheme, initially developed by Dang et al. [35]. The Dang-Scheme builds upon a prior protocol introduced by Wang et al. [36], which was designed to enhance the authentication of resource-constrained IoT devices, emphasizing improved security measures. It is essential to grasp the fundamentals of the Dang-Scheme as it is the foundation for our proposed protocol. The Dang-Scheme authentication protocol comprises three principal phases, as illustrated in Fig. 2.

Phase 1: Registration

The initial phase involves device enrollment into the system. Its primary objective is to register the device's unique identifier with the server. Upon completing this phase, as the server calculates and stores authentication data, the device reciprocates by providing secure cookie data, which will be utilized in subsequent authentication steps.

Phase 2: Authentication between Server and Device

Before the device gains access to the broader network, an authentication procedure is executed between the device and the server. During this phase, the device forwards its credentials or cookie data to the server. The server, in turn, scrutinizes these credentials to ascertain whether the connection is authorized. Verifying the connection's authenticity is imperative to ensure the device communicates with a legitimate server. Consequently, by the end of this phase, the device and the server authenticate each other, culminating in generating a pivotal session key.

Phase 3: Authentication between Two Devices

In IoT systems, inter-device communication is pervasive and frequent. Therefore, it becomes essential for devices to authenticate one another before exchanging data. This phase serves a purpose akin to the second step–validating each device's identity and establishing a default session key to safeguard subsequent message exchanges.

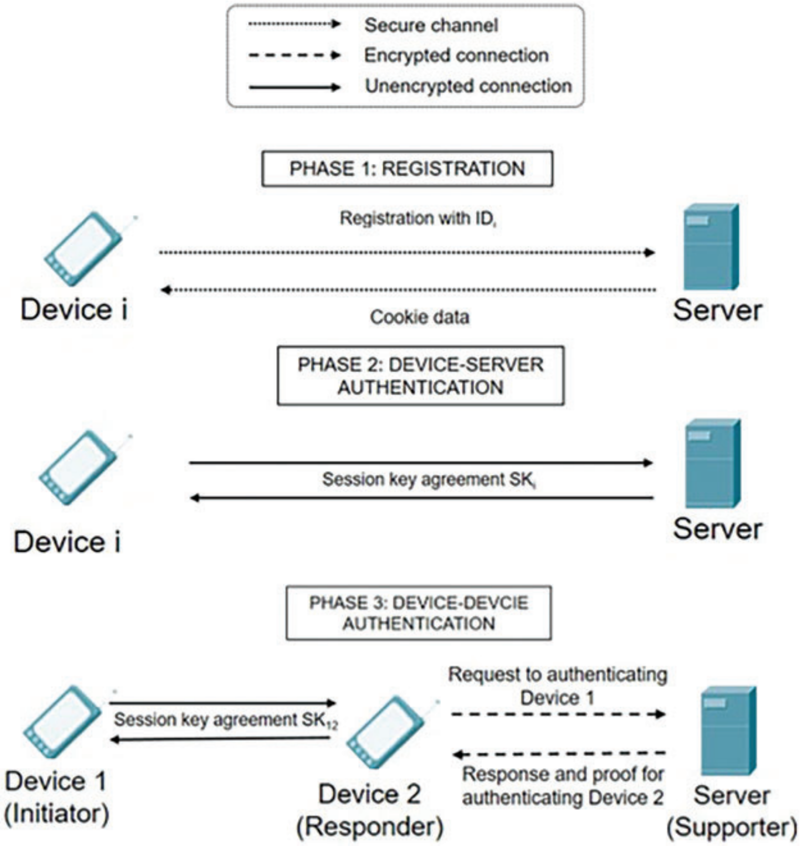

Secure channel
Encrypted connection
Unencrypted connection
PHASE 1: REGISTRATION
Registration with ID$_i$
Device i
Server
Cookie data
PHASE 2: DEVICE-SERVER AUTHENTICATION
Session key agreement SK$_i$
Device i
Server
PHASE 3: DEVICE-DEVCIE AUTHENTICATION
Session key agreement SK$_{12}$
Request to authenticating Device 1
Response and proof for authenticating Device 2
Device 1 (Initiator)
Device 2 (Responder)
Server (Supporter)

Figure 2: The authentication scheme proposed by Dang et al. [35]

Fig. 2 illustrates the Dang-Scheme authentication process, which relies partially on a centralized management model for overseeing connectivity, authentication, and access control in IoT device protocols. This approach is primarily tailored for users with constrained processing and storage capabilities and limited power resources. The Dang-Scheme employs an ECC-based mutual authentication protocol between devices to verify each other's identities. However, it is worth noting that in this scheme, Device 2 (the responder) exercises complete control to obtain the authentication key from the server. Moreover, as previously discussed, all inter-device connections are routed through the Device, which introduces inefficiencies. Consequently, in the ensuing section, we present an innovative approach that ensures security through liquid protocols that do not cover Brute force attacks.

4 Proposed Scheme

The research utilizes a centralized security framework to enable simultaneous and extensive deployment. This framework organizes security tasks and roles into a three-tier structure: collection, network, and application layers, as illustrated in Fig. 3. The study employs the ECC algorithm to secure sensitive identity data. ECC uses a pair of flexible public and private keys to effectively encrypt online communications, positioning it as a viable alternative to traditional Rivest–Shamir–Adleman (RSA) algorithms by leveraging elliptic curves to enhance public key security. Using ECC strengthens encryption and ensures efficient resource utilization. ECC is crucial for enhancing IoT systems with a robust security framework tailored to complex environments and interconnected devices, thereby ensuring secure and reliable interactions within the IoT infrastructure. This system builds on previous research incorporating D2D mutual authentication. Details of the design framework are provided in the following subsection.

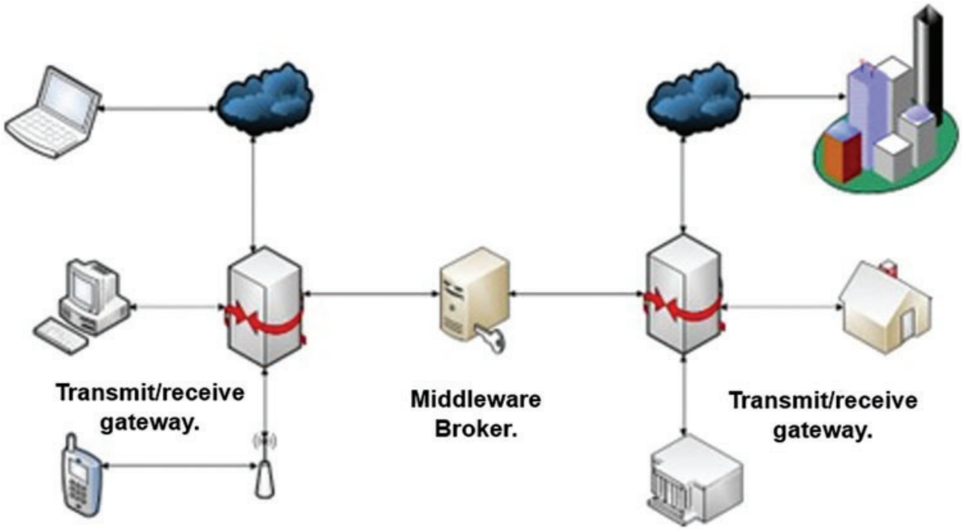


Figure 3: Proposed middleware security broker key for enhanced security and device/network isolation

The Fig. 4 shows the flowchart of a data transmission and identification processing scheme. The process outlined in the flowchart begins with the transmission of data from a URL or web application. Initially, the scheme performs scans simultaneously on the Device ID, Network ID, and Application ID. These IDs are then subjected to a Detection ID Process. If the detection succeeds, a Transaction ID key is generated or retrieved, subsequently used to receive the URL or web application. If the detection fails, the process halts, preventing data reception. The process ensures that only validated and authenticated data can proceed through the system, culminating in the successful transmission and reception of the intended web content.

