Access Management in Lightweight IoT: A Comprehensive review of ACE-OAuth framework

Master of Science in Technology Thesis
Master's Degree Programme in Information and Communication Technology
Department of Computing, Faculty of Technology
Cyber Security
2023

Author:
Bikesh Shrestha

Supervisors:
Dr. Tahir Mohammad
Prof. Jouni Isoaho

18.12.2023
Turku

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin Originality Check service.

Master of Science in Technology Thesis
Department of Computing, Faculty of Technology
University of Turku

Subject: Cyber Security
Programme: Master's Degree Programme in Information and Communication Technology
Author(s): Bikesh Shrestha
Title: Access Management in Lightweight IoT: A comprehensive review of ACE-OAuth framework
Supervisors: Dr. Tahir Mohammad & Prof. Jouni Isoaho
Number of pages: 74 pages
Date:18.12.2023

Abstract.

With the expansion of Internet of Things (IoT), the need for secure and scalable authentication and authorization mechanism for resource-constrained devices is becoming increasingly important. This thesis reviews the authentication and authorization mechanisms in resource-constrained Internet of Things (IoT) environments. The thesis focuses on the ACE-OAuth framework, which is a lightweight and scalable solution for access management in IoT. Traditional access management protocols are not well-suited for the resource-constrained environment of IoT devices. This makes the lightweight devices vulnerable to cyber-attacks and unauthorized access. This thesis explores the security mechanisms and standards, the protocol flow and comparison of ACE-OAuth profiles. It underlines their potential risks involved with the implementation. The thesis delves into the existing and emerging trends technologies of resource-constrained IoT and identifies limitations and potential threats in existing authentication and authorization methods.

Furthermore, comparative analysis of ACE profiles demonstrated that the DTLS profile enables constrained servers to effectively handle client authentication and authorization. The OSCORE provides enhanced security and non-repudiation due to the Proof-of-Possession (PoP) mechanism, requiring client to prove the possession of cryptographic key to generate the access token.

The key findings in this thesis, including security implications, strengths, and weaknesses for ACE-OAuth profiles are covered in-depth. It shows that the ACE-OAuth framework's strengths lie in its customization capabilities and scalability. This thesis demonstrates the practical applications and benefits of ACE-OAuth framework in diverse IoT deployments through implementation in smart home and factory use cases. Through these discussions, the research advances the application of authentication and authorization mechanisms and provides practical insights into overcoming the challenges in constrained IoT settings.

Key words: Authentication, authorization, resource-constrained IoT, enhancing, mechanisms, parameters, deployment, framework

List of Figures

List of Tables

Acronyms

IETF	Internet Engineering Task Force
DTLS	Datagram Transport Layer Security
PoP	Proof-of-Possession
OSCORE	Object Security for Constrained RESTful Environments
PSK	Pre-Shared Keys
CoAP	Constrained Application Protocol
CBOR	Concise Binary Object Representation
DoS	Denial-of-Service
LPKI	Lightweight Public Key Infrastructure
IBC	Identity-Based Cryptography
RS	Resource Server
AS	Authorization Server
SAML	Security Assertion Markup Language
CA	Certificate Authority
IBC	Identity-based cryptography
OIDC	OpenID Connect
CAC	Context-based Access Control
RBAC	Role Based Access Control
PBAM	Proxy-Based Access Management
TBAC	Token Based Access Control
AEAD	Authenticated Encryption with Associated Data
SDN	Software Defined Networking
LEAIoT	Lightweight Encryption Algorithm towards Low Latency Communication for Internet of Things
MitM	Man-in-the-Middle
FIM	Federated Integration Management
IdP	Identity Provider

Table of Contents

1 Introduction

The expansion of the Internet of Things (IoT) has enabled exchange of information across a range of devices. This collection of devices includes sensors, actuators, wearables, and smart home appliances. This evolution has revolutionized healthcare, transportation, and industrial automation sectors. The integration of IoT devices has generated complexities to security and privacy. Therefore, it is crucial to consider and address the challenges to ensure the reliability of IoT ecosystems [1].

Access management is an essential consideration in IoT systems. It involves controlling and governing interactions between authorized users, devices, and resources within the IoT framework [1]. An effective access management protects sensitive data and defends against malicious attacks.

To address authentication and authorization challenges in constrained environment, ACE-OAuth was designed for resource-constrained devices with limited computational power, memory, and energy. The traditional access management solutions may not be suitable for constrained IoT environment due to limited computational capabilities and resource requirements [2]. This study explores the properties of the ACE-OAuth mechanisms and their implications in constrained IoT ecosystem.

The ACE framework is based on OAuth 2.0 and designed for constrained IoT devices. It defines roles, profiles, and security considerations to facilitate the effective use of OAuth 2.0 in constrained devices [1]. It enables the development of new profiles and extensions to accommodate various IoT use cases. The framework introduces new parameters and encodings for OAuth 2.0 token and introspection endpoints. These features in ACE provide compact and efficient solutions for deployment in resource-constrained environments. The increasing number of IoT devices in different application increases simultaneously the requirement for robust security solution. Existing mechanisms have limitations; thus ACE-OAuth plays a vital role for bridging the gap This requires a comprehensive evaluation of its potential in enhancing IoT security.

This thesis will examine existing access management mechanisms and access their suitability for lightweight IoT devices. It will also investigate lightweight authentication protocols, efficient key management mechanisms, and scalable authorization models, which can enhance access management in resource constrained IoT devices.

1.1 Problem statement

Traditional authorization protocols such as OAuth 2.0 are not suitable for the resource-constrained environment of IoT devices. They require heavy bandwidth and resources, which lightweight devices cannot afford. Similarly, the dynamic and diverse nature of IoT demands flexible and scalable solutions that can handle various IoT device requirements and multiple devices without compromising performance and security. The traditional methods fall short of these issues. This makes them vulnerable to security breaches and unauthorized access [2].

ACE-OAuth is a promising solution to address these challenges by providing a lightweight, secure, and scalable access management framework for IoT deployments. This thesis aims to review the ACE-OAuth framework and its profiles comprehensively. It focuses on analyzing its security implications, comparing it with traditional methods, exploring deployment considerations, and identifying future research directions. The result of this thesis will contribute to the advancement of access management in lightweight IoT, providing valuable insights for researchers and practitioners in IoT security and access management.

1.2 Research questions

This thesis aims to investigate the authentication and authorization mechanisms within the ACE-OAuth framework comprehensively. The main questions that this thesis seeks to answer are as follows:

1. How does the ACE-OAuth framework address the unique security and privacy challenges of resource-constrained IoT devices?
2. How does the ACE-OAuth framework compare to other access management solutions for lightweight IoT devices in terms of security, efficiency, and scalability?
3. What are the key features and functionalities of the ACE-OAuth framework that make it suitable for resource-constrained IoT environments?

1.3 Research objectives

The objectives of this thesis are to investigate authentication and authorization mechanisms in constrained IoT environments. This involves evaluating current methods and analyzing the strengths and weaknesses of the ACE-OAuth framework. The thesis will also identify vulnerabilities and risks in existing authentication and authorization methods to mitigate potential threats.

Furthermore, the addition of new parameters in the ACE-OAuth framework will be investigated for security and scalability. This involves analyzing how these parameters strengthen authentication and defense against breaches and attacks. The benefits of using asymmetric keys and streamlined data encoding for enhanced security layers will also be explored.

In conclusion, this thesis aims to comprehensively investigate authentication and authorization mechanisms in constrained IoT environments within the ACE-OAuth framework.

1.4 Contribution

This thesis reviews existing methods for authenticating and authorizing lightweight IoT devices. This review highlighted the strengths and weaknesses of access management in IoT environments with limited resources.

A comparison of existing and emerging technologies is presented to assess how the advancement of the technology fills the gap or bridges the gap to boost authentication and authorization mechanisms for lightweight IoT.

The ACE-OAuth framework is explored regarding its security implications, protocol flow, and various profiles to evaluate its applicability in resource-constrained IoT environments. For this purpose, different profiles and additional parameters of the ACE-OAuth framework are studied, and a comprehensive review is done to understand better how the ACE-OAuth framework bridges the gap and suppresses the vulnerabilities from traditional methods. This involved an in-depth review regarding protocol flow, security standards and practice, potential risk, and countermeasures of the ACE OAuth framework. As a key finding of this thesis, the security implication of each profile of the ACE-OAuth framework and their strengths and

weaknesses are uncovered. That further clarifies a deeper understanding of ACE-OAuth and its mechanism and features to secure access management for IoT devices.

1.5 Structure of Thesis

Chapter 2: Literature review

This chapter describes existing and emerging trends, as well as the technology of resource-constrained IoT. The chapter discusses the background of lightweight IoT and access management role in securing IoT devices and data. Furthermore, this chapter delves into the challenges in lightweight IoT. It emphasizes on its limited computational capability, low memory, and constrained resources.

Furthermore, this section includes related works in lightweight devices where the effect of integrating existing and emerging technologies to bridge the gaps for constrained environments is described with examples of its implementation. The integration of technologies underscored drastic enhancements in access management for resource-constrained IoT in terms of security, scalability, and efficiency. Additionally, a comparison table illustrates the gaps of traditional technologies and directions of emerging trends for resource-constrained environments.

Chapter 3: Discussion: Authentication and Authorization of ACE-OAuth Framework Profiles

This chapter outlines the methods adopted for collecting data from various articles. The most important articles were obtained from IETF research papers. The selected articles are focused on lightweight IoT, access management in IoT and ACE-OAuth framework-related articles, and existing and emerging technologies in resource-constrained IoT devices.

Furthermore, this chapter highlights the basic protocol flow of the ACE-OAuth Framework, followed by authentication and authorization mechanisms and implementation guidelines. The DLTS profile, OSCORE profiles of ACE-OAuth are mentioned along with their protocol flow, potential risks involved in implementation, and countermeasures. The additional OAuth parameters for the ACE framework are also reviewed with its security standard and practices in this chapter.

Chapter 4: Implementation of Ace-OAuth framework: Smart Home and Factory use case.

This chapter includes demonstrating the ACE-OAuth framework in smart homes and smart factories. The focus of this example is to highlight the effect or boost in access management in lightweight IoT with the ACE-OAuth framework. It shows the security mechanisms involved in authenticating and authorizing access to resources. The case study also focuses on the role of ACE-OAuth in verifying authentic users or sources and ensuring secure communication with encryption and data decryption through secure communication channels.

Chapter 5: Key Findings: Security Implications and strength and weakness of ACEL-OAuth Profiles

This chapter discusses the key findings from articles review and results achieved from discussion section. The chapter focuses on security implications of ACE-OAuth framework, including bidirectional verification, access token protection, securing communication channel, granting credentials, and managing profiles. It also delves into the security implications of additional parameters in authentication and authorization such as the PoP key, CBOR encoding and DTLS profile. Additionally, the chapter examines the security implications of the OSCORE profile. It highlights the OSCORE establishment, confirmation of ownership, OSCORE message transmissions and non-repudiation assurance. Finally, the chapter concludes the summarizing the strengths and weaknesses of ACE-OAuth profiles.

Chapter 6: Conclusion and Future Recommendations

This chapter concludes the summary of a review of authentication and authorization mechanisms in constrained IoT environments focusing on the ACE-OAuth framework. This chapter emphasizes the importance of implementing strong countermeasures to mitigate security risks associated with lightweight IoT devices. It highlights the significance of integrating new parameters and encodings to enhance security, such as proof-of-possessions (PoP) keys, asymmetric keys, and Concise Binary objects (CBOR) for data encoding.

This chapter further advance security practices in lightweight IoT environments, such as reducing the burden on energy-constrained nodes, enhancing hardware security and tamper-resistant design, addressing challenges of irregular connectivity in IoT devices, integrating ACE-OAuth with edge computing devices in multi-tier IoT system, and strengthening privacy aspects in IoT security frameworks.

2 Literature review

The Internet of Things (IoT) is a network of physical devices that interconnect physical entities, devices, and systems over the Internet. These devices range from everyday devices such as sensors, actuators, appliances, vehicles, and wearables to industrial tools. IoT devices gather, process, and exchange data through networking and communication protocols. The growth of IoT increases convenience, efficiency and enhances safety and security that spans healthcare, agriculture, transportation, and urban planning [2].

Access management refers to the systematic governance and interactions between IoT devices, services, and users within the ecosystem. It involves various procedures, mechanisms, and protocols that determine which entities are authorized to access specific IoT resources, such as data, services, and functionalities [2]. Access management acts as a digital sentry, ensuring that only authorized entities can communicate within the IoT network. This control extends to both users and other IoT devices, creating a hierarchical framework that governs permissions and privileges.

2.1 The Role of Access Management in Securing IoT Devices and Data

The intrinsic nature of IoT, characterized by its vast and diverse system of interconnected devices, introduces complex security challenges. As the number of IoT devices surge, the potential attack for unauthorized access also intensifies. Therefore, access management is essential to ensure the security and integrity of IOT ecosystems.

Access management mechanisms enable administrators and users to gain control over the authorized users to interact with IoT devices and services. The authentication and authorization protocols of access management ensure that only authenticated and authorized devices gain access to the IoT network [2].

Furthermore, access management protects the privacy of IoT users and the data by distributing the privilege according to the user's roles and device capabilities. The information can only be accessed through identity verification and access control protocols. The users are restricted to access beyond the predetermined functionalities and responsibilities [2].

Effective access management has become a fundamental pillar that provides a strong foundation for maintaining data confidentiality and mitigating the risks associated with unauthorized access in IoT security. With the expansion of the IoT ecosystem, the integration of access management mechanisms has become an absolute necessity in building a secure and resilient IoT ecosystem.

2.2 Challenges in Lightweight IoT

The surge of Lightweight Internet of Things (IoT) devices within resource-constrained environments has created challenges in access management. These challenges originate from the distinctive characteristics of lightweight IoT devices, such as limited computational power, memory, and energy resources. These exceptional features of resource constrained IoT devices impose difficulties in implementing effective and efficient access control mechanisms. Scalability is another major concern as the number of devices grows simultaneously with the demands for access management [3].

Traditional authentication and authorization methods can be complex for lightweight IoT devices with their limited computational power and low memory. The existing methods rely on cryptographic keys and certificates that need more memory than the lightweight IoT devices. Other for the lightweight IoT devices is low battery power, the heavy nature of current authentication processes is not suitable as it drains the limited battery life [3]. Similarly, scalability issues arise with the growth of IoT devices, leading to potential bottlenecks in authentication and authorization processes.

The traditional methods fall short of adaptability as different devices has different requirements and traditional methods do not provide that flexibility to suit the diverse security requirements of different IoT devices. It is challenging for traditional methods to adapt the frequent changes that occurs in IoT environments with frequent manufacture and departure of IoT devices from different manufacturers [3].

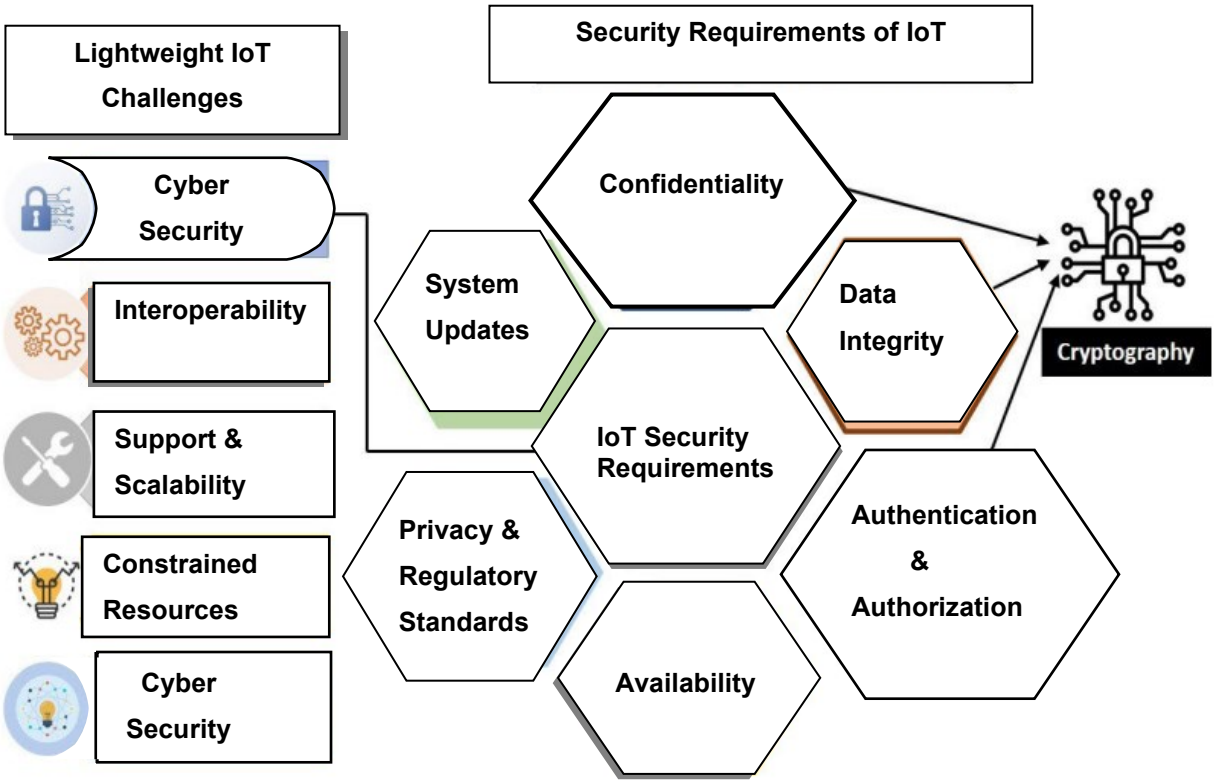

Figure 1. IoT Security Challenges [4]

Figure 1 shows the security concern of resource-constrained IoT devices with its challenges and security requirements. The transfer from server to sensor creates unprecedented challenges such as interoperability, sustainability, privacy, integrity, and confidentiality. Additionally, these devices are exposed to security attacks as they collect sensitive data by interacting with the physical world.

The development of new authentication and authorization methods **is** essential for the resource constrained IoT devices that enables efficient solution with limited computational power, memory, and energy resources. The new technology must be designed to handle and adjust the exponential increase of IoT devices to solve the scalability issues. Furthermore, the need for energy efficient approaches, adaptability to dynamic environment and customizable features for diverse scenarios urge the necessity of designing and implementing new technologies [4].

Additionally, ensuring robust security itself is a challenge, and the additional balance between access control and lightweight cryptographic protocols adds up to an obstacle. Key management for cryptographic operations becomes intricate, where security and efficiency are the main objectives to acquire.

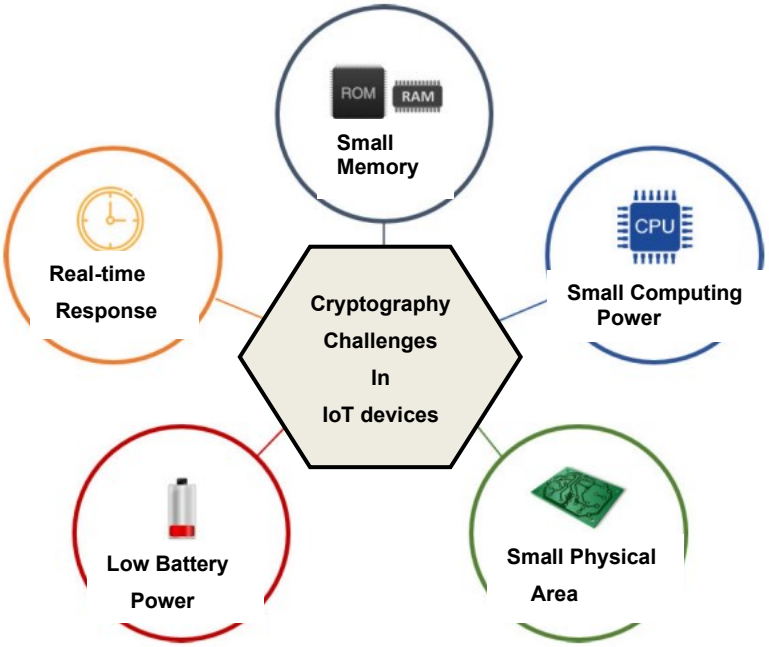

Figure 2. Cryptography challenge for lightweight IoT devices [4]

Figure 2. illustrates a resource constrained IoT device scenario where key challenges are limited memory, reduce computational power, small physical areas, low battery power, and a real-time response to implement conventional cryptography.

Access management systems are crucial for real-time applications and must be integrated with constrained lightweight IoT devices. The heterogeneity of IoT devices and protocols makes access management challenging. Adaptable access management solutions are needed to accommodate the diverse capabilities of IoT devices and protocols. These challenges are being addressed by the lightweight cryptography, role-based access control, efficient key management, and context-based access solutions [4]. These security mechanisms leverage a promising solution for securing and enhancing the functionality of lightweight IoT ecosystems.

