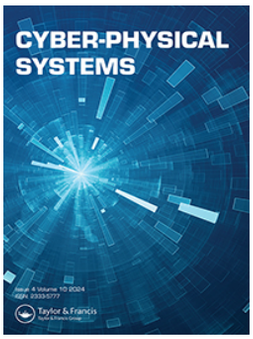

Cyber-Physical Systems

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tcyb20

Blockchain-based group signature for secure authentication of IoT systems in smart home environments

Mustafa Kara, Hisham R.J. Merzeh, Muhammed Ali Aydin & Hasan Hüseyin Balik

To cite this article: Mustafa Kara, Hisham R.J. Merzeh, Muhammed Ali Aydin & Hasan Hüseyin Balik (2024) Blockchain-based group signature for secure authentication of IoT systems in smart home environments, Cyber-Physical Systems, 10:4, 362-386, DOI: 10.1080/23335777.2024.2324798

To link to this article: https://doi.org/10.1080/23335777.2024.2324798

CYBER-PHYSICAL SYSTEMS
2024, VOL. 10, NO. 4, 362–386
https://doi.org/10.1080/23335777.2024.2324798

Taylor & Francis
Taylor & Francis Group

Blockchain-based group signature for secure authentication of IoT systems in smart home environments

Mustafa Kara[a], Hisham R.J. Merzeh[b], Muhammed Ali Aydin[c] and Hasan Hüseyin Balik[d]

[a]Department of Computer Engineering, Air Force Academy, National Defence University, Istanbul, Turkey; [b]Department of Computer Engineering, College of Electrical and Electronic Technology Middle Technical University, Baghdad, Iraq; [c]Department of Computer Engineering, Istanbul University-Cerrahpaşa, Istanbul, Turkey; [d]Department of Computer Engineering, Istanbul Aydin University, Istanbul, Turkey

ABSTRACT

Current solutions that rely on a single-server architecture have privacy, anonymity, integrity, and confidentiality limitations. Blockchain-based solutions can address some of these issues but face challenges regulating behaviour and protecting access policy privacy. This study proposes a new approach to securing a smart home environment to overcome these limitations.The proposed architecture is based on a group signature scheme, which allows multiple IoT devices to securely exchange keys and authenticate each other without needing a central authority. Our experimental results indicate the effectiveness and efficiency of the proposed architecture and provide insights into its security and privacy features compared to existing IoT authentication methods.

ARTICLE HISTORY
Received 12 September 2023
Accepted 21 February 2024

KEYWORDS
IoT; authentication; blockchain; group signature; key exchange

1. Introduction

Smart homes are becoming increasingly prevalent with the growth of the Internet of Things (IoT) technology [1–4]. Smart homes are equipped with advanced automation systems that allow homeowners to remotely control and monitor their home's appliances and systems, such as lighting, heating, and security [5]. Integrating IoT and smart homes allows homeowners to remotely control and monitor their home's appliances and systems through a smart home app or voice assistant. The homeowners utilise a smart home application to power down the lights, regulate the temperature, or monitor the status of their home security system when they are not present [6,7]. Also, integrating IoT and smart homes can provide homeowners with convenience and energy efficiency benefits. However, this also leaves these

CONTACT Mustafa Kara mkara@hho.msu.edu.tr Department of Computer Engineering, Air Force Academy, National Defence University, Istanbul, Turkey

devices vulnerable to hacking and exploitation, making security a critical concern [8]. IoT devices can be vulnerable to attacks if they have weak authentication mechanisms or lack other basic security measures. Hackers may also try to exploit vulnerabilities in the device's software or hardware to gain access. Furthermore, IoT devices are connected to the internet and often have access to sensitive data. Therefore, they can be vulnerable to cyber-attacks [9,10]. Hackers may try to gain access to an IoT device to steal sensitive information, disrupt the device's operation, or use it as a foothold to launch further attacks. To guarantee the safety of these smart home environments, finding a solution that can provide secure and transparent authentication for IoT devices [11–13].

Security is a significant issue in IoT systems [14,15]. The nature of smart devices, with limited resources, makes securing these smart devices a significant challenge. In recent years, there have been many research studies aimed at solving the security problems associated with smart homes [7,16,17]. Some solutions propose using a single server architecture to manage authentication, but these solutions are limited by their need for more privacy and anonymity. Another concern with IoT is the possibility of a vulnerable key exchange mechanism [18,19]. Lack of strong authentication, poorly designed key exchange algorithms, lack of forward secrecy, weak keys, key reuse issues, etc., can cause this vulnerability.

This paper proposes blockchain-based smart key exchange architecture to authenticate IoT systems securely. The proposed architecture set up a private blockchain network to implement a decentralised smart key exchange. A smart contract is also used for a self-executing contract that runs on the blockchain and contains the logic for key exchange between devices. The proposed architecture is used to facilitate the secure key exchange between devices. Therefore, the blockchain-based smart key exchange can provide a secure and transparent mechanism to authenticate IoT devices in smart homes. The experimental results demonstrate the effectiveness and efficiency of the proposed architecture and provide insights into its security and privacy features compared to existing IoT authentication methods provided in the literature.

The main contributions of this study can be outlined as follows:

- The introduced architecture aims secure remote mutual authentication between users and the home gateway. According to similar works in the literature, this is more secure and faster authentication approach for smart home systems.
- This paper also utilises a group signature for verification of messages between IoT smart devices and system users. The smart key exchange based on the blockchain provides a secure approach to create shared keys to encrypt transmitting messages and exchange them securely. The

group signature and smart Diffie Hellman Key Exchange (DHKE) are used to design a decentralised authentication model. As a result, it increases fault tolerance and resists the single point of failure, making the system more secure and reliable.

- The experiment results compares the proposed architecture with the *HomeChain* model [7]. According to comparisons, the proposed key exchange mechanism minimises the number of block generations, thus speeding up the authentication and key exchange process. As will be seen in the results, the proposed model is five times faster than the *HomeChain* model, which uses asynchronous communication and requires four transactions for each control order.

This study is structured as follows. Section 2 reviews the related works in the field. In Section 3, the proposed architecture of a smart authentication mechanism in IoT is presented along with the necessary background information. Section 4 presents the results and analysis of the proposed architecture. Finally, in Section 5, the paper concludes with a summary of the results.