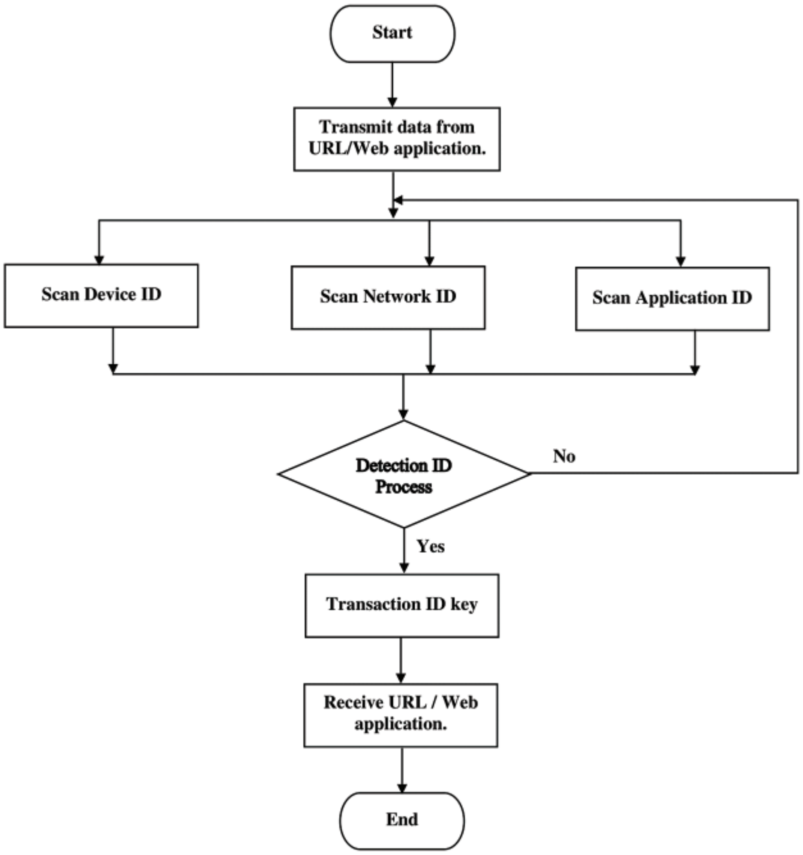


Figure 4: Flowchart process of middleware broker system

4.1 ECC Framework

The ECC framework encompasses encryption techniques and methodologies that leverage the properties of elliptic curves over finite fields to secure digital communications. Unlike traditional public key cryptography systems such as RSA, which depend on the factorization of large prime numbers, ECC provides equivalent security with smaller key sizes. This results in faster computations and reduced resource consumption. Here is a breakdown of the ECC framework:

4.1.1 Mathematical Foundation

ECC utilizes the algebraic properties of elliptic curves over finite fields as the basis for cryptographic algorithms. The provided code snippet defines a Java interface for elliptic curve parameters. Eq. (1) specifies the ECC curve secp256r1, the NIST P-256 curve. This interface, 'ECParameters,' outlines the essential components needed to fully define an elliptic curve used in cryptographic systems, especially those based on ECC.

$$y^2 = x^3 + ax + b(mod\,p) \tag{1}$$

4.1.2 Key Generation

In Elliptic Curve Cryptography (ECC), the generation of private and public keys is based on the mathematical properties of elliptic curves. The private key is chosen as a random integer, while the public key is obtained by multiplying this private key with a specific point on the curve known as the generator point. The 'generatorX()' function returns the x-coordinate of this generator point, and the 'generatorY()' function returns its y-coordinate. Furthermore, the 'order()' function provides the total number of points in the subgroup created by the generator point.

4.1.3 Encryption and Decryption

ECC uses asymmetric encryption techniques. In this approach, the sender encrypts a message using the receiver's public key, and only the receiver can decrypt it with their private key, ensuring a secure information exchange. This research developed a Java class, ECKey, which implements the Key interface. The ECKey class is designed to manage ECC keys, including public and private (secret) keys.

4.1.4 Digital Signatures

The ECC framework also supports digital signature mechanisms, allowing users to authenticate a message with their private key, which others can verify with the corresponding public key. The provided code is a Java implementation of an ECC system, encapsulated in the 'ECCryptoSystem' class, which complies with the 'CryptoSystem' interface specifications. This class enables the encryption and decryption of data using elliptic curve cryptography. This approach is preferred for secure communications due to its efficiency in utilizing smaller key sizes compared to traditional methods like RSA while still providing robust security.

4.1.5 Efficiency and Security

ECC offers enhanced security per bit compared to other public key cryptography methods, allowing for equivalent security levels with shorter key lengths. This increased efficiency makes ECC ideal for environments with limited bandwidth or processing power, such as mobile devices or IoT systems.

4.1.6 Standards and Protocols

ECC has been incorporated into various standards and protocols to ensure secure communication, including SSL/TLS for secure web browsing, SSH for secure remote access, and many other applications.

4.1.7 Implementation and Deployment

Implementing ECC requires careful selection of elliptic curve parameters, key generation methods, and cryptographic protocols to ensure security and interoperability. The ECC framework is recognized for its efficiency and robust security, making it increasingly popular in modern cryptographic applications, especially in environments with limited computational resources.

4.2 Middleware Broker Implementation

The architecture has been implemented and tested in a prototype environment to evaluate its scalability, security, and efficiency. The experiment employs NetBeans software for the Java Servlets platform within the middleware broker system, Ubuntu Server for MQTT brokering, and Raspberry Pi (RPi) for smart home device applications.

4.2.1 Middleware System Broker

Fig. 5 describes a secure messaging workflow in an IoT environment, ideal for smart home applications. It demonstrates the encryption, transmission, and decoding of messages between IoT devices using intermediary system brokers, IoT gateways, and MQTT brokers to queue messages. Below is an analysis of the components and processes depicted:

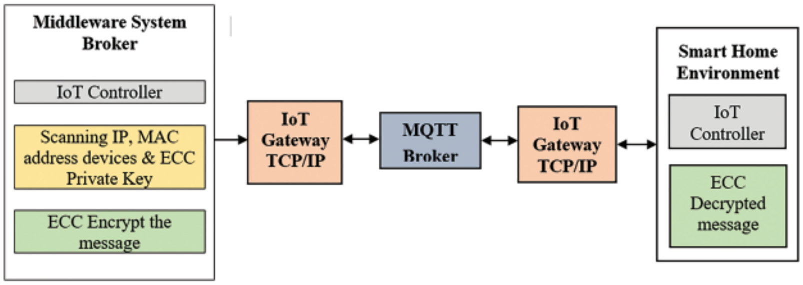


Figure 5: Block diagram of ECC security technique

4.2.2 Middleware System Broker-IoT Controller

This module is designed to scan devices' IP and MAC addresses and uses ECC private keys for enhanced security. It employs ECC to decrypt messages before transmission. The setup is divided into two modules, each detailed below.

(i) Module 1: Authentication Interface

Servlets serve as server-side Java platforms and programs designed to facilitate the seamless data exchange between a client and a web server. They achieve this by enabling interactive presentation and manipulation of data through dynamic web page input techniques. Servlets operate on the server side and function independently of any graphical user interface. In this context, a client program, typically a web browser or other Internet-enabled application, establishes a connection with a web server to make requests. Servlets, in turn, respond to these client requests by dynamically generating responses. A servlet request object represents each client request, while the resulting response is embodied in a servlet response object.

Servlets can handle multiple client requests concurrently, efficiently processing and synchronizing numerous requests simultaneously. Moreover, servlets can redirect requests to alternative servers or servlets, enhancing their versatility. To access a servlet, clients must issue a URL command pointing to the directory where the web server is hosted or to a location that simulates local access. Servlets, primarily written in Java, excel at implementing intricate business logic. This empowers clients to interact with relational databases via dynamic web pages, making them well-suited for developing complex business applications.

(ii) Module 2: Interface for IoT Controller

Creating an interface for an IoT Controller, mainly focusing on a module that improves user interaction with IoT devices, demands meticulous consideration of functionality, user experience, and security. This interface is a unified dashboard for users to manage their IoT devices effortlessly. It should be intuitive, providing direct access to device controls, real-time status updates, and configuration options. Key considerations include ensuring device compatibility, defining user access levels, supporting real-time data processing, and implementing strong security measures. The following sections will detail the technical implementation.

(a) Front-End Technologies:

The Java Servlet program was a middleware broker system on the NetBeans platform. Java servlets extend the capabilities of web servers by enabling dynamic generation of web content and responding to client requests. These platform-independent servlets are server-side components deployed on web servers to handle HTTP requests and generate corresponding responses. Their platform independence arises from being written in Java, allowing them to function on any server that supports the Java platform. Servlets execute on the server side, facilitating tasks like processing form data, interacting with databases, and dynamically generating web content before delivering it to clients. They follow a well-defined lifecycle that includes methods such as init(), service(), and destroy(), which are invoked by the servlet container (e.g., Apache Tomcat) at various stages. Managed by servlet containers—typically web servers or application servers implementing the Java Servlet API like Apache Tomcat—servlets are mapped to specific URLs within web applications through the web.xml deployment descriptor or annotations. Servlets maintain state across multiple requests using session tracking, cookies, and URL rewriting. Java Servlets are crucial in developing dynamic web applications by enabling server-side processing and client interaction via the HTTP protocol. They are particularly suited for managing smart home devices and processing their data in IoT applications. This setup is standard when multiple devices or systems must communicate with a central server for coordination, data aggregation, and processing.

(b) Back-End Integration:

RESTful API and MQTT are widely adopted protocols to enable communication between devices and servers, seamlessly integrating backend systems. The research chose MQTT because of its lightweight design and reliability in facilitating reliable communication in IoT and M2M contexts. MQTT operates on a publish-subscribe model, where subscribers can publish messages on a topic, and other subscribers can subscribe to this topic to receive relevant notifications. The protocol is optimized for performance over networks with limited bandwidth and high latency, effectively supporting asynchronous communication. It is particularly suitable for scenarios where devices must update their network status or conditions frequently, which may differ in consistency or quality.

(c) Security:

Security measures include using secure programming techniques, performing regular security evaluations, and complying with data protection laws like GDPR and CCPA.

4.2.3 MQTT Broker

MQTT is a lightweight publish-subscribe network protocol designed to transfer messages between devices efficiently. Suitable for environments with low bandwidth, high latency, or unreliable connections, MQTT is widely used in domains such as IoT, automation, and smart home systems. Established MQTT brokers such as Mosquitto, HiveMQ, and EMQX are compatible across multiple platforms, providing features for debugging, interface customization, and ensuring security through SSL/TLS encryption and user authentication. This review involves configuring the Mosquitto MQTT broker on an Ubuntu Server accessed via a web URL.