2.3 Existing Access Management Solutions in lightweight IoT Environment

The rapid growth of resource-constrained devices requires innovative solutions for access management. This literature review explores diverse strategies used to address access management challenges in resource constrained IoT environments.

2.3.1 Token-based Access Control (TBAC)

TBAC utilizes lightweight cryptographic tokens for authentication and authorization of users and devices. These tokens contain information about identity, permissions, and expiration time. Thus, it allows access to resources without requiring re-entering credentials every time.

The TBAC is designed for lightweight IoT devices due to its efficiency and security. The TBAC token are lightweight and requires minimal memory space. TBAC eliminates IoT devices to store and manage passwords which is suitable for limited resources devices. Hence, it improves security by reducing complexity and increasing flexibility and scalability. By supporting different range of devices and policies like role-based access control and least privilege access [5].

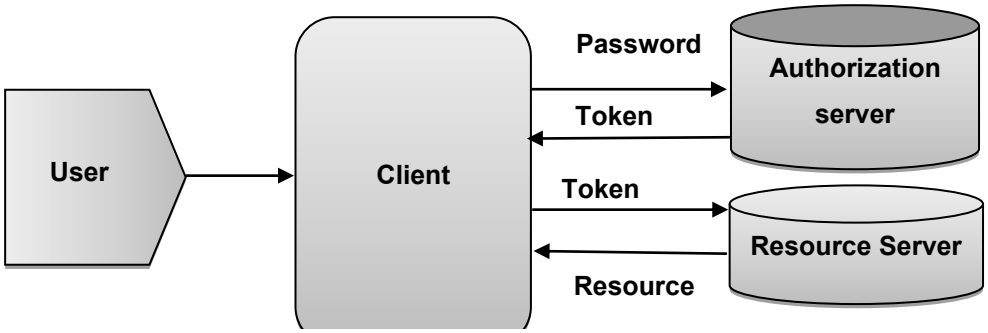

Figure 3. TBAC working mechanism in lightweight IoT [6]

The TBAC working mechanism in lightweight IoT is illustrated in figure 3. In a smart home, there exist various smart devices. For example, a smart bulb registers with TBAC server and the server generates a unique token for the smart bulb, which is stored in a database. After installing a smart light bulb in a home, the bulb connects to smart home system. The light bulb then sends the token to the home system which verifies the token and grants access to the system. After the successful access to the home system, the smart light bulb can receive commands and perform its task [6]. This applies to smart thermostat, wearable fitness tracker and different appliances inside the smart home.

The disadvantage of TBAC is that it relies on single key, which is vulnerable if it is compromised. The token is a complex cryptographic signature algorithm that require strong knowledge and understanding by developers. Its limitations restrict to manage clients from the server side [6].

2.3.2 Lightweight Public Key Infrastructure (LPKI)

LPKI is a security framework that enables a streamlined approach to managing digital certificates for device authentication. It utilizes public key cryptography which needs low computational and storage and is efficient and secure.

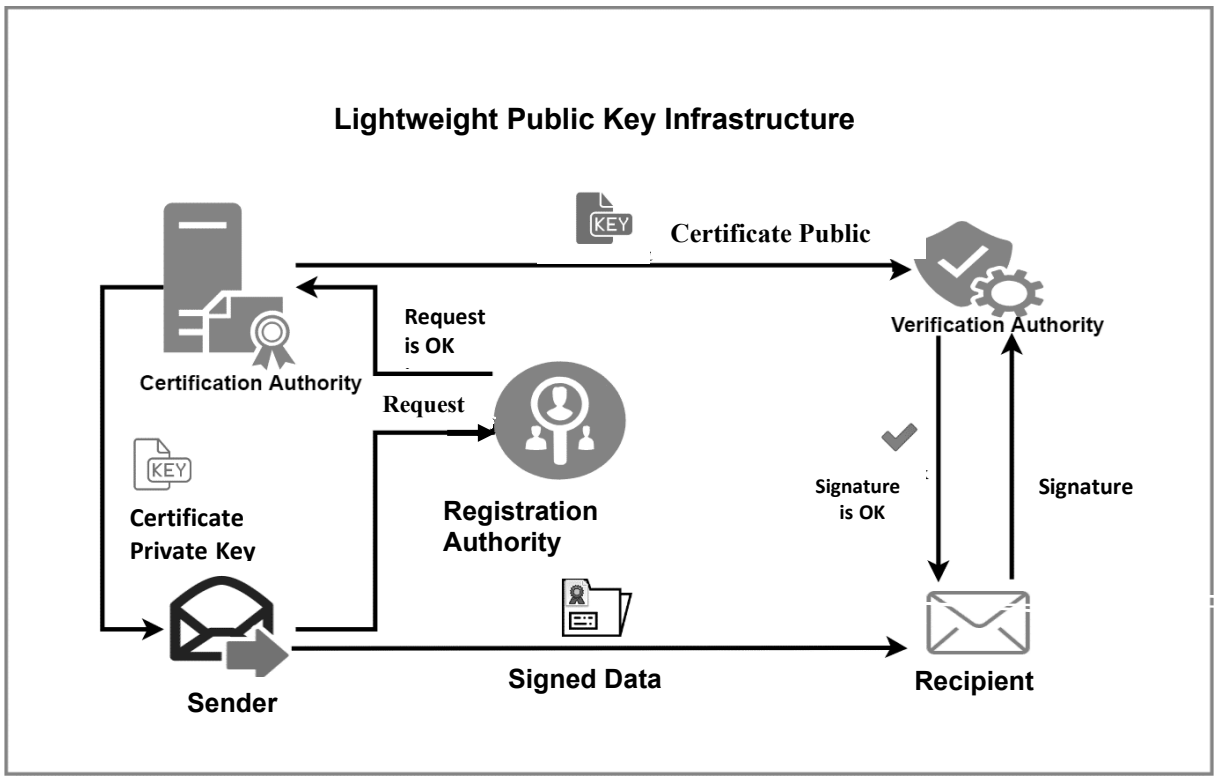

Figure 4. Lightweight Public key infrastructure [7]

LPKI can be used in mobile phones, embedded medical devices to smart home devices. The devices register with the LPKI server, which generates a public and private key pair for the device. The public key is stored in the database securely. The device sends a public key to the server to authenticate to a service. The server then verifies the public key against the stored key in the database. If the key matches, the device is authenticated and authorized to access. LPKI uses a centralized turn anchor, such as root certificate authority (CA), to identify the users and devices, making them suitable for lightweight devices [7].

LPKI ensures the security of health data in smart hospital. The encryption algorithm ensures the secure transfer of the patient data among IoT devices used in healthcare from malicious attacks. The decryption algorithm allows for proper use of data in healthcare applications.

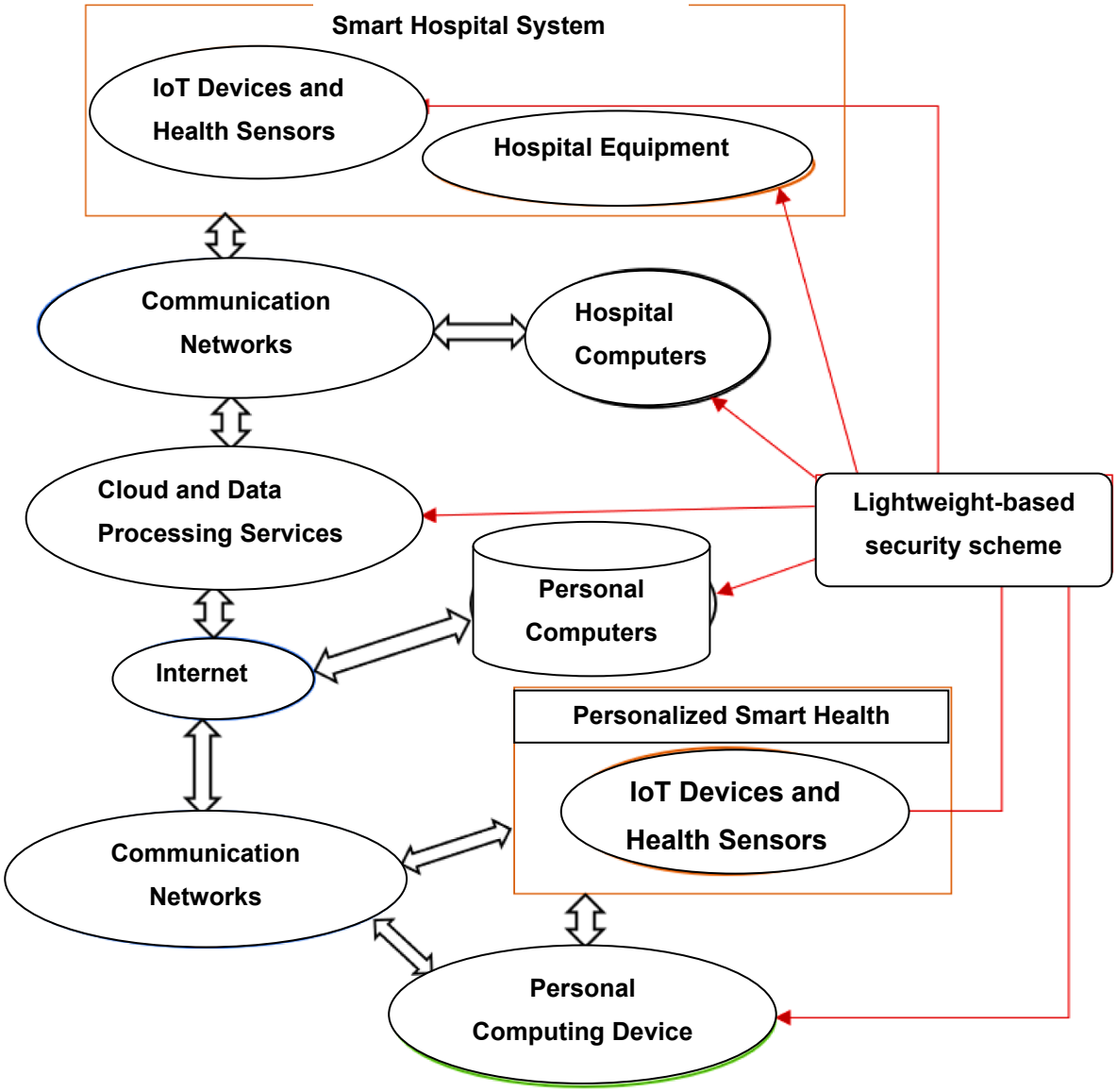

Figure 5. Lightweight-based security scheme for smart hospital system [8]

Figure 5 demonstrates the components of a smart health infrastructure that utilizes lightweight-based security scheme based on lightweight cryptography. This security scheme relies on a lightweight encryption algorithm towards low-latency communication for internet of things (LEAIoT) cryptographic primitive. Light encryption and decryption algorithm processes that data and manages key size that enable high execution speed. This mechanism ensures data privacy within communication networks.

For the LEAIoT encryption, a synthetic value is assigned to the plaintext. The next step involves symmetric key encryption using a private key n, which is known to both sender and

receiver. Then the ciphertext is encrypted using asymmetric linear block cipher, a public key, and a private key. The original plaintext is obtained by applying symmetric decryption using the SSK key which the modular inverse of n [8].

The lightweight design of LEAIoT addressees the complex communication requirement of the IoT-based healthcare environment. It provides efficiency with low hardware resources consumption. The symmetric and asymmetric encryption algorithms provide both speed for real-time data transfer and scalability for large-scale deployments. Thus, it establishes a strong defense mechanism for a lightweight-based security scheme, contributing to the confidentiality and authentication of the smart hospital system [9].

However, the LPKI is more complex to configure and manage than PKIs due to its lightweight cryptography and protocols. It supports specific range of devices. LPKI are susceptible to man-in-middle attacks. Thus, its security has limitations for lightweight PKIs.

2.3.3 Role-based Access Control (RBAC)

RBAC is a security model that controls access to the resources based on the roles assigned to uses and devices. The roles are predetermined by the administrator and includes the use of privileges and functions. This approach is simple to implement and understand. The RBAC is a flexible security mechanism and can be implemented for a wide range of devices. Its scalability supports managing many users and devices.

The security model of RBAC consists of users, roles, permissions, and resources as shown in the figure below. In RBAC, users are assigned to roles and depending on the roles of the users, permissions are granted. When users log in the system, they get the privileges according to their assigned roles [10]. Further, the users are authorized to specific data or resources.

RBAC defines a set of permissions associated with each role. These permissions determine the actions of the users for that role on the system’s resources. For example: A physician role might have permission to view and edit patient records, while the nurse might have permission to view patient records.

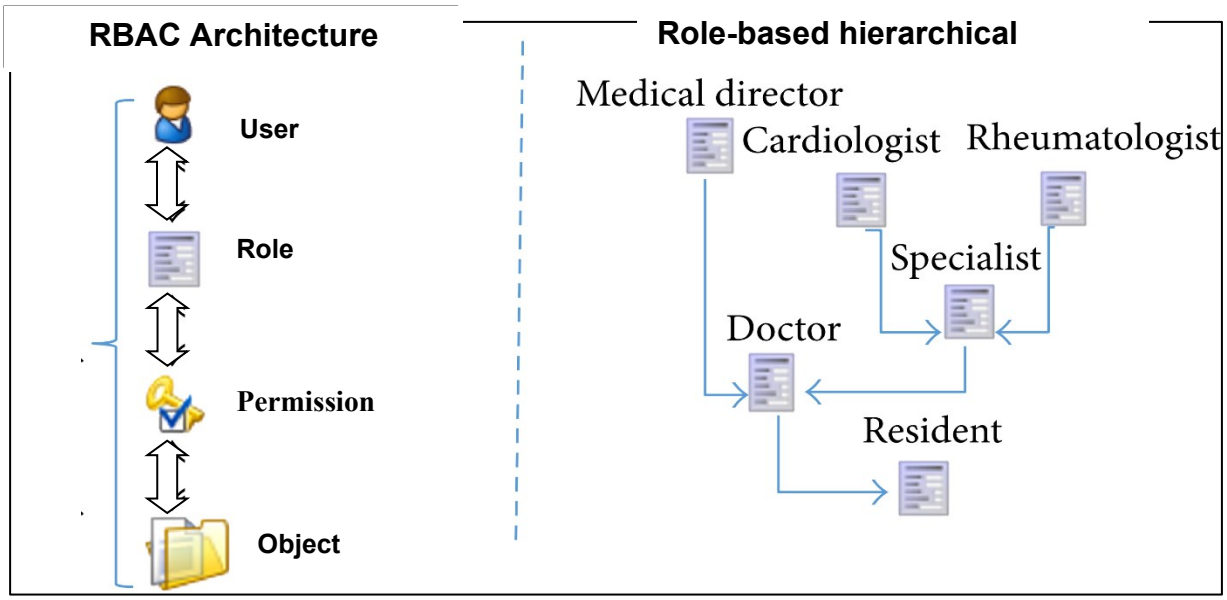

Figure 6. RBAC architecture and hierarchy of smart healthcare system [11]

Figure 6 shows the RBAC security model to gain to access to data. In RBAC, all users are authorized to access the data according to the predefined roles. For example, in a hospital, a doctor and nurses could get real time data access to the sensors of the patient being treated. Additionally, the doctors could have access to medical devices and but not the nurses.

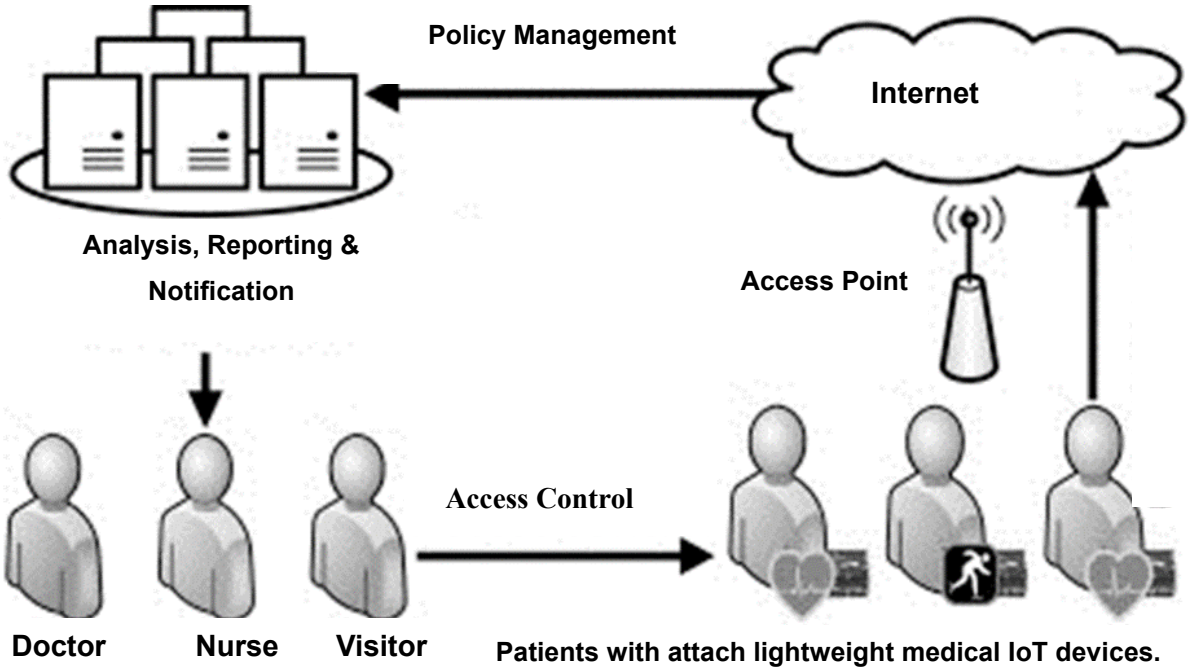

Figure 7. Role-based access control architecture in healthcare [11].

Figure 7 shows different users with different roles and access points to the data. Also, the roles of the users could be changed, and permissions could be extended or seized in the healthcare system. For example, nurses and doctors might need repeated access to a resource. The doctors and nurses treating different patients might need to access sensors attached to them. They might need access to different door locks and building infrastructure to which they have authorized access. RBAC security model enables the management of roles and responsibilities for users [11].

RBAC has limitations when it comes to large-scale organizations as its complexity rises with the scale. Its flexibility requires a granular access control system that might not be suitable for organizations with limited resources. The RBAC is vulnerable to role spoofing attacks. Thus, it is essential to evaluate the needs and limitations of RBAC before implementation.

2.3.4 Proxy-based Access Management (PBAM)

PBAM is a security mechanism that utilizes proxies to control access to resources. The proxies could be hardware and software that acts as an intermediary between clients and server. The proxies intercept the traffic between them. These proxies manage authentication and authorization processes of lightweight IoT devices that help to minimize resource burden [12].

In PBAM a client requests access to a resource and the request is sent to a proxy. The proxy then authenticates the client and authorizes to access the requested resource. The proxy transfers the request to the server. The server responds to the request and that response is forwarded back to the client [12]. Using this mechanism PBAM can facilitate a wide range of devices.

Figure 8. Proxy-based Access Control [12]

Figure 8 illustrates a proxy-based access control scheme for implantable medical devices (IMD) programmers to reduce computation burden and power consumption. These IMDs could be an insulin pump to diagnose and monitor patients' conditions, a pacemaker to regulate heart beating using electrical pulses, and a neurostimulator to send impulses to the spine to treat chronic pain and disorders. The IMD are lightweight devices that have low computational and low battery power. The data from the sensor attached to either a patient or a person is transferred to a proxy device, which could be a smartphone to handle complex cryptographic tasks for access control [12]. The communication between proxy devices and IMD utilises lightweight symmetric encryption for data integrity and confidentiality.

The PBAM has certain limitations. It can disrupt the performance due to traffic overflow in the proxy. The PBAM system is complex to configure and manage. Its vulnerability lies in the proxy, as it handles all the traffic flow. The attacker could gain access to resources if the proxy is compromised.

2.3.5 Context-aware Access Control (CAC)

CAC is a framework that dynamically grants or denies permissions to access resources based on the context of the request. This contextual information includes the user’s identity,

location, device, and time of day. CAC is more flexible than RBAC, leveraging wireless communication channel attributes, Physical Layer Security techniques for lightweight IoT access management. These techniques exploit channel randomness to establish secure communication links, enhancing access control without intensive cryptographic operations [14].

In CAC the user requests access to a resource. The CAC system collects context information of the request and evaluates the contextual information with its predefined policies. If the request is matched the system grants access to the resource [14]. Thus, it helps organizations to comply with the regulations to protect sensitive data and improve usability and risk of unauthorized access.

The limitation of CAC lies in design and implementation. It is expensive to maintain and purchase. The system could cause an overflow of performance as it requires evaluation of the context information of each request. The CAC system creates a profile of the user and collects personal information. These data can be used to track the activities of users [14].