2. Related work

This section reviews the current highest level of development, performance, or technology achieved in smart home environments and the relevant contributions related to the integration of blockchain and IoT. For example, Chao Lin et al. [7] incorporated group signature and blockchain technology to ensure access policies' traceability and privacy protection. The related study, Homechain, utilises the group signature to verify a group request. By utilising these methods, Homechain can anonymously authenticate group members. All control requests and responses are stored in blocks in the blockchain. This approach can provide a high-security level, but it is time-consuming since the block creation needs mining, in addition to the money cost of this process. Furthermore, there needs to be a precise key exchange mechanism to be used in this approach. There is no direct channel between entities and home gateways. Therefore a synchronisation problem will occur since the requests and responses will receive indirectly through the smart contract without any notification from the sender entities to the receiver entities. Our proposed approach is a novel authentication model against Homechain. Lee et al. [11] proposed an Ethereum-based solution for smart homes that address confidentiality, integrity, and authentication issues related to IoT devices and centralised gateways. However, their design should have considered the additional computational complexity that blockchain introduces. As a result, while their model includes features such as private blockchain use, confidentiality, availability, prevention of Distributed Denial-of-Service (DDoS) attacks, and cloud storage, it lacks certain features such as hardware implementation, integrity, immutability,

scalability, minimising response time, and cost-effectiveness. This study contributes a blockchain-based architecture for a smart home gateway network, which aims to address current issues with centralised security network architectures and combat future attacks on smart home gateways. However, the proposed architecture is vulnerable as the gateway can be a single point of failure, and there needs to be an approach designed to tackle this issue. Dang et al. [20] preferred using Ganache, Remix, and web3.js architecture for a Smart Home-based IoT Blockchain (SHIB) to tackle challenges related to data privacy, trust access control, and system scalability in their study. They suggested an IoT gateway to connect a group of IoT devices to a blockchain network within the smart home. However, their approach could be more complex as each user and IoT device is associated with only one subject-object pair. In addition, the gateway may require substantial computing power to handle significant transactions. Finally, although they use smart contracts, they do not apply encryption, anonymisation, or evaluate the security of the smart contract.

Ammi et al. [12] propose an innovative blockchain-based solution for secure smart home systems, which employs a combination of hyper ledger fabric and hyper ledger composer. This could lead to the development of different applications with high security and reliability for smart homes. However, possible threats to the study include message modification, replay attacks, and eavesdropping. Security goals such as availability, integrity, authenticity, and confidentiality can be compromised [21]. Ren et al. [22] proposed an identity-based proxy aggregate signature (IBPAS) scheme to enhance the efficiency of signature verification, which could save storage space and reduce communication bandwidth. However, most research has centred on securing cloud data using blockchain technology [6,23]. The appropriate application of blockchain technology adds an extra layer of security to cloud data, and users can maintain their trust while outsourcing information.

Consequently, legitimate users can access the relevant information without any issues. Gauhar et al. [24] suggested an access control framework named xDBAuth, which employs smart contracts to manage internal and external users and IoT devices. xDBAuth imposes lower computational overhead than other methods and can handle high throughput in numerous concurrent requests. In addition, they have some cyber security properties like authentication and non-repudiation, with low confidentiality based on Transport Layer Security (TLS) and unknown blockchain technology.

Using a blockchain for group signature verification provides several benefits. Firstly, it ensures that each device is part of the authorised group, as only the authorised devices have access to the group's public key. Secondly, it provides a secure and tamper-proof record of all the transactions related to the group signature, which makes it difficult to forge or modify the signature. Finally, it provides transparency, as all the transactions related to the group signature can be publicly viewed and audited. Zhang and Lee [23] proposed a group

signature-based authentication scheme for blockchain-based mobile-edge computing to validate blocks created with blockchain. Jiang et al. [25] proposed a group signature-based authentication approach for vehicular ad-hoc networks, utilising trusted region authority to achieve conditional privacy and anonymity. Zhang et al. [26] proposed a group session key and batch group signature-based authentication scheme for vehicular ad hoc networks to achieve computation efficiency. This work proposes a multiple-group model where each group creates a group signature for all members to sign transactions and communicate anonymously with other remote devices. While these previous studies address security and privacy concerns, critical challenges still require other solutions.

In this study, the proposed innovative architecture offers a promising solution to the security challenges that smart homes and IoT systems face. A blockchain-based smart key exchange can authenticate Internet of Things (IoT) devices by allowing them to exchange keys for communication securely. Based on the features of similar works, our architecture demonstrates superior performance, low computational costs, and increased resilience to security threats. The results and analysis section provides a detailed comparison and performance evaluation.

3. The proposed architecture: smart authentication mechanism in IoT

With the rising number of connected devices in IoT, security has become a major concern. This study proposes an authenticated communication architecture for smart home environments to address the insecure harvesting and exfiltration of sensitive information from these devices [7]. While there are solutions available to authenticate nodes in smart home networks [4,6,17], the proposed architecture should ensure the security of a key exchange mechanism using solid and trusted keys, forward secrecy, and verify the authenticity of the keys being exchanged. The solution is based on a group signature scheme, which allows multiple IoT devices to securely exchange keys and authenticate each other without needing a central authority. The group signature is verified using a blockchain, providing secure and transparent authentication while ensuring each device is part of the authorised group. In addition, blockchain has several key characteristics, such as decentralisation (meaning it operates across a network of many different nodes rather than relying on a central authority), traceability (allowing for a transparent and auditable history of transactions), and anonymity (allowing users to remain anonymous while still being able to transact). These features help to solve problems related to poor trust between evaluation nodes, unreliable data sharing, and unclear trust relationships. Also, the proposed architecture can be more secure than traditional authentication methods because the keys are stored on the blockchain, a distributed and immutable ledger. This means that a single entity cannot

easily compromise or alter the keys, and the authentication process is transparent and verifiable.

This can provide synchronised real-time communication between system devices. Encrypted data transmitted between connected devices concerns user privacy and complies with GDPR. The distributed architecture used in this approach provides high fault tolerance and resists many security threats like DDoS, single point of failure, and MITM.

This chapter outlines designing and implementing a secure authentication system that protects IoT devices and their transmitted data. The proposed architecture combines blockchain and group signatures for efficient and effective authentication and a smart key exchange mechanism.

3.1. Centralized smart home architecture

The household appliance collection with wireless communication interfaces forms the home network. Every home has a wired, wireless, or both network and data collected from each device are sent to a central station, which we call the home sink or hub. Each home network smart device is a node with reasonable computation and communication abilities. The home hub can be any device, such as a smart metre, PC, tablet, or smartphone, with data storage capacity, the ability to perform local processing, and communication with devices outside the home network [27].

The traditional centralised model is depicted in Figure 1. A centralised smart home communication architecture is used in smart homes to manage the communication between different smart devices and appliances within the home.

In this architecture, a central hub or controller is used to manage the communication between the various devices in the home rather than having each

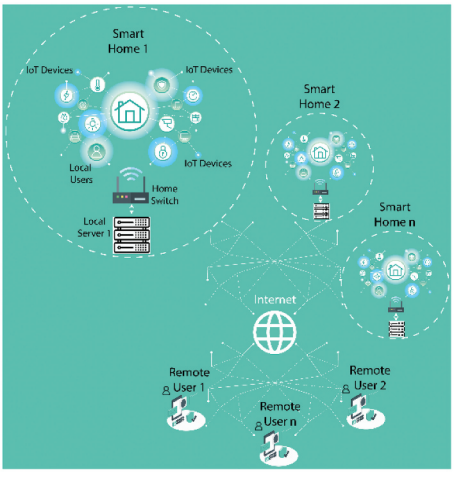

Figure 1. Centralized smart home communication architecture.

device communicate directly with other devices. One potential disadvantage of a centralised smart home communication architecture is that it can create a single point of failure. The single point of failure and low fault tolerance can threaten a single server architecture [17]. If the central server or controller fails, all of the devices in the home may be affected, and it may be difficult to access or control these devices until the issue is resolved.