4.2.4 Smart Home Environment-IoT Controller

In a typical Smart Home Environment, various IoT devices are integrated to create a networked ecosystem that enhances comfort, efficiency, and security within the living space. At the core of this setup, the IoT Controller acts as the central hub, facilitating communication, control, and automation of the diverse smart devices. This study utilized a Raspberry Pi (RPi) as a prototype for the smart home system. This depiction represents a simplified process within a smart home or IoT system, where operations are received and processed. Only those operations identified as device-related undergo decryption for further action, ensuring security by decrypting only essential information.

4.3 Middleware Broker System Algorithm

An algorithm is a systematic collection of sequential instructions or rules designed to solve a problem or perform a particular task. In computer science and mathematics, algorithms provide explicit guidelines for various computational processes, such as data management, automated reasoning, and more. They simplify complex issues by breaking them down into manageable steps, offering a structured approach to problem-solving. Algorithms are fundamental to computer science, underpinning software development, artificial intelligence, and data analytics. Executed via programming languages, they enable automation and streamline problem-solving across diverse computational environments. The provided pseudocode, described in the algorithm below, is tailored for a smart home application that primarily manages encrypted messages to control smart home device operations. While it adopts a Python-like syntax, the pseudocode includes non-standard operations and syntax. Additionally, the algorithm can determine the data size and track the process's duration from start to finish.

Pseudocode System Model Application:

```
on connect(client,userdata,flags,rc):
rc ← str(rc)
client.subscribe ← smarthome/light
ec = newEllipticCurve(newsecp256r1( ))
ECCryptoSystemcs = newECCryptoSystem(ec)
encryptedOnOFF = cs.encrypt(bytes,bytes.length,privateKeys.elementAt(index))
command+=encryptedOnOFF
on message(client,userdata,msg):
msg ← str(msg.payload)
Size(bytes) ← str(sys.getsizeof(msg.payload))
start_time ← int(round(time.time( ) * 1000))
message ← str(msg.payload.decode("utf-8"))
cmd ← echo + message|ecc-d-sk
stream ← os.popen(cmd)
output ← stream.read( )
output ← stream.read( )
output ← stream.read( )
if output.replace is ON then
output.replace ← light ON
end if
os.system ← (/home/pi/turnofff.sh)
duration_time ← int(round(time.time( ) * 1000)) – start_time
Duration(ms) ← str (duration_ time)
```

4.4 Security Measures, Processing Time, and Data Size

This research will evaluate the system's effectiveness in defending against common IoT challenges such as unauthorized access, data breaches, and system manipulation at the middleware broker level.

It will also compare the system's encryption methodologies, authentication systems, and access control tactics with those discussed in previous studies. During processing, the analysis will measure the time required to perform essential security tasks, including encryption, decryption, authentication, and critical oversight. The study will also assess how processing time affects the overall efficiency and responsiveness of IoT devices using this protocol, comparing these times with those reported for similar protocols under different operational loads in the literature. Regarding data size, the investigation will examine the volume of data transmitted during the system's operation, including any additional load imposed by the ECC framework. The study will evaluate the system's capacity to manage data, particularly in environments with extensive IoT systems featuring frequent device communications. Furthermore, it will compare the system's data size requirements with the security methods employed during data processing. All findings will be based on the middleware broker system established in the initial design phase.

4.5 Energy Consumption

"Energy consumption" refers to using energy by a process, device, or system. In the context of technology, especially regarding IoT devices or security protocols, energy consumption indicates the electrical power required by these devices or protocols. It includes the energy required to maintain device functionality, process data, implement security measures such as encryption and decryption, and facilitate communication with other devices or networks. Minimizing energy consumption is often a key goal in designing IoT devices and protocols to extend battery life and reduce operating costs, especially in large-scale deployments. The energy consumption of the router was calculated using the formula:

$$\text{Total Energy Consumed} = \text{Power} \times \text{Processing Time}.$$

With the router's power consumption rated at 0.008 kW and assuming continuous operation over 24 h, the total energy consumed is determined as follows:

$$\text{Total Energy Consumed} = 0.008\,\text{kW} \times 24\ \text{h} = 0.192\ \text{kWh}.$$

This calculation estimates the router's energy usage over a full day, offering a more accurate basis for evaluating long-term operational energy requirements.

4.6 Rationale behind the Selection of Dang-Scheme among Other Schemes

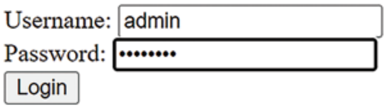
The rationale behind selecting the Dang-Scheme, among other schemes in the paper, is that it builds upon a prior protocol designed to enhance the authentication of resource-constrained IoT devices, focusing on improving security measures. The Dang-Scheme is adapted because it incorporates ECC-based mutual authentication between devices. It is crucial for verifying device identities and establishing secure session keys, particularly in environments with constrained processing, storage capabilities, and limited power resources. However, the paper acknowledges certain inefficiencies in the Dang-Scheme, specifically regarding its reliance on a centralized management model and the control one device (the responder) exercised to obtain the server's authentication key. These aspects prompted the researchers to propose an innovative approach that addresses these inefficiencies while ensuring security. This rationale underlines the need for a scheme that can provide robust security while accommodating the limitations of IoT devices. Thus, the Dang-Scheme was selected and adapted as a foundation for further improvement.

5 Security Analysis

In this section, we establish the robustness and security of the proposed authentication protocol by conducting an extensive security analysis of the scheme. Our scrutiny centers primarily on middleware brokers, mainly when two devices communicate through asymmetric authentication.

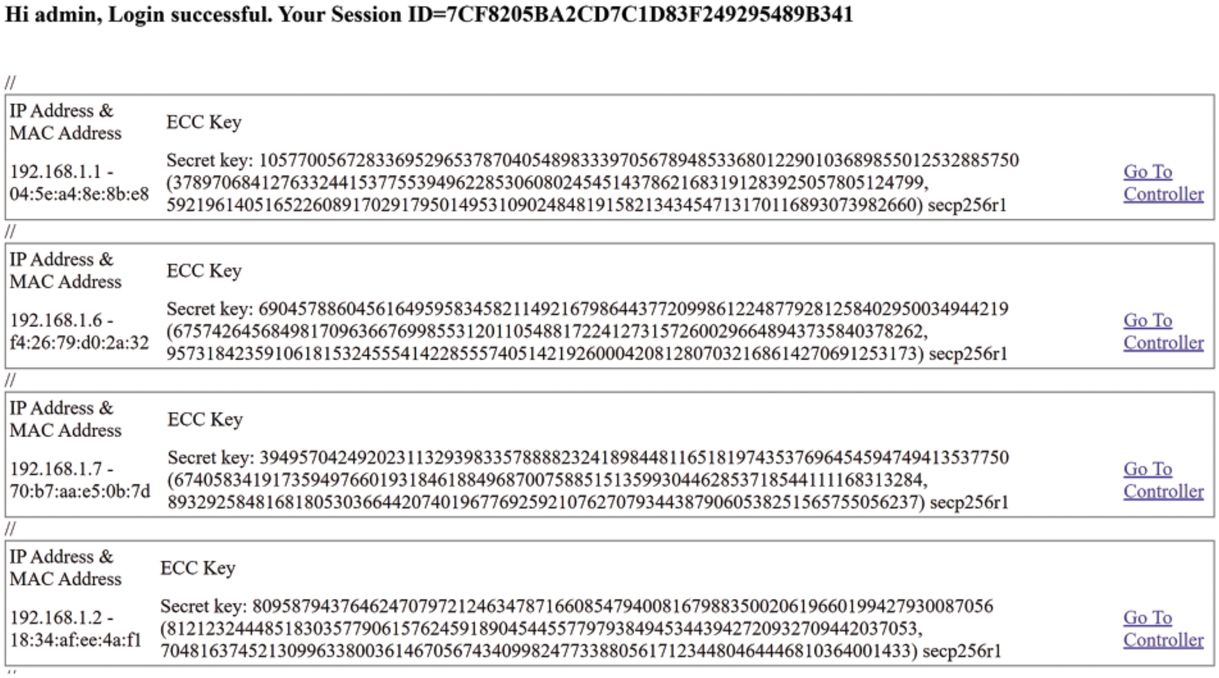
5.1 *Encryption and Decryption*

The authentication interface, illustrated in Figs. 6 and 7, is part of a security protocol activated during user login. This process involves collecting various IDs associated with the user's device, network, and application. Simultaneously, the system generates a unique secret or private key for each device in the smart home network using the ECC algorithm. The effectiveness of this module is assessed based on its ability to withstand four key security threats.



