Improving Smart Home Security: Using Blockchain-Based Situation-Aware Access Control

by

Zhicheng Lin

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2023 by the
Graduate Supervisory Committee:

Stephen S. Yau, Chair
Jaejong Baek
Samira Ghayekhloo

ARIZONA STATE UNIVERSITY

December 2023

ABSTRACT

The evolution of technology, including the proliferation of the Internet of Things (IoT), advanced sensors, intelligent systems, and more, has paved the way for the establishment of smart homes. These homes bring a new era of automation with interconnected devices, offering increased services. However, they also introduce data security and device management challenges. Current smart home technologies are susceptible to security violations, leaving users vulnerable to data compromise, privacy invasions, and physical risks. These systems often fall short in implementing stringent data security safeguards, and the user control process is complex.

In this thesis, an approach is presented to improve smart home security by integrating private blockchain technology with situational awareness access control. Using blockchain technology ensures transparency and immutability in data transactions. Transparency from the blockchain enables meticulous tracking of data access, modifications, and policy changes. The immutability of blockchain is utilized to strengthen the integrity of data, deterring, and preventing unauthorized alterations. While the designed solution leverages these specific blockchain features, it consciously does not employ blockchain's decentralization due to the limited computational resources of IoT devices and the focused requirement for centralized management within a smart home context. Additionally, situational awareness facilitates the dynamic adaptation of access policies.

The strategies in this thesis excel beyond existing solutions, providing fine-grained access control, reliable transaction data storage, data ownership, audibility, transparency, access policy, and immutability. This approach is thoroughly evaluated against existing smart home security improvement solutions.

DEDICATION

This work is dedicated to my mother and father. Their support during my studies has been really important. Thank you.

ACKNOWLEDGMENTS

Special thanks to the committee chair and my advisor, Professor Stephen S. Yau, for his invaluable guidance and encouragement throughout my time at Arizona State University, which has been vital to both the realization of this thesis and the broader academic experience. Gratitude is also directed to committee members, Professors Jaejong Baek and Samira Ghayekhloo, for their insightful contributions and dedication, which have been instrumental in shaping this work.

Acknowledgment is due for the financial support provided by the National Science Foundation's CyberCorps: Scholarship for Service. The generous backing throughout the tenure at Arizona State University has been truly significant.

I would like to thank my friend Dustin Cristos for his invaluable assistance in helping with setting up the parameters of the blockchain, ensuring it handles all possible interactions.

TABLE OF CONTENTS

Page

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Algorithms Page

CHAPTER 1

INTRODUCTION

Smart homes have evolved due to the convergence of various technologies, including IoT devices, advanced sensors, data analytics, etc. Such homes utilize a network of interconnected devices to offer tailored services, drawing on the rich data these devices gather [1]. To enable specific features, users often input personal details like names, addresses, and contact information. These devices can monitor and document user habits and preferences [2]. Simultaneously, integrated sensors capture a combination of audio-visual content and environmental metrics [3][4]. As the number of integrated devices in the smart home ecosystem increases, so does the volume of data, heightening the risk of potential misuse.

In the absence of stringent regulations and transparent practices, corporations might capitalize on selling this data to external entities, potentially resulting in unsolicited advertising [5]. Because a considerable amount of data is stored in a database system it becomes a target for attackers. A data breach could potentially expose a large amount of information, creating a possible risk [6]. But it does not end there - unauthorized access to devices could lead to the extraction of sensitive data [7][8][9]. And attackers do not just take data - they go even further. They could interfere with the operation of devices, which might allow them access to residences. When significant amounts of data from devices are exposed to those that contain information like smart hubs, security cameras, or smart locks it becomes possible to create profiles of residents. This poses risks to individuals and their privacy. Moreover, when multiple occupants share a dwelling, managing device access permissions for homeowners, residents, service providers and guests becomes a complex task. Since most data is stored in a centralized database system,

it becomes a magnet for potential cyberattacks, this urges the need for improvement in smart home security.

The methods currently employed for remote access to smart home devices often rely on authentication methods that lack robust security and there is a need for greater control over data ownership and access permissions, as well as increased transparency in record maintenance. Some alternative strategies maintain records of data access, modifications, and policy changes in a single database, but this can increase vulnerability. Attackers could potentially access this database and alter records without the knowledge of users [10]. Therefore, securely storing access records, data modifications, and policy changes for all devices in a smart home environment is essential. It not only improves the quality of services provided but also becomes a valuable resource for handling security incidents effectively. An approach to improving smart home security should securely authenticate users, provide fine-grained access control and data ownership, and maintain records for all interactions with a mixture of IoT devices and policy changes. Blockchain offers a valuable solution to these challenges in smart homes by providing transparency for data transactions, immutability for data content, timestamps, nonce values for proof of work, and storage of hash values in a Merkle tree root structure. However, traditional blockchain models, which are typically decentralized and require significant computational resources to achieve consensus, are not ideally suited for smart homes. These models do not inherently support the fine-grained access control necessary for smart environments and are too resource-intensive for the typically less powerful IoT devices.

In this thesis, a focused application of blockchain is proposed to improve smart home security, utilizing only those attributes that contribute to transaction transparency and data integrity, while omitting the decentralized consensus aspect

of the technology. By integrating smart contracts with situation-aware access control (S-AC), the approach provides automated transaction processing and role-specific access management, adapting to the unique requirements of smart homes. Blockchain technology is utilized in this framework to record interactions between users and devices, with each interaction captured as an individual block. This selective use of blockchain preserves the chain's resistance to unauthorized modifications, and when combined with cryptographic techniques—encrypting data with public keys for storage and decrypting with private keys for access—it greatly improves the overall security of data within the smart home ecosystem.

The contribution of this thesis is summarized as follows:

1. First, this thesis delves into the foundational principles of blockchain, smart contracts, and smart homes.
2. It performs a systematic review of the challenges associated with smart homes, blockchain, and existing solutions to solve them and the drawbacks.
3. Presents an approach that combines a lightweight private blockchain with situational awareness to provide fine-grained access control to smart home systems.
4. Employs a smart contract stored on the lightweight private blockchain to monitor all users, manage device interactions, and adjust access policies through a situational awareness access control scheme. All successful user requests are encrypted using public keys and then recorded on the blockchain to improve security for smart home users.
5. Finally, analyze the main results and compare them with existing smart home solutions.

The surge in smart home adoption brings unparalleled convenience, but it also introduces unique security challenges. As these interconnected systems become

commonplace, the threat of unauthorized access could lead to privacy breaches or even personal harm. While smart homes offer varied functionalities, from security cameras to smart refrigerators, each device requires specific access permissions, which may change depending on the user and situation. This study is motivated by the need to address these concerns. The objective is to develop a robust framework that not only counters external threats but also intelligently manages device permissions based on situational factors. By ensuring a safe and user-friendly smart home experience.

To provide an overview this article is organized as follows; In chapter two, the background information on home environments and insights into blockchain technology are discussed. The third chapter focuses on existing research conducted in the field of smart homes addressing solutions for managing security. Moving forward, the fourth chapter explains the architecture of the approach, while chapter five delves into the details of the approach. As for chapter six, the innovations brought about by the research and their benefits specifically within smart home settings. To illustrate this approach further, an example is presented in chapter seven. Chapter eight evaluates the features and analysis. The summary of the research results and prospective areas for further investigation are offered in chapter nine.

CHAPTER 2

BACKGROUND

Introduced in 2008 by Satoshi Nakamoto [11], blockchain is a technology that comprises chained blocks in a distributed ledger. This platform eliminates the need for a third party when performing transactions. Transactions are securely stored in the blockchain as individual blocks. Each block contains transaction data, a hash value linked to the previous block, a timestamp marking the transaction time, a Merkle tree ensuring data integrity [12], and a nonce value for proof of work. See Figure 1 for an illustration.

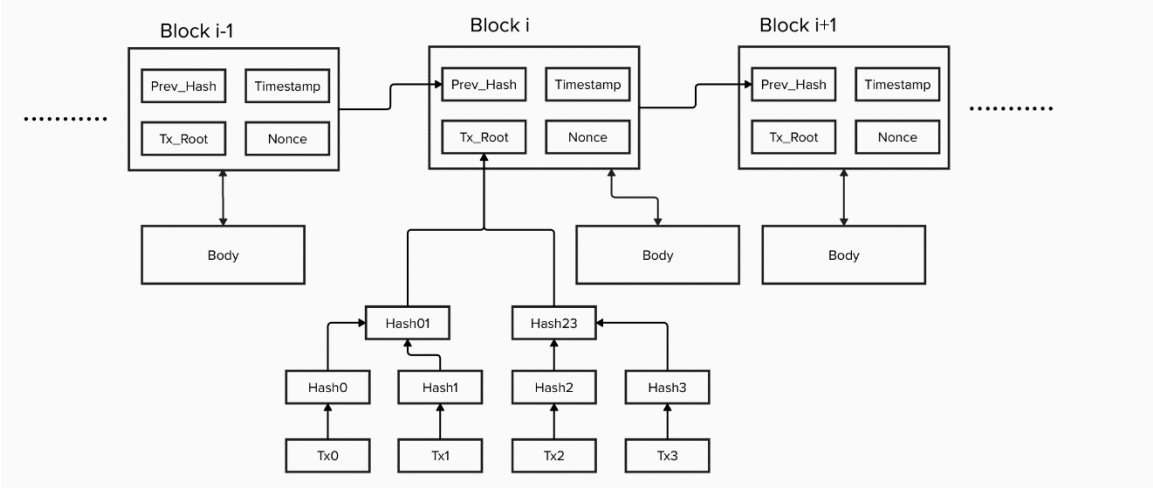

Figure 1: Structure of Blockchain [13]

Blockchain networks can be differentiated based on their permission model, which dictates who has the authority to add new blocks. In permissionless(or public) blockchains, anyone can contribute a new block, exemplified by platforms like Bitcoin. Conversely, permissioned(or private) blockchains restrict block publishing to selected users or entities. Public blockchains are entirely open, allowing anyone to join, publish blocks, and participate without seeking approvals[12]. However, the openness of these networks means they're susceptible to potential malicious actors trying to disrupt the system. To combat this, consensus mechanisms like Proof-of-

Work (PoW) [12] or Proof-of-Stake (PoS) [15] are employed. On the other hand, private blockchains are similar to public blockchain but they are tailored for specific groups, organizations, or individuals. Here, only vetted and authorized users have the privilege to add blocks. Furthermore, read access can be limited to a select group. Given that all participants in a private blockchain are known and trusted, there is reduced need for resource-intensive consensus mechanisms. If any user acts against the network's interests, their permissions can be swiftly withdrawn.