The system admin will deploy the smart contract in the blockchain platform. The blockchain platform execution and maintenance are considered the responsibility of the company owner of the blockchain platform. All users and home gateways can execute their transactions through the smart contract. Moreover, there is no need for individuals to have blockchain at home. One blockchain with a proposed smart contract can manage all smart homes in this model. Every home has its own gateways and group members. In addition, all smart homes can connect to the same platform. All communication channels are considered to be not secure in this study. However, the proposed study assumes that the system admin will do the first execution and deployment of the smart contract in a specific condition. The secure channel is created between the communicated parties using smart DHKE, and the exchanged messages are verified with the group signature. The blockchain is only used to implement the group signature and smart DHKE and not to store the exchanged messages. Instead, the blockchain stores the crucial data used for registration and authentication, the encrypted channel created by smart DHKE is used to transmit the messages, and the group signature is used to verify these messages.

3.2. System architecture

The proposed architecture is based on a group signature scheme, which allows multiple IoT devices to securely exchange keys and authenticate each other without requiring a central authority. The system provides a smart key exchange approach to encrypt transmitted data. A brief explanation is given below for all the proposed architecture before defining all phases in detail.

The first step is getting the public key of the home gateway HGW_{puk} from the blockchain. The user implements a related step, which sends public key requests (Get HGW_{puk}) attached to the Home gateway address to the smart contract. HGW_{puk} is stored in the blockchain in the trusted registration table during the registration phase. Next, after the user gets the HGW_{puk}, the user generates a secret key S_{key} by multiplying its private key U_{prk} with HGW_{puk}.

Then the user generates a message *M*containing the control order *Order* (for example, access request, control request) and time stamp (*timestamp*) signed with the private key of the group signature gsk_i. For every user has its own private key for corresponding signature. This message is encrypted with a secret key S_{key} to generate *MSG*. Then, the user sends the encrypted message *MSG* to the Home

gateway attached with the user address $U_{address}$. Then the Home gateway receives the encrypted message *MSG* from the user. Firstly, it extracts the $U_{address}$ that is attached to the message *MSG*, then send the $U_{address}$ to the smart contract with (get U_{puk}) requests asking for that user's U_{puk} that is stored in the blockchain in a trusted registration table. After receiving the U_{puk} from the smart contract, the Home gateway generates its own S_{key} by multiplying its private key HGW_{prk} with U_{puk} Since S_{key} generated is similar according to the DHKE principle. After that, the Home gateway uses S_{key} to decrypt the *MSG*. Then, the Home gateway verifies the group signature G_{sig} and *timestamp*. After verifying the G_{sig}, the Home gateway sends the *Order* to the IoT smart device for execution. The phases consist of Initialization, Registration, Authenticated Key Exchange, and Public Key Update.

3.2.1. Initialization

The system public parameters *PP* are initialized by the group manager in the home gateway, using Initialization Algorithm 1 and Enroll Algorithm 2 to obtain the gsk_i and group public key gs_{puk} Each member *i* is allocated gsk_i by the group manager for signing their transactions. The gs_{puk} is used to verify transactions and stored in the home gateway. Additionally, the group manager invokes public parameter generation to obtain the public parameter (*q, G1, G2, GT, e, P1, P2, H(·)*) as explained in Algorithm 1. This parameter is used by the group manager *GM* to generate *PP*, the group manager's private key GM_{sk}, tracing key *tk*, and group public key gs_{puk} to initialize parameters. That is employed in the integrated encryption scheme using elliptic curves (*ECIES*) [28].

Algorithm 1. Initialization Process

Inputs : ***q, P1, P2;*** initialization parameters by group manager
Output : ***(PP,*** GM_{sk}, ***tk,*** gs_{puk})//system public parameters, group manager's private key, tracing key, group public key
Initialization:
d, s, u = random()
calculate:
G1, G2, G_T = cyclic groups of order *q* //G_T is a bilinear pairing
D = d. P1
S = s. P2
U = u. P1
GM_{sk} = (d, s)//group manager's private key
tk = u//tracing key
gs_{puk} = *(D, S,U)//*the group public key
PP = (q,G1,G2,GT, e, P1, P2,H(·)), //system public parameters

Algorithm 2 is used for enrolment, and it gets the *PP* and GM_{sk} to generate the group member's tag Tag_i and gsk_i.

Algorithm 2. Enroll Process

Inputs: PP, $\mathbf{GM}_{sk}$ *//system public parameters, group manager's private key*
Output: Tag_i, $\boldsymbol{gsk}_i$
Initialize:
x_i= random()
(q,G1,G2,GT, e, P1, P2,H(·)) = PP//*extract parameters from PP*

(*Continued*)

Algorithm 2. Enroll Process

Inputs: PP, $\mathbf{GM_{sk}}$ *//system public parameters, group manager's private key*
Output: Tag_i, $\boldsymbol{gsk_i}$
Initialize:
x_i= random()
(q,G1,G2,GT, e, P1, P2,H(·)) = PP//*extract parameters from PP*
(d,s) = GM_{sk} //extract parameters from GM_{sk}
Calculate:
$Z_i = (d - xi)(s.\ xi)^{-1}.\ P1$
$Tag_i = H(x_i \cdot Z_i)$ //group user's tag
$gsk_i = (x_i, Z_i)$ // group user's private key

3.2.2. Registration

The registration phase includes the registration of the group members and the home gateway. The initialisation phase that generates the group signature private key gsk_i for every user device. Every device generates a new Elliptic curve (EC) public/private key pairs. U_{prk} must be securely stored, and U_{puk} is stored in the registration trusted list mapped with its *Ethereum Address* in the blockchain network. The group member asks for a registration *token* from the group manager. The group user utilizes its blockchain address, user device ID D_{id}, Group ID *Gid* set by the group manager, using U_{puk} which generates using EC. The group manager use his signature to sign the registration token, as shown in Equation 1.

$$RGToken = Sign\left(U_{address}|D_{id}|Gid|U_{puk}, GM_{sk}\right) \tag{1}$$

The group member gets the registration token *RGToken* and gsk_i from the group manager. Then, the corresponding group user sends their information ($U_{address}$, D_{id}, *Gid*, U_{puk}) with the registration token *RGToken* to the smart contract for registration as an *RG_req*, as shown in Equation 2. The smart contract checks the information, verifies the registration token *RGToken* to approve the registration request, and maps $U_{address}$ with the U_{puk} and *Gid*.

$$RG\_Req = U_{address}, D_{id}, Gid, U_{puk}, RGToken \tag{2}$$

3.2.3. Authentication and smart key exchange

This section describes an authenticated key exchange communication process between a user device, home gateway, smart contract, and IoT devices. The process starts when the user sends a request to the smart contract to get the public key HGW_{puk}. Then, the smart contract maps the request with a trusted registration table and send back HGW_{puk} to the user. The user calculates a S_{key}, as shown in Equation 3. Then user generates a message *M* containing *Order*, *timestamp*, and group signature G_{sig} of the message, as shown in Equation 4. The message is encrypted using AES encryption with S_{key} named *MSG* as explained in Equation 5. After that, sent *MSG* to the home gateway. Next, the Home gateway

receives the *MSG* attached with $U_{address}$ then sends a request to the smart contract with the related address to get U_{puk} and calculates the S_{key}, as shown in Equation 6. After decrypting the message using AES decryption with S_{key}, as shown in Equation 7, the home gateway verifies G_{sig} and *timestamp*, then sends the *Order* to the IoT device for execution. The IoT device sends feedback *f* to the home gateway, which generates a new message *fM* with the *f*, *timestamp*, and Message Authentication Code (*MAC*) of the (*f*|*timestamp*), as shown in Equation 8. The message is AES encrypted with the S_{key}, as shown in Equation 9 and sent back to the corresponding user. The user decrypts the message using AES algorithm, as shown in Equation 10. As a result, the user verifies the *MAC* and *timestamp*, and according to verification results, accepts the *f*.