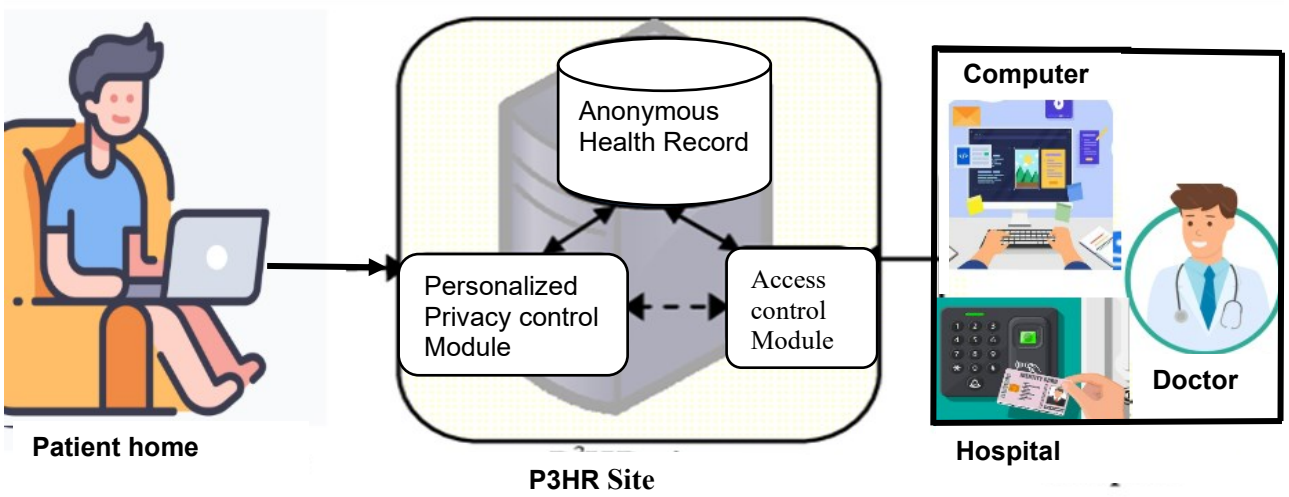

Figure 9. Privacy management architecture personal health record system [15]

Figure 9 shows that CAC can be used to protect patient records and sensitive personal data by allowing only authorized user in authorized location. For example, A hospital can grant doctors access to patient records only when they are in the hospital and with authorized devices.

Similarly, a hospital using CAC can grant visitors access to a patient room only when the patient approves it. The bank and government agencies can utilize the CAC to control user's privileges and maintain the scalability and availability of specific data assigned to them [15].

2.3.1 Comparison of Existing Access Management Methods for Lightweight IoT Environments

The comparison of existing access management methods for resources constrained IoT Environments are mentioned in the table below. It highlights the summary of advantages and disadvantages of each method based on the literature review.

Method	**Advantages**	**Disadvantages**
Token-based Access Control (TBAC)	Lightweight, efficient, secure, supports different ranges of devices and policies	Relies on single key, complex cryptographic signature algorithm, restricts to manage clients from server side
Lightweight Public Key Infrastructure (LPKI)	Low computational and storage requirements, efficient, secure, can be used in various devices	Complex to configure and mange, limited range of supported devices, susceptible to man-in-the-middle attacks
Role-based Access Control (RBAC)	Simple to implement and understand, flexible, scalable	Complexity rises with scale, requires granular access control system, vulnerable to role spoofing attacks
Proxy-based Management (PBAM)	Minimizes resource burden, facilitates a wide range of devices	Can disrupt performance due to traffic overflow, complex to configure, and manage, vulnerable if proxy is compromised.
Context-aware Access Control (CAC)	Flexible, can be used to comply with regulations, protect sensitive data, improve usability and risk of unauthorized access	Expensive to maintain and purchase, can cause performance overflow, collects personal information that can be used to track users

Table 1. Summary existing access management methods for lightweight IoT

Table 1 illustrates the key advantages and disadvantages of existing access management methods for lightweight IoT environments. It provides an overview of their suitability and limitations in resource constrained IoT scenarios. It also highlights that each method has its distinct approach with its strengths and weaknesses essential for securing resource constrained IoT devices.

2.4 Related works in Lightweight Devices

The literature reviews show that both established and emerging technologies have unique characteristics. However, there is a lack of fully utilizing the potential synergies among these technologies. By combining these technologies, we can bridge existing gaps and leverage their collective attributes to enhance security protocols and improve user accessibility. For example, the integration of token-based access control with blockchain technology and the combination of role-based access control with OAuth 2.0. These examples demonstrate how these technologies can work together to achieve better results.

2.4.1 Integration of Token-based access control with blockchain technology

Integrating token-based access control with blockchain technology has the potential to enhance security and accountability in Internet of Things environments. This approach combines the strengths of token-based authentication and authorization with the blockchain's decentralized and unchangeable nature.

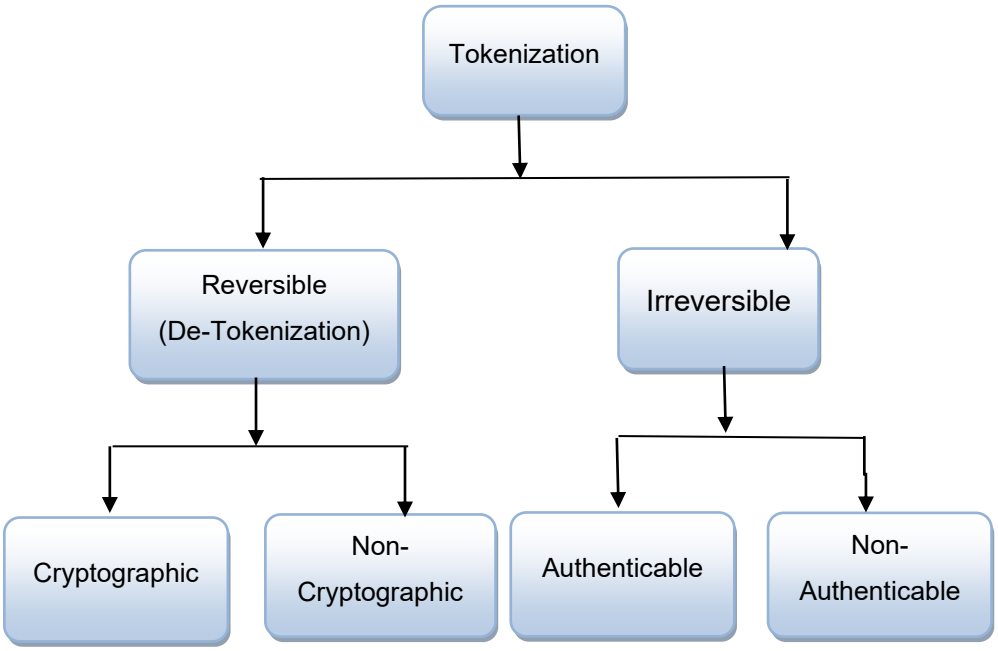

Figure 10. Token payment classification [16].

The above figure 10 shows the token-based access control relies on lightweight cryptographic tokens to manage authentication and authorization processes. These tokens are exchanged between devices and users during communication, allowing secure access to resources without the need to constantly transmit sensitive credentials. This approach reduces the risk of credential theft and lessens the computational burden on resource-limited devices, which is particularly important in IoT scenarios where devices often have limited processing power and memory [16].

On the contrary, blockchain technology introduces a decentralized and tamper-resistant ledger that systematically records all transactions and events in a transparent and unmodifiable manner. Each access occurrence, including actions such as authentication and authorization, can be securely entered into the blockchain. This creates an immutable record of the identities interacting with resources and the corresponding chronological markers [16]. This guarantees a strong accountability mechanism and provides a comprehensive record, which is crucial in regulatory compliance, forensic examination, and the coordination of incident responses.

By integrating token-based access control with blockchain, the security and accountability of access management are significantly strengthened. Tokens are used to enable authentication and authorization processes. When a user or device wants to access a resource, a token is created and exchanged. This token contains the necessary permissions and is validated by the recipient before granting access [17]. This process maintains the lightweight and efficient nature of token-based access control.

Every access event, along with its relevant metadata, is recorded as a transaction on the blockchain. These records are distributed across multiple nodes, making it highly challenging for malicious actors to tamper with the historical access data. This feature guarantees the integrity of access events. The blockchain's transparent nature allows authorized parties to review the access events and permissions assigned to different users and devices [17].

This transparency enhances accountability and facilitates audits to ensure that access control policies are being followed correctly. Blockchain's decentralized architecture eliminates the need for a central authority or administrator to manage access control data. This is particularly advantageous in scenarios where a single point of failure or vulnerability could compromise the entire access management system [17].

Blockchain's encryption and consensus mechanisms further enhance the security of access control data. This is especially beneficial in scenarios where unauthorized access attempts or breaches need to be detected and prevented in real-time.

Integrating token-based access control with blockchain addresses the challenges of secure authentication, authorization, and accountability in a lightweight and efficient manner. It ensures that only authorized entities can access resources while maintaining a secure and tamper-proof record of access events. This combined approach is valuable not only in IoT environments but also in various other applications where maintaining a trustworthy record of access activities is essential for security and compliance.

2.4.2 Integration of RBAC and OAuth 2.0

The rapid growth of the Internet of Things (IoT) has created a need for effective access management strategies. This article explores a new approach by combining two established mechanisms: the Role-Based Access Control (RBAC) framework and the OAuth 2.0 protocol. The main goal is to develop a dynamic solution that can address the challenges of lightweight IoT environments.

By combining RBAC's structured role allocation with OAuth 2.0's strong authentication, the article aims to demonstrate a successful way to enhance security and user-friendly.

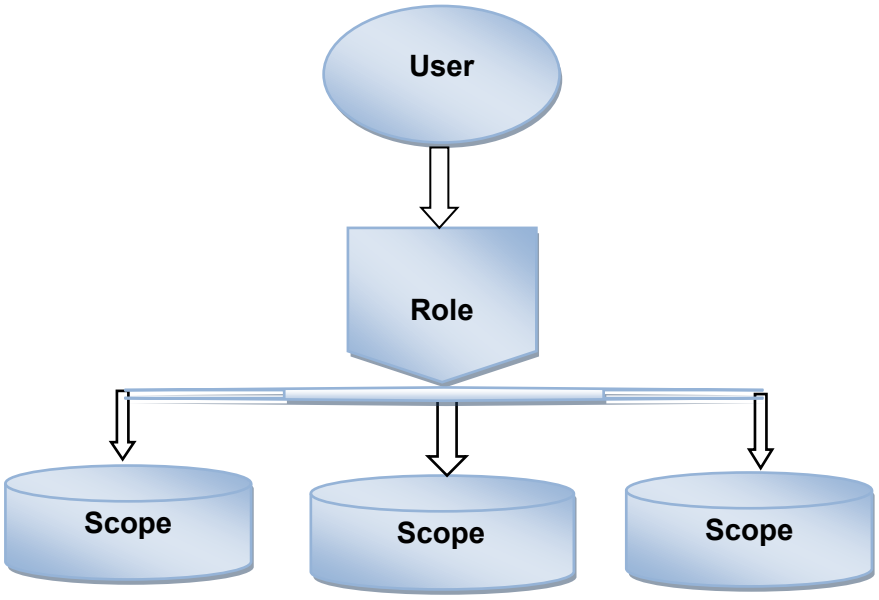

Figure 11. Proposed Role-Based Access Control. [18]

Figure 11 illustrates a proposed OAuth 2.0 authorization flow for controlling the relation between each service. Here each user is authenticated and authorized according to their role that is integrated with scope. This integration allows a convenient change of role for users and third-party integration. This approach simplifies the authorization process in resource-constrained environments.

Role-Based Access Control (RBAC) is a widely recognized access management model that organizes users and devices into specific roles. In the realm of lightweight IoT devices, this framework provides a structured approach to categorize devices according to their functional purpose. By managing access permissions in a coordinated manner, this system simplifies administrative tasks and ensures the security of IoT interactions [18].

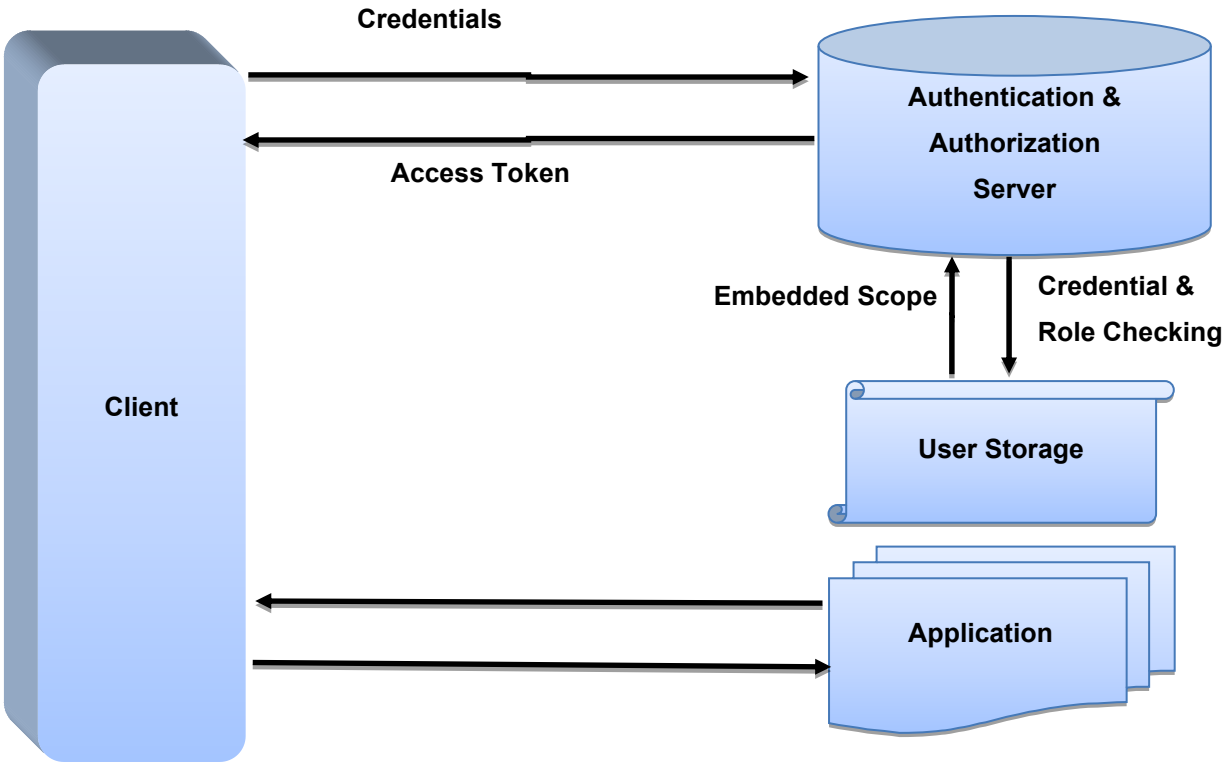

Figure 12. Proposed OAuth 2.0 Authorization Flow [18]

Figure 12 illustrates the approach described in article [18], which initiates with a client or third-party application authentication process by providing credentials to the server via an API call from the application's interface. The server verifies the credentials by cross-referencing with the user storage server. If the user is validated, the server gathers all necessary access permissions and roles of users which is included in the scope. The scope is sent to the authentication and authorization server to create an access token.

The generated access token is then issued to the client. This token grants permission to the client to access the resources within the application. Additionally, the application inspects the embedded scope in the access token to confirm the specific access rights granted to the user.

This approach simplifies the authorization process and ensures that each scope is implemented to its respective resource access. Thus, it enhances the security and control of privileges [18].

The main idea is to introduce authorization tokens to access gateways without the risk of exposing sensitive credentials. The proposal suggests integrating OAuth 2.0's token mechanism to create a secure and seamless user authentication system that can be used across multiple IoT.

2.4.3 Convergence of RBAC and OAuth 2.0

This research focuses on the seamless combination of RBAC and OAuth 2.0. In this approach, IoT devices are assigned specific roles, each with well-defined access privileges. At the same time, OAuth 2.0's cryptographic tokens allow users to authenticate once, granting them access to various IoT domains without the hassle of managing multiple credentials [18]. This harmonious integration creates a comprehensive access management framework that enhances device security and improves user convenience.

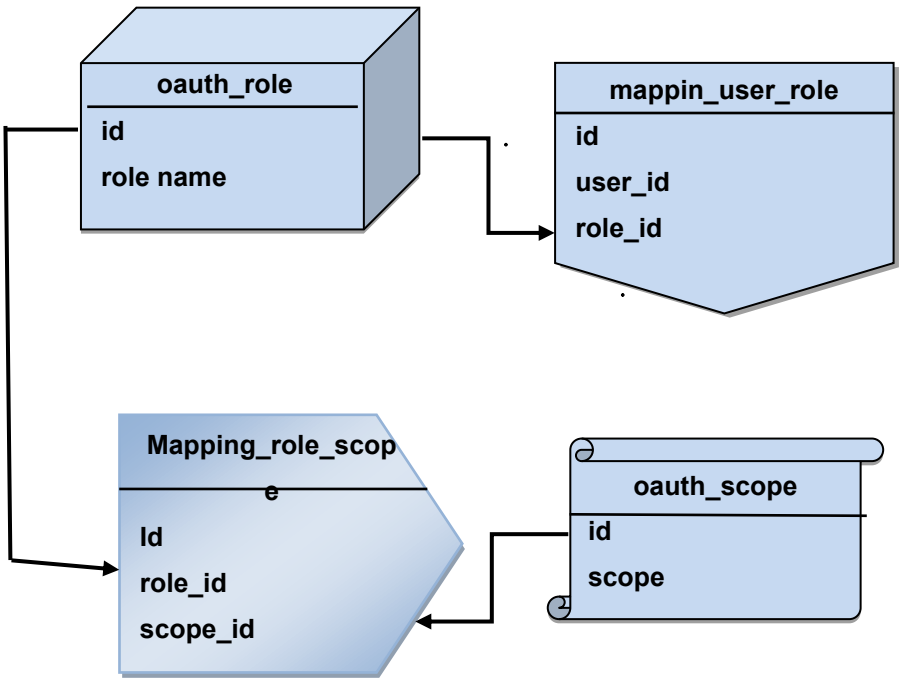

Figure 13. Proposed Role Model for RBAC [18].

Figure 13 shows that each scope is mapped to the role. It is implemented in a way where one role can have multiple scopes. This similar principle applies to the relationship between users and roles, where a user can have multiple roles [18].

The combination of these solutions provides enhanced security and user experience. RBAC ensures that devices operate within authorized parameters, preventing unauthorized access. OAuth 2.0's tokens simplify user authentication, reducing the need for constant credential management. This creates an IoT ecosystem that is both secure and user-friendly [18].

The literature review analysis reveals the hidden potential of combining RBAC and OAuth 2.0 for lightweight IoT access control. This combination offers a solution to the ongoing challenge of balancing security and convenience.

The practical applications of this integration are reassuring and with the advancement of IoT devices, the demands for exploring the incorporation of emerging technologies to enhance its effectiveness rise steadily.

2.5 Emerging Technology for lightweight IoT access management

Access management is evolving to adapt to limited processing power and memory of lightweight IoT devices. Emerging trends and cutting-edge technologies are changing the future of IoT access management mechanisms. The software defined networking and lightweight cryptography-based solutions are being adopted to enhance communication, scalability, adaptability and enforce security policies.

Lightweight cryptography is designed to secure IoT devices with limited computational resources. These solutions offer optimized cryptographic algorithms and protocols while ensuring robust security. By implementing lightweight cryptographic primitives for authentication, encryption, and key exchange, IoT devices can achieve secure access management without excessive computational overhead [19].

For example, a wearable device manufacturer uses lightweight cryptography algorithms to secure communication between its devices and cloud-based healthcare platforms. It uses lightweight cryptography algorithms designed for resource constrained IoT devices [19]. Since, lightweight cryptography algorithms consume less energy, have lower computational

overhead, and are actively maintained and update. It ensures protection against cyberattacks without burdening the device resources.