Expanding upon the groundwork of technology smart contracts were developed to simplify, validate, and enforce contract negotiations and executions. Nick Szabo introduced them as a transaction protocol intended to function as a contract, within the marketplace [16]. The core concept of a smart contract is based on logical conditions. If condition one is met, then it will execute a process defined by the condition, if not, then it goes to the next condition and this process keeps going until it meets one. As for blockchain networks, smart contracts can help verify transactions by re-computing the previous hash values and comparing them with the existing ones [12].

The idea behind a home is to create a comfortable and convenient living environment [17]. Within this home setup, there are hybrid devices that generate data, such as personal information, behavioral patterns, and location details. This data, from each device can be uploaded to the home system. Shared with third-party service providers. This enables them to provide services based on the data gathered [18][19][20]. When data is not well protected, it can become susceptible to unauthorized access, breaches, and even disruptions from minor system issues, potentially leading to violations of privacy or exploitation of confidential data. The significance of smart home security goes beyond merely defending against physical break-ins; it extends to the protection of digital information. Regular updates of the

latest security patches are essential for the smart home system to defend against evolving cyber threats. Devices that gather health-related details, like heart rate or sleep patterns, could, in the event of a breach, expose vast amounts of personal data.

Smart homes offer many advantages, from tracking vital signs to learning exercise habits. For instance, if the system identifies poor sleep patterns, an analysis report is promptly sent to the health service provider, facilitating timely consultations and interventions. These devices have the innate ability to adapt based on the user's behaviors, providing dedicated experiences. However, this personalization means the data stored becomes increasingly granular and personal. Such detailed profiles, if accessed by malicious users, could reveal deeply personal aspects of an individual's life and routines. To ensure safety of a smart home, continuous monitoring and timely system improvements are necessary.

CHAPTER 3

CURRENT STATE OF THE ART

3.1 Traditional Challenges

A variety of smart devices, some of which are supplied by third parties, are interacted with by the smart home system. These technologies enable authorized users to remotely access and manage these smart devices. At present, the majority of these systems utilize a straightforward authentication method involving a login process with usernames and passwords. This is then supplemented by the use of cookies for any subsequent user actions. While this approach is relatively user-friendly for the average smart home user, it is not as secure as signature-based authentication. The persistent use of weak or default passwords [21] remains an ongoing issue. A compromised authentication server poses a significant risk as it may expose user login credentials since it stores all such information. The risk is further amplified by the common practice of password reuse, making it easier for attackers to guess authorized usernames and passwords and potentially endangering users with multiple accounts [22].

Multi-factor authentication offers a potential solution to these vulnerabilities, but its usage is not widespread. As of 2023, less than 10% of Google accounts use two-factor authentication, which when activated, reportedly reduces Google account hacks by 50% [23]. However, using a cell phone for multi-factor authentication poses its own problem: if a user loses their phone, they might temporarily lose access to their smart home devices and could potentially be locked out of their house. Furthermore, current authentication systems mostly operate on a binary access model predicated on usernames and passwords. Users are thus either granted full control over home devices or denied access entirely. Such a model fails to offer a nuanced control mechanism allowing for differential access levels to distinct devices

based on user-specific needs. Smart home environments often necessitate varying degrees of access control for individual users and IoT devices. Unfortunately, current methods do not address this requirement effectively and do not maintain user activity logs, a significant facet of data security management [24].

Signature-based authentication offers a higher level of security compared to username and password-based authentication due to the use of lengthy, randomly generated cryptographic keys. These keys are challenging to guess or forge. Nonetheless, not all signature-based schemes are applicable to smart homes due to the high computational costs associated with anonymous signatures [25][26]. Moreover, this authentication method does not inherently offer robust access control management across different users and devices.

3.2 Lightweight Blockchain

Blockchain technology has the potential to offer secure data ownership and immutable records, making it appealing for applications in smart home security systems. One of the significant challenges associated with blockchain is the computing and time complexity that results from the mining processes necessary to reach a consensus among network nodes [12]. This considerable energy requirement can be a deterrent, making blockchain-based smart home security systems unsuitable for some environments. A lightweight blockchain solution is introduced for integrating IoT devices. This solution emphasizes energy-saving design, efficient use of limited resources, and improved security. It also aims to reduce processing and networking overhead for IoT [27], utilize less computing power [28], decrease block validation time [29], and minimize blockchain size specifically for IoT applications [30]. A study identified as [25] presents a case in point, where the researchers propose authentication protocols and develop a new approach to mutual authentication systems tailored for smart home environments. The security of the

smart home network is improved by this mutual authentication method, which enables devices to confirm each other's identities. The strategy is usefulness in real-world applications is, however, constrained by high administrative expenses and significant transaction costs. On the other hand, a different study, denoted as [26], presents an approach that focuses on the preservation of privacy for sensitive data within smart homes. Homomorphic encryption is integrated into the blockchain structure to maintain the privacy of sensitive data, which allows computations to be performed on encrypted data without the need for decryption. Despite these promising developments, the applicability of such techniques depends largely on the specific demands and constraints of individual smart home environments.

Several studies [31][32] have mitigated security concerns with smart devices through threat analysis and data storage within the blockchain, and by protecting data integrity and availability in smart homes by implementing blockchain at the gateway layer. However, these improvements still do not offer effective control over users and devices. The study in [33] tackles centralization issues in IoT within smart homes by deploying hash chain-based access control, and the work in [34] predict users' behaviors in smart homes using five different prediction algorithms based on historical data. Most blockchain-based approaches to solving smart home issues often lack fine-grained access control and management between users and devices [35][29]. Despite these advancements, the mentioned approaches still fall short in providing effective user management, control over devices, complete data ownership, and adaptable policy changes under varying situations. The detailed comparison is shown in table 1.

Table 1: Current Smart Home Solutions

Title	Disadvantages	Advantages
A Lightweight Payment Verification Protocol for Blockchain Transactions on IoT Devices [27]	-	Compatibility with low-end devices and reduced communication and processing overhead.
A Review of Lightweight Blockchain Technology Implementation to the Internet of Things [28]	Limited block validation, it only holds partial blockchain histories.	Reduced computational power and storage space, and improved scalability of IoT data transactions.
Towards an Optimized BlockChain for IoT [29]	Issues with data ownership, fault tolerance and latency under different network conditions.	Hierarchical structure for IoT and Blockchain, distributed trust method, and reduced validation overhead.
Fusion chain: A decentralized lightweight blockchain for IOT security and privacy [30]	Issues with latency, resilience to attacks, scalability, and interoperability for heterogeneous IoT.	Leveraging IPFS to reduce the size of the blockchain, using PBFT consensus for lower power consumption, and ensuring data privacy with PKI.
HomeChain: A Blockchain-Based Secure Mutual Authentication System for Smart Homes [25]	High computation and communication costs.	Authentication method that ensures anonymity, traceability, confidentiality, and resistance to various attacks.
Homomorphic Consortium Blockchain for Smart Home System Sensitive Data Privacy Preserving [26]	-	Processing sensitive data without revealing the original information and using distributed consensus for efficient data access.
A blockchain-based smart home gateway architecture for preventing data forgery [32]	Does not address computational complexity, scalability, latency of IoT devices, or privacy concerns for storing sensitive data.	Countering patch file forgery, zero-day attack, blockchain 51% attack, and DDoS attack.
Blockchain for IoT security and privacy: The case study of a smart home [35]	Does not address how to handle malicious devices in a smart home or the scalability of the framework.	Eliminating the need for POW, maintaining security and privacy, ensuring secure access control, and providing user control and auditability of transactions.
Our Approach	None Found.	Addresses smart home security directly.

Based on the discussions presented in this chapter, it is evident that leveraging parts of blockchain technology, smart contracts, public and private keys, and situation-aware access control, could significantly improve smart home security. This approach ensures comprehensive data ownership and control for users, providing a reliable system for the storage of modification records, access logs, and policy amendment histories. Furthermore, it is designed to assign users different access permissions tailored to various circumstances, effectively creating a robust and versatile system for smart home environments.

CHAPTER 4

ARCHITECTURE OF THE APPROACH

This chapter discusses the specifics of the smart home environments under consideration, elucidating the roles and functions of each entity within these environments. The system architecture, as depicted in Figure 2, involves multiple entities: Homeowner, IoT devices, Visitors, Public and Private Keys, Local Data Storage, and Smart Contracts.