Username:	admin
Password:	••••••••

Figure 6: Module 1 showcases the authentication interface display located at the login wall



Hi admin, Login successful. Your Session ID=7CF8205BA2CD7C1D83F249295489B341

//

IP Address & MAC Address	ECC Key	
192.168.1.1 - 04:5e:a4:8e:8b:e8	Secret key: 105770056728336952965378704054898333970567894853368012290103689855012532885750 (37897068412763324415377553949622853060802454514378621683191283925057805124799, 59219614051652260891702917950149531090248481915821343454713170116893073982660) secp256r1	Go To Controller

//

IP Address & MAC Address	ECC Key	
192.168.1.6 - f4:26:79:d0:2a:32	Secret key: 69045788604561649595834582114921679864437720998612248779281258402950034944219 (67574264568498170963667699855312011054881722412731572600296648943735840378262, 95731842359106181532455541422855574051421926000420812807032168614270691253173) secp256r1	Go To Controller

//

IP Address & MAC Address	ECC Key	
192.168.1.7 - 70:b7:aa:e5:0b:7d	Secret key: 39495704249202311329398335788882324189844811651819743537696454594749413537750 (67405834191735949766019318461884968700758851513599304462853718544111168313284, 89329258481681805303664420740196776925921076270793443879060538251565755056237) secp256r1	Go To Controller

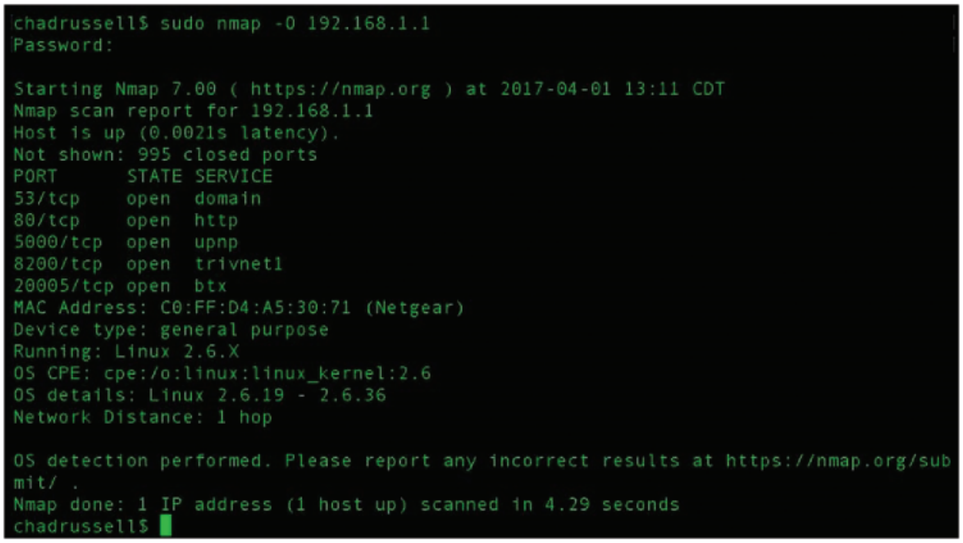
//

IP Address & MAC Address	ECC Key	
192.168.1.2 - 18:34:af:ee:4a:f1	Secret key: 80958794376462470797212463478716608547940081679883500206196601994279300087056 (81212324448518303577906157624591890454455779793849453443942720932709442037053, 70481637452130996338003614670567434099824773388056171234480464446810364001433) secp256r1	Go To Controller

Figure 7: On the wall among the list of smart home devices is the authentication interface display featured in Module 1

5.1.1 *Brute Force Attacks*

Brute force attacks involve an attacker systematically testing every possible combination of keys to break encryption until the correct one is found. The robustness of ECC makes these attacks largely ineffective, as the vast number of potential combinations makes the process extremely time-consuming. The Middleware Broker System has implemented specialized login protocols to enhance its defences against such attacks, thereby increasing its security beyond the levels provided by the Dang-Scheme. While not detailed here, these additional security measures likely introduce complexities that significantly hinder the success of brute force attacks, indicating a more secure authentication infrastructure within the Middleware Broker System. Fig. 8 captures the terminal output of a Network Mapper (Nmap) scan, a tool designed for network exploration and auditing. Nmap identifies the devices running on the network and the services they offer, essentially mapping the network infrastructure. It often targets the default gateway in a LAN. Network administrators use this data to determine which services are active on a network device, and it helps security experts in vulnerability assessments. Nmap evaluation is usually the first step before conducting more intrusive security tests, such as brute force attacks with tools like Hydra.



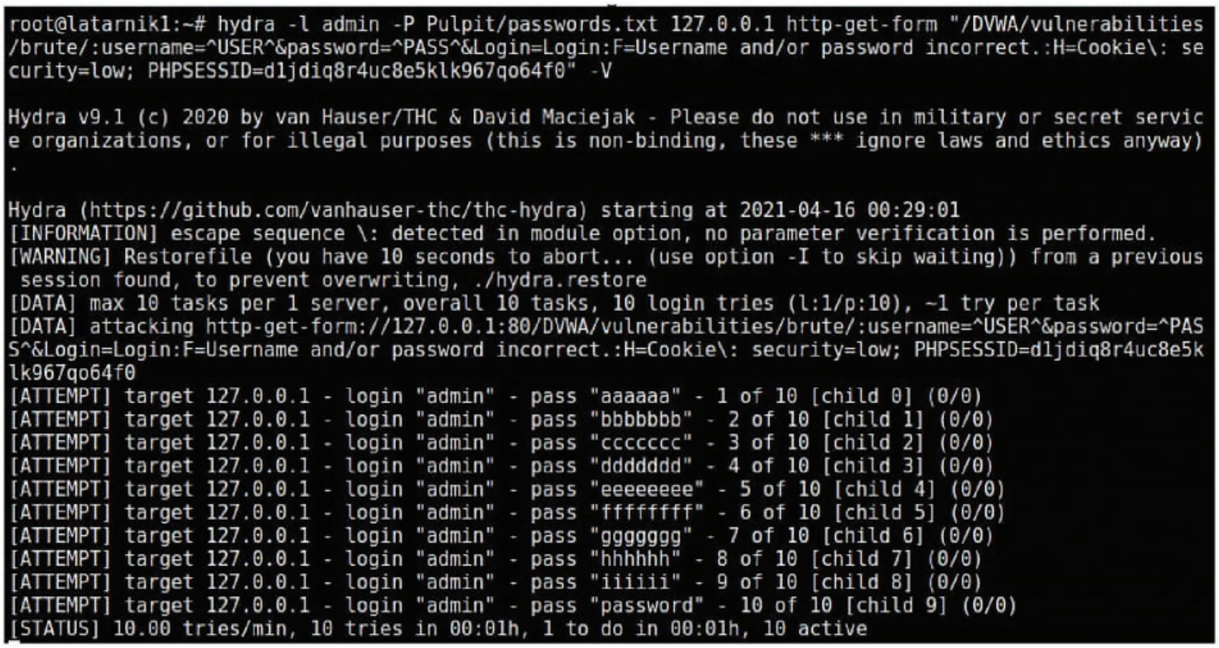
```
chadrussell$ sudo nmap -O 192.168.1.1
Password:

Starting Nmap 7.00 ( https://nmap.org ) at 2017-04-01 13:11 CDT
Nmap scan report for 192.168.1.1
Host is up (0.0021s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
5000/tcp  open  upnp
8200/tcp  open  trivnet1
20005/tcp open  btx
MAC Address: C0:FF:D4:A5:30:71 (Netgear)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.19 - 2.6.36
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.29 seconds
chadrussell$ 
```

Figure 8: The output shows the results of using the Nmap network scanning tool

Fig. 9 illustrates the terminal where Hydra is used, a brute-force password-cracking tool that systematically attempts various username and password combinations to infiltrate the system. Such operations are legal only in a regulated environment for authorized security assessments. The illustrated results highlight the tool's effectiveness in breaching security and underscore the necessity of a stringent password policy. If a brute-force attack occurs, results like those shown in Fig. 10 can be expected. Tools like Hydra are integral to security and penetration testing, enabling practitioners to identify vulnerable passwords and enhance the system's defences.