The proposed architecture consists of 14 steps.

Step 1: The user sends a request to contain the home gateway Ethereum Address to the smart contract, asking for the public key HGW_{puk}. The smart contract will map the address with a trusted registration table to get HGW_{puk} and send it back to the user.

Step 2: After receiving HGW_{puk}, by the user device, S_{key} is calculated as:

$$S_{key} = U_{prk}.HGW_{puk} \tag{3}$$

Step 3: User device generate a massege *MSG* contain *Order* and *timestamp*, signed with group signature as shown in the equation:

$$M = Order|ts|G_{sig}(Order|ts, gsk_i) \tag{4}$$

ts:timestamp

Step 4: The user device encrypt the message *M* using AES encryption with S_{key} to generate *MSG*, then send the *MSG* with $U_{address}$.

$$MSG = AES_{ENC}(M, S_{key}) \tag{5}$$

Step 5: After receiving the *MSG* attached with $U_{address}$ by the Home gateway. The Home gateway sends a request to the smart contract asking for U_{puk}. The smart contract will map the address with the trusted registration table; if there is a matching U_{puk} with the $U_{address}$, The U_{puk} send back to the Home gateway.

Step 6: The Home gateway calculates S_{key} using its HGW_{prk} and U_{puk}.

$$S_{key} = HGW_{prk}.U_{puk} \tag{6}$$

Step 7: Decrypting the encrypted message using secret key as:

$$M = AES_{DEC}(MSG, S_{key}) \tag{7}$$

Step 8: Verify G_{sig} and *timestamp*. If the verification pass, the *Order* can be accepted.

Step 9: *Order* send to the IoT device for execution. The IoT device receives the *Order* and executes it.

Step 10: After executing the *Order* the IoT device send its new status as a feedback *f* to the home gateway.

Step 11: The Home gateway receives *f* and generates a new *fM* to the user containing *f*, *timestamp*, and the MAC of the (*f* | *timestamp)*.

$$fM = f|timestamp|MAC\left(f|timestamp, HGW_{prk}\right) \tag{8}$$

Step 12: Encrypt *fM* using AES encryption with S_{key} and send it back to the corresponding user device.

$$MSG = AES_{ENC}\left(fM, S_{key}\right) \tag{9}$$

Step 13: The user receives the feedback message *MSG* from the Home gateway. Decrypt the message using AES decryption with S_{key} as:

$$fM = AES_{DEC}\left(MSG, S_{key}\right) \tag{10}$$

Step 14: The user device verify the MAC and the *timestamp*. If verified, the feedback *f* will be accepted.

3.2.4. Public Key Update

All registered members and Home gateways can update their public key periodically after ending of the expiration time of the public key. This process can be implemented securely by sending the new public key signed with the device signature with a time stamp.

$$u = U_{address}, D_{id}, Gid, nU_{puk}, ts, U_{sig} \tag{11}$$

u = update
nU_{puk}*= new user public key*
U_{sig} *= user signature*
ts = timestamp

The smart contract checks the $U_{address}$ and D_{id} in the trusted registration table then verify the signature s to update the public key. The public key of the user devices and home gateways stored in the blockchain must be updated periodically for security reasons. In case the private key of one device is disclosed. The device needs to generate a new key pair, store the private key in the device and update the public key in the blockchain. The security of the communication between user devices and home gateways in a smart contract system is of utmost importance. For this reason, the public keys of these devices, which are stored in the blockchain, must be updated regularly. If the private key of a device is compromised, it is crucial that the device generates a new key pair and securely stores the private key. Additionally, the updated public key must be reflected in the blockchain to ensure the system's integrity. Regular updates

to the public keys help maintain the system's security and prevent potential threats from malicious actors.

3.3. System interaction

Figure 2 shows the timeline of the data transfer with the key exchange mechanism. The detailed timeline data transmitted through the system component is shown in detail. It shows the implementation of the DHKE using a smart contract to encrypt and decrypt the messages. The system consists mainly of a user, smart contract, the home gateway, and IoT device. The user is one of the home members authenticated to control home devices through the user's device. The smart contract is used to manage the registration phase, store the public keys in the blockchain, and respond to public key requests and updates. The home gateway receives user access control requests and sends the control order to the Smart Home device. It is also responsible for decrypting user messages and verifying the group signature of the requests.

The home gateway sends the order to the smart home device for execution, then receives the order feedback on the new status of the home device. Finally, it sent back the feedback to the user's device. The IoT represents the Smart home device ready to execute the user's requests or send the sensor data to the user's device through the home gateway.

The system connects all homes in one decentralised system. However, the proposed architectures consider a single home as a local group of devices and users in addition to their local home gateway. If one of the homes' gateway is down, this does not mean the authentication system is also down. The related

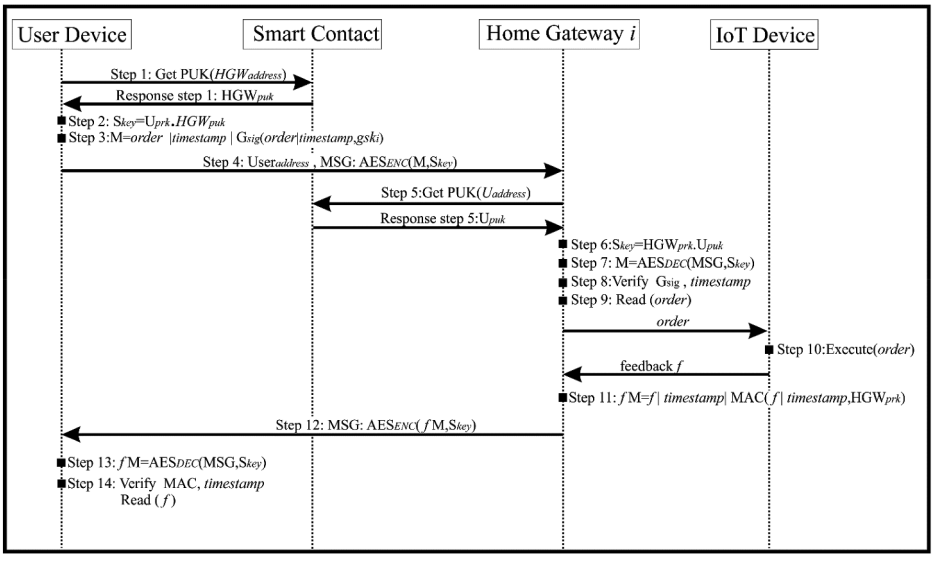

Figure 2. Rgtoken request time in second.

home has been affected for a while, but the entire authentication system will not be affected and will work effectively.

3.4. System limitation

System limitations refer to the inherent limitations of a particular system, whether it be a computer system, software system, or any other type of system. These limitations can include hardware constraints, such as limited memory or processing power, or software constraints, such as limitations in the design or functionality of a particular program or application. Some examples of system limitations include the security of the private keys, scalability and reliability of the smart contract, and system compatibility.