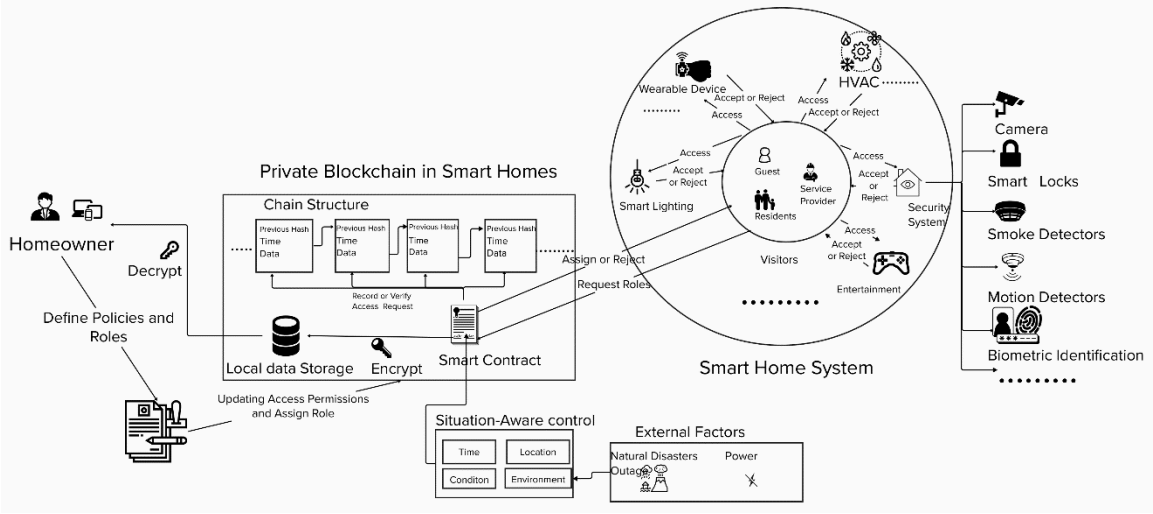

Figure 2: System Architecture of a Smart Home

- The homeowner possesses the highest level of access privilege, allowing them to access device data, define and maintain access control policies, manage various IoT devices, and assign roles to users, including visitors. While most homeowners are not security professionals and might be unaware of the potential vulnerabilities when setting permissions, the default security settings are determined by security experts. Homeowners can then modify these settings based on their needs. It is beneficial to consult with security professionals to prevent potential risks, given their expertise in the field. The homeowner also validates user identities, grants permissions according to their respective roles, and importantly, holds the private key required for

decrypting data, emphasizing their critical role in maintaining the overall security of the smart home system.

- Internet of Things devices play a role, as elements within the smart home ecosystem. These tangible devices, connected to the internet, enable communication among themselves. Collect valuable data to optimize their performance, in meeting residents' requirements. Given their limited resources, these IoT devices only store the most recent transactions, while older data is transferred to local storage. These devices have defined access permissions based on user roles, which are verified through the blockchain network using smart contracts. This verification process assesses the identity and role of users before permitting interaction, thereby contributing to the security and efficiency of the smart home system.
- Visitors within the smart home ecosystem are categorized into three top roles: residents, service providers, and guests, each possessing basic access permissions. The homeowner will assign a more specific role under the three top roles to give more access permissions. Interaction with IoT devices is contingent upon the legitimacy of these access rights, as defined by the visitor's role. Upon successful interaction with a device, a record of the event is stored within the blockchain, thus maintaining a clear and immutable transaction history.
- Public and private keys are used to encrypt and decrypt data stored on local devices before being uploaded to local data storage, as referenced in [36] for key generation. Each device has a public key stored within it, and when an interaction occurs, this key is used to encrypt the data. If there are changes to the key, the homeowner can upload the updated key to the smart home system, and all devices will adjust accordingly.

- Local data storage preserves all interaction data exchanged between users and IoT devices. Given that IoT devices cannot perpetually retain all accumulated data, it becomes necessary to transfer and securely store this information within a local storage facility.

The smart contract is a program stored in the blockchain, capable of execution based on logical conditions. Within the system model, each transaction can trigger the smart contract for tasks like user registration, device registration, revoking user permissions, updating access policies, and recording interactions between users and devices. It can also dynamically adjust permissions based on the circumstances. Every operation performed by the smart contract generates a record, which is then uploaded to the blockchain network as a block. These records, marking user interactions with the smart home system, offer high reliability and transparency. However, the homeowner can only execute actions such as user registration, device registration, revoking user permissions, and updating access policies. In scenarios where a user seeks to utilize a device or access its data, the IoT device communicates this information to the smart contract. The smart contract then verifies from the data recorded on the blockchain network whether the user possesses the necessary permissions for the requested operation. The pseudocode of the algorithms for the smart contracts are shown below.

Algorithm 1: Predefined structure of the blockchain:

```
DEFINE BLOCKCHAIN_STRUCTURE with fields:
  Previous_Hash, TimeStamp, Data
DEFINE ACL with fields:
  device_identifier, Model, Operating_system, Hardware_feature, device_latitude,
device_longitude, sensitive_level, role_permissions, Scheduling, Name, Contact,
user_identifier, role, level, Action, Time_duration
DEFINE DEVICE with fields:
  Model, device_identifier,role_permissions,position
DEFINE USERS with fields:
  Name, user_identifier, role, level, position
CREATE an empty list named Blockchain
```

The initialization phase, invoked by the homeowner for device registration, also defines the device's level of sensitivity and identifies the user roles that can access it.

Algorithm 2: Device Initialization

```
def initialize_devices():
   Initialize more as "1"
   While more is "1":
      Display "please enter the device information"

      Prompt user for device model, operating system, hardware feature, latitude
in the house, longitude in the house, and sensitivity level (0,1,2,3)
      Prompt user for roles that can access the device data (residents, guest,
service_provider or leave blank for owner only) and store in permission_input

      If permission_input is empty:
         Set permission_input to "owner"
      Else:
         Add "owner," to the beginning of permission_input

      Generate a unique identifier for the device
      Set scheduling to "24"
      Create an ACL data with the provided inputs and additional default values
      Get the current time in the specified format
      Create a BLOCKCHAIN_STRUCTURE block with previous hash, current time,
and the ACL data
      Append the block to the Blockchain

      Prompt user to add more devices or exit and store choice in more
      Display the value of more
```

The initialization phase is invoked by the homeowner for user registration. It assigns specific roles to users, determining which devices they can access based on these assigned roles.

Algorithm 3: User Initialization

```
def initialize_users():
    Set more to "1"

    While more equals "1":
        Prompt the user for their name and contact information

        Generate a unique identifier for the user
        Prompt the user for their role (residents, guest, service_provider) and store it in role

        Set level to "0"

        Create ACL data with default values and the provided inputs
        Get the current date and time in the specified format
        Create a BLOCKCHAIN_STRUCTURE block with a previous hash, current time, and the ACL data
        Append the block to the Blockchain
        Prompt the user to add more users or exit and store the choice in more
        Display the value of more
```

Regarding device removal, the system will scan all the blocks to identify which devices are accessible within the smart home. Users can then select which device they wish to remove. Once chosen, that device will become permanently inaccessible after the block information is updated.

Algorithm 4: Rmove Device

```
def remove_devices():
    1. Create an empty list called list_devices.
    2. Iterate over the Blockchain:
       - If a device is found, store its details in list_devices.
    3. Create an empty list called list_devices2.
    4. Identify unique devices from list_devices and store their positions in list_devices2.
    5. Filter out repeated devices from list_devices2 based on the Blockchain.
    6. Create an empty list called valid.
    7. Display devices that are not revoked and store their identifiers in valid.
    8. Ask the user to input a device identifier to revoke.
    9. If the input identifier matches a valid device:
       - Create a new record with "revoked" status.
       - Add the record to the Blockchain.
```

Regarding user removal, the system will scan all the blocks to identify which users remain within the smart home system. The homeowner can then select which user they wish to remove. Once chosen, the "role" attached to the user will be "revoked", ensuring that the user can no longer access any devices.

Algorithm 5: Remove Users

```
def remove_users():
    1. Create an empty list called list_users.
    2. Iterate over the Blockchain:
       - If a user (not an owner) is found, store its details in list_users.
    3. Create an empty list called list_users2.
    4. Identify unique users from list_users and store their positions in list_users2.
    5. Filter out repeated users from list_users2 based on the Blockchain.
    6. Create an empty list called valid.
    7. Display users that are not revoked and store their identifiers in valid.
    8. Ask the user to input a user identifier to revoke.
    9. If the input identifier matches a valid user:
       - Create a new record with "revoked" status.
       - Add the record to the Blockchain.
```

For permission checks, whenever users attempt to access any devices, the system will verify their role and the permissions associated with it, along with the device permissions. If they match, the user can access the device; otherwise, access will be denied.

Algorithm 6: Permission Check

```
def permission_check(user,device):
    Initialize I,check to 0
    While i is less than the length of roles:
        If user's role matches the current role in roles:
            Display "role match"
            Display user's level and device's sensitive level
            If user's level is greater than or equal to device's sensitive level:
                Display "Access granted"
                Create a new data record to log the interaction
                Get the current date and time
                Create a new block with this data and add to the Blockchain
                Set i to a large number to exit the loop
                Set check to 1
            Else:
                Display "level is not enough"
                Call the downgrade function with user and device
        Else If i is at the last role in roles:
            Display "not right role"
        Increment i by 1
```

Regarding updating user information, the homeowner can assign a new role to the user and either elevate or reduce their permissions to access more or fewer devices, respectively. If a user is granted additional roles, they can access some devices beyond their initial permissions, but not all, depending on the role, to better manage access control.

Algorithm 7: Update User Information

```
def upgrade():
    1. Create an empty list named list_users.
    2. Iterate over the Blockchain:
       - If a non-owner user is found, store its details in list_users.

    3. Create another empty list named list_users2.
    4. Identify unique users from list_users and store their positions in list_users2.

    5. Filter out repeated users from list_users2 based on the Blockchain.

    6. Create an empty list named valid.
    7. Display users that are not revoked and store their identifiers in valid.

    8. Ask the user for an identifier of a user they wish to upgrade and store it in
id_input.

    9. Iterate over list_users2:
       - If the input identifier matches a valid user:
          - Ask if the user permissions should be upgraded or downgraded.
          - If "up" and not already at max level:
             - Optionally, assign new roles.
             - Increase the user's level by 1.
             - Log this upgrade in the Blockchain.
          - If "down" and not already at lowest level:
             - Decrease the user's level by 1.
             - Log this downgrade in the Blockchain.
```

Regarding updating device information, the homeowner can update the roles associated with the device and modify the permission level. If a device is initially accessible only by residents, the homeowner can extend its accessibility by adding an additional role. Consequently, users who may not be residents can still access the device if they possess the appropriate permission level.