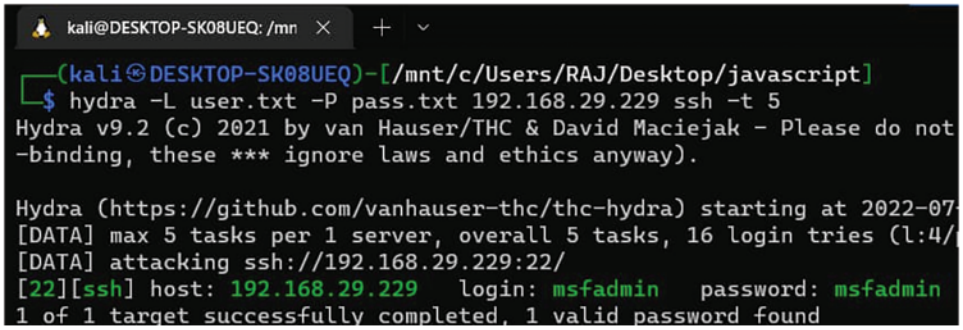
5.1.2 Man-in-the-Middle (MitM) Attacks

A man-in-the-middle (MitM) attack is a security breach where a hacker covertly intercepts and possibly manipulates data between two unsuspecting parties. Such attacks can compromise the integrity and confidentiality of the communication, allowing the attacker to eavesdrop or impersonate one of the entities. Employing ECC can significantly reduce the risk of these attacks by encrypting the data, making intercepted messages challenging to decipher. To further strengthen the system against this vulnerability, a strategy has been implemented that assigns a unique private key to each device within the smart home network, ensuring secure and authenticated communication channels.



```
root@latarnik1:~# hydra -l admin -P Pulpit/passwords.txt 127.0.0.1 http-get-form "/DVWA/vulnerabilities
/brute/:username=^USER^&password=^PASS^&Login=Login:F=Username and/or password incorrect.:H=Cookie\: se
curity=low; PHPSESSID=dljdiq8r4uc8e5klk967qo64f0" -V

Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret servic
e organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway)
.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-04-16 00:29:01
[INFORMATION] escape sequence \: detected in module option, no parameter verification is performed.
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous
 session found, to prevent overwriting, ./hydra.restore
[DATA] max 10 tasks per 1 server, overall 10 tasks, 10 login tries (l:1/p:10), ~1 try per task
[DATA] attacking http-get-form://127.0.0.1:80/DVWA/vulnerabilities/brute/:username=^USER^&password=^PAS
S^&Login=Login:F=Username and/or password incorrect.:H=Cookie\: security=low; PHPSESSID=dljdiq8r4uc8e5k
lk967qo64f0
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "aaaaaa" - 1 of 10 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "bbbbbbb" - 2 of 10 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "ccccccc" - 3 of 10 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "ddddddd" - 4 of 10 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "eeeeeeee" - 5 of 10 [child 4] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "ffffffff" - 6 of 10 [child 5] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "ggggggg" - 7 of 10 [child 6] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "hhhhhh" - 8 of 10 [child 7] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "iiiiii" - 9 of 10 [child 8] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "password" - 10 of 10 [child 9] (0/0)
[STATUS] 10.00 tries/min, 10 tries in 00:01h, 1 to do in 00:01h, 10 active
```

Figure 9: The displayed output from Hydra showcases the successful application of a brute-force attack, where the tool has determined valid access credentials



```
┌──(kali㉿DESKTOP-SK08UEQ)-[/mnt/c/Users/RAJ/Desktop/javascript]
└─$ hydra -L user.txt -P pass.txt 192.168.29.229 ssh -t 5
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not
-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-07
[DATA] max 5 tasks per 1 server, overall 5 tasks, 16 login tries (l:4/
[DATA] attacking ssh://192.168.29.229:22/
[22][ssh] host: 192.168.29.229   login: msfadmin   password: msfadmin
1 of 1 target successfully completed, 1 valid password found
```

Figure 10: The output from Hydra, a brute-force password-cracking tool, demonstrates an attempted security breach

5.1.3 Replay Attacks

Replay attacks involve an attacker duplicating or delaying genuine data transmissions to deceive the recipient into performing unintended actions. Systems often use time stamping or unique session identifiers to mitigate these threats. A session ID is generated upon successful user authentication in the Middleware Broker System. This ID is a protective measure to differentiate and authenticate each session, effectively preventing replay attacks.

5.1.4 Impersonation Attacks

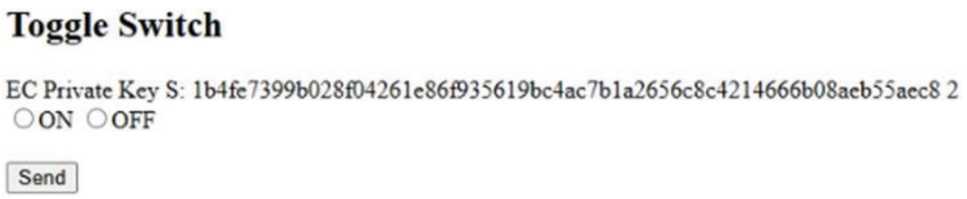
Impersonation attacks occur when a malicious actor pretends to be a legitimate user or device. Using ECC to generate unique keys for each device helps ensure that only verified devices can securely communicate on the network. To further enhance security, the Middleware Broker System identifies each component in the smart home network using a combination of IP and MAC addresses. Additionally, a unique private key is generated for each device and updated every time a user logs in, strengthening the overall security measures. Table 2 provides an overview of the proposed system's security performance compared to previous studies. The analysis underscores the robustness of the proposed protocol against a spectrum of attacks. It ensures data integrity by interrupting the authentication process whenever a transmitted message is tampered with or modified. Moreover, this model yields another crucial outcome from its use of asymmetric authentication between devices.

Table 2: Security comparisons with the previous scheme

No.	Property	Ours	Dang-Scheme
1	Asymmetric authentication	√	
2	D2D authentication	√	√
3	Resistance to impersonation attack	√	√
4	Resistance to brute force attack	√	
5	Resistance to man-in-the-middle attack	√	√

5.2 Authentication

An impersonation attack occurs when a malicious actor pretends to be someone else to gain unauthorized access to a system, steal sensitive data, or carry out harmful actions. These attacks often involve tactics such as IP spoofing or using fake credentials. To combat such threats, authentication verifies the identity of an individual, system, or entity by comparing provided credentials, such as passwords, biometric data, or cryptographic keys, with records or certifications from trusted sources. Authentication is crucial for security, controlling access to systems and data, preventing unauthorized use, and safeguarding against risks like identity theft and impersonation attacks. Figs. 11 and 12 reference in Module 2: Interface for IoT Controller, illustrate the authentication process via a user interface. Fig. 11 shows the synchronization of a device's secret or private key before a user initiates an action. Upon selecting a function, Fig. 12 displays the system's final output, including the device's IP, MAC addresses, and public key, finalizing the authentication process and ensuring secure communication within the IoT framework.

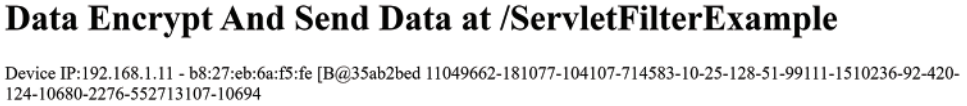


Toggle Switch

EC Private Key S: 1b4fe7399b028f04261e86f935619bc4ac7b1a2656c8c4214666b08aeb55aec8 2

○ON ○OFF

Figure 11: Module 2 features the Interface for IoT controller display, positioned on the wall to enable users to select device applications



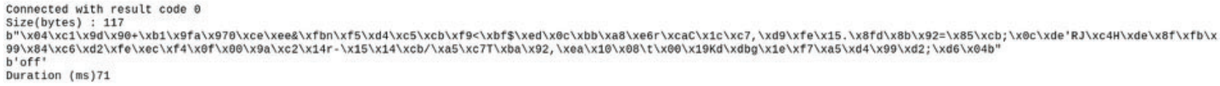
Data Encrypt And Send Data at /ServletFilterExample

Device IP:192.168.1.11 - b8:27:eb:6a:f5:fe [B@35ab2bed 11049662-181077-104107-714583-10-25-128-51-99111-1510236-92-420-124-10680-2276-552713107-10694

Figure 12: Module 2 presents the user interface for the IoT controller display, positioned on the wall after users select applications for their devices

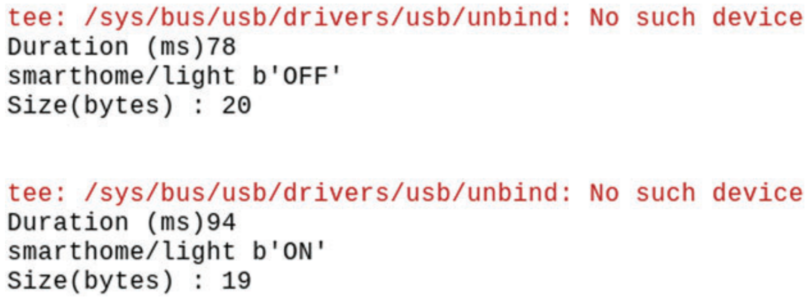
5.3 Integrity and Confidentiality

Data security is important to protect sensitive information from unauthorized access or change. Key aspects such as integrity and confidentiality are fundamental in this endeavor. Integrity ensures data accuracy, consistency, and resistance to modification, maintaining its authenticity and reliability over time. Confidentiality protects sensitive data from disclosure or unauthorized access by restricting information to those with valid permission. These principles are essential to maintaining trust and complying with regulatory requirements across the Financial, Healthcare, and cybersecurity sectors. These concepts are demonstrated through prototypes of the RPi, as shown in Figs. 13 and 14. Fig. 14 confirms the integrity and confidentiality of the data, as illustrated in Fig. 13. Despite an initial problem in which the RPi system was unable to decrypt the public key sent from the broker's intermediary system—suggesting a possible compromise of the user's public key—Fig. 13 confirms that the transmitted public key remains unchanged, accurate and accessible only to authorized parties, ensuring successful decryption by the RPi system.



```
Connected with result code 0
Size(bytes) : 117
b"\x04\xc1\x9d\x90+\xb1\x9fa\x970\xce\xee&\xfbn\xf5\xd4\xc5\xcb\xf9<\xbf$\xed\x0c\xbb\xa8\xe6r\xcaC\x1c\xc7,\xd9\xfe\x15.\x8fd\x8b\x92=\x85\xcb;\x0c\xde'RJ\xc4H\xde\x8f\xfb\x
99\x84\xc6\xd2\xfe\xec\xf4\x0f\x00\x9a\xc2\x14r-\x15\x14\xcb/\xa5\xc7T\xba\x92,\xea\x10\x08\t\x00\x19Kd\xdbg\x1e\xf7\xa5\xd4\x99\xd2;\xd6\x04b"
b'off'
Duration (ms)71
```

Figure 13: The RPi wall display showcases the data integrity following the description



```
tee: /sys/bus/usb/drivers/usb/unbind: No such device
Duration (ms)78
smarthome/light b'OFF'
Size(bytes) : 20

tee: /sys/bus/usb/drivers/usb/unbind: No such device
Duration (ms)94
smarthome/light b'ON'
Size(bytes) : 19
```

Figure 14: The data shown on the RPi wall did not undergo successful decryption

5.4 Detail of Simulation

Elliptic Curve Cryptography (ECC) offers robust security with shorter key lengths than traditional algorithms like RSA, making it ideal for resource-constrained environments like IoT. Its key generation, encryption, and decryption efficiency supports secure data transmission, which is crucial for IoT's limited computational capabilities. Hash chain algorithms complement ECC by ensuring data integrity and authenticity through lightweight security mechanisms, essential for IoT devices with restricted processing power and energy. Combining ECC and hash chains enhances security while maintaining efficiency, making it an effective solution for IoT applications where security and resource management are critical.