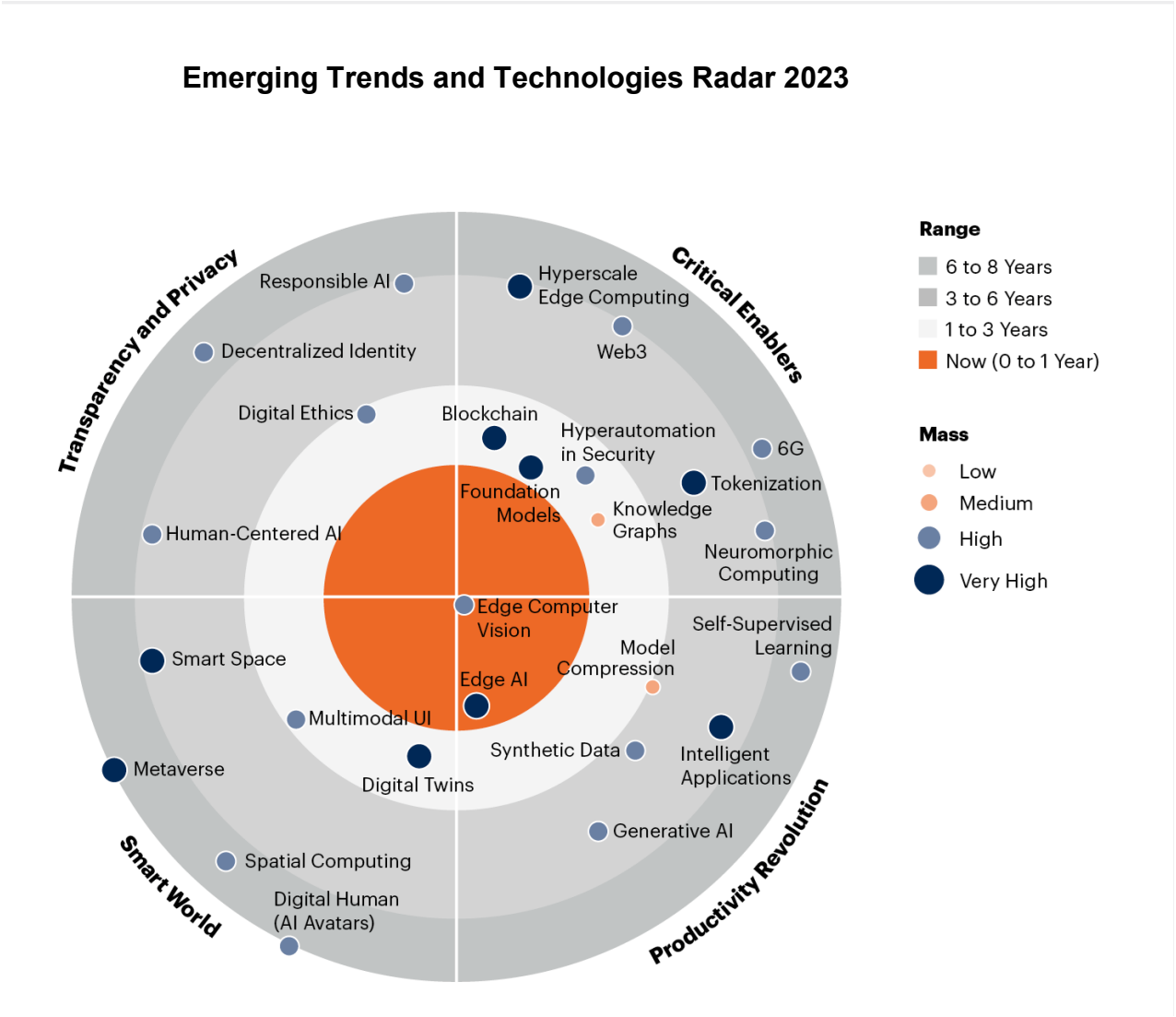

Figure 14. Emerging Technologies and Trends Radar in IoTs [19]

Figure 14 illustrates the emerging trends and technologies and their expected time for full implementation. According to it, emerging technologies greatly impact business and society in the next decade. The emerging technologies and the digital business transformation must go side by side to grave the potential benefits. On the contrary, the risk lies in the fact that few technologies reach the full-scale implementation. It further illustrates that these technologies provide individuals with more control over their identities and data. The major characteristic of new technologies focuses on decentralized identity, AI-driven representation, edge-fog computing, and blockchain-based mechanisms [19].

2.5.1 Blockchain and Distributed Ledger Technologies

Blockchain technology is a potential solution for enhancing the security and integrity of resource-constrained IoT access management. Its decentralized and tamper-resistant nature ensures trust and maintaining auditable records. Blockchain can provide secure device identity, authentication, and access control, particularly when a centralized authority might be vulnerable to compromise [20]. This technology ensures secure and auditable access management, crucial in lightweight IoT environments with interconnected numerous devices with limited computational power.

Blockchain can be used in lightweight IoT devices to store sensor data, and device configuration data. Blockchain enables lightweight devices to share data securely with cloud servers and with third parties. The lightweight IoT devices do not have the resources to run complex transaction processing systems. Blockchain can help to address this challenge by providing a decentralized and efficient way to process transactions [20].

The emerging trends in blockchain for resource-constrained devices are lightweight blockchain protocols, privacy-preserving blockchain solutions to protect the privacy of users and their data, and scalable blockchain networks designed to process large number of transactions efficiently.

Blockchain enhances the security and efficiency of access management for lightweight IoT devices. Blockchain enables secure storage of access control information such as user identities and permissions. The information can be encrypted and stored on the blockchain to prevent tampering of data and unauthorized access.

Furthermore, Blockchain can be implemented to create self-sovereign identities which are controlled by users. This provides more control over private data and makes it easier to share with others. The decentralized nature of blockchain can use cryptographic techniques to secure access management while reducing reliance on a central identity provider. The access rules and permissions could be implemented into smart contracts that allow devices to interact automatically without the need for central authority. Thus, it reduces the trust dependencies on a single entity.

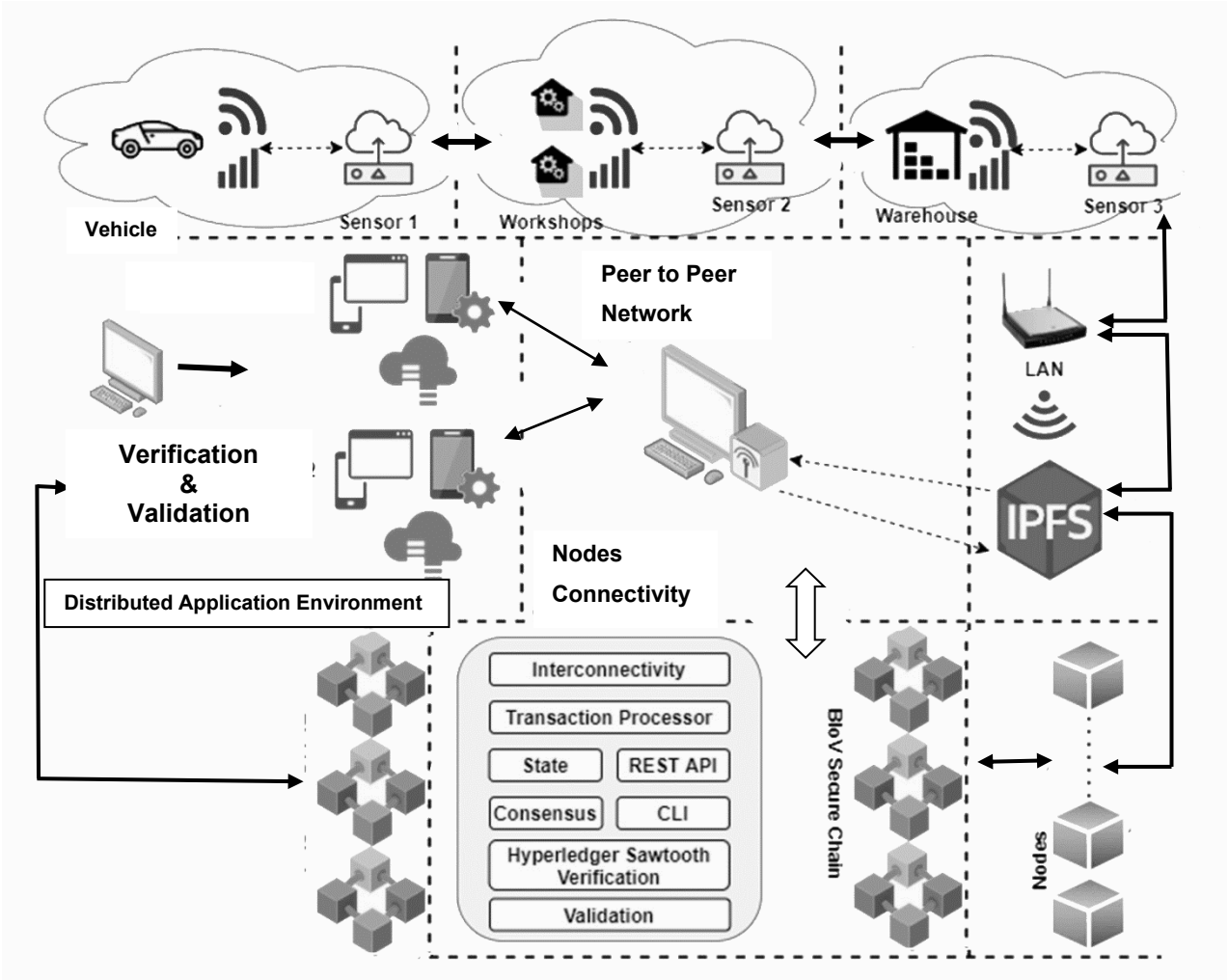

Figure 15. Blockchain Lightweight Internet of Vehicle architecture [20]

Figure 15 illustrates a blockchain serverless network environment designed to manage vehicle data with network sensors (radio-frequency identification sensors). The collected data on the internet of vehicles is sent to workshops for examination and analysis. These workshops identify the potential risks and issues within the data and propose a solution. The records are then transmitted to the warehouse for processing and storage. All transactions are received and delivered over LAN to maintain ledger security.

The blockchain Hyperledger verifies and validates the connectivity of each device and mode. The P2P distributed network connects between data transmission and computational nodes facilitating efficient data exchange. A Hyperledger sawtooth-enabled multiple-validation mechanism is implemented to maintain the integrity, transparency, and validity of digital signature [20].

One of the examples of the distributed ledger technology (DLT) implementation for lightweight IoT is IOTA. It is a DTL that is designed for devices with low resource requirements and scalability. It uses Tangle to allow high transaction throughput which eliminates the need for validators. Tangle refers to acyclic graph structure that allows for high transaction without validation. Each transaction is represented as node and each node is connected to previous node, creating a chain of transactions. [22]

Thus, the tangle can handle multiple transactions simultaneously without the need for block sizes. IOTA can be used in different IoT applications such as smart cities, autonomous vehicles, and supply chain management as shown in figure below.

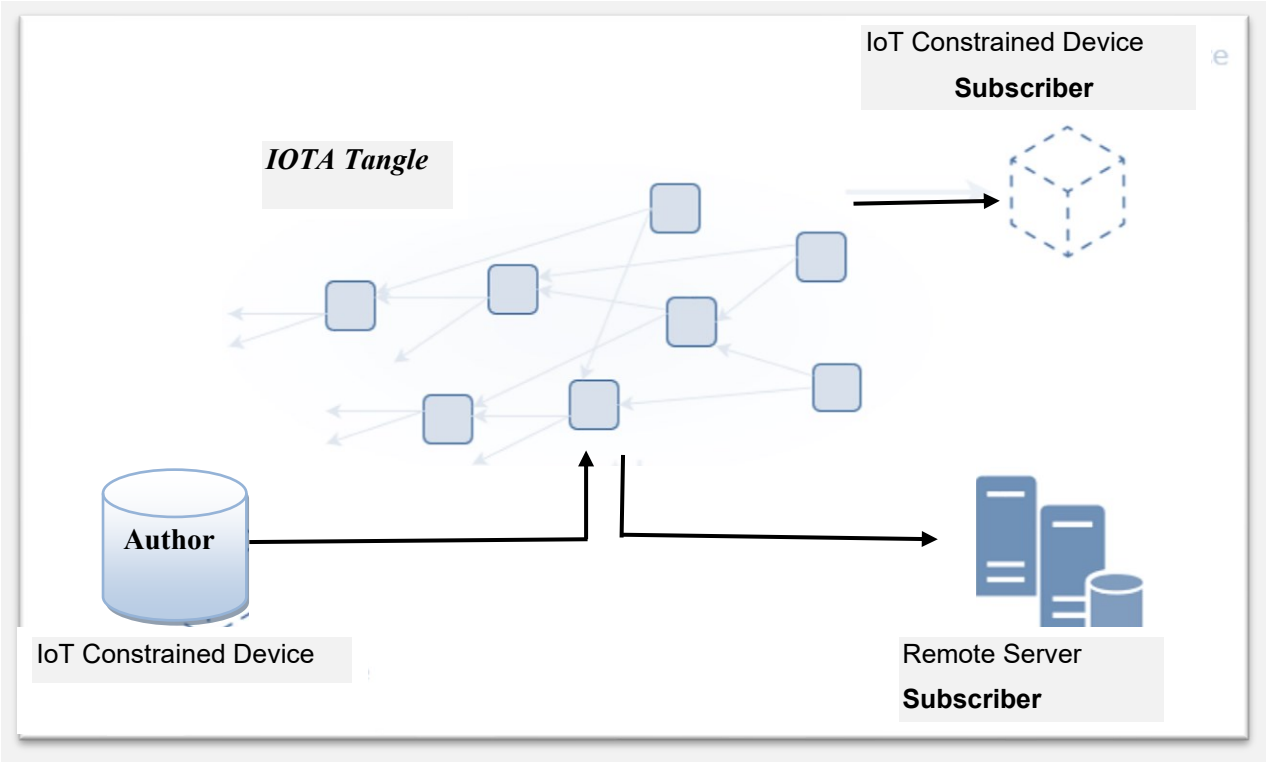

Figure 16. Constrained devices communication through IOTA Tangle [22]

Figure 16 shows the high-level system architecture of IOTA Tangle. In the above figure, IoT constrained devices represented as authors that collects the data and transmits to the Tangle and the remote servers access the Tangle to retrieve the data. Tangel utilizes symmetric cryptographic key to encrypt the message and to transmit the data securely to the server.

The encryption is further backed up by Message authentication code to verify the data integrity. These two verifications are performed to retrieve the data. The public key is used for

deriving the next message index and included in the message chain. On the receiver side the signature and public key are used to authenticate the recipient.

For example, in smart cities, IOTA can be used to enable secure micropayments between IoT devices such as parking meters and streetlights. In autonomous vehicles, IOTA can be used to facilitate secure and real-time data sharing between vehicles, traffic lights and road sensors.

2.5.2 Zero Trust Architecture and Federated Identity Management

The Zero Trust architecture is a new approach of the security model, and its principle is applied to the vast IoT ecosystem. Zero Trust emphasizes the principle of "never trust, always verify." The devices are not automatically trusted based on their location or origin but are continuously authenticated and verified before access is granted. The application of Zero Trust principles in IoT access management mitigates the risks associated with compromised devices or unauthorized. ZTA microsegment the devices and each segment are isolated from others which prevent attackers from moving across the network.

ZTA provides the least privilege access to lightweight IoT devices. The users are limited to the limited permission to perform tasks assigned to them. This technique minimizes the exploitation of vulnerabilities in applications. ZTA monitors the lightweight IOT devices for any suspicious activities and restrains the service before damage is done [23].

Zero Trust Architecture is an essential approach for controlling access in IoT devices. The main principle of ZTA for access management in IoT involves the use of least privilege to limit the access of IoT devices, validating and monitoring user identity, device type and micro segmentation to separate from other part of the network. The ZTA provides narrow surface for attacks. It provides extra control and oversight of IoT devices through strong authentication.

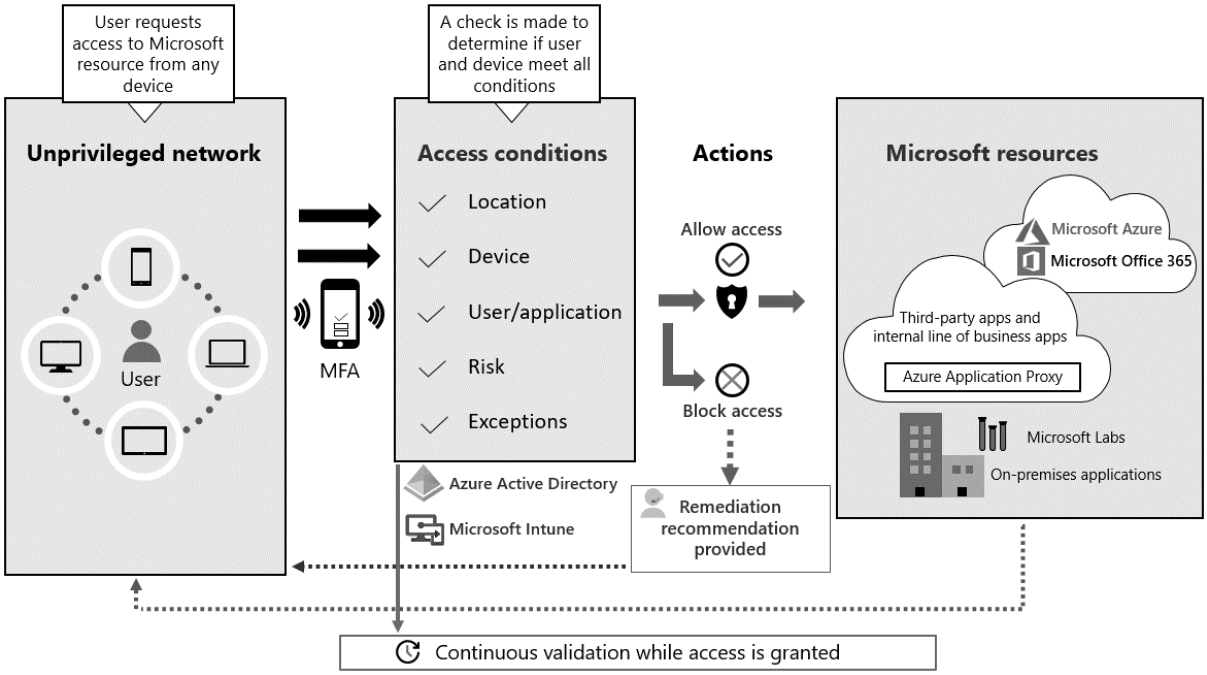

Figure 17. Microsoft Zero Trust Architecture [24]

Figure 13 shows the zero-trust architecture of a company where the user using various IoT devices is being validated with predetermined protocols. If the authentication is successful, the access is granted to resources. If not, then the trust is broken, and access is denied to the resources.

For example: A manufacturing company can implement ZTA to secure its IoT network. ZTA in a company provide device identity and authentication. Each device in the network is uniquely identified and authenticated using a combination of digital certificates and hardware-based security modules. ZTA helps to implement granular access control policies and restrict access to resources based on device identity, role, rand context. It also continuously monitors and verifies devices behaviour to detect suspicious activities. ZTA acts by isolating compromised devices from the network and revoking the access privileges.

Federated identity management (FIM) allows IoT devices to authenticate and access resources across multiple domains without the need for separate credentials for each domain. This approach enhances usability and minimizes the management burden on users and devices. By leveraging standards like OAuth 2.0 and OpenID Connect, federated identity management enables secure cross-domain access while maintaining user privacy. FIM works by using a third-party identity provider [25].

The identity provider (IdP) is responsible for authenticating users and providing with a token to authorize to a service. Lightweight IoT devices can implement the token to avoid the use of credentials. This feature of federated identity management in lightweight IoT devices improves security, reduces the complexity of authentication and authorization, and increases usability [25].

For example, a smart home platform provider can implement federated identity management FIM to enable users to authenticate and access their smart home devices using their existing credentials from other trusted providers such as Google, Amazon, and Facebook.

The FIM solution is integrated with existing identity providers to leverage their authentication capabilities. The token-based authentication mechanisms are used to securely exchange identity information between providers. It enables to share relevant identity attributes with smart home services based on access control policies [25]. Thus, FIM solutions establish standards like SAML and OAuth to ensure interoperability.

2.5.3 Machine Learning and AI in Access Management

Machine learning and artificial intelligence (AI) techniques can enhance IoT access management by enabling intelligent decision-making and adaptable access control. These technologies can detect suspicious activities in device behaviour, and potential threats, and change access permissions based on real-time data.

Machine learning and AI is a shift towards more adaptable, secure, and efficient approaches to lightweight IoT access management. By leveraging blockchain, Zero Trust principles, federated identity management, and machine learning, the IoT ecosystem can address the unique challenges posed by lightweight devices while ensuring robust security and access control [26].