Algorithm 8: Update IoT Device Information

```
def update_devices():
    1. Create an empty list named list_devices.
    2. Iterate over the Blockchain:
       - If a device is found, store its details in list_devices.
    3. Create another empty list named list_devices2.
    4. Identify unique devices from list_devices and store their positions in
list_devices2.

    5. Filter out repeated devices from list_devices2 based on the Blockchain.
    6. Create an empty list named valid.
    7. Display devices that are not revoked and store their identifiers in valid.
    8. Ask the user for an identifier of a device they wish to update and store it in
id_input.
    9. Iterate over list_devices2:
       - If the input identifier matches a valid device:
          - Ask the user to provide a sensitivity level and store it.
          - Ask the user for roles that can access the device.
          - Update the device's properties with the provided values.
          - Log this update in the Blockchain.
          - Display a message indicating the device was updated.
```

The smart contract manages three entities that are initialized when the system is deployed.

1. IoT Device Table: The smart contract oversees a catalog of device identifiers, data sensitivity levels, and permissions that dictate which users can access the data. Table 2 presents a sample of what this IoT device list may resemble.

Table 2: Example of IoT Device Table

Identifier	Device Model	Operating System	Hardware Feature	Device latitude	Device longitude	Sensitive Level	Role permissions	Scheduling
1	Smart Fridge(RF22N9781SR)	Tizen OS	Wi-Fi, Bluetooth, speakers	24.814657	1.546873	1	owner, residents	24 hours
2	Robot Vacuum(i755020)	iRobot proprietary OS	Camera, Wi-Fi	19.277960	22.992185	0	owner,residents	9:00-20:00

2. User Table: The smart contract supervises a registry of user identifiers and the roles assigned to them. Table 3 provides an illustration of a potential user list. This assignment of roles dictates the access permissions of users, determining the devices they can interact with, the "null" in the Time_duration field means the access time is unlimited.

Table 3: Example of User Table

Name	Contact	Identifier	Role	Level	Action	Time_duration
Tom	812-309-8401, tom123@gmail.com	80d8e7e0-10b0-11ee-a56f-99f00722327b	guest	0	User added	null
John	841-420-2441 John123@gmail.com	c61aba1c-13bd-11ee-be56-0242ac120002	residents	2	User removed	null

3. Most Recent Transactions: The smart contract maintains records of the latest successful user interactions with devices, and all the old ones are stored in local data storage due to the limited resources of IoT devices. A user accessing a smart lock to open the front door is considered one such interaction. The interaction data, encrypted with the public key, can only be decrypted, and viewed using the homeowner's private key.

CHAPTER 5

APPROACH

This chapter discusses the overall steps for the approach. In Figure 2 consists of four integral stages: Initialization, Classify Users, User Registration, and Situation-Aware Access Control. Building upon the private blockchain foundation by Xue et al. [37], the approach in this study expands upon this basis by introducing more finely grained access control. Additionally, it ensures adaptability to accommodate varying user needs. Figure 3 shows the summary of the overall steps in the approach.

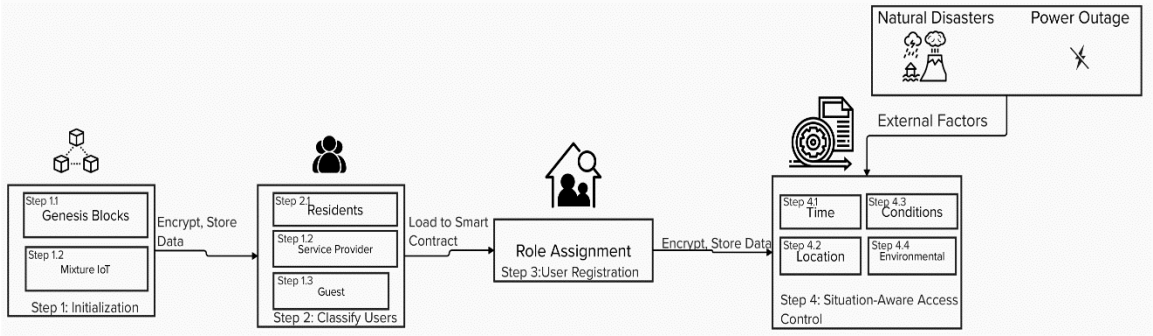

Figure 3: Overall Steps

1. Creating the genesis block, registering a mixture of IoT devices, deploying the smart contract, and encrypting all the data are covered in detail in Step1.
2. Defining general access permissions for residents, service providers, and guests. Discussing more specific roles for individual users. Additionally, exploring how sensitivity levels correspond to access permissions. The details can be found in Step 2.
3. Registering users on the blockchain network, assigning roles associated with access permissions, managing large groups of people, and encrypting all data stored in the devices. The details can be found in Step 3.
4. Regarding Time, devices operate automatically based on set schedules and restrict access outside of designated hours. For Location, users can only

access devices within specified areas. Conditional Access adjusts user permissions. Unauthorized device access leads to reduced permissions, while demonstrated trustworthiness may lead to upgrade permissions. In terms of Environmental considerations, the smart home is equipped with a backup power source to respond during natural disasters or power outages. All interactions between devices and users are encrypted and recorded, with more details available in Step 4 on explaining Time, Location, Conditional, and Environmental.

Step 1: Initialization

The structure of the blockchain setup is depicted in Figure 4, and $n \in N$. A private blockchain is implemented to enable secure storage and interactions among devices. During the initialization phase, the homeowner will be the genesis block by registering the name, contact information, assigning a unique ID, role, and sensitive level; the subsequent blocks will consist of a mixture of IoT devices. Each device has a unique identifier and a device profile that includes characteristics such as the device model, operating system, available hardware features, and the device's physical location (expressed in latitude and longitude). Upon registration to the blockchain network, these devices are considered trusted, thereby excluding all unregistered devices from participating in the smart home environment. The homeowner will deploy the smart contract, as shown in algorithms 1-8, on the blockchain. These algorithms make it easier to add and remove devices and users, update device and user information, and verify permissions associated with users and devices. Once the homeowner block and the device blocks have been added to the blockchain network, the plaintext data contained in the devices will be encrypted using the homeowner's public key. To view the plaintext, one would require the homeowner's private key.

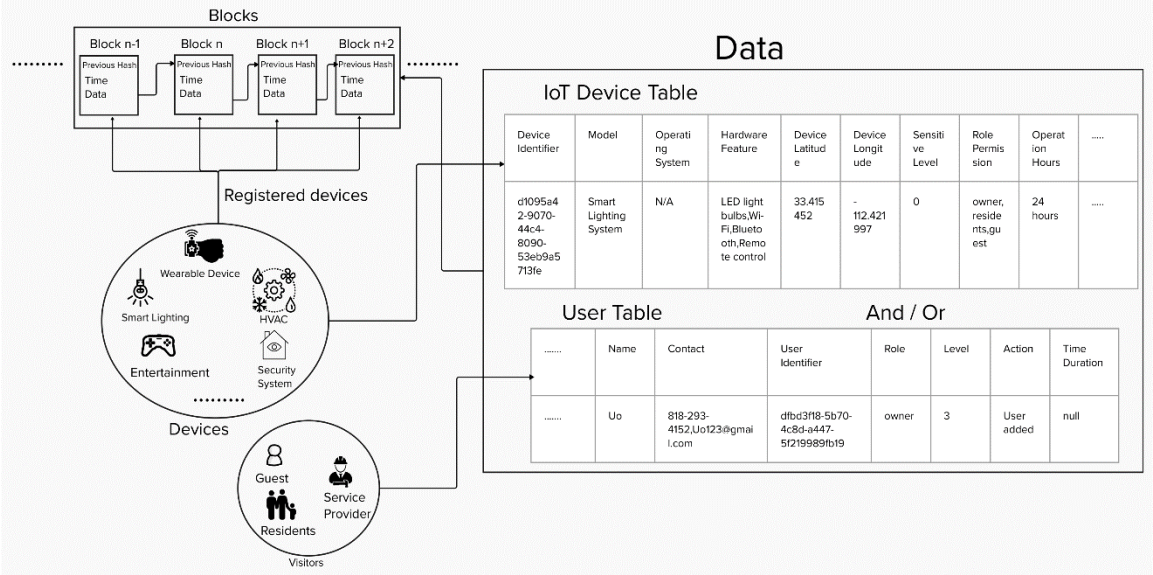

Figure 4: Blockchain Infrastructure

Step 2: Classify Users

During this phase, the homeowner will undertake several tasks including identifying user roles, categorizing device types, defining permissions for each role, implementing rules in the smart contract for automation, and updating these rules to accommodate the dynamic nature of devices, users, and environments. Table 4 illustrates some possible types of users.

Table 4: User Classifications

Residents	Service Providers	Guest
Children	Maintenance & Repair	Frequent Visitors
Adult Residents	Housekeepers	Occasional Guests
Teenager	Gardeners	Event Attendees
Elderly	Homecare Professionals	Overnight Guests
….	….	….