6 Performance and Efficiency Analysis

This section evaluates the proposed protocol's performance and efficiency, focusing on energy consumption and processing time. The "Energy Consumption" and "Processing Time Analysis" analyses have been consolidated to provide a clear and unified comparison of the protocols under study.

6.1 Consolidated Results for Energy Consumption and Processing Time

The "Energy Consumption" and "Processing Time" data are presented together to streamline the presentation and reduce duplication. This unified approach facilitates direct comparison and enhances the clarity of the findings.

6.1.1 Combined Analysis of Device-to-Server and Device-to-Device Authentication

The proposed protocol was evaluated for its energy efficiency and processing time across device-to-server and device-to-device (D2D) authentication scenarios. Key metrics include total energy consumption and the time required for authentication processes.

Energy Consumption Analysis

A deeper examination of computational load involves assessing the power consumption of both protocols. Given that the proposed protocol leverages elliptic curve cryptography, our analysis is based on a specific configuration of these cryptographic algorithms. Table 3 outlines the cryptographic algorithm configuration, including estimated energy consumption for operations in both schemes. According to our findings, the Dang-Scheme, utilizing Curve m-221, incurs an energy cost of 9480 μJ [32]. Conversely, the proposed scheme also employs elliptic curve cryptography (ECDLP and ECDH) with the curve shown in Fig. 14. Fig. 15 represents the energy consumption analysis of the proposed ECC-integrated MQTT protocol for device-server authentication within the Middleware Broker System. This analysis highlights the energy efficiency of the ECC-based security framework compared to other authentication schemes. This reduction in energy consumption underscores the protocol's efficiency and suitability for resource-constrained IoT devices, contributing to its practical feasibility.

Curve attributes: a = −5.4, b = 16.8, Curve: $y^2 = x^3 + (-5.4)\,x + 16.8$, Point J = (3.2|5.68), Point K = (−1.6|−4.62), Point L = J + K = (3.007|−5.267)

The Interpreting the Curve is:

(i) X-Axis (Time in mS):

-This axis represents the time it takes for the communication between the middleware broker and the device (RPi) to occur.

(ii) Y-Axis (Energy Consumption in μJ):

-This axis shows the energy consumption measured in microjoules (μJ) for the respective duration on the X-axis.

(iii) Curve Representation:

-The curve on the graph illustrates the energy consumption trend of the ECC operations over time.

-A lower position on the Y-axis indicates lower energy consumption for a given time duration on the X-axis, which translates to higher energy efficiency.

(iv) Key Observations:

-The Middleware Broker System consumes 3456 μJ for a communication duration of 18 ms, highlighting its efficiency.

Table 3: Energy consumption of ECC operations

Notation	Protocol	Time in ms	Energy consumption (μJ)
Elliptic curve cryptography	Proposed protocol	18	3456
	Dang-Scheme	1019	9480

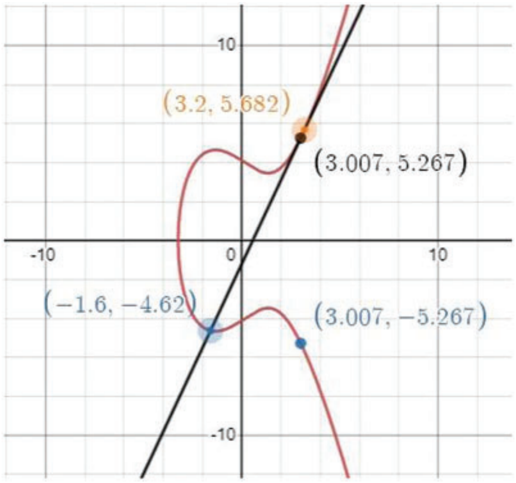


Figure 15: The proposed scheme curves

The curve representation depicts the energy consumption trend of ECC operations, with lower positions on the Y-axis indicating higher energy efficiency for shorter communication times. The consolidated results in Table 3 demonstrate that the proposed protocol significantly reduces energy consumption compared to the Dang-Scheme. Specifically, the Middleware Broker System achieves energy savings of up to 77.04%, highlighting its suitability for resource-constrained IoT environments.

Fig. 16 has been revised to include energy consumption and processing time data, providing a clear visual performance comparison across the evaluated protocols.

Table 4 compares energy consumption across various schemes during two D2D operation phases: Total Device Request and Accepted. It underscores the effectiveness of these schemes in optimizing energy management during D2D operations and verification. The table evaluates performance and efficiency based on energy consumption, with lower values indicating better energy efficiency. Energy usage is influenced by duration, ECC curve, and the type of data transmitted. Notably, the Middleware Broker System achieved an 87.68% efficiency in managing total device requests and a 71.88% efficiency during the verification phase, surpassing the performance of other schemes. The increase in data secrecy detailed in the paper is calculated by incorporating Elliptic Curve Cryptography (ECC) into the Message Queue Telemetry Transport (MQTT) communication framework. This integration enables secure data encryption before transmission. According to the paper, this approach significantly enhances data security, resulting in an 87.68% improvement in data secrecy. It likely stems from a comparative analysis of data protection levels before and after the implementation of ECC, potentially using metrics such as the success rate of unauthorized data access or the strength of the encryption in resisting cryptographic attacks within the MQTT protocols publish/subscribe communication process in smart homes. The exact computation would involve evaluating the improvements in encryption strength and the reduced vulnerabilities, leading to the reported increase in data secrecy.

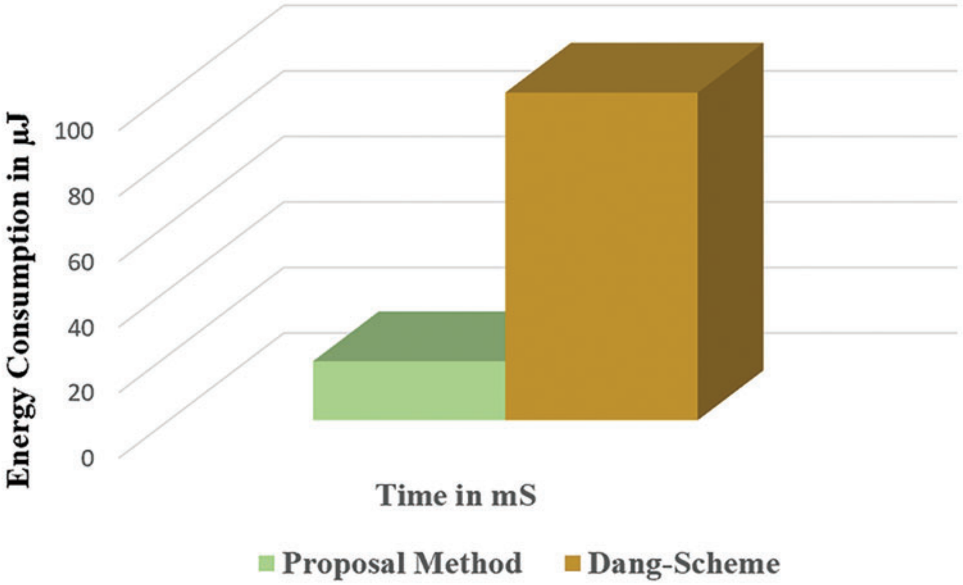


Figure 16: Time and energy comparison: Proposal method *vs*. Dang-Scheme

Table 4: Comparative analysis of D2D energy use

Phase	Operation	Proposed protocol	Dang-Scheme
D2D authentication	Total device request (µJ)	4032	28,704
	Total device accepted (µJ)	7488	19,143

Processing Time Analysis

As shown in Table 5, the proposed protocol outperforms the Dang-Scheme in the D2D authentication process, with processing times of 74 ms for total device requests and 21 ms for accepted devices, compared to 417 and 225 ms, respectively.

Table 5: Processing time represents a system's total duration to complete a task or operation

Phase	Device	Proposed protocol	Dang-Scheme
D2D authentication	Total device request (µJ)	74 ms	417 ms
	Total device accepted (µJ)	21 ms	225 ms

6.2 Summary of Key Findings

The combined analysis of "Energy Consumption" and "Processing Time" demonstrates the proposed protocol's energy use and processing speed efficiency, making it highly suitable for deployment in IoT environments. The presentation is streamlined by consolidating these results into a single section, highlighting the key findings more effectively.