The proposed architecture is a key exchange mechanism using DHKE with a smart contract, which aims to provide secure communication between a user, home gateway, and IoT smart device. However, the model has certain limitations and system limitations that should be considered. One of the limitations is related to the security of the private keys. If the private keys are disclosed or compromised, the security of the communication will be severely impacted. Therefore, it is essential to implement strong key management practices and secure storage of private keys to mitigate this risk. Another limitation is the scalability of the system. As the number of devices and users increases, the amount of data stored in the blockchain and the number of requests sent to the smart contract will also increase. This can lead to slow response times and high resource consumption, impacting the system's performance. Scalability refers to a software system's ability to efficiently handle increasing amounts of data or a growing number of users. If a system has limitations in handling large amounts of data or many users, it can limit scalability. As the amount of data or the number of users increases, the system may become slow, unresponsive, or even crash. This can negatively impact the user experience and result in the loss of valuable data. Therefore, it is vital to design and develop the system with scalability in mind to ensure its scalability, considering potential future increases in data or users.

Additionally, the reliability of the smart contract is a concern. If the smart contract is hacked or fails to execute correctly, the communication between the devices will be impacted. Therefore, it is essential to implement strong security measures and regularly audit the smart contract to ensure its reliability. Finally, the model may have limitations regarding adaptability to changing technology. As new technologies and security threats emerge, the model may need to be updated and improved to remain secure and effective.

As a solution to the proposed architecture, the blockchain model is used. A blockchain refers to a distributed ledger or database that keeps a secure and unalterable record of transactional data. The blockchain relies on a peer-

to-peer network to achieve complete decentralisation. Each node in the network maintains a copy of the ledger to prevent a single point of failure. Any changes to the ledger are updated and validated across all copies simultaneously. The blockchain ledger consists of numerous blocks, each with two sections. The first section contains the transactions or data (that the database needs to store), which could include a range of types, such as financial transactions, health records, system logs, and traffic data. The second section is the header, which contains information about the block, including the timestamp, a hash of its transactions, and the preceding block's hash. Therefore, the blocks are interlinked, forming an ordered chain [29]. The longer the chain becomes, the more challenging it is to manipulate. If malicious users want to modify or replace a transaction within a block, they must first alter all subsequent blocks as they are connected via their respective hashes. Furthermore, they must change the blockchain version stored on each participating node. Blockchains can either be permissioned (private) or permissionless (public) [17].

The first category of blockchain is permissioned (private), which imposes restrictions on consensus contributors. Only trusted actors are authorised to validate transactions; little computation is required to reach a consensus. It is not time or energy-intensive and maintains transactional privacy, as only authorised participants can access them. In contrast, the second type (public blockchain) utilises unlimited anonymous nodes that securely communicate based on cryptography [8]. Each node is represented by a pair of private/public keys, and any actor can read, write, and validate transactions in the blockchain. Consensus is achieved when 51% of the nodes are honest, making the blockchain secure. However, permissionless blockchain networks are typically energy and time-consuming because they require significant computation to strengthen the system's security. Both private and public blockchains can implement this model, but with some differences explained below. Public and private blockchains are two types of blockchain networks with distinct features and use.

3.4.1. Private blockchain

Private blockchains are networks that are restricted to a specific group of participants. Access to the network is controlled by a central authority or a group of authorised participants. They are also more secure, as the network is only accessible to a specific group of participants, reducing the risk of malicious attacks or hacking attempts. They are useful for applications that require a high degree of security, such as financial transactions and supply chain management. Private blockchains are also suitable for use cases where the participants need to be identified such as in healthcare, government and enterprise applications [30].

3.4.2. Public Blockchain

Public blockchains are decentralised networks that are open to anyone. They do not have a single entity that controls them. They are transparent, meaning that the transactions and the ledger are visible to anyone who joins the network. They are also secure, as they use cryptography and consensus mechanisms to ensure that the data is valid and cannot be tampered with. Bitcoin and Ethereum are examples of public blockchains [31]. Public blockchains are suitable for use cases where transparency and decentralisation are essential such as supply chains, digital identity, and voting systems.

In summary, public blockchains are open, decentralised, and transparent, while private blockchains are restricted, centralised, and more secure. The choice between a public or a private blockchain depends on the specific use case and the required level of security and privacy.

4. Results and analysis

In this chapter, we analyse the proposed architecture's result regarding the security property, formal verification, performance evaluation, time complexity, computational complexity, message size, and communication cost. We compare it with other existing related work. We discussed the result and detailed analysis to show the proposed architecture's robustness and reliability in both offline and online modes. This model was implemented using a computer as a blockchain server, raspberry pi 3B+ as a home gateway, and the same computer as a group member and group manager. The Operating systems used are windows 10 and RPI OS 64bit. Solidity is used as a programming language for smart contract programming. Truffle and Ganache for simulation of the Ethereum blockchain. Node JS, Web3, JavaScript, and python for back and front-end programming.

4.1. Experimental setup

In this study, we will analyse the results of our experiment regarding the time consumption of various authentication processes. We will focus on the token request, registration, public key (PK) request, PK update, and the total request-response time. By analysing these results, we aim to shed light on the efficiency and performance of the proposed architecture. Also, the time consumption is discussed according to the experimental result.

4.1.1. Token request

The token request time was measured in our experiment and the results are shown in Figure 2 after 100 iterations. The figure provides valuable insights into the time consumption of the token request process. Additionally, Figure 3 presents the normal distribution of the request time, highlighting the distribution of time consumption for the token request process. From the figure, it can

be seen that the majority of requests take around 6 milliseconds to complete, from making the request to receiving the response. This information helps us understand the performance of the token request process and its consistency over multiple iterations.

The Gaussian distribution, commonly called the normal distribution or bell curve, is a probability distribution that characterises the distribution of various natural phenomena, such as human heights, test scores, and measurement errors [32]. The bell-shaped normal distribution curve is defined by two parameters, the mean (μ) and the standard deviation (σ). The mean represents the central point around which the data is distributed, while the standard deviation measures the spread of the data. A higher standard deviation indicates a more spread-out distribution of the data. The normal distribution is symmetrical around the mean, with 68% of the data falling within one standard deviation of the mean, 95% within two standard deviations, and 99.7% within three standard deviations. The normal distribution is proper in many applications, such as in statistics and probability theory, and it has many essential properties that make it useful in these areas. For example, the normal distribution is used to model random, and measurement errors, test hypotheses about population means, and make predictions about future events based on past data. In addition, a normal distribution is used in many fields, such as finance, economics, engineering, and the natural sciences. For example, in finance, the normal distribution is used to model the distribution of returns on investments; in economics, it is used to model income distribution.