First, the homeowner will identify the user roles that will interact with the smart home system. The three major roles are residents, service providers, and guests. Each of these roles serves as the basis for determining access rights and restrictions, and they can have some subcategories to define more precise access

permissions for users. Below is a breakdown of common permissions for each user role and some subcategories:

- Residents: They can access all the devices defined at sensitivity levels ranging from 0 (non-sensitive) to 1 (low-sensitive) and have limited access to level 3 (medium-sensitive).
    - Adult and Elderly: Live in the home and might have slightly restricted access compared to the homeowner.
    - Teenagers: Restricted access to certain devices or areas, no access to liquor cabinets.
    - Children: Have access to their own rooms' devices and some common areas only. Restrictions could be set on the internet, TV content, etc.
- Service Providers: They have access to specific devices necessary to perform their tasks, as well as a limited set of device data such as diagnostics, maintenance history, or configuration settings. Their default access level is the same as the guest level.
    - Maintenance & Repair: Individuals like plumbers, or HVAC technicians. They have temporary access to specific devices or areas they need to work on.
    - Housekeepers and Gardeners: Regular cleaning staff who can access most areas but no access to personal devices or some private areas.
    - Homecare Professionals: Access to medical devices and certain areas of the house.
- Guests: They have limited access to the smart home system, with control over devices in common areas or guest rooms. They can only access devices defined at sensitivity level 0 (non-sensitive) and have limited access to level 1 (low-sensitive). Their access time is often restricted.

- Frequent Visitors: They are more close friends or family, have more access than a typical guest.
- Event Attendees and Occasional Guests: For parties or other events, only have access to common areas and no bedrooms or private spaces.
- Overnight Guests: Temporary access to guest rooms and related devices.

In the second step, devices and data types are categorized based on their sensitivity and significance. This crucial step helps assign the right permissions to each role, ensuring data security. Below is a breakdown of the data classification based on sensitivity:

- Non-sensitive Data: This refers to data that, if disclosed, would not pose any harm or risk to the individuals involved.
- Low Sensitivity Data: This category includes data that is not overly sensitive but is not intended for public viewing.
- Medium Sensitivity Data: Unauthorized access to this data type could potentially disrupt the household or reveal more detailed behavior patterns.
- High Sensitivity Data: Disclosure of this data could cause substantial harm.

The three roles are the basic default levels, and sub-categories provide narrower levels. Additional categories can be added based on the homeowner's needs. The third part involves defining permissions. Based on the outcomes of the previous two steps, the homeowner can specify permissions associated with each role. This includes determining which devices residents, service providers or guests can control and the type of data they can access.

Once user roles and rules are defined, they are encoded into the blockchain's smart contracts, ensuring the automatic enforcement of access control policies. Lastly, in response to the evolving nature of the smart home environment, the homeowner will update the rules to accommodate new devices, users, or changing circumstances. This allows the smart home system to adapt to meet the homeowner's needs continually.

Step 3: User Registration

In the initial setup of the smart home system, user accounts are created for each individual requiring access. Each account includes the user's name, contact information, and a unique identifier for differentiation. After setting up these accounts, the homeowner may assign additional roles to users that are subcategories of the three primary roles, based on their relationship with the homeowner. Homeowners also designate a sensitivity level for device access to users. If the duration of permissions is marked as "null," it implies that the user's access time is unlimited. Users are restricted to interacting with devices beyond their permissions, promoting a safe and efficient smart home setup. After registering all users, their data is stored in blocks on the blockchain network. Devices encrypt the data using the homeowner's public key to maintain confidentiality. To decrypt the ciphertext, the homeowner's private key is necessary.

In situations where a large group of people is expected for a party or event, the homeowner will establish a general access permission, categorized under roles such as "guest" or "event attendees," as described in Step 2. Upon the arrival of the attendees, the homeowner will assign this general role to them, granting access to common devices within the house. The access duration, as defined by the homeowner in the smart contract, will ensure that the system automatically revokes their roles after the event, in parameter with the specified "Time_duration". All

interactions will be logged, and any necessary adjustments to permissions will be made, as detailed in Step 4.

Step 4: Situation-Aware Access Control

S-AC adopts user access privileges in response to various factors and conditions. Prior studies, such as [38], have focused on automating the creation of situation-awareness agents for service-based systems. Other research efforts have explored capturing temporal and logical relationships among contexts and situations to better analyze and coordinate services [39], providing both developmental and runtime support for context acquisition, situation analysis, and action scheduling [40], as well as establishing and maintaining situation-aware access control for safeguarding private information in service advertisements and requests [41]. Collectively, these advancements improved the smart home user experience. There are four main parts: location, time, conditions, and environment. In the smart home environment, everyone wears a pin that tracks their physical location in the house. Below is a breakdown of each condition:

- Time: Devices can automatically be triggered based on the scheduling. The entertainment systems will be forbidden during school hours or bedtime to ensure children's focus on academics or rest. The following demonstrates that devices can only be triggered during the time of the day.

U: A set of all users in the smart home environment

D:A set of all devices in the smart home environment

T: A set of all possible time intervals during which device can be triggered.

Q: U x D -> {T,F}: This represents if the user has permission to access the device or not.

R: D x T -> {T,F}: This represents the device is accessible during that time interval.

P: U x D x T -> {T,F}: This represents the user has permission to access the device during the specified time interval.

P(u,d,t) = Q(u,d) ∧ R(d,t): This represents, a user "u" has permission to access a device "d" specified by time interval "t" if and only if the user has permission to access the device and it is in the specified time.

∀d ∈ D, ∃t ∈ T| R(d,t) The meaning of this is that every device has at least a one-time interval during which it can be triggered.

∀u ∈ U, ∃d ∈ D | Q(u,d): This represents for every user, there exists at least one device that the user has permission to access.

∀d ∈ D, ∃u ∈ U | Q(u,d): This represents for every device, there exists at least one user who has permission to access that device.

- Location: These rules regulate access within the smart home, either permitting or denying entry to designated areas. The following illustrates that devices can be accessed only in specific areas of the house.

U, D, and Q remain the same.

L1 = Longitude, L2 = Latitude: This represents the set of all possible longitude and latitude coordinates.

R: D x L1 x L2 -> {T,F}: This maps device and location to true or false, represents if device is in the specific location or not for access.

P: U x D x L1 x L2 -> {T,F}: This represents the user has permission to access the device at that specific location.

P(u,d,l1,l2) = Q(u,d) Λ R(d,l1,l2): This represents, a user 'u' has permission to access a device 'd' at a location specified by longitude 'l1' and latitude 'l2' if and only if the user has permission to access the device and it is in the specified location.

$\forall d \in D, \exists l1 \in L1, \exists l2 \in L2 \mid R(d,l1,l2)$: This represents for every device, there exists at least one location where the device can be found.

$\forall u \in U, \exists d \in D \mid Q(u,d)$: This represents for every user, there exists at least one device that the user has permission to access.

$\forall d \in D, \exists u \in U \mid Q(u,d)$: This represents for every device, there exists at least one user who has permission to access that device.

The definition stays the same for the below:

$\forall u \in U, \exists d \in D \mid Q(u,d)$

$\forall d \in D, \exists u \in U \mid Q(u,d)$

- Conditional Access: If users attempt to access something they should not, and the access permission of the device exceeds the user's granted permissions, the device will send this information to the smart contract. The contract will then automatically downgrade the user's sensitivity level permission by one to prevent any further malicious behavior. If the action was accidental, the homeowner can restore the permission to its previous state. Moreover, if the homeowner believes the individual is trustworthy, that user's sensitivity level access permission can be elevated by one. The following demonstrates how elevating and downgrading permissions work.

U and D remain the same.

O: The owner of the house.

T(U) denotes trustworthiness of user U, T(U) = 1, User is trustworthy by O, T(U) = 0 otherwise.

P(U) represents the permission level of U, ranging from 0 (non-sensitive access) to 3 (high sensitive access)

A(U, D) represents the action of U attempting to access device D.

S(D) represents the sensitivity of device D, ranging from 0 (non-sensitive) to 3 (high sensitivity).

If O thinks U is trustworthy, T(U) will be set to 1, and when this happens, the permission level of user U will be incremented by 1, however P(U) cannot exceed 3, as shown below:

$T(U)=1 \rightarrow P(U)=P(U)+1, P(U) \leq 3$

If U attempts to access a device D that they should not (i.e., the permission level P(U) is less than the sensitivity level of the device, S(D)), then the permission level of user U, P(U), is decreased by one. However, P(U) cannot go below 0, as shown below:

$A(U,D) \wedge P(U)<S(D) \rightarrow P(U)=P(U)-1, P(U) \geq 0$

- Environmental-based access: As for external factors, it will be demonstrated with real life scenarios. In the house with multiple residents, they will often want to use the same device at the same time. Thus, access permissions will be altered based on a predefined schedule or on a first-come, first-served basis.

U and D remain the same

E = Event Access Time, which can be a specific time instance when a user tries to access a device.

Assuming all users have access to the devices

t(u,d,e) represents a user "u" accessing a device "d" at a specific event access time "e".

$\forall u,u' \in U, \forall d \in D, \forall e \in E$ if $t(u,d,e) < t(u',d,e)$, u has priority over u'. This represents if user "u" accessed device "d" at an earlier time than user "u'" did, then "u" is given priority over "u'" for accessing the device.

Power outages and natural disasters are also considered by the smart home system. In the scenario of disasters, the essential security features of the smart home need to remain operational for an extended period of time. With crucial devices such as security cameras, alarms, fire safety equipment, and others, the system comes packed with a backup power supply in order to cope with the resulting issues of power outages. The backup power supply relies mostly on batteries. In the event of natural disasters, all the sensors around the house will detect potential threats and alert the homeowner. Hazards such as short circuits can arise due to flooding, and to prevent this, the main water or power supply will shut down. Additionally, alerts will be sent to emergency services to get quick assistance. All the devices will come with integrated safety protocols, for instance, if a smoke detector is activated when no one is at home, it can send alerts to both the homeowner's phone and local fire departments to handle such issues.