6.3 Analysis of Data Communication Efficiency

Examining data communication efficiency involves understanding how data is transmitted across a network while optimizing performance and user experience. This assessment considers various factors, including bandwidth, latency, throughput, error rates, and protocols. Protocols, such as TCP/IP, UDP, HTTP, and FTP, are particularly significant as they set the rules for data transmission. In this study, the HTTP protocol used within the Java Servlet platform is especially relevant. Table 6 evaluates data usage during encryption and decryption across various protocols to determine their effectiveness in specific scenarios. Our approach simplifies the process by requiring only asymmetric authentication, using a 116-bit private key and a 117-bit public key (ON and OFF states), and consuming only 7488 μJ of energy. This significant reduction in power consumption is crucial, allowing the device to manage authentication requests from the source device with fewer resources. This feature is precious in defending against Distributed DDoS attacks, where attackers overwhelm the target with numerous bogus requests. In such cases, the target device must conserve resources until the attack is detected, a strength demonstrated during the D2D authentication phase. Conversely, the Dang-Scheme method incurs higher energy costs due to handling timestamps and data lengths, totalling 19,143 μJ.

Table 6: Data length of operation and messages exchanged

Data	Protocol	Length
Encryption/decryption operations	Proposed Scheme	116-bit to ON and 117-bit to OFF
	Dang-Scheme	14 data blocks of 128-bit

7 Scalability and Complexity Analysis

The proposed system is designed with scalability, particularly for IoT environments requiring efficient processing and minimal latency. Scalability here refers to the system's ability to maintain performance and security standards as the number of connected devices or data volume increases. The gateway's capacity to handle a growing number of IoT devices supports the system's scalability, assuming that the gateway's processing power and memory can be scaled accordingly. Computationally, the system relies on ECC, which offers an efficient time complexity based on the bit-length of cryptographic keys. It ensures rapid cryptographic operations even as the network grows, with most computational tasks offloading to the gateway to ease the burden on individual IoT devices. Space complexity is also optimized, with ECC's smaller key sizes reducing memory requirements and a design that processes only essential bits during authentication, minimizing the memory footprint. The system's scalability is contingent on the gateway's capacity, network bandwidth, and the basic cryptographic capabilities of IoT devices. The proposed system can scale effectively under these conditions, maintaining high performance and security levels even as the network expands. Future research should investigate the system's scalability in more extensive networks and adaptability across various IoT scenarios.

8 Discussions

The middleware broker system architecture and the ECC framework present various applications and benefits across multiple domains. Implementing and testing this architecture using a Raspberry Pi in smart home environments enhances security, scalability, and efficiency by enabling secure messaging workflows through IoT gateways and MQTT brokers. IoT systems, in general, benefit significantly from this approach, as the elliptic curve cryptography ensures robust security with reduced key sizes and faster computations, making it ideal for environments with limited computational resources. Additionally, ad hoc wireless and peer-to-peer (P2P) networks gain from quickly establishing secure communication channels facilitated by the proposed D2D authentication protocols. These address challenges like limited device capabilities and the urgent need for rapid, secure connections. Furthermore, the approach is particularly beneficial for smart cities, where the unique security challenges of interconnected devices in complex environments are effectively managed through the novel authentication methodology and the ECC framework, ensuring secure and efficient communication. These advantages make the middleware broker system and ECC framework highly applicable and beneficial across these domains.

9 Comparative Evaluation

To comprehensively understand the improvements introduced by our proposed solution, we conducted a comparative evaluation against several related works in IoT security and smart home environments. The comparison focuses on key metrics such as energy consumption, computational efficiency, and security.

9.1 Energy Consumption

Energy efficiency is critical in IoT environments, particularly for devices with limited power resources. We compared our proposed solution's energy consumption with related works, specifically the protocols developed by the Dang Scheme [32]. Our results demonstrate a significant reduction in energy usage. For instance, during the device-to-device (D2D) authentication process, our solution consumed only 4032 μJ, compared to 28,704 μJ reported by Dang Scheme [32]. This 85% reduction highlights the superior energy efficiency of our approach, making it highly suitable for resource-constrained environments.

9.2 Computational Efficiency

We also evaluated the computational overhead associated with our solution compared to other protocols. Computational efficiency is vital for maintaining system responsiveness, especially in real-time applications. Using Elliptic Curve Cryptography (ECC) in our protocol significantly reduces the computational burden. For example, during the device-server authentication process, our solution completed the operation in just 18 ms, compared to 1019 ms for the protocol proposed by Dang-Scheme [32]. This drastic improvement in processing speed, by over 98%, underscores the practicality of our solution in real-world applications.

9.3 Security

Security is paramount in protecting IoT systems from various cyber threats. We compared the security features of our solution with those of other protocols, focusing on resistance to brute force, man-in-the-middle, and replay attacks. Our solution, which integrates dynamic key management and session ID generation, provides robust protection against these threats. While the protocol by Dang et al. [32] addresses some of these concerns, it lacks the comprehensive defence mechanisms found in our approach. For example, our protocol's use of ECC for generating unique private keys for each session significantly reduces the risk of impersonation attacks, which was a noted vulnerability in Dang et al. [32].

9.4 Summary of Comparative Results

The comparative analysis reveals that our proposed solution outperforms existing protocols regarding energy efficiency and computational speed and offers enhanced security features. These improvements are crucial for deploying secure, efficient, and reliable IoT systems in smart home environments.

10 Novel Reflected in the Main Contributions

The novelty of this research is reflected in several key contributions. We propose a novel, lightweight communication proxy designed to enhance the scalability and security of MQTT-based IoT networks. Central to this innovation is the adaptation and enhancement of the Dang-Scheme, integrating Elliptic Curve Cryptography (ECC) to significantly strengthen device authentication, data confidentiality, and energy efficiency. The proposed solution achieves an 87.68% increase in data secrecy and up to 77.04% energy savings during publish/subscribe communications. Furthermore, the Middleware Broker System dynamically manages transaction keys and session IDs, providing robust defences against common cyber threats like impersonation and brute-force attacks. These innovations collectively represent a substantial advancement in IoT security, offering a scalable and efficient solution tailored for resource-constrained environments.

11 Conclusion

The rapid advancement of smart homes, smart cities, and various IoT sectors is becoming a cornerstone of the future digital landscape, introducing new security challenges that differ from those in previous technological eras. This study addresses these challenges by proposing an innovative authentication methodology designed explicitly for asymmetric authentication, enabling seamless device-to-server communication within IoT ecosystems. A middleware security agent that authorizes devices efficiently manages the authentication process, facilitating communication between them and reducing the server's computational load, a key aspect of our approach. This innovation's heart is a novel authentication protocol combining simplicity with robust security, leveraging Elliptic Curve Cryptography (ECC) for its strong encryption capabilities and efficient operations. Rigorous security analysis has proven the robustness of our proposed scheme against common cyberattacks in IoT environments while also demonstrating exceptional energy efficiency, consuming just 4032 μJ during device verification and optimizing memory usage for session and expiration keys. Simulation results further highlight the proposal's efficiency, with the gateway managing most computational tasks, which instils confidence in the practicality of our system for real-world applications, especially in domains like smart cities and sustainable environments. As the IoT landscape evolves, ensuring security and efficiency remains paramount, and our authentication protocols stand out as a beacon of innovation and resilience in this dynamic field. Moreover, this study has demonstrated the effectiveness of integrating ECC and hash chain algorithms to enhance security for IoT systems, particularly in resource-constrained environments. However, challenges remain in optimizing computational efficiency and scalability across diverse IoT networks. Future research should focus on refining these algorithms to reduce computational overhead, explore scalability, and address potential vulnerabilities in real-world scenarios, emphasizing the interdisciplinary nature of collaboration needed to fully understand the implications of integrating AI with cryptographic security protocols in IoT. The proposed solution significantly enhances MQTT-based IoT network security, offering a scalable and efficient framework that advances IoT security, particularly in modern, resource-constrained environments.

Acknowledgement: The authors would like to thank Universiti Sains Malaysia (USM), the Ministry of Higher Education Malaysia for providing the research grant, Fundamental Research Grant Scheme and Ajman University, United Arab Emirates that helped to carry out this research.

Funding Statement: This work was supported through Universiti Sains Malaysia (USM) and the Ministry of Higher Education Malaysia providing the research grant, Fundamental Research Grant Scheme (FRGS–Grant No. FRGS/1/2020/TK0/USM/02/1).

Author Contributions: Study conception and design: Zainatul Yushaniza Mohamed Yusoff, Mohamad Khairi Ishak, Lukman A. B. Rahim, Mohd Shahrimie Mohd Asaari; data collection: Zainatul Yushaniza Mohamed Yusoff; analysis and interpretation of results: Zainatul Yushaniza Mohamed Yusoff, Mohamad Khairi Ishak, Lukman A. B. Rahim, Mohd Shahrimie Mohd Asaari; draft manuscript preparation: Zainatul Yushaniza Mohamed Yusoff, Mohamad Khairi Ishak, Mohd Shahrimie Mohd Asaari. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: This article does not involve data availability, and this section is not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Rizou, E. A. Egyptiadou, Y. Ishibashi, and K. E. Psannis, “Preserving minors’ data protection in IoT-based smart homes according to GDPR considering cross-border issues,” *J. Commun.*, vol. 17, no. 3, pp. 180–187, 2022. doi: 10.12720/jcm.17.3.180-187.

[2] V. Nyangaresi, “Lightweight anonymous authentication protocol for resource-constrained smart home devices based on Elliptic curve cryptography,” *J. Syst. Archit.*, vol. 13, Oct. 2022, Art. no. 102763. doi: 10.1016/j.sysarc.2022.102763.