In summary, the normal distribution is a widely used probability distribution that describes the distribution of many naturally occurring phenomena. It is defined by the mean and standard deviation and is represented by a bell-shaped curve. The normal distribution has many important properties and is widely used in many fields to model and predict data. Using statistical analysis for experimental result data is essential because it

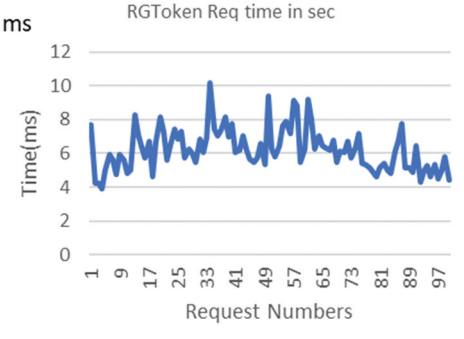

Figure 3. Rgtoken request time in second.

provides valuable insights into the distribution of the timing data. The goal is to determine if the data follows a normal distribution, which is a fundamental assumption in many statistical tests. In this case, the analysis shows that the timing data is predictable and not randomly occurring, with a normal distribution around the average value. As a result, researchers understand and make informed conclusions about the experiment and the data collected.

4.2. Registration

The registration process was also measured in our experiment, with the results shown in Figure 4 after 100 iterations. The registration process involves the creation of a block, which is a time-consuming task.

However, it is only necessary to perform this process once for each device involved. Figure 5 presents the normal distribution of the registration time, showing the distribution of time consumption for the registration process. The average time for completing the registration process and receiving an acknowledgement is approximately 1045 milliseconds. This information is useful for understanding the performance and consistency of the registration process and can be used for optimising the process in the future.

4.3. Public key request

The public key (PK) request time was also measured in our experiment and the results are shown in Figures 6 and 7 These figures provide valuable insights into the time consumption of the PK request process. The average time for completing a PK request and receiving a response is approximately 14 milliseconds. This information helps us understand the performance of the PK request process and can be used for optimising the process in the future.

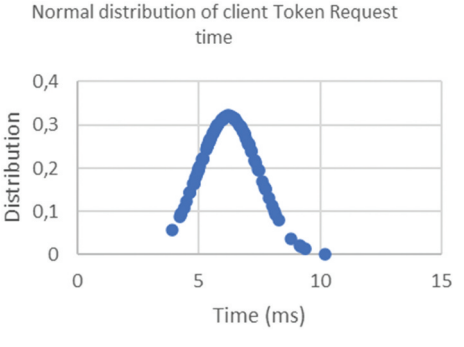

Figure 4. Registration time in millisecond.

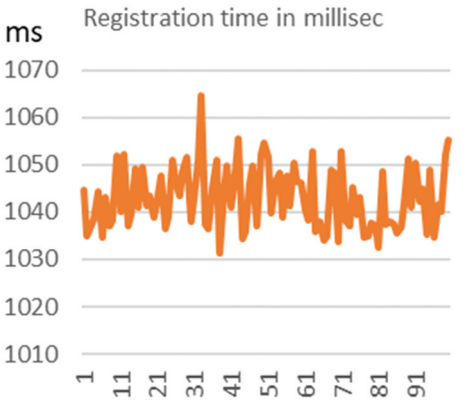

Figure 5. Registration time in millisecond.

4.4. Public key update

The public key (PK) update time was measured in our experiment, with the results shown in Figures 8 and 9 These figures provide valuable insights into the time consumption of the PK update process. The average time for completing a PK update and receiving an acknowledgement is approximately 1035 milliseconds. This information helps us understand the performance of the PK update process and can be used for optimising the process in the future.

4.5. Total request-response time

The total request-response time was also measured in our experiment, with the results shown in Figures 10, 11 , and 12 These figures provide a comprehensive view of the time consumption for the complete request-response process for any control order. The average time for completing the full request-response process is approximately 90 milliseconds. This information helps us understand

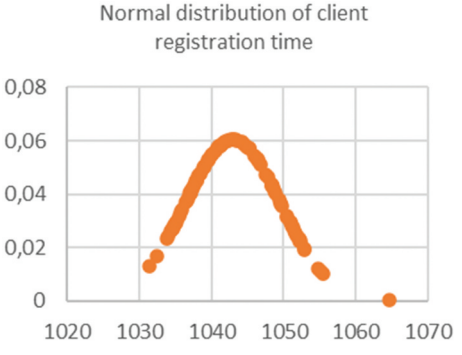

Figure 6. Normal distribution of client registration time.

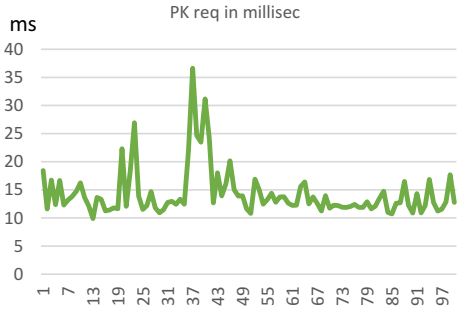

Figure 7. Public key request in time.

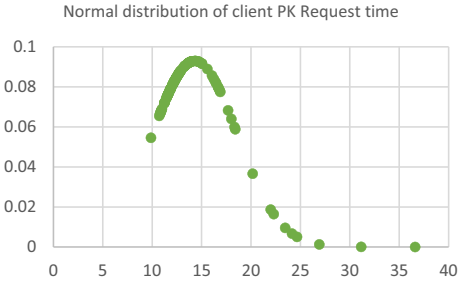

Figure 8. Normal distribution of client PK request time.

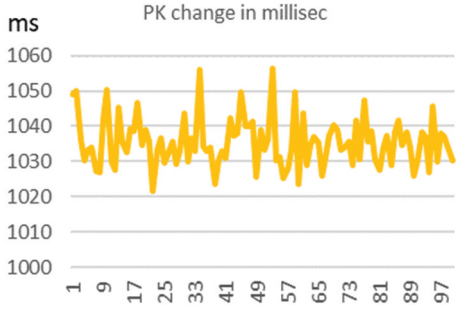

Figure 9. Public key update time.

the overall performance of the system and can be used for optimising the processes in the future.

4.6. Performance comparison

The HomeChain experimental result in Figure 13 me gateway, which is about 500 ms compared to our model, takes about 90 ms on average for key exchange between the user device, and home gateway, and about 100 in average for key exchange between the smart device and home gateway. The total key exchange time of the proposed architecture is about 190 ms on average. Accordingly, our

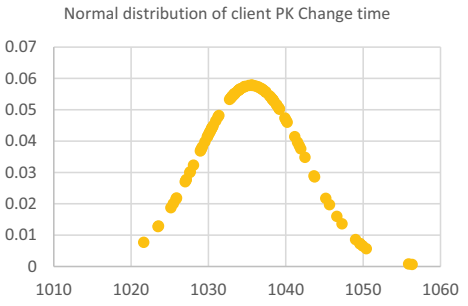

Figure 10. Total request-response time.

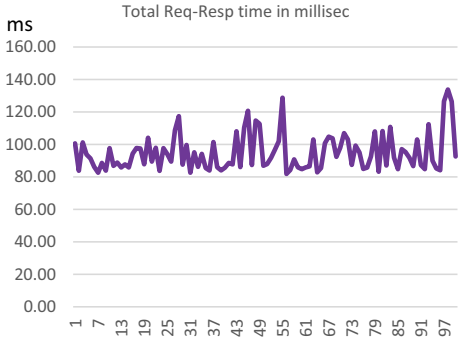

Figure 11. Total request-response time.