In real-world scenarios of situation-aware access control, the system, recognizing it is 7 a.m. from its internal clock, prompts the bedroom's smart blinds to open, allowing sunlight to rouse the residents. Children in the house have their wake-up time set a bit later, at 8 a.m. For location-based access, the homeowner's pin detects his/her location within the house. If the homeowner goes for a morning jog, the system, noting the owner's absence, locks all doors and activates the security system. In the afternoon, when the kids return from school, they each access the snack drawer once. The system then locks it for the day to ensure each child only gets one snack. If the homeowner hosts a book club, friends are assigned the "book_reading" role under the guest category with access from 16:00-18:00.

When guests arrive, they can ring the bell, and the door will automatically grant entry. For nighttime conditional access, the kids' permissions are restricted due to bedtime. If the homeowner returns late and triggers the security alarm, it can be deactivated via their phone. However, this event is logged by the system, which then resets the security status.

CHAPTER 6

INNOVATIONS OF OUR APPROACH

In this chapter I will talk about the advancements in my research. It involves implementing access control and adjusting user permissions based on situations. I'll explain how these changes are successfully integrated into the blockchain network emphasizing their impact on the performance and security of the system.

5.1 Fine-Grained Access Control

The approach in this study demonstrated how to use blockchain to handle situation awareness in smart homes. This is achieved by leveraging roles to designate user accessibility, coupled with sensitivity levels that outline which user roles can interact with devices. These aspects are primarily governed by the Smart Contract, which stores the IoT Device Table and User Table data. This repository of information empowers homeowners with the ability to customize device particulars, update device-associated policies, and amend user access permissions depending on the roles they've been assigned.

During the system initialization phase, when the homeowner assimilates devices and users into the smart home environment, they simultaneously instantiate users' access permissions. This is accomplished by attributing specific roles to each user that align directly with the predetermined access policy. In circumstances where the homeowner wishes to update any device or user access permissions, the process is straightforward. This can be done by invoking the upgrade() and update_devices() functionalities. If the homeowner wants to update the profile information, they should invoke remove_users() and remove_devices() before reintroducing them into the system. This ensures that the modifications are accurately reflected within the smart home setup.

5.2 Adjust Under Various Circumstances

The approach showcases innovative robust adaptability to a diverse array of conditions, including time (devices can be automatically triggered based on the time of day), location (devices activate in response to user presence), conditional access (updating users' access permissions), and environmental factors (responding to external changes), effectively modifying user permissions as needed. Upon the integration of all devices and users into the smart home system by the homeowner, their respective information becomes securely stored within the blockchain network structure. When users want to engage with IoT devices, the device initiates a request to the smart contract. This request serves to verify whether they hold the requisite permissions, as informed by the data residing in the blockchain network. If they possess the necessary authorization, the smart contract will grant access and simultaneously record the interaction within the blockchain network. If users lack sufficient permissions, the smart contract will downgrade the user's permission to prevent any possible malicious interactions. The smart contract will record such actions in the blockchain network, and the homeowner will be notified. If it turns out to be an accident, then the homeowner can restore the permission and temporarily grant permission for a specific duration to that user to access the device if the homeowner believes he or she is trustworthy.

By utilizing the blockchain structure for storing device information, user details, access logs, and modifications to access permissions, all this information is securely safeguarded, thus creating an immutable, transparent, and timestamped record of the actions performed. By merging situation-based access control with blockchain technology, a more secure smart home environment is created, offering users improved control over their devices.

CHAPTER 7

AN ILLUSTRATIVE EXAMPLE

In this chapter, an example is presented to illustrate how to overall approach for improving smart home security by using blockchain and situation awareness access control. I will use multiple residents living environments in smart homes as an example.

Consider a smart home with four users: Uo, Uf, Ug, and Us. These users operate various devices, including a smart lighting system, security camera, smart thermostat, smart lock, window sensors, alarm, and sprinkler system. In this scenario, Uo is the homeowner with sensitive level as "3", responsible for overseeing everything. The smart lighting system adjusts lights based on the time of day and room illumination. The security camera provides live feed access to users. The thermostat adjusts the temperature automatically. The smart lock can be locked and unlocked remotely. The window sensors, alarm, and sprinkler system serve as fundamental components of the home security environment. The approach works well for all kinds of smart devices, from lights and speakers to printers, home hubs, fans, and entertainment systems. All these smart devices require a user-initiated request for access or control. Each device will provide feedback to the user, whether in the form of requested data or device status. The deployment of blockchain and a smart contract determines user access to these devices.

The homeowner, Uo, is initialized and acts as the genesis block in the blockchain, following the Step 1 for Initialization. Uo triggers initialize_devices() to add all the devices. Uo sets the smart lighting system to be accessible by everyone, while the smart thermostat, alarm, window sensors, smart lock, and sprinkler system are made accessible only to the homeowner and residents. The security camera is only accessible to the owner. Service providers can access relevant devices when

providing service. Uo sets the data sensitivity levels: 0 for the smart lighting system, 3 for the security camera and smart lock, and 2 for the smart thermostat, alarm, window sensors, and sprinkler system. Following Step 2, User Classification of the overall approach, Uo assigns permissions to each role, determining access based on data sensitivity levels. Once defined, this information is stored in the blockchain for verification when users request device use or data access.

To register Uf, Ug, and Us in Step 3, User Registration, three requests are sent to add them as users to the blockchain by invoking initialize_users(), after which the homeowner assigns them roles as residents, guests, or service providers alone with sensitive level as "0". Following user and device initialization, Uo can add more users or devices by invoking initialize_devices(), initialize_users(), remove_users(), and remove_devices() as needed. The user table and IoT device table created in this example are shown in Table 5 and Table 6. Uf is registered as a residents, Ug as a guest, and Us as a service provider. Therefore, their device access is based on the roles previously described.

Table 5: Illustrative Example of IoT Device Table

Identifier	Device Model	Operating System	Hardware Feature	Device_latitude	Device_longitude	Sensitive Level	Role_permissions	Scheduling
d1095a42-9070-44c4-8090-53eb9a5713fe	Smart Lighting	N/A	Wi-Fi,Bluetooth	33.415452	-112.421997	0	owner, residents,guest,overnight	24 hours
23a34f21-1650-430c-b94e-84565b96c3b6	Security Camera	N/A	Camera, Wi-Fi,	33.415422	-112.421905	3	owner	24 hours
47bf37c3-1acd-40a3-	Learning Ther	Google	sensors	33.415436	-112.421980	2	owner, residents,john	24 hours

91f6-47d06679ec81	mostat	Nest OS						
7bb0ae42-f301-47e3-99e4-d892ec92adff	Smart Lock	N/A	Bluetooth, Wi-Fi DoorSense sensor	33.415471	-112.421910	3	owner, residents,john	24 hours
43c6f115-a419-4be9-b57c-d26cfcf1e5e8	Alarm System	N/A	Sensors, Wi-Fi, Alarm , remote control	33.415445	-112.421857	3	owner, residents	24 hours
b3c4aef7-acdf-4a90-ab68-265073910e69	Sprinkler System	Embedded OS	Sensors, Wi-Fi, Water sprinklers	33.415422	-112.421883	3	owner, residents	24 hours
da75f501-62e2-467d-82d3-f3705a39830b	Window Sensors	N/A	Sensors, Bluetooth	33.415462	-112.421980	3	owner, residents	24 hours

Table 6: Illustrative Example of User Table

Name	Contact	Identifier	Role	Level	Action	Time_duration
Uo	818-293-4152, Uo123@gmail.com	dfbd3f18-5b70-4c8d-a447	owner	3	User added	null
Uf	841-231-2001 Uf123@gmail.com	6262b2ec-e9d3-4f61-90e9	residents	0	User added	null
Ug	526-243-2061 Ug123@gmail.com	d7cbc3a4-e02d-44b6-baa4	guest	0	User added	null
Us	126-632-2119 Us123@gmail.com	cf4755eb-e18f-4f7f-bd54	service_provider,john	0	User added	null

Once all devices and users are registered, the homeowner can define more specific roles and extend permissions to the users based on specific circumstances as explained in Step 2. Consider a scenario where a guest is invited to stay overnight. Under normal conditions, this guest would not have any permission to access the devices in the room. However, given their overnight stay, the homeowner assigns a role called “overnight” and grants them permission to use the devices in their assigned bedroom, while denying access to devices in other residents' rooms. Similarly, if a service provider is called in to repair devices in the house, the homeowner will assign a specific role with specific access to the door locks and relevant devices necessary for their task. However, if the service provider attempts to access devices unrelated to their job, the smart contract will check if their role matches or not, then downgrade their sensitive level access permission and notify the homeowner.

In this scenario, assume that Uo and Uf are away from home when Uo arranges for a service provider to repair the thermostat system. The service provider, Us, arrives at the house but lacks the necessary permissions to access the lock and enter. The homeowner can remotely assign him a specific role called “john” and elevate the sensitive level access permission in that service provider's profile, which has previously been entered into the blockchain. Then, the homeowner can update the devices to allow “john” to access them, enabling him to unlock the door. This interaction is recorded as a block in the blockchain. If the service provider is not on the blockchain network for any reason, Uo or Uf can remotely remove the old data and enter the new service provider's information into the blockchain. This is done by calling the functions remove_users() and initialize_users(), updating Us's information. In Step 4, Us needs access to the smart thermostat and related devices for repair. Initially, he can access the thermostat, but related devices, defaulted for

homeowners and residents, are off-limits. However, he can request the homeowner to grant additional access permissions by adding a “john” role to the devices. This access is granted only for the duration of the service and will be time limited.

In terms of time, the system first verifies if the service provider's visit falls within the service time frame. If it does, the system moves on to check permissions. However, if it falls outside of the service time frame, access to all devices is denied and the Us permissions are revoked through the function remove_users(). The system checks whether the role matches and if the sensitivity level is equal to or greater than the device's sensitivity level. If the roles do not match, even with sufficient sensitivity level access permission, it is rejected. If the roles do match, but the sensitivity level is lower than the device's level, the service provider’s level is downgraded and recorded in the blockchain network.