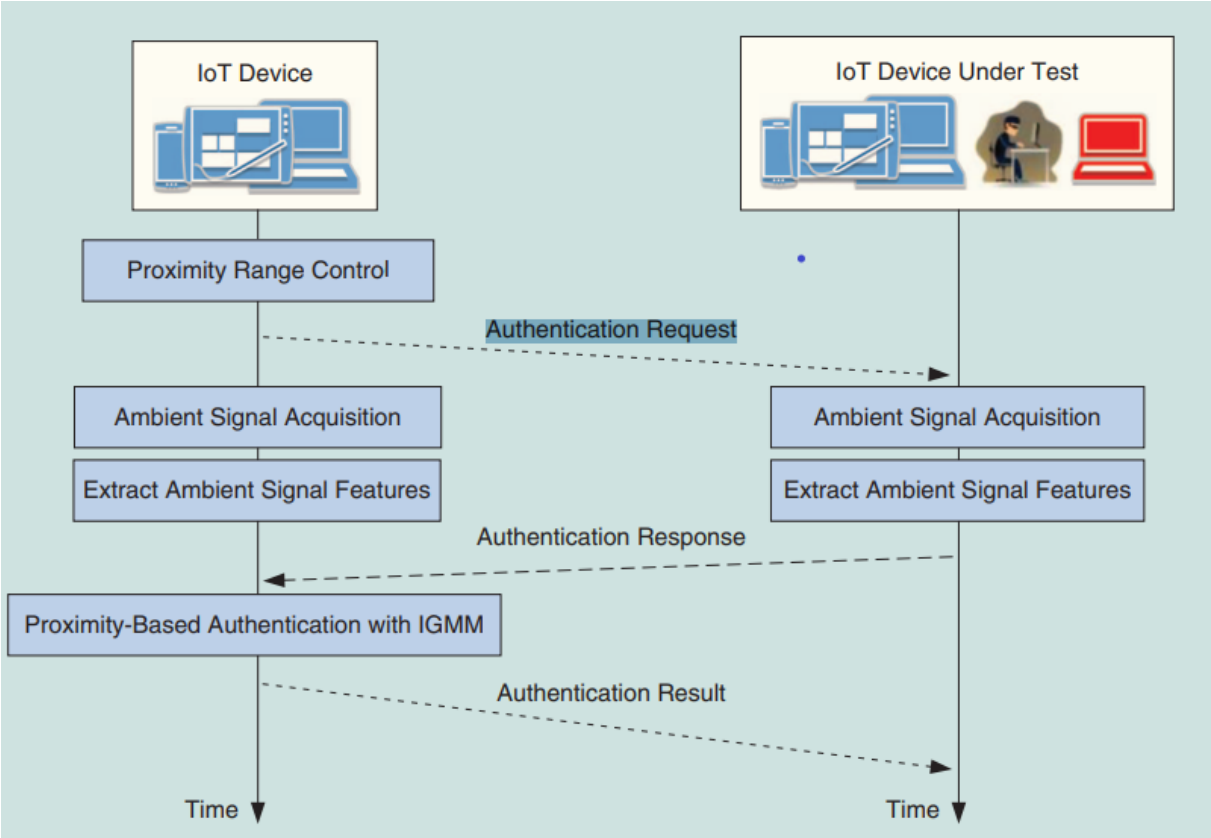

Figure 18. Machine learning based authentication in IoT System [26]

Figure 18 demonstrates machine learning model analyse user behaviour and pattern to identify suspicious activities using ML techniques such as support vector machines (SVMs), K-Nearest Neighbours (K-NNs) and Neutral Networks (NNs) for intrusion detection. Similarly, AI algorithms assess user behaviour and context to authenticate the legitimacy of access requests.

IoT devices, such as sensors and wearable devices, have constrained resources and low computational ability, which is a huge challenge for intrusion detection techniques [26]. The machine learning enables lightweight access control mechanisms that consume less energy and require low computational power.

ML and AI enable real-time monitoring of device behavior and user interactions to identify and address security threats. For example, lightweight IoT platforms such as Particle and Tuya use ML and AI to implement access policies based on historical data and contextual information. This enables the possibility of active and responsive access management in resource-constrained IoT devices, which is also capable of adapting to diverse environments.

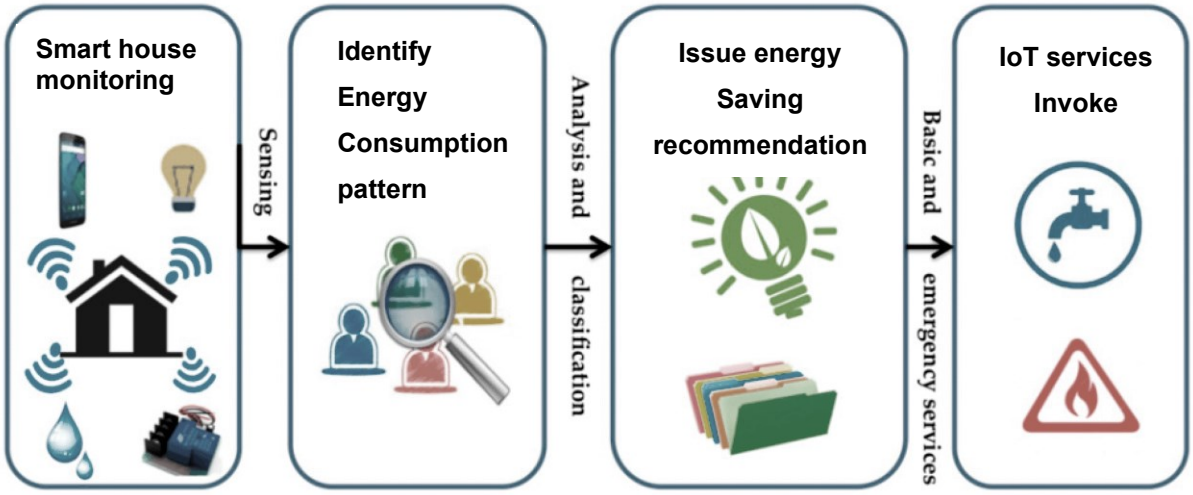

Figure 19. Machine learning in Home Energy Management System [27]

Figure 19 illustrates the machine learning implementation in smart homes to manage energy. In this energy management system, the classification and regression of supervised learning and clustering in unsupervised learning techniques are used. The services are implemented according to the data analyzed to save energy.

Machine learning and AI can be implemented in various other smart home devices. The systems use machine learning algorithms to analyze device behavior and user interaction to adjust access permissions. The system collects data such as network activity, resource usage, and error logs. The algorithms analyze data to identify potential risks and unauthorized access attempts. It also analyses the pattern of user behavior such as time of data, location, and frequency of interactions [27]. If devices show suspicious behavior, the system restricts access to resources or requires additional authentication.

2.5.4 Fog and Edge computing in Access Management

Fog computing is a distributed computing model that extends cloud services in proximity to the devices where data is created and managed. This new approach, where computational capabilities are extended to the edge of the network, helps for better interaction among IoT devices and reduces delays. Fog nodes are located near IoT devices, which enable instant data processing and analysis, leading to improved access management.

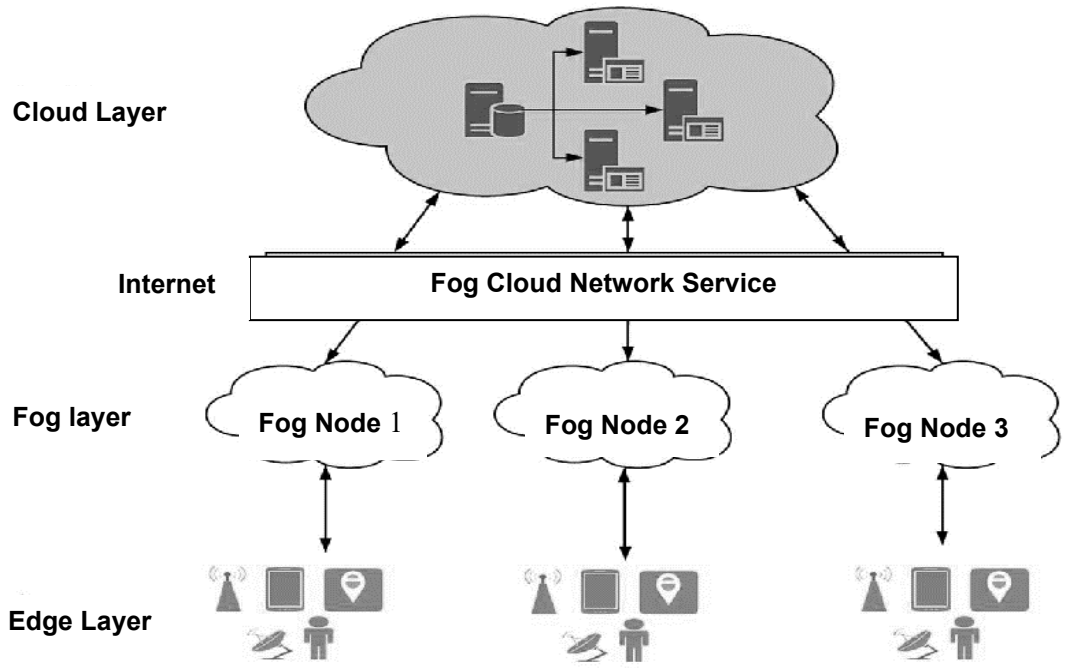

Figure 20. Architecture of fog computing [29]

Figure 20 highlights a fog architecture providing a more efficient way for authentication and authorization by reducing the need to send data to central cloud servers. This reduces potential points of attack. These nodes gather and process the data from IoT devices before sending to the cloud. This helps to minimize the data size, which improves latency. Additionally, the fog nodes help to identify patterns and trends in the data and improve decision-making in real-time [29]. The fog nodes can cache data from the cloud and other fog nodes. This drastically reduces the time to access data.

Furthermore, edge computing expands on the technique of distributing computational resources near IoT devices. This enables efficient data processing and immediate responses at the network's edge, which eliminates the need to send data to remote data centers [29]. The edge nodes can authenticate and authorize the device by verifying its certificate.

The edge nodes can also perform access control access control by enforcing the policies on resources IoT devices are allowed to access. For example, an edge node could have a predetermined protocol to only permit IoT devices to access a specific data and task during certain time of a day [29].

Data encryption is important for lightweight IoT devices. The edge node enables data encryption before storing it on a device. Similarly, edge nodes can monitor the data received

and sent by devices to identify any suspicious activity and take action to mitigate the threat [20]. Additionally, edge computing can also implement zero-trust security and machine learning-based access control for access management in lightweight devices.

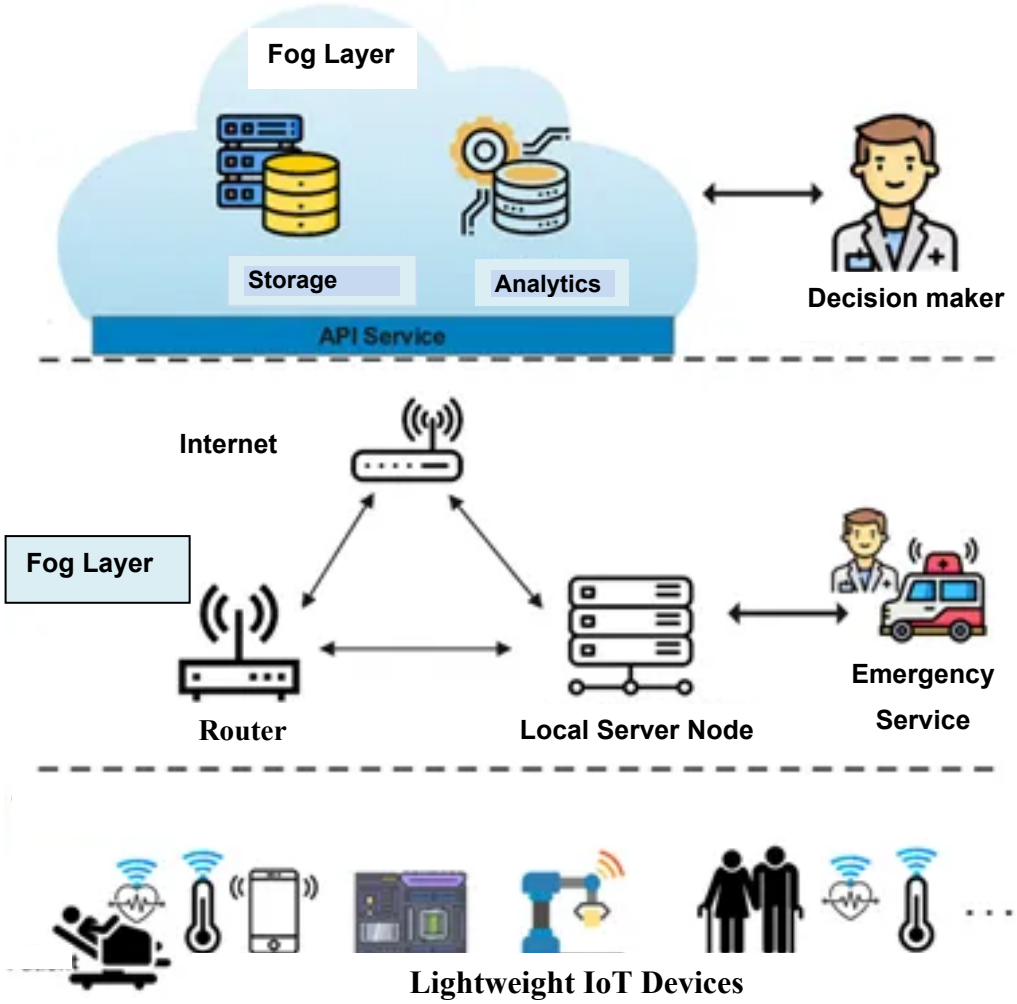

Figure 21. Fog computing in healthcare [30]

Figure 21 demonstrates the implementation of fog computing in healthcare. It highlights the importance of real-time processing and immediate response in healthcare sector.

Fog computing in healthcare application supports real-time image analysis, early disease detection and personalized treatment. It enables services such as remote patients monitoring, wearable devices, with low latency, mobility assistance and location awareness.

The fog computing in smart homes enables handling of multiple devices and sensors. IoT devices from different manufacturers with different hardware limitations and requirements especially lightweight IoT devices with constrained resources is difficult to handle. Fog computing solves this issue by integrating all devices into a unified platform. For example,

home security application consolidates all sensors such as smart locks, and video recorders are integrated into the fog platform. If a motion sensor detects suspicious movement, real-time video analytics can dispatch a cleaning robot with a camera for investigation and send report to the house owner [30].

Fog and edge computing can be used to address several challenges such as latency, security, and privacy of access management in IoT. Fog and edge computing enable real-time access control decision by processing data closer to the devices. This eliminates the data to transfer to cloud for analysis.

2.5.5 Homomorphic and Searchable Encryption-Based Solutions

Homomorphic encryption enables computations on encrypted data, ensuring privacy while processing data in encrypted form. This technology can be applied to access management scenarios, allowing authentication and authorization operations to be performed on encrypted credentials. Searchable encryption enhances data privacy by enabling secure search over encrypted data. Combining these techniques homomorphic and searchable encryption-based solutions are designed for secure IoT access management, ensuring privacy and confidentiality.

These emerging technologies signify the efforts to address the intricate access management challenges posed by lightweight IoT devices. Fog and edge computing, SDN, lightweight cryptography, and advanced encryption techniques collectively contribute to creating adaptable and secure access management mechanisms tailored to the unique requirements of lightweight IoT ecosystems [31].

Homomorphic and searchable encryption plays a vital role in enhancing access management in IoT by addressing security and privacy concerns of IoT devices. Homomorphic encryption allows computations on encrypted information without the need for decryption.

This enables access to sensitive IoT information without revealing the content during data analysis. This allows authorized users in lightweight IoT settings to perform computations on encrypted data without the need to decrypt them.

On the other hand, searchable encryption empowers search of encrypted data, protecting privacy during search operations and eliminating data exposure during data search. This is essential for fine-grained access control in lightweight IoT scenarios where protecting data is

a challenge [32]. Additionally, the access management can be further enhanced by integrating searchable encryption with attribute-based access control in lightweight IoT environment. The access to IoT data is determined by specific attributes that limits that access to only specified users.

Furthermore, searchable encryption schemes support multi-user search. This feature contributes to effective access control by allowing multiple authorized users to access the data without exceeding their permissions [32]. The integration of Homomorphic encryption and searchable encryption into lightweight IoT system strengthens access management by providing strong foundation for secure data processing, dynamic access policies and privacy-preserving search operations.

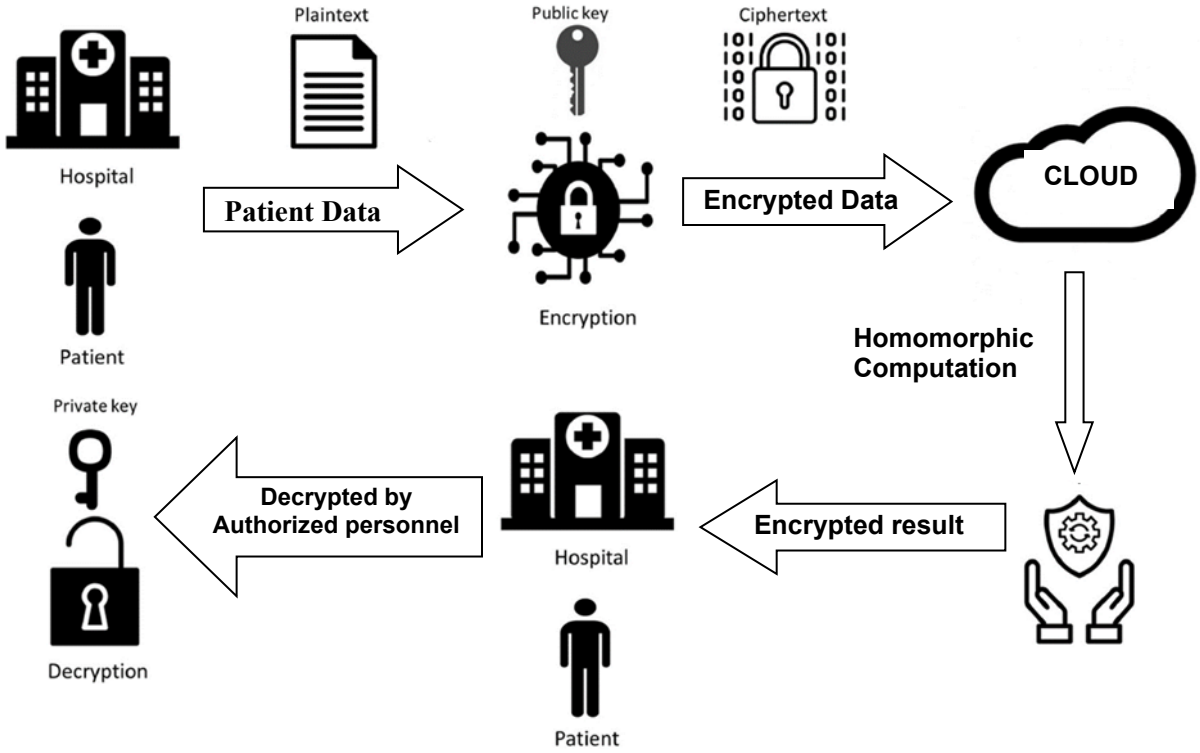

Figure 22. Homomorphic and searchable encryption architecture [32]

Figure 22 highlights homomorphic encryption implementation where a patient or doctor sends encrypted data to a cloud service provider for analysis. The cloud uses homomorphic encryption to perform operations on encrypted data and provides encrypted results. The results are decrypted by the client to retrieve the original information.

Hence, the homomorphic encryption can be implemented in medical devices to protect the sensitive data of patient collected from IoT- enabled wearable devices and medical sensors

[32]. It allows to perform computation on encrypted data without decrypting them. It ensures privacy and confidentiality of patient information throughout the data processing process.

Similarly, the searchable encryption can be implemented to protect the privacy of sensitive data collected from vast network of IoT devices. It enables to search and retrieve encrypted data without decrypting the dataset which reduces the data exposure. Additionally, the searchable encryption algorithm is more scalable and flexible for large-scale datasets, making it suitable for smart city applications [32].

2.6 Comparison of existing and emerging trends and technologies

The table below is the comparison of existing access management solutions with emerging trends and technologies in lightweight IoTs.

Characteristic	Existing Access Management Solutions	Emerging Trends And Technologies
Solution type	Centralized	Decentralized, Token-based
Architecture	Server-client	Peer-to-Peer, Blockchain
Authorization methods	Role-based access control (RBAC), attribute-based access control (ABAC)	Context-aware access control, machine learning-based access control, RBAC for Lightweight IoT, Federated Identity Management, Proxy-based Access Management
Lightweight	No	Yes
Scalability	Limited	High
Security	Moderate	High
Privacy	Moderate	High
Complexity	High	Low
Cost	High	Low
Examples	AWS IoT Core, Azure IoT Hub, Google Cloud IoT Core	EdgeX Foundry, OpenIoT, FIWARE

Table 2. Comparison of old and new technologies for Lightweight devices.

Table 2. illustrates the enhancement and change in methods and usability with the modern technology. Lightweight IoT devices due to their limited resources pose obstacles for developers. The new advancements in decentralized access management, lightweight authentication and authorization protocol and machine learning-based security are developing to fill the gap and facilitate secure and scalable access management solutions for resource-constrained devices.

The decentralized access management system transfers the obligation of managing access permissions to the devices themselves. This eliminates the need for a centralized server, which are vulnerable to attackers.

The lightweight authentication and authorization protocols use efficient cryptographic algorithms that can be implemented by the resource-constrained devices. There are three main features of lightweight cryptography. They are physical, performance, and security. The physical features include the cost, area, logic blocks, memory, and energy consumption [33]. The performance focuses on latency and throughput of computing power.

The security characteristics of lightweight cryptography focuses on minimum security strengths (bits), related key and multi-keys, side-channel, and fault-injection attacks. These characteristics provide the resource-constrained IoT devices to obtain a strong internal structure, and simple key generation with low computation [33].