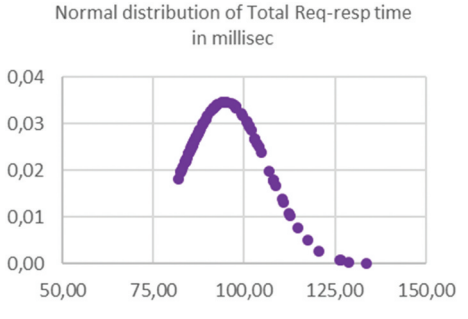

Figure 12. Normal distribution of total request-response time in millisec.

model is faster than the HomeChain model. The reason is that the HomeChain model uses an asynchronous communication approach that needs to create a new block for every transaction. In contrast, any control order with request and response in the HomeChain approach needs four transactions. This process is time-consuming and does not provide any instant notification for any new transaction to the targeted party. Any party must frequently check the blockchain through the smart contract for the new transaction. Our proposed approach replaced asynchronous communication with synchronous communication by creating a secure encrypted channel with the session key. The key

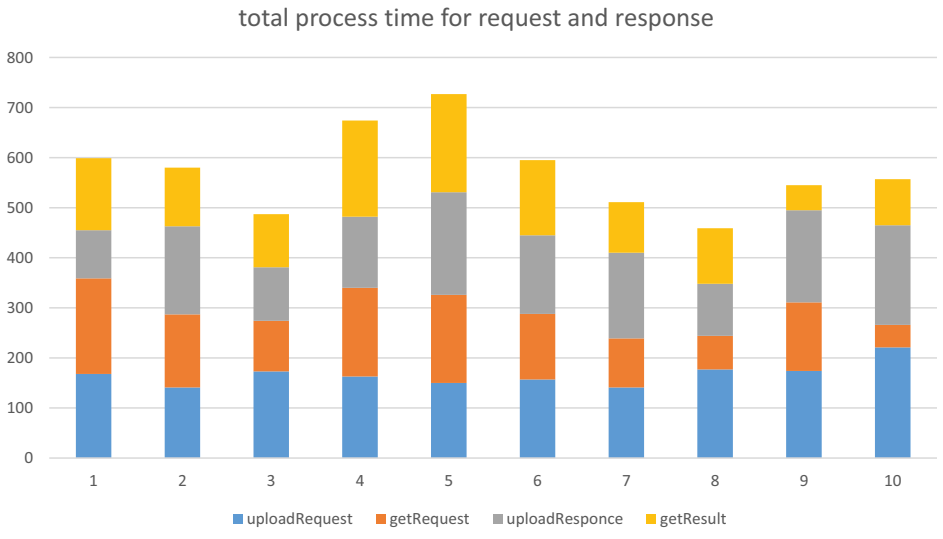

Figure 13. Total process time for request and response.

exchange mechanism is based on the blockchain and DHKE to create a session key. The proposed key exchange mechanism does not need new block creation because it is based on the private key and the registration data stored in the blockchain that contains the public key of each communicated party. Minimizing the number of block creation, and confining it with the registration phase instead of the authentication phase, speeds up the authentication and key exchange process.

This study analysed the algorithm's time complexity using Big O notation, using the same calculation time complexity as HomeChain. The algorithms are divided according to the complexity of the functions used. For example, encryption and Decryption are $O(n^2)$, the one-time hash function, Random number generation, and bitwise operation are $O(n)$, while some small fixed calculation operations are $O(1)$. The algorithm does not use complex operations or outer nested loops with a very high O notation. As a result, all calculations and process operations are lightweight and suitable for devices with limited resources.

The HomeChain experimental result in Figure 8 shows the average time for request and response of any control order between the user and the home gateway. Each order consists of four transactions (uploadRequest, getRequest, uploadResponce, getResult) of about 500 ms. Our model takes an average of 90ms for key exchange between the user device and home gateway. Accordingly, our model is five time faster than the HomeChain model. This is because the HomeChain model uses an asynchronous communication approach that needs to create a new block for each transaction. Any control order with request and response in the HomeChain approach needs four transactions. This process is time-consuming and needs to provide the targeted

party with instant notification of any new transactions. This allows any party to frequently check the blockchain for the new transaction via smart contract. Our proposed approach replaced asynchronous communication with synchronous communication by creating a secure encrypted channel with the session key. The key exchange mechanism is based on blockchain and DHKE to generate a session key. The proposed key exchange mechanism does not need to create new blocks because it relies on the private key and the record data stored in the blockchain, which includes the public key of each party transmitted. Minimizing the number of block generations and limiting it to the registration phase instead of the authentication phase speeds up the authentication and key exchange process.

5. Conclusion

The integration of IoT technology into smart homes is making life more convenient, secure, and energy-efficient for homeowners. However, it also opens up security vulnerabilities and leaves smart homes vulnerable to hacking and exploitation. In order to mitigate these risks, it is important to find a solution that provides secure and transparent authentication for IoT devices. In this study, a novel approach was proposed using a blockchain-based smart key exchange mechanism to authenticate IoT systems in smart homes.

The proposed architecture utilises a group signature scheme and blockchain technology to ensure the legitimacy of each IoT device, and this offers many advantages, such as enhanced security, improved scalability, and increased privacy. The authentication information is stored on the blockchain, which contains a public ledger accessible to everyone. This way, public keys or other registered information in the system can be preserved from hacking, theft, or unauthorised access, considering the users' anonymity. Through experimentation, the effectiveness and efficiency of this approach have been demonstrated, providing improved security, scalability, and privacy. Key contributions of this study include a focus on authentication protocols for IoT devices in the smart home setting, utilising a public ledger for storing sensitive information, and a lightweight solution that is highly performant and has low computation costs, making it ideal for resource-limited devices. In conclusion, the suggested blockchain-based smart key exchange mechanism presents a transparent and secure method of authenticating IoT devices in smart homes.

The unique characteristics of IoT, including its large size and other factors, make it nearly impractical to establish a centralised authentication system that can guarantee a secure identification and authentication of devices while also ensuring data integrity, availability, and resilience against single points of failure. In response to this challenge, decentralised systems based on blockchain technology have been utilised to overcome the limitations of centralised systems. The problem arises when all smart homes are connected to a centralised

server that allows remote users to access their smart homes. A single point of failure could threaten the system and other security problems like DDoS. For this reason, decentralised systems connect smart homes with remote users securely. The blockchain-based authentication system provides remote (online) authentication between remote users and the local edge server of the smart home. Even if the blockchain is time and energy-consuming, it can overcome the problems of the centralised architecture of the authentication systems.