In another scenario, both Uo and Uf are at home, and Uo invites Ug for dinner. Prior to Ug's arrival, Uo enters the guest information, and the smart contract will automatically execute the function initialize_users(). This action adds the guest's information to the blockchain network. Upon the guest's arrival at the entrance, the smart contract will check the person's information with that stored in the blockchain to verify identity, subsequently assigning the person a guest role for a specified duration. With this role assigned, the smart contract notifies Uo or Uf of the guest's arrival, enabling them to unlock the door. Within the house, Ug can only access common area devices typically located in shared spaces like the living room, kitchen, and restrooms. These devices can be utilized by all occupants and include appliances like smart lights, smart TVs, smart locks for restrooms, entertainment systems, hairdryers, and more. While Ug can interact with these devices, they cannot access any device not defined in the role, nor can they access devices beyond sensitivity level 0.

After dinner, Uf invites Ug to stay overnight, and the smart contract calls the functions update_devices() to update device information. This is done by adding "overnight" in the device “role_permissionst" and adding “overnight” in Ug’s “role”. All these are recorded as blocks in the blockchain. Now, Ug can access the guest room and the devices in the room, but still cannot access any device with a sensitivity level beyond their current sensitivity level. During the night, if Ug wants to use the hairdryer but Uo also wants to use it, the smart contract grants permission based on a first-come-first-serve basis. If Ug wants to use the entertainment system in the living room during the night, even though Ug has sufficient permissions, the smart contract also considers the time and the event fact. As using the entertainment system late in the evening could potentially disturb other residents due to the noise, the smart contract calls the functions update_devices() for the entertainment system to remove all roles in the "role_permissionst" except for the owner. This modification is also recorded in the blockchain network. Once it is daytime again, the smart contract updates the permissions and reinstates them. On the following day, when Ug leaves, the smart contract calls the function remove_users() to remove Ug's permissions and update_devices() to remove the “overnight” role. This revocation is recorded as described in Step 4.

Let's envision a situation where Ug begins to act maliciously during the stay, aiming to change the sensitivity level of the security camera data to "0", thus allowing universal access. To achieve this, Ug would have to modify all the previous hash values stored in each block of the blockchain network. This task becomes exponentially difficult as the size of the network increases, making the computational cost of such an attack steep. Even if Ug succeeds in changing previous hash values and lowers the sensitivity level to "0", any data they can access is still encrypted. Despite having the required permissions, Ug does not have Uo's private key needed

to decrypt the data. So, they cannot read any information, ensuring the security of the smart home is preserved.

CHAPTER 8

EVALUATION

This chapter takes a detailed look at the practical efficiency of the framework in smart home environments. It examines its robustness, scalability, and adaptability under varying circumstances, along with how the smart contract adjusts to different situations. The framework's capabilities are evaluated through simulations and comparisons with existing solutions, highlighting its ability to manage complex permissions, maintain data transparency, and respect user autonomy. This comprehensive evaluation serves to not only affirm the theoretical significance of the framework but also to ascertain its real-world applicability.

7.1 Prototype Execution

To evaluate the operation of the smart contracts, I created a prototype of the smart home blockchain system on an Ubuntu 21.10 machine with 2 cores and 8GB of RAM using Python. I prepared all the information regarding devices and users to be stored in the blockchain. Following the deployment of the smart contract, I employed a Python script to assess operational complexity across various trials with extensive input. Nevertheless, as block sizes increase, time estimation becomes challenging. The outcomes, derived from the logical steps and the complexity of the procedure, are presented in Table 7.

Table 7: Process Timing for Smart Contract Algorithms

	initialization	initilize_user	revoke_device	revoke_user	permission_check	all_users	all_devices
Process Timing	Low	Low	Low	Low	Medium	Low	Low

Utilizing the same machine, I also measured the time taken for each encryption and decryption process, as well as the time taken for hashing with previous blocks. The encryption process is used prior to IoT devices uploading data

into the blockchain, utilizing a public key to secure the data. As for decryption, whenever users wish to access data and their requests are approved, the homeowner's private key is used to decrypt the ciphertext. Based on the complexity of the process and the logical steps, the results are presented in Table 8 below.

Table 8: Process Timing for Cryptographic Operations

	Encryption	Decryption	Hashing Previous Block
Process Timing	Low	Low	Low

Regarding the overall approach, this being a situation-awareness-based access control, the total time taken for each situation condition is displayed in Table 9 below. Note that the time for location, timing, condition, and event factors are not included for users to request data or get the result sent back to the user. The runtime of less intensive operations has been disregarded, and as all the rules are preconfigured, the setup time is also omitted as it does not affect the system's regular usage.

Table 9: Process Timing for Situation-Aware Analysis

	Location	Time	Condition	Event
Process Timing	Low	Low	Medium	Medium
Process Timing Maximum	Medium	Medium	High	High
Process Timing Minimum	Low	Low	Low	Low

From the conducted simulations, it is evident that the proposed method can manage both small and large smart home environments with several residents. The block parameters, encompassing comprehensive device and user details, ensure adaptability across varied scenarios. As block sizes grow, the time complexity remains relatively moderate. This is primarily because IoT devices offload data to local storage, thereby optimizing storage needs. The design effectively addresses security concerns, with more specifics to be discussed in the subsequent section.

7.2 Comparison of Smart Home Solutions

In this section, I will compare the overall costs of my approach with those of other smart home system solutions. Although my method is more time-consuming compared to the conventional username and password-based authentication, it offers more robust security and control. Due to the chain-like transaction structure of the blockchain, the initialization of devices, role assignment to users, information updates under different conditions, and permission checks will have more computation power. However, each operation only takes a few seconds, resulting in a minimal delay in smart home system transactions.

While signature-based authentication offers a higher level of security than username and password authentication, it is associated with high computational costs and is not universally compatible with all smart home environments. The time and computational expenses inherent to the blockchain structure can be deemed worthwhile considering the control it offers over device management, data protection, transparency between user and IoT device interactions, data immutability, and fine-grained access control under various circumstances.

While my framework is more complex than the traditional username and password or signature-based approach, it ensures heightened security. It offers additional features such as reliable data transaction records and automatic permission adjustment under different conditions. Compared to existing solutions, the significant features of our approach are shown in Table 10. We employ a qualitative representation for clarity, elucidating how our approach fulfills all requisite features while others might fall short. Due to the potential enormity of blocks, accurately measuring cost and energy consumption can be challenging.

Table 10: Comparison of Approach to Smart Home Environments

	[21] [22] Password	[25] HomeChain Blockchain Authentication	[26] Homomorphic Blockchain	[31] [32] Blockchain Gateway	[37] Blockchain Access control	Our Approach
Data Ownership	No	Yes	Yes	Yes	Yes	Yes
Access Policy	No	No	No	No	No	Yes
Adjust Permissions	No	No	No	No	No	Yes
Auditability	No	Yes	Yes	Yes	Yes	Yes
Transparency	No	Yes	Yes	Yes	Yes	Yes
Immutability	No	Yes	Yes	Yes	Yes	Yes

Data ownership empowers users with control over their own data or information. The dynamic access policy and adaptive permissions, in the approach, enable fine-grained control for different users and IoT devices, adjusting their permissions under varying circumstances for added convenience. In contrast, other methods do not provide such flexibility - they neither adjust access permissions in response to different situations nor record modifications of permission changes or interactions between users and smart devices.

One more thing is that my strategy offers the blockchain’s inherent auditability, transparency, and immutability, which is essential for monitoring all transactions between users and IoT devices. This protects the accuracy of data records and helps manage security events as they happen in environments like smart homes. By identifying the affected device, the most recent user, the incident's timing, and the accessed information, homeowners can quickly pinpoint the cause of a problem.

CHAPTER 9

CONCLUSION AND FUTURE WORK

Improving the security of data generated by IoT devices and managing these devices effectively is crucial for improving smart home security. In this thesis, an approach is presented to achieve fine-grained access control within smart home ecosystems, leveraging the integration of private blockchain technology and situational awareness. The use of blockchain fosters transparency and instills a robust, unalterable framework for data exchange. Through the implementation of smart contracts, processes are streamlined, roles are effectively assigned to users, and permissions tied to each role are adapted responsively to circumstances. The data shared between users and IoT devices are encrypted before being uploaded to the blockchain. This methodology offers two layers of defense: one by storing transaction data in a blockchain network, and another by using cryptographic keys to encrypt the data. These measures significantly improves overall security while providing an intelligent, real-time adaptive response to various circumstances.

For future research, there is a vast spectrum of domains to delve into, aiming to refine and optimize the system in several areas. Foremost, handling the increased system complexity is paramount to maintain superior performance while ensuring minimal latency. With the escalating concerns regarding data protection, there is a compelling need to probe deeper into advanced data security methodologies. Techniques such as differential privacy, homomorphic encryption, and zero-knowledge proofs offer promising solutions to improve the blockchain's inherent security features. Moreover, as networks become more complicated, there is potential in analyzing network behavior and architecture to further streamline data flow. On the forefront of technological evolution, Artificial Intelligence (AI) and Machine Learning (ML) stand out. Incorporating these can greatly augment

situational awareness in smart home environments. By harnessing the predictive and analytical capabilities of ML and AI, we can improve the precision of situation detection and access control decisions. Additionally, evaluating the physical security of devices and the infrastructure, and integrating robust AI-driven algorithms can mitigate vulnerabilities. In summary, the amalgamation of advanced encryption methods, sophisticated network analysis, and the prowess of AI and ML will be central to future explorations.

REFERENCES

[1] S. Ivanović, S. Milivojša, T. Erić and M. Vidaković, "Collection and Analysis of System Usage Data in Smart Home Automation Systems," 2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), Berlin, Germany, 2017, pp. 65-66, doi: 10.1109/ICCE-Berlin.2017.8210592.