Additionally, machine learning-based security solutions help to identify and mitigate security threats and can be executed to implement security protocols to meet specific device requirements according to its physical structure, performance, and security strengths.

Table 2 shows that the traditional methods are generally centralized, server-client-based, and depend on shared secrets, passwords, and certificates for authentication and authorization. These mechanisms do not meet the requirements of lightweight devices in many cases.

On the contrary, the decentralized, peer-to-peer, and multi-factor authentication, biometric authentication, blockchain-based authentication, and machine learning-based access control-driven new emerging technologies are providing a better solution for authentication and authorization of resource-constrained devices. The new emerging technologies push the boundaries of existing technologies in terms of security, scalability, integrity, privacy, cost-effectiveness, and usability.

3 Discussion: Authentication and Authorization mechanism of ACE-OAuth framework Profiles

The ACE-OAuth is based on the OAuth 2.0 framework and the Constrained Application Protocol (CoAP). It includes additional profiles and extensions that enhance its ability to support different IoTs requirements. This framework provides a standardized way to implement authentication and authorization in IoT environments to serve different devices and network capability ranges from low power devices such as battery-powered devices with restricted power to mains- powered devices.

The framework includes three components: client, authorization server (AS) and resource server (RS). Client refers to an IoT device that is requesting access to a secure resource on AS server issuing access token to the client and RS server host the secure resources. The framework includes several message flows, such as the authorization grant flow, the client credentials flow, and the resource owner password credentials flow [34].

3.1 Methodology

To conduct a systematic review a thorough analysis of security aspects of the ACE OAuth framework and a comparative assessment with other profiles and additional parameters will be conducted. For this process, various research papers and published articles will be analysed and assessed to further deepen the understanding of the ACE-OAuth framework.

3.1.1 Inclusion and Exclusion Criteria

The articles regarding the building blocks and security mechanisms involved in the ACE-OAuth and OAuth 2.0 were considered for this research. The inclusion is based on publications within the specific period, with primary concern for the authentication and authorization for lightweight IoTs from IETF publications. The articles that do not contain the key terms like authentication, authorization, constrained IoTs, and security aspects of the OAuth framework are excluded. Furthermore, the articles before 2016 are not considered for this research. Additionally, authors work with related topic articles are considered to search for gaps and understand the fundamentals of the framework.

3.1.2 Data Synthesis and Analysis

The extracted information included the improved framework, security measures, and comparison studies with established theories. The ACE-OAuth framework was analysed with a qualitative content analysis approach to identify any patterns, trends, and developments.

A detailed analysis and assessment of selected articles will be conducted regarding the application to ACE OAuth security. The inclusion criteria will entail a broad category comprising all permissions methods, especially those relevant to this research, and originated primarily from IETF publications. Data collection will involve the structured collection of relevant information from purposely chosen articles for future reference.

3.1.3 Comparison of the ACE-OAuth Framework and identifying gaps

A comparative analysis will be carried out to investigate the development and changes in the OAuth 2.0 framework of the ACE. In this assessment, different framework developments will be reviewed, focusing on how permission has evolved over time. The data will be analysed to determine trends and gaps within the ACE-OAuth framework. It will help to further areas for improvement, and to compare robustness of the framework.

3.2 **Basic Protocol Flow**

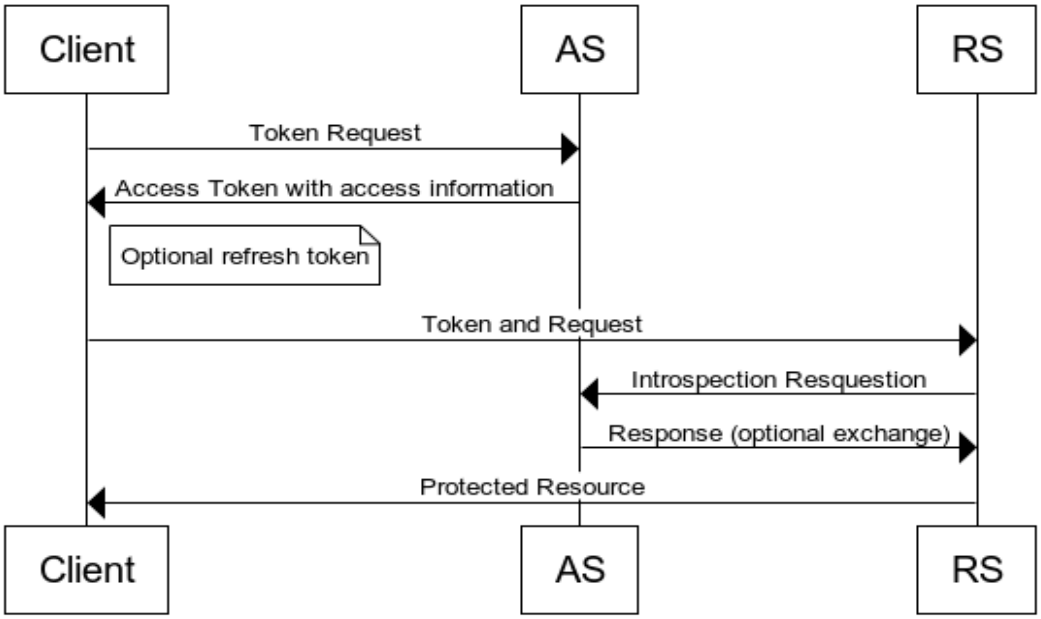

Figure 23. ACE-OAuth Protocol flow [34]

Figure 23 shows communication-security mechanisms between the client, AS, and RS. Based on these mechanisms, the ACE-OAuth framework can utilize various protocols. The framework includes profiles for different IoT cases and communication protocols.

These profiles include ACE for constrained environments using CoAP, CoAP-DTLS, and CoAP-OSCORE for constrained environments [34]. Thus, the ACE-OAuth framework provides a standard approach for incorporating authentication and authorization mechanisms like mutual authentication and TLS support in IoT environments. The ACE-OAuth framework utilizes CoAP and its diverse profiles and features to ensures security and adaptability.

3.3 Authentication and Authorization mechanism

The ACE-OAuth framework uses various mechanisms for authentication and authorization within an IoT environment. Its security features include mutual authentication, token binding, utilization of pre-shared secrets (PSKs) and Transport Layer Security (TLS).

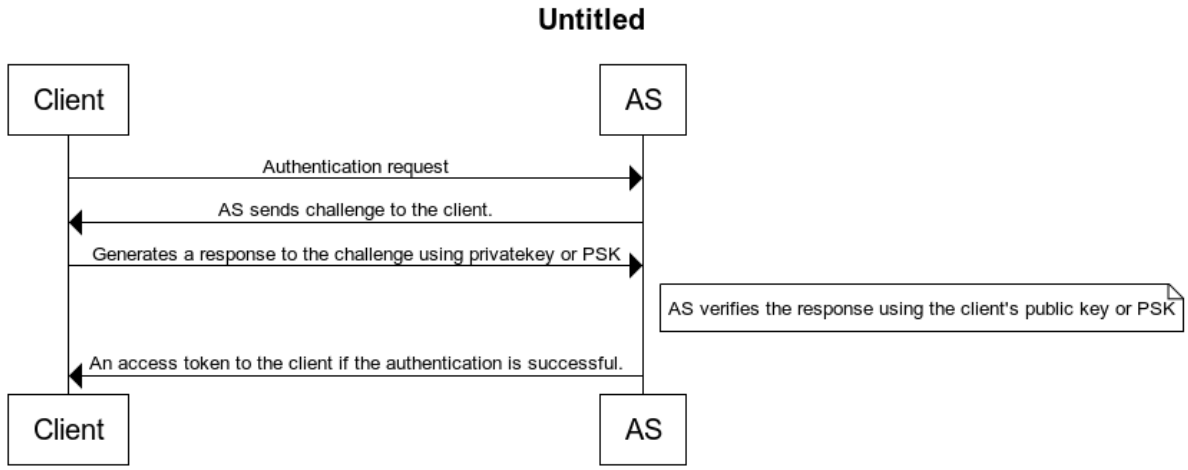

Figure 24. ACE-OAuth Authentication and Authorization mechanism. [34]

Figure 24 illustrates the mutual authentication process in the ACE-OAuth framework. Mutual authentication is a key security feature of the ACE-OAuth framework. It validates that the client and the authorization server (AS) authenticate each other before access tokens are issued. The PSK is used for the authentication process. The PSK uses cryptographic algorithms such as RSA, ECDSA, or HMAC.

The mutual authentication process in the ACE-OAuth framework involves the following steps:

1. The authentication process initiates with a request to AS from a client.
2. The AS sends a challenge to the client.
3. The client generates a response to the challenge using its private key and sends it to the AS.
4. The AS verifies the response using the client's public key. If the authentication is successful, an access token is sent to the client.

Token binding certifies that access tokens are bound to the client's TLS connection and prevent unauthorized access. In this process the token binding key (TBK) that is derived from the client's TLS connection using cryptographic algorithms such as SHA-256 or HKDF [3].

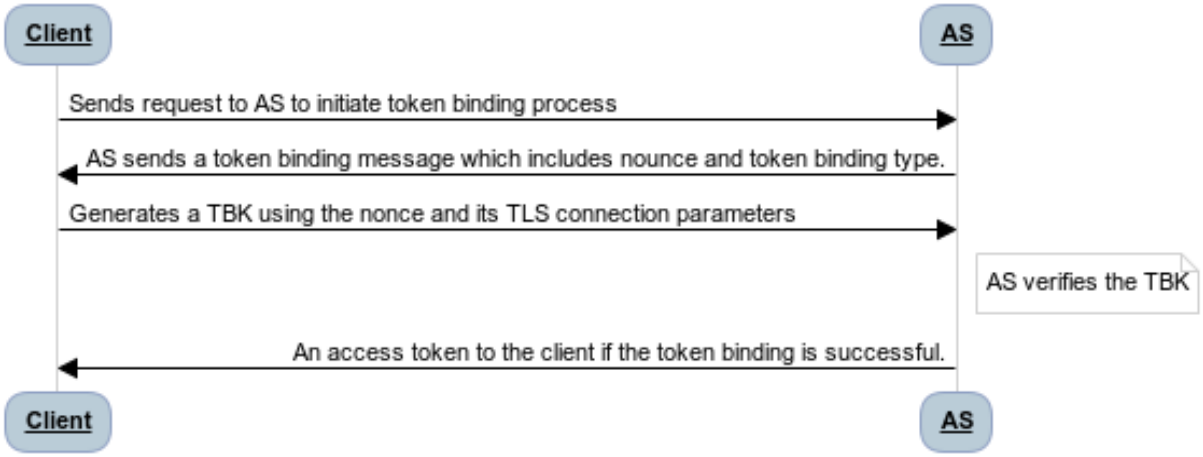

Figure 25. Mutual Authentication process in ACE-OAuth [34]

Figure 25 shows the token binding process in the ACE-OAuth framework. The token binding process is mentioned below:

1. The client sends a request to the AS.
2. In response, AS sends a token binding message to the client with a nonce and a token binding type.
3. The client uses nonce and its TLS connection parameters and transmit it to the AS.

4. If successful, the AS generates an access token and send it to the client after validating the TBK.

PSKs are used in the ACE-OAuth framework to authenticate the client and AS during mutual authentication. These symmetric keys are shared between the two parties where digital certificates are not practical.

The ACE-OAuth framework uses Transport Layer Security to ensure secure communication between the client and AS. TLS provides encryption, integrity, and authentication services for secure communication between the client and AS. The TLS protects the messages from eavesdropping, tampering, and replay attacks [35].

Token revocation and expiration are important security features of the ACE-OAuth framework. AS generates access tokens. These tokens have a limited lifespan and are revoked by the AS when they are no longer needed or when the client's authorization is revoked [35]. Hence, the tokens cannot be used by unauthorized users. This minimizes the risk of token theft or misuse.

Thus, the mutual authentication, PSKs, token revocation and expiration, and TLS support in ACE-OAuth helps to establish a standard approach for ensuring security and adaptability in resource-constrained environments.

3.3.1 Implementation guidelines

The guidelines for implementing secure security mechanisms in IoT environments are described below:

1. **Understanding the constraints of the IoT environments**

The protocol CoAP is designed to reduce bandwidth and resource usage compared to HTTP. It is essential to understand the limitations of the IoT environment in which the framework is suitable to deploy including the network bandwidth, and devices type and security requirements.

2. **Profile Selection**

Different profiles in the ACE-OAuth framework are designed to meet specific requirements of IoT deployments such as the ACE-CoAP profile for CoAP base deployments and the ACE-MQTT profile for MQTT-based deployments.

3. **Implementing building blocks**

   CoAP used within the ACE-OAuth framework also supports DTLS for secure communication. The OAuth 2.0 uses access tokens to authorize clients. It provides limited privileges to resources with the permission of the resource owner.

   This separates the user authentication from the access authorization, which improves both security and scalability. IoT devices rely on cryptographic protocols to ratify data privacy and integrity. Therefore, it is crucial to manage the cryptographic keys from generation to storage, rotation, and deletion [35].

4. **Secure Bootstrapping**

   The first step in securing IoT devices is ensuring the credentials. It is essential to test and validate the implementation by securely exchanging keys. Devices must be designed to be tamper-proof and resistant to compromise. IoT devices handle sensitive data. It is essential to implement privacy-preserving mechanisms such as encrypting data to provide additional protection.

5. **Firmware and Software Updates**

   The ability to update device firmware and software helps to ensure security vulnerabilities can be patched and new security features can be added over time. The strengths of the ACE-OAuth framework with CoAP, such as the use of lightweight protocols and token-based authorization, align closely with these best practices. By focusing on both the common principles of cybersecurity and the unique challenges of IoT environments, the framework provides a foundation for stronger and more adaptable security in IoT deployments.

3.3.2 Summary

The ACE-OAuth framework is built upon the strong foundation of OAuth 2.0. It extends the capabilities to address specific requirements of IoT devices through extensions and profiling. The CoAP is integrated into ACE-OAuth which runs above the UDP protocols. The integration minimizes overhead by reducing the number of message exchanges making it a better option in an environment where HTTP is not practical. The ACE-OAuth framework provides compatibility and the capability of handling communications overflow through integration with CoAP.

In addition, its core framework enables the creation of customized profiles for specific security protocols and underlying transports. This feature improves interoperability and ensures that implementations of the same profile can work together effectively. This is particularly important in an environment with a variety of IoT devices, including powerful mobile devices. These devices can support multiple profiles and can easily interact with a wide range of constrained devices.

In conclusion, ACE-OAuth emerges as a standard solution to handle the complex challenges of IoT environments. It addresses resource limitations, accommodates various deployment scenarios, and supports a wide range of authorization scenarios. ACE-OAuth is a comprehensive framework for authentication and authorization in IoT domains. ACE-OAuth prioritizes security and adaptability which makes it a crucial component in IoT security and access control by integrating OAuth 2.0 and CoAP.

3.4 Additional OAuth Parameters for ACE framework

The proof-of-possession (PoP) key enhances security for constrained environments by providing a mechanism for authenticating clients and resource servers and protecting against token theft and replay attacks.

In the OAuth 2.0 flow, a client requests an access token from an authorization server (AS) and then uses that token to access protected resources on a resource server (RS). In a constrained Internet of Things (IoT) device, it is not possible to store a long-lived access token securely. In such scenario, PoP key is used to authenticate to the AS and the RS [36].

3.4.1 Parameters for the Token Endpoint

This section allows for a Proof of Possession key to be included in an access token from a token endpoint in the ACE framework. The AS verifies that the client has the matching key. This mechanism of AS is determined by the profiles used in this specification.

```
Header: POST (Code=0.02)
Uri-Host: "as.example.com"
Uri-Path: "token"
Content-Format: application/ace+cbor
Payload:
{
   / req_cnf / 4 : {
     / COSE_Key / 1 : {
     / kty /  1 : 2 /EC2/,
     / kid /  2 : h'11',
     / crv / -1 : 1 /P-256/,
     / x /   -2 : h'BAC5B11CAD8F99F9C72B05CF4B9E26D24
                  4DC189F745228255A219A86D6A09EFF',
     / y /   -3 : h'20138BF82DC1B6D562BE0FA54AB7804A3
                  A64B6D72CCFED6B6FB6ED28BBFC117E'
      }
   }
 }
```

Figure 26. Access Token Bound Request bound to an asymmetric Key [36]

The above figure 26 shows a request for an access token using the ‘req_cnf’ parameter which is used to ask for a specific public key as a Proof-of-Possession key [36]. The information is in CBOR diagnostic notation.

For token requests, the client generates a PoP key and sends it to the AS. The token is sent to the client by the AS, having chosen the PoP key. The client and RS use the PoP key to authenticate each other and secure against token theft and replay attacks.

```
Header: Created (Code=2.01)
Content-Format: application/ace+cbor
Payload:
{
  / access_token / 1 : h'4A5015DF686428/...
   (remainder of CWT omitted for brevity;
   CWT contains COSE_Key in the "cnf" claim)/',
  / cnf / 8 : {
   / COSE_Key / 1 : {
      / kty / 1 : 4 / Symmetric /,
      / kid / 2 : h'DFD1AA97',
      / k /  -1 : h'849B5786457C1491BE3A76DCEA6C427108'
    }
  }
}
```

Figure 27. AS Response with an Access Token [36].

A main advantage of the PoP key is that it allows authentication of the client without the need for long-term storage of access tokens. This minimizes the chances of token theft and replay attacks, which are common in constrained environments.

This feature enhances the use of a PoP key in providing non-repudiation. Non-repudiation refers to the inability of a party to deny having performed a certain action, such as accessing a protected resource. By using a PoP key, the client and RS can verify the ownership of a

specific key [36]. This helps resolve disputes between the two parties regarding whether a particular action has been performed.

Overall, the PoP key enhances security for constrained environments by providing a mechanism to authenticate the clients and resource servers and protect against attacks. There is a potential security risk of key compromise, but the benefits of enhanced security outweigh the risks.

3.4.2 The potential security risks associated with implementation.

While the changes proposed are designed to enhance security for constrained environments, there are some potential security risks associated with implementing these changes. Some of the key risks include:

1. Key compromise: One of the main risks associated with using proof-of-possession (PoP) keys is the risk of key compromise. If an attacker obtains the PoP key, they can use it to impersonate the client or resource server and gain unauthorized access to protected resources.

2. Replay attacks: Another risk associated with using PoP keys is the risk of replay attacks. If an attacker can intercept a PoP key, they can use it to replay a previous request and gain unauthorized access to protected resources.

3. Denial-of-service attacks: The use of PoP keys can also make it easier for attackers to launch denial-of-service (DoS) attacks. By flooding the AS or RS with requests for PoP keys, an attacker can overwhelm the system and prevent legitimate users from accessing protected resources.

4. Implementation errors: Finally, there is a risk of implementation errors when implementing the changes proposed in article RFC 9201. If the changes are not implemented correctly, they could introduce new security vulnerabilities or weaken existing security measures [36].

Overall, the additional parameters enhance security for constrained environments, it is important to carefully consider the potential security risks associated with implementing these changes and take steps to mitigate these risks. This may include implementing additional security measures, such as rate limiting or intrusion detection systems, to prevent attacks and monitor suspicious activity.

3.4.3 Security standards and practices

The suggested changes align with existing security standards and practices in several ways:

1. The proposed are designed to extend the OAuth 2.0 framework for use in constrained environments. OAuth 2.0 is a widely adopted standard for authorization and authentication, and the changes proposed in RFC 9201 build on this existing framework.

2. CBOR is a compact binary format that is designed to be more efficient than JSON (JavaScript Object Notation) for use in constrained environments. The use of CBOR aligns with best practices for optimizing network performance and reducing bandwidth usage.

3. The use of PoP keys is a best practice for securing access tokens in constrained environments. PoP keys provide a way to authenticate clients and resource servers without requiring the storage of long-lived access tokens, which can be vulnerable to theft or replay attacks.

4. Asymmetric cryptography enables the use of asymmetric cryptography for PoP keys. Asymmetric cryptography is a best practice for securing communications and is widely used in other security protocols, such as SSL/TLS [36].

Thus. these changes are achieved by introducing the use of CBOR, implementing PoP keys for authentication, enabling the use of asymmetric cryptography, and undergoing a thorough review and approval process through the Standards Track on the OAuth 2.0 framework.

3.5 **Datagram Transport Layer Security (DTLS) Profile**

The DTLS profile involves a client and a resource server using the Constrained Application Protocol (CoAP) over DTLS for communication. DTLS is a protocol designed to secure datagram-based communications, such as User Datagram Protocol (UDP). It is lightweight thus making it suitable for resource-constrained devices.

In this profile, specific protocol flows, message formats, and security considerations are outlined for the ACE profile. It uses DTLS version 1.2 or later to ensure secure communication between entities in a constrained network.

This can be achieved using either raw public keys or pre-shared keys. The profile allows a resource-constrained server to delegate the management of authorization information to a trusted host, which may have fewer constraints on processing power and memory [37].