[3] J. Sicato, P. Sharma, V. Loia, and J. Park, “VPNFilter malware analysis on cyber threat in smart home network,” *Appl. Sci.*, vol. 9, no. 13, 2019, Art. no. 2763. doi: 10.3390/app9132763.

[4] N. Amraoui and B. Zouari, “Securing the operation of smart home systems: A literature review,” *J. Reliab. Intell. Environ.*, vol. 8, no. 1, pp. 67–74, 2022. doi: 10.1007/s40860-021-00160-3.

[5] Y. Guo, Z. Zhang, and Y. Guo, “Secfhome: Secure remote authentication in the fog-enabled smart home environment,” *Comput. Netw.*, vol. 207, 2022, Art. no. 108818. doi: 10.1016/j.comnet.2022.108818.

[6] W. Yan, Z. Wang, H. Wang, W. Wang, J. Li and X. Gui, “Survey on recent smart gateways for smart home: Systems, technologies, and challenges,” *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 6, 2022, Art. no. e4067. doi: 10.1002/ett.4067.

[7] G. Goyal, P. Liu, and S. Sural, “Securing smart home IoT systems with attribute-based access control,” in *Proc. 2022 ACM Workshop Secur. Trustworthy Cyber-Phys. Syst.*, Baltimore, MD, USA, ACM, Apr. 27, 2022, pp. 37–46. doi: 10.1145/3510547.3517920.

[8] A. Huszti, S. Kovács, and N. Oláh, “Scalable, password-based and threshold authentication for smart homes,” *Int. J. Inf. Secur.*, vol. 21, no. 4, pp. 707–723, 2022. doi: 10.1007/s10207-022-00578-7.

[9] H. Muhammad *et al.*, “Secure authentication protocol for home area network in smart grid-based smart cities,” *Comput. Elect. Eng.*, vol. 108, 2023, Art. no. 108721. doi: 10.1016/j.compeleceng.2023.108721.

[10] Y. Xiao, Y. Jia, C. Liu, A. Alrawais, M. Rekik and Z. Shan, “Homeshield: A credential-less authentication framework for smart home systems,” *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7903–7918, 2020. doi: 10.1109/JIOT.2020.3003621.

[11] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar and K. R. Choo, “HomeChain: A blockchain-based secure mutual authentication system for smart homes,” *IEEE Internet Things J.*, vol. 7, no. 2, pp. 818–829, 2020. doi: 10.1109/JIOT.2019.2944400.

[12] L. Yang, X. Liu, and W. Gong, “Secure smart home systems: A blockchain perspective,” in *39th IEEE Conf. Comput. Commun., INFOCOM Workshops 2020*, Toronto, ON, Canada, IEEE, Jul. 6–9, 2020, pp. 1003–1008. doi: 10.1109/INFOCOMWKSHPS50562.2020.9162648.

[13] X. Qin, Y. Huang, and X. Li, “An ECC-based access control scheme with lightweight decryption and conditional authentication for data sharing in vehicular networks,” *Soft Comput.*, vol. 24, pp. 18881–18891, Dec. 2020. doi: 10.1007/s00500-020-05117-x.

[14] S. Ji, R. Huang, J. Shen, X. Jin, and Y. Cho, “A certificates sign encryption scheme for smart home networks,” *Concur Comput. Pract. Exp.*, vol. 33, no. 7, 2021. doi: 10.1002/cpe.5081.

[15] Z. Huang, L. Zhang, X. Meng, and K. R. Choo, “Key-free authentication protocol against subverted indoor smart devices for smart home,” *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1039–1047, 2020. doi: 10.1109/JIOT.2019.2948622.

[16] M. Alshahrani and I. Traore, “Secure mutual authentication and automated access control for IoT smart home using cumulative keyed-hash chain,” *J. Inf. Secur. Appl.*, vol. 45, pp. 156–175, 2019. doi: 10.1016/j.jisa.2019.02.003.

[17] S. Dey and A. Hossain, “Session-key establishment and authentication in a smart home network using public key cryptography,” *IEEE Sens. Lett.*, vol. 3, no. 4, pp. 1–4, 2019. doi: 10.1109/LSENS.2019.2905020.

[18] H. Luo, C. Wang, H. Luo, F. Zhang, F. Lin and G. Xu, “G2F: A secure user authentication for rapid smart home IoT management,” *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10884–10895, 2021. doi: 10.1109/JIOT.2021.3050710.

[19] P. Gope and B. Sikdar, “Lightweight and privacy-preserving two-factor authentication scheme for IoT devices,” *IEEE Internet Things J.*, vol. 6, no. 1, pp. 580–589, 2019. doi: 10.1109/JIOT.2018.2846299.

[20] W. Iqbal *et al.*, “ALAM: Anonymous lightweight authentication mechanism for SDN-enabled smart homes,” *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9622–9633, 2021. doi: 10.1109/JIOT.2020.3024058.

[21] W. M. Kang, S. Y. Moon, and J. H. Park, “An enhanced security framework for home appliances in smart home,” *Hum. Centric Comput. Inf. Sci.*, vol. 7, no. 1, 2017, Art. no. 6. doi: 10.1186/s13673-017-0087-4.

[22] M. Wazid, A. K. Das, V. Bhat, and A. V. Vasilakos, “LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment,” *J. Netw. Comput. Appl.*, vol. 150, 2020, Art. no. 102496. doi: 10.1016/j.jnca.2019.102496.

[23] S. Challa *et al.*, “Secure signature-based authenticated key establishment scheme for future IoT applications,” *IEEE Access*, vol. 5, pp. 3028–3043, 2017. doi: 10.1109/ACCESS.2017.2676119.

[24] C. Pham, T. Nguyen, and T. Dang, “Resource-constrained IoT authentication protocol: An ECC-based hybrid scheme for device-to-server and device-to-device communications,” in *Future Data and Security Engineering*, Springer, Cham, Nov. 2019, pp. 446–466. doi: 10.1007/978-3-030-35653-8_30.

[25] S. S. Chowdhury, S. Sarkar, S. Syamal, S. Sengupta, and P. Nag, “IoT-based smart security and home automation system,” in *2019 IEEE 10th Ann. Ubiquitous Comput., Electron. & Mobile Commun. Conf. (UEMCON)*, Oct. 2019, pp. 1158–1161. doi: 10.1109/UEMCON47517.2019.8992994.

[26] A. Singh, D. Gupta, and N. Mittal, “Enhancing home security systems using IoT,” in *2019 3rd Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Jun. 2019, pp. 133–137. doi: 10.1109/ICECA.2019.8821833.

[27] H. Vasudev, V. Deshpande, D. Das, and S. K. Das, “A lightweight mutual authentication protocol for V2V communication in the internet of vehicles,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6709–6717, 2020. doi: 10.1109/TVT.2020.2986585.

[28] S. Garg, K. Kaur, G. Kaddoum, J. J. P. C. Rodrigues, and M. Guizani, “Secure and lightweight authentication scheme for smart metering infrastructure in smart grid,” *IEEE Trans. Ind. Inform.*, vol. 1, 2019. doi: 10.1109/TII.2019.

[29] A. Kumar, P. Kumar, and R. Agrawal, “A face recognition method in the IoT for security appliances in smart homes, offices and cities,” in *2019 3rd Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Mar. 2019, pp. 964–968. doi: 10.1109/ICCMC.2019.8819790.

[30] K. L. Raju, V. Chandrani, S. S. Begum, and M. P. Devi, “Home automation and security system with node MCU using Internet of Things,” in *2019 Int. Conf. Vis. Towards Emerg. Trends Commun. Netw. (ViTECoN)*, Vellore, India, 2019, pp. 1–5. doi: 10.1109/ViTECoN.2019.8899540.

[31] Z. Vahdati, A. Ghasempour, M. Salehi, and S. M. Yasin, “Comparison of ECC and RSA algorithms in IoT devices,” *J. Theor. Appl. Inform. Technol.*, vol. 97, 2019, Art. no. 4293.

[32] T. Sasi, A. H. Lashkari, R. Lu, P. Xiong, and S. Iqbal, “A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges,” *J. Inform. Intell.*, vol. 98, 2023, Art. no. 65. doi: 10.1016/j.jiixd.2023.12.001.

[33] IBM, “What is an attack surface?,” 2024. Accessed: Aug. 27, 2024. [Online]. Available: https://www.ibm.com/topics/attack-surface

[34] J. M. Borky and T. H. Bradley, “Protecting information with cybersecurity,” in *Effective Model-Based Systems Engineering*. Springer, Sep. 9, 2018, pp. 345–404. doi: 10.1007/978-3-319-95669-5_10.

[35] T. K. Dang, C. D. M. Pham, and T. L. P. Nguyen, “A pragmatic elliptic curve cryptography-based extension for energy-efficient device-to-device communications in smart cities,” *Sustain. Cities Soc.*, vol. 56, 2020, Art. no. 102097. doi: 10.1016/j.scs.2020.102097.

[36] K-H. Wang, C-M. Chen, W. Fang, and T-Y. Wu, “A secure authentication scheme for Internet of Things,” *Pervasive Mob. Comput.*, vol. 42, pp. 15–26, 2017. doi: 10.1016/j.pmcj.2017.09.004.