In conclusion, the proposed key exchange mechanism in the HomeChain experiment showed significant improvement in terms of speed and efficiency compared to the HomeChain approach. The average time for request and response was significantly reduced to 90 ms on average, which is faster than the HomeChain model. This is due to the implementation of synchronous communication and the use of blockchain and DHKE to generate a session key. The proposed architecture minimises the number of block generations and limits it to the registration phase, making the authentication and key exchange process faster and more efficient. The results of this experiment demonstrate the potential of using blockchain and DHKE for secure and efficient key exchange in IoT systems. The comparison shows significant results compared with other related works regarding security, reliability, performance, and privacy consideration. The experimental result shows that the proposed architecture is five times faster than the model in HomeChain, in addition to the synchronous communication between the user and home gateways through the secure channel. Moreover, the key exchange management based on a smart key exchange approach creates a secure communication channel between users and the home gateway. All these features in the proposed architecture make IoT smart systems better and more reliable.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Almusaylim ZA, Zaman N. A review on smart home present state and challenges: linked to context-awareness internet of things (IoT). Wirel Network. 2019;25(6):3193–3204. doi:10.1007/s11276-018-1712-5

[2] Kamran M, Khan HU, Nisar W, et al. Blockchain and internet of things: a bibliometric study. Comp Elec Eng. 2020;81:106525. doi: 10.1016/j.compeleceng.2019.106525

[3] Kashihara M, Bhuyan S, Taenaka D, et al. A survey on blockchain, SDN and NFV for the smart-home security. Internet Things. 2022;20:100588. doi: 10.1016/j.iot.2022.100588

[4] Mocrii D, Chen Y, Musilek P. IoT-based smart homes: a review of system architecture, software, communications, privacy and security. Internet Things. 2018;1:81–98. doi: 10.1016/j.iot.2018.08.009

[5] Douha NYR, Bhuyan M, Kashihara S, Fall D, Taenaka Y, and Kadobayashi Y. A survey on blockchain, SDN and NFV for the smart-home security. Internet Things (Netherlands). 2022;20(April):100588. doi: 10.1016/j.iot.2022.100588
[6] Guo Y, Zhang Z, Guo Y. SecFHome: secure remote authentication in fog-enabled smart home environment. Comput Network. 2021;207(September):108818. doi: 10.1016/j.comnet.2022.108818
[7] Lin C, He D, Kumar N, et al. HomeChain: a blockchain-based secure mutual authentication system for smart homes. IEEE Internet Things J. 2020;7(2):818–829. doi: 10.1109/JIOT.2019.2944400
[8] Khan MA, Salah K. IoT security: review, blockchain solutions, and open challenges. Futur Gener Comput Syst. 2018;82:395–411. doi: 10.1016/j.future.2017.11.022
[9] Zhang Z, Zhou S. A decentralized strongly secure attribute-based encryption and authentication scheme for distributed internet of Mobile things ☆. Comput Network. 2021;201(July):108553. doi: 10.1016/j.comnet.2021.108553
[10] Cvitic I, Perakovic D, Gupta BB, et al. Boosting-based DDoS detection in internet of things systems. IEEE Internet Things J. 2022;9(3):2109–2123. doi: 10.1109/JIOT.2021.3090909
[11] Lee Y, Rathore S, Park JH, et al. A blockchain based smart home gateway architecture for preventing data forgery. Human-Centric Comput Inf Sci. 2020;10(1). doi: 10.1186/s13673-020-0214-5
[12] Ammi M, Alarabi S, Benkhelifa E. Customized blockchain-based architecture for secure smart home for lightweight IoT. Inf Process Manag. 2021;58(3):102482. doi: 10.1016/j.ipm.2020.102482
[13] Pirayesh J, Giaretta A, Conti M, et al. A PLS-HECC-based device authentication and key agreement scheme for smart home networks. Comput Network. 2022;216 (June):109077. doi: 10.1016/j.comnet.2022.109077
[14] Roychoudhury P, Roychoudhury B, Saikia DK. Provably secure group authentication and key agreement for machine type communication using Chebyshev's polynomial. Comput Commun. 2018;127(February):146–157. doi: 10.1016/j.comcom.2018.06.005
[15] Homes SS, Iqbal W, Abbas H, et al. ALAM: Anonymous Lightweight Authentication Mechanism for SDN-Enabled Smart Homes. IEEE Int Things J. 2021;8(12):9622–9633. doi: 10.1109/JIOT.2020.3024058
[16] Rashid A, Masood A, Khan R. Zone of trust: blockchain assisted IoT authentication to support cross-communication between bubbles of trusted IoTs. Cluster Comput. 2022;6(1):237–254. doi: 10.1007/s10586-022-03583-6
[17] Hammi MT, Hammi B, Bellot P, et al. Bubbles of trust: a decentralized blockchain-based authentication system for IoT. Comput Secur. 2018;78:126–142. doi: 10.1016/j.cose.2018.06.004
[18] Bang AO, Rao UP, Visconti A, et al. An IoT inventory before deployment: a survey on IoT protocols, communication technologies, vulnerabilities, attacks, and future research directions. Comput Secur. 2022;123:102914. doi: 10.1016/j.cose.2022.102914
[19] AbuAlghanam O, Qatawneh M, Almobaideen W, et al. A new hierarchical architecture and protocol for key distribution in the context of IoT-based smart cities. J Inf Secur Appl. 2022;67(May):103173. doi: 10.1016/j.jisa.2022.103173
[20] Dang TLN, Nguyen MS. An approach to data privacy in smart home using blockchain technology. Proc Conf Adv Comput Appl Acomp. 2018;58–64. doi: 10.1109/ACOMP.2018.00017
[21] García-Vázquez F, Guerrero-Osuna HA, Ornelas-Vargas G, et al. Design and implementation of the e-switch for a smart home. Sensors. 2021;21(11):1–17. doi: 10.3390/s21113811

[22] Ren Y, Leng Y, Qi J, et al. Multiple cloud storage mechanism based on blockchain in smart homes. Futur Gener Comput Syst. 2021;115:304–313. doi: 10.1016/j.future.2020.09.019

[23] Zhang S, Lee JH. A group signature and authentication scheme for blockchain-based mobile-edge computing. IEEE Internet Things J. 2020;7(5):4557–4565. doi: 10.1109/JIOT.2019.2960027

[24] Ali G, Ahmad N, Cao Y, et al. xDbauth: blockchain based cross domain authentication and authorization framework for internet of things. IEEE Access. 2020;8:58800–58816. doi: 10.1109/ACCESS.2020.2982542

[25] Jiang Y, Ge S, Shen X. AAAS: an anonymous authentication scheme based on group signature in VANETs. IEEE Access. 2020;8:98986–98998. doi: 10.1109/ACCESS.2020.2997840

[26] Zhang C, Xue X, Feng L, et al. Group-signature and group session key combined safety message authentication protocol for VANETs. IEEE Access. 2019;7:178310–178320. doi: 10.1109/ACCESS.2019.2958356

[27] Risteska Stojkoska BL, Trivodaliev KV. A review of internet of things for smart home: challenges and solutions. J Clean Prod. 2017;140:1454–1464. doi: 10.1016/j.jclepro.2016.10.006

[28] Oleshchuk V. Internet of things and privacy preserving technologies. ProcVeh Technol Inf Theory Aerosp Electron Syst Technol Wirel VITAE. 2009;336–340. doi: 10.1109/WIRELESSVITAE.2009.5172470

[29] Zhang R, Xue R, Liu L. Security and privacy on blockchain. ACM Comput Surv. 2019;52(3):1–34. doi: 10.1145/3316481

[30] Yang R, Wakefield R, Lyu S, et al. Public and private blockchain in construction business process and information integration. Autom Constr. 2020;118(February):103276. doi: 10.1016/j.autcon.2020.103276

[31] Esposito C, Ficco M, Gupta BB. Blockchain-based authentication and authorization for smart city applications. Inf Process Manag. 2021;58(2):102468. doi: 10.1016/j.ipm.2020.102468

[32] Stahl S. The evolution of the normal distribution. Math Mag. 2006;79(2):96–113. doi: 10.1080/0025570x.2006.11953386