3.5.1 Protocol Flow

The ACE framework ensures mutual authentication between the client and server before any application data is exchanged by using the Constrained Application Protocol (CoAP) over DTLS. The client and server use DTLS to establish a secure communication channel, and then they use CoAP to exchange messages. The following steps are involved in the mutual authentication process:

```
2.01 Created
Content-Format: application/ace+cbor
Max-Age: 3560
Payload:
{
  / access_token / 1 : b64'SlAV32hk'/...
   (remainder of CWT omitted for brevity;
   CWT contains the client's RPK in the cnf claim)/,
  / expires_in /  2 : 3600,
  / rs_cnf /     41 : {
    / COSE_Key /  1 : {
      / kty /  1 : / EC2 /   2,
      / crv / -1 : / P-256 / 1,
      / x /   -2 : h'd7cc072de2205bdc1537/.../',
      / y /   -3 : h'f95e1d4b851a2cc80fff/.../'
    }
  }
}
```

Figure 28. Example of access token response from Authorized server to Client [37].

1. The client retrieves access tokens from the Authorization server using the CoAP protocol. At first, the client sends a request to the AS and the AS responds with an access token for authorization.

2. After receiving the access token, it initiates the DTLS channel setup with the resource server. The DTLS channel allows secure communication between the client and the RS.

3. After the DTLS channel is established, the client can send an authorized request to the RS which includes the access token. The RS verifies the access token to provide access to the resource.

4. The resource server sends the protected resource to the client through the DTLS channel. The protected resource is encrypted and authenticated using the DTLS protocol to assure data integrity.

Figure 29. Overview of DTLS protocol [37]

Figure 29 illustrates the protocol flow from the DTLS profile. The figure shows the role for the client (C), resource server (RS), and authorization server (AS) involved in the protocol flow. The arrows represent the message flow between entities. The access token is retrieved by the client from the AS and is used to establish a secure DTLS channel between the client and RS. The authorized request is sent over the DTLS channel, and the protected resources is sent back to the client over the same channel.

3.5.2 Security standards and practices

Encryption DTLS provides encryption in the ACE framework by encrypting all data sent over the channel, ensuring that it cannot be intercepted by an attacker. Symmetric encryption algorithms such as AES are used during the handshake process. After the encryption keys are established, the data are encrypted and decrypted using those keys.

DTLS uses digital certificates to authenticate the client and RS to verify the authenticity. The client and resource server exchange digital certificates to verify the identities during the handshake process. The certificates are issued by certificate authorities (CA) and contains information about the entity's identity, public key, and other attributes. The sequence numbers and timestamps are included in each message exchange over the channel [37]. The resource server keeps track of them. The messages are received if the timestamps and sequence numbers are not repeating and expected and vice-versa.

This mechanism is used to encrypt and decrypt the data sent over the channel. It is based on the Diffie-Hellman key exchange algorithm that allows the client and RS to generate a shared secret without transmitting it over the channel. The shared secret is used to derive the encryption keys used for the DTLS session [37].

Message authentication codes (MACs) are used to ensure that the data transmitted over the channel are not tampered with. For this, the shared secret is used for MACs during the DTLS handshake process. The MACs are computed using a cryptographic hash function such as HMAC-SHA256 and are included in every message transmitted through the channel. The receiver can verify the integrity of the message by computing the MAC using the shared secret key and comparing it to the MAC included in the message [37].

After the completion of the DTLS handshake, the client and server can securely exchange CoAP messages. The client includes its access token in the CoAP message to prove its authorization for accessing protected resources hosted by the server [37]. The server verifies the access token and responds with the requested resource or an error message if the access token is invalid.

3.5.3 Potential risks and Countermeasures

The ACE framework should be aware of several potential risks or attacks, including Denial of Service (DoS) attacks, replay attacks, and man-in-the-middle attacks. The resource-constrained devices that use DTLS are vulnerable to DoS attacks because the handshake protocol requires creating an internal state within the device. This vulnerability is concerning if an attacker intercepts the initial cookie exchange and injects malicious messages with a valid cookie to proceed with the handshake.

Similarly, the unprotected authorization information endpoint on the resource server is susceptible to attacks, as attackers could flood the constrained resource server's internal storage with intercepted or retrieved valid access tokens. To mitigate this, the resource server should establish a time limit for unused access tokens, after which they will be deleted.[38]

Replay attacks can be prevented by incorporating sequence numbers and timestamps into each message transmitted over the channel. These measures add an extra layer of security against unauthorized replay attempts.

DTLS is susceptible to man-in-the-middle attacks (MitM). In such attacks, an attacker intercepts and manipulates the DTLS messages exchanged between the client and server. Such attacks can be prevented by implementing digital certificates for server and client authentication and utilizing pre-shared keys or raw public keys for client authentication.

A critical aspect of DTLS security is key management. It is essential to store and distribute the cryptographic keys securely to ensure secure communication between the client and server. Weak keys or compromised keys lead to significant security vulnerabilities [38]. This underscores the importance of strong key management practices.

3.5.4 Countermeasures

To mitigate potential risks such as DoS attacks, the resource server should set a time limit for access tokens. If these tokens are not used for an extended period, they should be deleted. This precaution prevents attackers from flooding the server's internal storage with intercepted or retrieved valid access tokens. Furthermore, the ACE framework protects against man-in-the-middle attacks by using digital certificates for client and resource server authentication during the DTLS handshake. These certificates are issued by trusted Certificate Authorities (CAs) which verify the identities of the entities involved and establish secure communication [38].

In addition, secure key management practices are crucial for generating, storing, and distributing cryptographic keys within the ACE framework. These keys must be securely generated and shared only with authorized parties. The rate limiting and throttling mechanisms can be implemented within the ACE framework to prevent DDoS attacks. These techniques regulate the frequency and volume of client requests, providing an additional layer of protection [38].

In conclusion, the ACE framework utilizes secure key management, digital certificates, and pre-shared keys and implements rate limit and throttle to mitigate potential vulnerabilities of DTLS. These measures ensure the security and integrity of communication in a constrained network between the client and server.

3.6 Object Security for Constrained RESTful Environments (OSCORE) Profile

The goal of OSCORE is to provide confidentiality, integrity, and protection against replay attacks for messages exchanged between devices in constrained environments for RESTful interactions. OSCORE utilizes CBOR Object Signing and Encryption (COSE) to secure CoAP messages. It establishes a secure binding between requests and responses to prevent tampering or interception during transmission.

OSCORE is designed for devices with limited processing power, memory, and energy resources. Its lightweight cryptographic algorithms minimize message size between devices. Additionally, OSCORE comprises a proof-of-possession mechanism that ensure only authorized devices get access to the specified resource [39].

3.6.1 Protocol Flow

The OSCORE profile establishes secure communication with a set of guidelines and mechanisms that use OSCORE and proof-of-possession for a key obtained by the client and bound to an OAuth 2.0 access token. It is designed for devices with limited processing power, memory, and energy resources.

The OSCORE profile utilizes OSCORE to provide end-to-end security for RESTful communication between a client and a resource server. It uses CBOR Object Signing and Encryption (COSE) to secure CoAP messages exchanged between the client and the resource server. Additionally, the profile includes a proof-of-possession mechanism that guarantees only authorized devices can access the resources they are permitted to access [39].

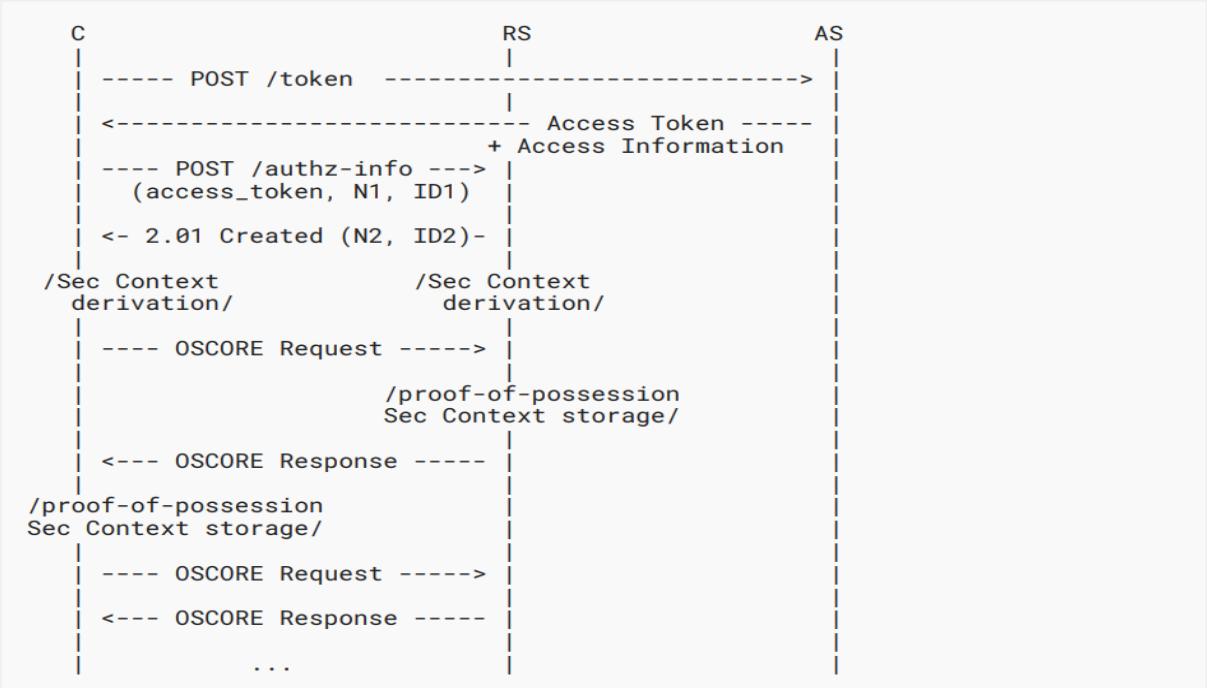

Figure 30. OSCORE Protocol Overview [39]

The protocol uses the ACE framework and OSCORE in the following steps:

- The client sends a POST request to the Authorization Server (AS) for an access token.
- The AS responds by providing the client with an access token and access information.
- The client then sends a POST request to the RS, including the access token, a nonce (N1), and an identifier (ID1).
- The RS verifies the access token and responds with a message, including a nonce (N2) and an identifier (ID2).[4]
- The client and RS establish a secure context using OSCORE. This includes keying material from the Proof of Possession (PoP) key and relevant parameters.
- The client sends a request to the RS using OSCORE, including the PoP key as proof of possession.

3.6.2 Security Mechanism

The OSCORE strengthens security by establishing secure channel for communications between a client and a resource server in a RESTful setup. All the messages shared between them are authenticated to prevent disruption during communication.

OSCORE integrates the access token with a proof-of-possession key to ensure that only devices with proper authorization can get to the resources. The client uses this key to validate its access to the resource server, and then the resource server verifies the key to confirm the client's authorization for the requested resources.

Additionally, OSCORE uses CBOR Object Signing and Encryption (COSE) to secure the messages. CoAP exchanges between the client and the resource server. COSE checks the integrity of message and ensures confidentiality, integrity, and protection against replay attacks.

In the OSCORE profile, the Authorization Server (AS), client, and Resource Server (RS) interact. The client sends a POST request to the AS asking for an access token. The AS then replies with an access token and important access info. The client sends another POST request to the RS, packing in the access token, a nonce (N1), and an identifier (ID1). The RS checks the access token and returns with a message, holding a nonce (N2) and an identifier (ID2).

The client and RS get a secure configuration going through OSCORE. This involves getting keying material from the PoP key and other key parameters. The client then sends a request to the RS using OSCORE, tagging along the PoP key as proof of possession. The RS checks the PoP key and sends a message back through OSCORE.

The client and RS continue the secure message exchange via OSCORE, using the PoP key as proof of possession for each message. The communication in the OSCORE profile is configured to provide end-to-end security and validate possession [39]. This guarantees that the authorized devices can reach the assigned resources.

4 Implementation of Ace-OAuth: Smart Home and Factory use case

To conduct a systematic review a thorough analysis of security aspects of the ACE OAuth framework and a comparative assessment with other profiles and additional parameters were conducted. For this process, various research papers and published articles were analysed and assessed to further deepen the understanding of the ACE-OAuth framework.

4.1 ACE-OAuth in Smart Home

The ACE-OAuth can play a vital role for the authentication and authorization of devices such as smart thermostats, smart lights, and smart locks in a smart home. These devices require access to the resources either from cloud storage services or from other devices. For this purpose, ACE-OAuth provides the necessary credentials and permissions. It enables house owners to control and manage devices, appliances, and systems remotely and securely [40]. Traditional OAuth 2.0 is not suitable for resource constrained environment of smart homes as they require heavy bandwidth and computational power.

ACE-OAuth addressed the limitations of OAuth 2.0 in IoT environments by utilizing simple and efficient lightweight protocol designed for resource-constrained environments. ACE-OAuth use the CoAP for message exchange which is lightweight application layer protocol that is suitable for lightweight IoT devices.

In addition to its lightweight design, ACE-OAuth protects sensitive data and prevent unauthorized access utilizing CBOR (light version of JSON). CBOR-based secure message format is used to secure PoP token. It is used by users to provide possession of secret to resource server to access the resource. It can accommodate multiple devices and resources. In the smart home there could be growing number of smart devices and applications [40].

For example, smart home consists of smart doors, locks, and sensors. The administrator controls the devices and could share among different visitors that requires dynamic access control policies. ACE-OAuth enables flexibility and scalability without compromising security and performance.

In addition to scalability, ACE-OAuth provides self-sovereignty, utilizing decentralized identifier (DID) and verifiable credentials (VC) to protect user privacy and IoT devices. DID provides a secure and encrypted communication channel while VC provides digital verification of identities of users [41].

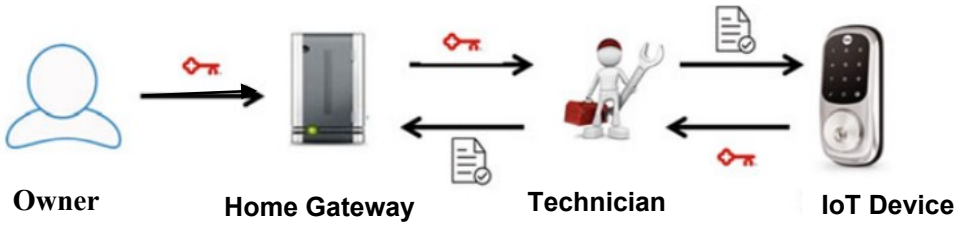

Figure 31. Implementation of ACE-OAuth in smart home [41]

Figure 31 illustrates a scenario of the implementation of ACE-OAuth-based delegation method in smart home to authorize technician to access home door or devices. This method utilizes decentralized identifiers and pre-shared key to establish trust between technicians and IoT devices. The IoT device is registered with authorization server with unique identifier and a pre-shared key. The technician initiates the authorization process by sending a request to AS. The AS verifies the technician's identity and credentials and grants access token for the door lock or device.

The technician then uses PoP access token to interact with the IoT device with further enhance security by limiting the technician's right to access the devices. This eliminates the need for traditional credential management and ensure mutual identity between technicians and IoT devices [41]. This case study focuses on how the Ace-OAuth framework enhances security for smart homes and its role in protecting smart home devices from unauthorized access.

4.1.1 Security of ACE-OAuth

ACE-OAuth has a strong authentication method. It uses Proof-of-Possession tokens to authenticate the devices. These tokens verify whether the devices could be provided the privilege to access the resource or not. ACE-OAuth using the CoAP protocol for communication. CoAP provides encryption and integrity protection that ensures secure communication for constrained IoT. This feature allows users to have complete control over authorization. It enables users to manage privileges and determine device access to specific resources within the smart home ecosystem.

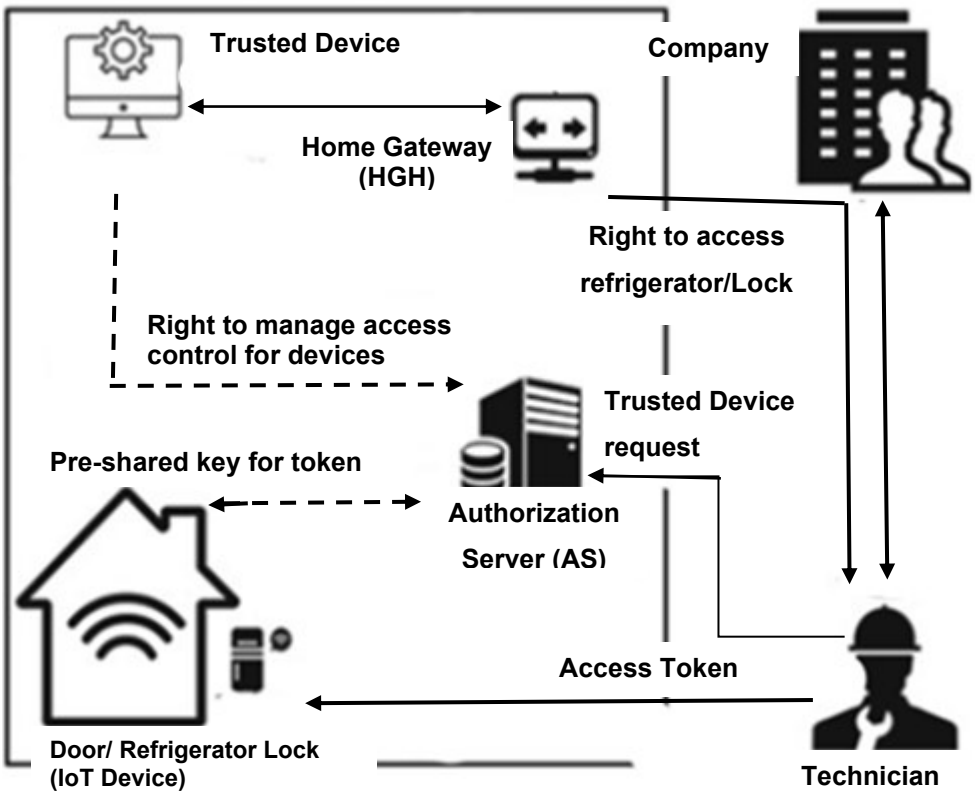

Figure 32. ACE-OAuth Authentication and Authorization Mechanism in smart home [41]

Figure 32. illustrates the authentication and authorization Mechanism in ACE-OAuth. In ACE-OAuth, when a device in a smart home wants to access a resource from a resource server, the server requires authentication. The device provides a Proof of Possession (PoP) token as a response. The resource server verifies the PoP token, and if it is valid, the authentication is successful. The resource server then allows access to the requested resources [41].

In the case of authorization, there are Role-based access control (RBAC), Attribute-based access control (ABAC), and Time-based access control (TBAC) security mechanisms implemented by the ACE-OAuth framework. The RBAC grants authority to users to assign roles to the devices. Each device possesses a predefined set of permissions to access the resources [41]. ABAC enables users to assign rules and regulations regarding access based on devices and user attributes.

The TBAC security mechanism is implemented to establish a rule for accessing the devices according to time parameters. For instance, a rule could be established that a device can only access the home's lock between certain periods in a day [42]. This mechanism is commonly

used in several buildings in which the main door is locked after a fixed time and can only be opened with a physical key.

The multi-layered access control authorization mechanism ensures that resources are only accessible to authorized devices. It makes it difficult for unauthorized devices to gain access to resources in the home. It also provides full authorization control, so that device owners can control that devices have access to specific resources [42].

4.2 Possible threats and attack and ACE-OAuth mitigation scenario

The table 3 includes different threats and attacks that could occur in smart home. It also includes the security approach and mechanisms adopted by ACE-OAuth to prevent and restraint such attacks.

Attacks / Threats	Smart Homes	Ace-OAuth Approach
Unauthorized Access	Smart IoT devices such as smart locks, cameras, and other devices withing smart home can be accessed without authorization to steal data and use for unauthorized purposes.	It utilizes challenges-response authentication for device verification addressing the limitations of OAuth 2.0 in resource-constrained environments.
Man-in-the-Middle Attacks	Attacker can disrupt communications between smart devices and servers and intercept data.	Utilizes encryption to protect communications between devices and server for secure communication.
Denial-of-Service Attacks	IoT devices are flooded with traffic causing operation to halt and unavailable to users.	Employs token-based access to limit device access periods to restrain the impact of compromised tokens.
Malware	Malware can be installed on smart devices to steal data, disrupt operation and for stalking.	Implements a secure protocol for authentication and authorization to protect device from malware.
Physical Attacks	Smart devices can be tampered and stolen.	Implements physical access control such as alert administrator, record log, revoke token, report suspicious activities and unknown location.

Table 3. ACE-OAuth approach to mitigate attacks in smart home.

The table below demonstrates the security mechanisms and advanced features implemented by ACE-OAuth to further enhance access management in Smart home.