[2] T. Perumal, Y. L. Chui, M. A. B. Ahmadon and S. Yamaguchi, "IoT based activity recognition among smart home residents," 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE), Nagoya, Japan, 2017, pp. 1-2, doi: 10.1109/GCCE.2017.8229478.

[3] H. Niu, D. Nguyen, K. Yonekawa, M. Kurokawa, S. Wada and K. Yoshihara, "Multi-source Transfer Learning for Human Activity Recognition in Smart Homes," 2020 IEEE International Conference on Smart Computing (SMARTCOMP), Bologna, Italy, 2020, pp. 274-277, doi: 10.1109/SMARTCOMP50058.2020.00063.

[4] M. Rokonuzzaman, M. I. Akash, M. Khatun Mishu, W. -S. Tan, M. A. Hannan and N. Amin, "IoT-based Distribution and Control System for Smart Home Applications," 2022 IEEE 12th Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 2022, pp. 95-98, doi: 10.1109/ISCAIE54458.2022.9794497.

[5] González, V., Sánchez, L., Lanza, J., Santana, J. R., Sotres, P., & García, A. E. (2023). On the use of Blockchain to enable a highly scalable Internet of Things Data Marketplace. Internet of Things, 22, 100722. https://doi.org/10.1016/j.iot.2023.100722

[6] P. Srujana, K. R. Krishna, S. Madhavi, C. Sharma, N. Satheesh and G. Poshamallu, "Internet of Things Based Smart Home Security Analysis System," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICECONF57129.2023.10083624.

[7] J. Srinivas, A. K. Das, N. Kumar and J. J. P. C. Rodrigues, "Cloud Centric Authentication for Wearable Healthcare Monitoring System," in IEEE Transactions on Dependable and Secure Computing, vol. 17, no. 5, pp. 942-956, 1 Sept.-Oct. 2020, doi: 10.1109/TDSC.2018.2828306.

[8] A. Iqbal, J. Olegård, R. Ghimire, S. Jamshir and A. Shalaginov, "Smart Home Forensics: An Exploratory Study on Smart Plug Forensic Analysis," 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 2020, pp. 2283-2290, doi: 10.1109/BigData50022.2020.9378183.

[9] T. Cultice, D. Ionel and H. Thapliyal, "Smart Home Sensor Anomaly Detection Using Convolutional Autoencoder Neural Network," 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), Chennai, India, 2020, pp. 67-70, doi: 10.1109/iSES50453.2020.00026.

[10] Huang, Danny Yuxing, et al. "IoT Inspector." Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 4, no. 2, 2020, pp. 1–21. Crossref, doi:10.1145/3397333.

[11] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Available at https://Bitcoin.org/Bitcoin.pdf.

[12] Yaga, D., Mell, P., Roby, N. and Scarfone, K. (2018), Blockchain Technology Overview, NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.IR.8202

[13] [1] "Eli5: Blockchains explained in simple terms," crypto.bi - Cryptography, network security, programming, Bitcoin, ELI5 cryptocurrencies, cryptocurrency Tools, Tutorials and Guides, https://crypto.bi/eli5-blockchain/.

[14] Types of Blockchain: Public, Private, or Something in Between | Foley & Lardner LLP," www.foley.com. https://www.foley.com/en/insights/publications/2021/08/types-of-blockchain-public-private-between

[15] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen and E. Dutkiewicz, "Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities," in IEEE Access, vol. 7, pp. 85727-85745, 2019, doi: 10.1109/ACCESS.2019.2925010.

[16] N. Szabo, "Smart Contracts: Building Blocks for Digital Markets," EXTROPY: The Journal of Transhumanist Thought,(16), 1996.

[17] D. Vasicek, J. Jalowiczor, L. Sevcik and M. Voznak, "IoT Smart Home Concept," 2018 26th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2018, pp. 1-4, doi: 10.1109/TELFOR.2018.8612078.

[18] P. Liu, T. Lu and N. Gu, "Behavior Analysis and Prediction of Disabled in Smart Home," 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 2021, pp. 1027-1032, doi: 10.1109/CSCWD49262.2021.9437754.

[19] W. Jiang, X. Liu, X. Liu, Y. Wang, S. Lv and F. Ye, "A new behavior-assisted semantic recognition method for smart home," in China Communications, vol. 17, no. 6, pp. 26-36, June 2020, doi: 10.23919/JCC.2020.06.003.

[20] J. Ramesh, A. R. Al-Ali, A. Al Nabulsi, A. Osman and M. Shaaban, "Deep Learning Approach for Smart Home Appliances Monitoring and Classification," 2022 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2022, pp. 1-5, doi: 10.1109/ICCE53296.2022.9730441.

[21] M. Awad, Z. Al-Qudah, S. Idwan and A. H. Jallad, "Password security: Password behavior analysis at a small university," 2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, United Arab Emirates, 2016, pp. 1-4, doi: 10.1109/ICEDSA.2016.7818558.

[22] I. Mannuela, J. Putri, Michael and M. S. Anggreainy, "Level of Password Vulnerability," 2021 1st International Conference on Computer Science and Artificial

Intelligence (ICCSAI), Jakarta, Indonesia, 2021, pp. 351-354, doi: 10.1109/ICCSAI53272.2021.9609778.

[23]"17 essential multi-factor authentication (MFA) statistics [2023]," Zippia, https://www.zippia.com/advice/mfa-statistics/#:~:text=Less%20than%2010%25%20of%20Google,Google%20account%20hacks%20by%2050%25.

[24] Z. N. Mohammad, F. Farha, A. O. M. Abuassba, S. Yang and F. Zhou, "Access control and authorization in smart homes: A survey," in Tsinghua Science and Technology, vol. 26, no. 6, pp. 906-917, Dec. 2021, doi: 10.26599/TST.2021.9010001.

[25] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar and K. -K. R. Choo, "HomeChain: A Blockchain-Based Secure Mutual Authentication System for Smart Homes," in IEEE Internet of Things Journal, vol. 7, no. 2, pp. 818-829, Feb. 2020, doi: 10.1109/JIOT.2019.2944400.

[26] She, Z. -H. Gu, X. -K. Lyu, Q. Liu, Z. Tian and W. Liu, "Homomorphic Consortium Blockchain for Smart Home System Sensitive Data Privacy Preserving," in IEEE Access, vol. 7, pp. 62058-62070, 2019, doi: 10.1109/ACCESS.2019.2916345.

[27] A. Pouraghily and T. Wolf, "A Lightweight Payment Verification Protocol for Blockchain Transactions on IoT Devices," 2019 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 2019, pp. 617-623, doi: 10.1109/ICCNC.2019.8685545.

[28] D. Hanggoro and R. F. Sari, "A Review of Lightweight Blockchain Technology Implementation to the Internet of Things," 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129), Depok, West Java, Indonesia, 2019, pp. 275-280, doi: 10.1109/R10-HTC47129.2019.9042431.

[29] A. Dorri, S. S. Kanhere and R. Jurdak, "Towards an Optimized BlockChain for IoT," 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), Pittsburgh, PA, USA, 2017, pp. 173-178.

[30] D. Na and S. Park, "Fusion chain: A decentralized lightweight blockchain for IOT security and privacy," MDPI, https://www.mdpi.com/2079-9292/10/4/391.

[31] Ratkovic, Nada. "Improving Home Security Using Blockchain." International Journal of Computations, Information and Manufacturing (IJCIM) 2.1 (2022).

[32] Lee, Younghun, et al. "A blockchain-based smart home gateway architecture for preventing data forgery." Human-centric Computing and Information Sciences 10.1 (2020): 1-14.

[33] D. Kim, "A Reverse Sequence Hash Chain-based Access Control for a Smart Home System," 2020 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2020, pp. 1-4, doi: 10.1109/ICCE46568.2020.9043090.

[34] S. E. Wu, J. B. Rendall, M. J. Smith, S. Y. Zhu, J. H. Xu, H. G. Wang, Q. Yang,

and P. Qin, “Survey on prediction algorithms in smart homes,” IEEE Internet of Things Journal, vol. 4, no. 3, pp. 636-644, Jun, 2017

[35] A. Dorri, S. S. Kanhere, R. Jurdak and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 2017, pp. 618-623, doi: 10.1109/PERCOMW.2017.7917634.

[36] A. Lazarov and P. Petrova, "Crypto Genetic Approach in Information Security," 2022 22nd International Symposium on Electrical Apparatus and Technologies (SIELA), Bourgas, Bulgaria, 2022, pp. 1-5, doi: 10.1109/SIELA54794.2022.9845775.

[37] J. Xue, C. Xu and Y. Zhang, "Private Blockchain-Based Secure Access Control for Smart Home Systems," KSII Transactions on Internet and Information Systems, vol. 12, no. 12, pp. 6057-6078, 2018. DOI: 10.3837/tiis.2018.12.024.

[38] S. S. Yau, H. Gong, D. Huang, W. Gao, and L. Zhu, “Automated Agent Synthesis for
Situation Awareness in Service-based Systems,” Proceedings of 30th Annual International Computer Software and Application Conference (COMPSAC '06), September, 2006, pp. 503-510.

[39] S. S. Yau, D. Huang, H. Gong and H. Davulcu, "Situation-Awareness for Adaptable Service Coordination in Service-based Systems," Proceedings of 29th Annual International Computer Software and Application Conference (COMPSAC '05), July, 2005, pp. 107-112.

[40] S. S. Yau, D. Huang, H. Gong and S. Seth, "Development and Runtime Support for Situation-Aware Application Software in Ubiquitous Computing Environments," Proceedings of 28th Annual International Computer Software and Application Conference (COMPSAC '04), September, 2004, pp. 452-457.

[41] S. S. Yau and J. Liu, “A situation-aware access control based privacy-preserving service matchmaking approach for service-oriented architecture,” Proceedings of International Conference on Web Services (ICWS), July, 2007, pp. 1056-1063.