Security Mechanisms	ACE-OAuth Authentication and Authorization Mechanism in Smart Home
Authentication	Devices uses a Proof-of-Possessions token to ensure only authorized devices gain access.
Communication	The CoAP protocol with encryption and integrity protection ensures secure communication for constrained IoT and grants user complete control.
Role-based Access Control	Grants authority to users to assign roles to devices with predefined permissions
Attribute-Based Access Control	Enables user to assign rules based on devices and user attributes.
Time-Based Access Control	Establishes rules for devices access-based on time parameters to enhance security in time-sensitive environments.

Table 4. ACE-OAuth security mechanisms for smart home

4.3 ACE-OAuth in Smart Factory

Smart factories use advanced technologies to enable smooth operations through seamless device interaction. Different devices with different capabilities and resources pose a significant obstacle to maintaining a secure authentication and authorization of the devices. ACE-OAuth provides a framework for strong access control that is adapted to the needs of different IoT devices to handle this. For example, if a sensor needs to store temperature data in the factory's cloud storage. ACE-OAuth plays a vital role in authenticating the sensor and granting it the necessary permissions.

Similarly, an actuator might require authentication to be controlled by the factory's production planning software. ACE-OAuth ensures that only authorized devices can interact with critical resources, bolstering overall security as shown in figure.

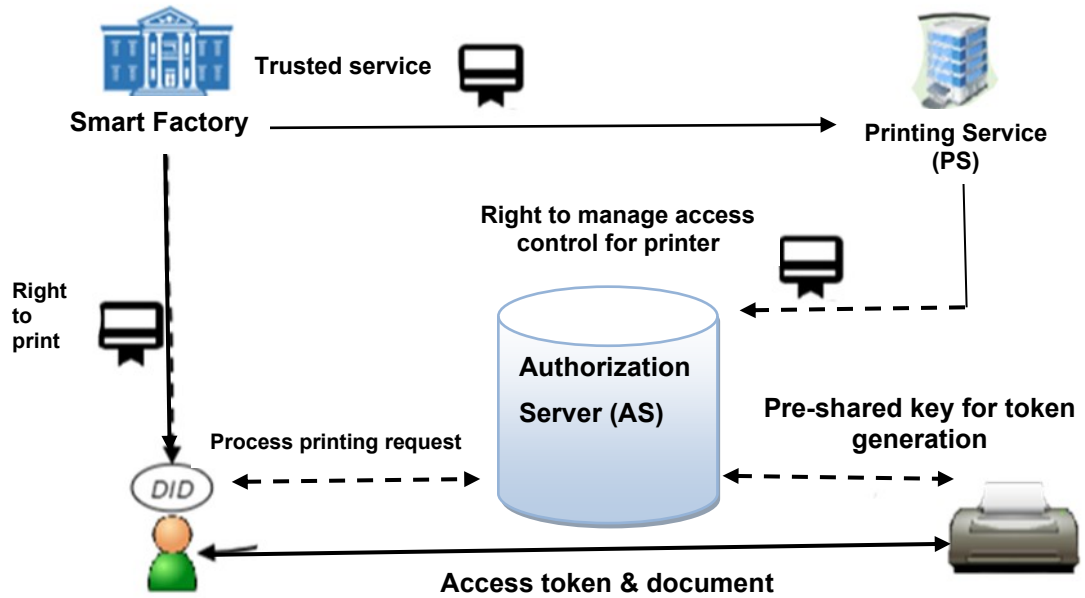

Figure 33. Access management in smart factory [45]

Figure 33 demonstrates ACE-OAuth profile for a scenario where the employee requires to use a IoT device (printer) within smart factory system. The employee using a computer proposes the use of printer and requests AS to issue proofs indicating trust and authorization. The AS uses the credentials to create proofs. Further AS requests the employee to provide proof to the right to access the printer. After the verification of proofs, the employee sends proof of right to access the printer within smart factory system. The AS verifies the proof and issues a Proof-of-possession (PoP) access token to the employee. The employee applies the PoP access token to access to the printer [45].

4.3.1 Security mechanisms

The ACE-OAuth framework uses a challenge-response mechanism to authenticate devices. This means that devices must prove their identity to the authorization server before they can access protected resources.

1. **Authentication**

The challenge-response mechanism works as follows:

- At first, the device requests an access token from the authorization server.

- The authorization server challenges the device with a random string.
- The device receives the challenges and encrypts it with its private key and sends the encrypted challenge back to the authorization server.
- The authorization server decrypts the response and verifies it with the original challenge. If the challenges match, the authorization server provides an access token to the device.

2. **Authorization**

After a successful authentication, the authorization process takes place. In the ACE-OAuth framework, scopes play a vital role in granting authorization to devices to access protected resources. Scopes refers to the level of privileges a user or client possesses to a particular resource or account. It empowers the factory to regulate what devices can access specific resources.

For example, the factory can create a scope that allows devices to read sensor data. The factory might then grant this scope to all the sensor devices in the factory. A device must include the scope to request a resource. The authorization server then verifies whether the device has the requested scope. The authorization server authorizes access to the device if the device possesses the specified scope [45].

3. **Token-based access**

The ACE-OAuth framework uses access tokens to grant devices access to protected resources. Access tokens expire after a certain period. This helps to protect the factory's resources from unauthorized access. If an access token is compromised, the factory can revoke the access token. This will disable access to protected resources for the compromised device.[9]

4. **Revocation**

The ACE-OAuth framework allows the factory to revoke access tokens. The factory can immediately disable access to protected resources if a device is compromised. To revoke an access token, the factory sends a request to the authorization server. The authorization server will then invalidate the access token.

5. **Auditing**

The ACE-OAuth framework provides an audit trail of all access requests. This allows the factory to track which devices are accessing protected resources and when. The audit trail can be used to investigate security incidents and to identify unauthorized access to protected resources [45].

The ACE-OAuth framework provides several security benefits for smart factories. By using the ACE-OAuth framework, smart factories can protect their resources from unauthorized access and improve their overall security posture.

4.4 Possible threats in Smart Factory

The smart factories are adopting new advanced technologies to improve efficiency, productivity, and flexibility. The interconnectivity introduces new issues and challenges making in vulnerable to attacks and threats. These attacks include the unauthorized access, denial-of-service attacks, man-in-middle attacks, malware infections and physical attacks. The ACE-OAuth framework implements various approaches to cater the unique requirement of industrial environments which are mentioned in the table below.

Attacks / Threats	Smart Factory Approach	ACE-OAuth Approach
Unauthorized Access	Unauthorized access to industrial sensors, systems, and devices.	Implements strong authentication methods that ensures only authorized devices can access critical industrial resources.
Denial-of-Service Attacks	Disruption of operation due to traffic overflow.	Utilizes short-lived tokens for access, minimizing window for unauthorized access to critical industrial resources.
Man-in-the middle Attacks	Interception of communication between devices and servers	Utilizes encryption to secure communication and prevent unauthorized access of data in industrial networks.
Malware	Malware can be installed through various ways and could be used for malicious activities.	Utilizes a secure protocol for authentication and authorization to protect devices from malware and secure industrial operations.
Physical Attacks	Physically tampering of devices	Integrate physical access controls and entry points by integrating surveillance systems.

Table 5. ACE-OAuth approach to prevent attacks and threats in Smart Factory

The ACE-OAuth approach provides a strong framework to address diverse security challenges and protect industrial resources. The ACE-OAuth access management mechanisms in smart factory are mentioned in the table below.

Security Mechanisms	**ACE-OAuth Access Management in Smart Factory**
Authentication	Utilizes mechanisms such as device registration, access request, challenge generation, response generation and validation an access control enforcement.
Authorization	Scopes play a vital role that grants predefined authority to devices that enables factory to regulate resource access.
Token-Based Access	Short-lived access token and implement token revoke mechanism in case of suspicion.
Auditing	Provides an audit trail of access requests that helps to track device access to resources and examine security incidents.

Table 6. ACE-OAuth security mechanisms for Smart Factory

Smart factories are increasingly vulnerable to cyber-attacks due to their interconnectivity and dependencies on automation. These attacks could be data breaches to physical tampering of devices. ACE-OAuth Access Management provides a secure framework to authenticate authorize and control access to resources while mitigating the threats as shown in the above table 6.

5 Key Findings: Security implications and strength and weaknesses of ACE-OAuth Profiles

The ACE-OAuth framework is customized to overcome the challenges raised by resource-constrained IoT environments. The comparison of security attributes with conventional authentication and authorization methods of the ACE-OAuth framework are described below.

The traditional systems rely on one-way username-password exchanges. ACE-OAuth mandates mutual authentication between the client and the server that helps to boost security. In OAuth 2.0, bearer tokens pose a risk as anyone in possession can use them. ACE-OAuth employs proof-of-possession tokens that require the client to demonstrate control of a cryptographic key which adds an extra layer of security.

The ACE-OAuth allows tokens to be bound to a key, making it challenging for an attacker to exploit them. The attacker needs both the token and the key. Traditional methods need a public-key infrastructure, often impractical in resource-constrained IoT setups. ACE-OAuth provides symmetric cryptography with pre-shared keys for constrained-resource devices.

ACE-OAuth adeptly addresses scenarios with limited user interfaces. Traditional solutions require user interaction for access authorization, which is unfeasible in the IoT context. ACE-OAuth provides a flexible, lightweight, and secure solution tailored to meet the specific demands and limitations of IoT-constrained environments. The security level depends on how the ACE-OAuth framework is utilized and implemented in practice.

5.1 Security Considerations

The ACE OAuth framework incorporates security features and mechanisms to enhance communication in restricted environments. The security consideration for lightweight IoT is described in the following section:

1. **Bidirectional Verification**

The framework comprises both the client and the Authorization Server to verify each other's identities. This verification process occurs through the exchange of credentials and configuration parameters. It occurs when both the client and RS are registered with the AS.

2. **Access Token Protection**

   The AS provides an endpoint where access token requests can be submitted. These access tokens serve as proof of authorization granting clients access to protected resources on the RS. The framework empowers the AS to enhance the functionality of this endpoint thereby facilitating key sharing or public key exchanges between clients and RS.

3. **Securing Communication channel**

   The AS offers an endpoint for submitting requests for access tokens. These access tokens act as evidence of authorization enabling clients to reach protected resources on the RS. The framework allows the AS to improve how this token endpoint works making it easier, for clients and RS to establish shared keys or exchange keys.

4. **Granting credential**

   In the ACE framework, the AS endows credentials and related information to facilitate mutual authentication between the client and the RS. This is crucial, as it cannot be assumed that the client and the RS are part of a shared key infrastructure in constrained environments.

5. **Managing Profiles**

   The AS oversees the alignment of compatible profile choices between a client and an RS, based on the assumptions defined by various deployment settings. This supports various situations encountered in constrained environments. The goal of new parameters is to enhance security for constrained environments by enabling the utilization of proof-of-possession keys. These keys play a vital role in authenticating clients and resource servers and preventing token theft and replay attacks.

5.1.1 Security Implications of Additional Parameters

The PoP key is generated by the client and transmitted to the authorization server (AS) as part of the token request. The AS then selects a PoP key and dispatches it back to the client in the token response. The client and resource server subsequently employ the PoP key to authenticate each other and fortify against token theft and replay attacks.

The importance of security implications of the new parameters is the facilitation of asymmetric keys for Pop. Asymmetric keys provide enhanced security for brute-force attacks and can be implemented for non-repudiation.

The CBOR for data encoding helps in reducing the size of messages transmitted over constrained networks. Hence, the introduction of new parameters enhances security for constrained environments by executing PoP keys and CBOR encoding.

5.1.2 Security Implications of DTLS Profile

The security considerations regarding the DTLS profile for authentication and authorization in constrained environments are discussed in the Security Considerations section. The key security implications are mentioned below:

1. **Effective Key Management**

   DTLS requires the precise handling of cryptographic keys to guarantee secure communication between the client and server. The authorization server validates the key provided to the resource server is correctly connected to the client.

2. **Authentication Integrity**

   Mutual authentication is established in the ACE framework before the exchange of application data. DTLS enables mutual authentication by allowing the client and server to validate the pre-shared key during the DTLS handshake.

3. **Authorization Validation**

   The client acquires an access token from the authorization server to substantiate its authorization to access protected resources hosted by the resource server. The authorization server ensures the access token is connected to a PoP key and cryptography key is associated with the client.

4. **Denial of Service (DoS) Attacks:**

   DTLS is vulnerable to DoS attacks, such as flooding attacks and resource exhaustion attacks. The rate limiting and throttling should be implemented to restrain such attacks.

5. **Implementation and Configuration challenges**

   The system's safety depends on how DTLS is set up and the configuration of its codes. It is crucial to follow secure programming practices, routinely search for weaknesses,

and carry out thorough penetration testing for added security. This will help to address and resolve the possible gaps promptly. within a constrained environment.

5.1.3 Security Implications of OSCORE Profile

The OSCORE profile leverages the ACE framework and communication security and proof-of-possession for a key owned by the client. This key is bound to an OAuth 2.0 access token. The following is a step-by-step description of the security mechanisms provided by OSCORE:

1. **Authentication and Authorization**

   The client initiates a request for an access token from the Authorization Server (AS) using the ACE framework. The AS then authenticates the client and authorizes the client's access to protected resources on the Resource Server (RS) by issuing an OAuth 2.0 access token.

2. **OSCORE establishment**

   The client and RS use OSCORE to establish a shared security context. This process includes the exchange of a series of messages that serve to establish the security context that determines the algorithms and keys to implement for encryption and authentication.

3. **Confirmation of Ownership**

   The client and RS are involved in a verification process to ensure that they have established the same security context. This involves conducting a proof-of-possession check. This step verifies that the OSCORE request and response successfully pass verification and that the RS authentication is valid.

4. **OSCORE Message Transmission**

   The client dispatches a request to the RS using OSCORE. The request is encrypted and authenticated using the OSCORE AEAD algorithm. This ensures confidentiality and integrity protection for the payload of the message. The RS responds to the client also using OSCORE. This response is likewise encrypted and authenticated using the OSCORE AEAD algorithm.

5. **Non-Repudiation Assurance**

   The key obtained by the client is permanently linked to the OAuth 2.0 access token. This linkage ensures that the client cannot deny sending a particular message, providing a strong foundation for non-repudiation of the message exchange.

   The security mechanisms integrated into this profile comprise the proof of possession, communication security, authorization, confidentiality, integrity, and non-repudiation. These mechanisms collaborate seamlessly to guarantee secure communication between a client and a resource server within a constrained environment.

5.2 Strengths and Weaknesses

OAuth 2.0 in Constrained Environment

Strengths:

- It is based on OAuth 2.0 which provides a strong foundation that allows seamless integration in constrained environments.
- It is adaptable to customization for specific IoT cases by utilizing different grant scopes.
- It supports diverse authentication methods like username/password, client certificates, and tokens that can be used in various IoT deployments.
- The framework is designed to handle large-scale IoT deployments and is highly scalable.

Weaknesses:

- The implementation of OAuth 2.0 involves complex framework components that require a comprehensive understanding and might face challenges during implementation.

- IoT devices with limited processing power, memory, and energy might encounter difficulties while implementing OAuth 2.0.
- It is susceptible to security risks including token theft, token replay attacks, and man-in-the-middle attacks; requires robust security measures.

Additional OAuth Parameters in Authentication and Authorization

Strengths:

- It enhances security in authentication and authorization processes for constrained devices and systems.
- It facilitates smooth operability and interaction between systems.
- It provides greater flexibility in authentication and authorization processes within lightweight environments by introducing additional parameters.

Weaknesses:

- Implementing additional parameters and claims can add complexity to the authentication and authorization process for developers.
- The utilization of the additional parameters is not widely adopted which limits the applicability.

DTLS Profile

Strengths:

- DTLS ensures the security of data transmitted over the channel and prevents unauthorized manipulation.
- It allows the delegation of client authentication and authorization which reduces the server processing load and increase the scalability.
- It provides a versatile key management option by supporting raw public keys and pre-shared keys.

Weaknesses:

- The devices implementing DTLS might be susceptible to Denial-of-Service attacks due to the need for internal state creation during the handshake protocol.
- The reliance on digital certificates might add complexity and resource intensity.
- The suitability of the software may depend on the processing power and memory capacity of the device, which could make it unsuitable for certain types of devices.

OSCORE Profile in Authentication and Authorization

Strengths:

- The implementation of OSCORE ensures the integrity and confidentiality of data transmissions and restrains data tampering.
- It implements a strong proof-of-possession mechanism that enforces strict authorization protocols to ensure only authorized device gets the access privilege.
- It utilizes a symmetric key-based access token that improves authorization security.

Weaknesses:

- It is vulnerable to breaches if the key is compromised, leading to unauthorized access to resources.
- The introduction of the PoP key increases the complexity and requires cautious implementation to prevent vulnerabilities.
- The effectiveness of OSCORE might decrease in traditional computing environments with fewer resource constraints that limit its applicability.

6 Conclusion and Future Recommendations

The review of authentication and authorization mechanisms in constrained IoT environments is focused on the ACE-OAuth framework that is based on the OAuth 2.0 framework.

The thesis provides a detailed analysis of authentication and authorization methods and evaluates the strengths and weaknesses of Pre-Shared Keys, Public Key Infrastructure, Lightweight PKI, access control mechanisms, OAuth 2.0, CoAP, and DTLS.

The objective of the thesis is to conduct a comparative analysis of the ACE-OAuth framework in contrast to its various profiles and traditional OAuth 2.0 framework. ACE-OAuth includes features such as mutual authentication, proof of possession, token binding, and support for pre-shared keys. These features simplify the challenges faced by devices with limited user interfaces.

Further, the security implications arising from the ACE-OAuth framework with CoAP and DTLS are analysed. It emphasizes the necessity of robust key management, secure authentication, and comprehensive authorization protocols. The security risks also involve brute-force attacks, key compromise, token theft, replay attacks, and denial-of-service threats. Thus, it highlights the importance of implementing strong countermeasures and proactive security measures. This paper emphasizes the importance of implementing strong countermeasures to restrain these risks.

Incorporating new parameters and encodings enhanced the security mechanisms of ACE-OAuth framework for authentication and authorization in lightweight IoT devices. These advancements enable the utilization of proof-of-possession (PoP) keys, fortifying authentication and providing a robust defence against token-related breaches and replay attacks. Additionally, the adoption of asymmetric keys and streamlined use of Concise Binary Object Representation (CBOR) for data encoding further enhance security layers.

This thesis analyses authentication and authorization mechanisms designed for constrained IoT environments. It evaluates the ACE-OAuth framework and highlights how it improves IoT security over conventional OAuth 2.0. These findings significantly contribute to the advancement of security practices in lightweight IoT environments.

Traditional system uses a one-way username-password mechanism from the client to the server. The ACE-OAuth requires mutual authentication to establish trust between the client and the server. This enhanced security boosts the strength of the framework.

As the scope and complexity of IoT networks grow, there has been an increasing demand for scalable authentication and authorization methods. Thus, future investigations into reducing the computational burden on energy-constrained nodes might be considered as a prospective research direction.

Future research should consider how secure operations can be facilitated across a heterogenous IoT ecosystem where several different IoT devices are designed by various manufacturers using different communication protocols. It involves incorporation and thorough adjustment of procedures such as ACE-OAuth.

Since the IoT devices are physically accessible they are vulnerable to tampering and direct access attacks. Research should focus on hardware security and tamper-resistant design. It is also important to address the challenges of irregular connectivity in IoT devices while maintaining security. Therefore, it is essential to conduct research on reliable real-time threat detection mechanisms and integration with ACE-OAuth in the context of IoT systems operation. As these systems deal with an enormous amount of data flow.

The integration of edge computing devices with smaller IoT devices in IoT architecture poses additional security challenges for multi-tier systems, especially those involving resource-constrained devices needs additional research. The data collected and processed through IoT devices should be secured for data privacy. Therefore, future studies may focus on strengthening the privacy aspect in current IoT security frameworks.

References

[1] Agrawal, M., Zhou, J., & Chang, D. (2019). A survey on lightweight authenticated encryption and challenges for securing industrial IoT. Security and Privacy Trends in the Industrial Internet of Things, 71-94.

[2] Bertin, E., Hussein, D., Sengul, C., & Frey, V. (2019). Access control on the Internet of Things: a survey of existing approaches and open research questions. Annals of telecommunications, 74, 375-388.

[3] Höglund, J., Khurshid, A., & Raza, S. (2023). AC-SIF: ACE Access Control for Standardized Secure IoT Firmware Updates. In Public Key Infrastructure and Its Applications for Resource-Constrained IoT ,54, pp. 54-62.

[3] Frustaci, M., Pace, P., & Aloi, G. (2017, September). Securing the IoT world: Issues and perspectives. In 2017 IEEE Conference on Standards for Communications and Networking (CSCN) (pp. 246-251). IEEE.

[4] Thakor, V. A., Razzaque, M. A., & Khandaker, M. R. (2021). Lightweight cryptography algorithms for resource constrained IoT devices: A review, comparison, and research opportunities. IEEE Access, 9, 28177-28193.

[5] Thilagam, K., Beno, A., Lakshmi, M. V., Wilfred, C. B., George, S. M., Karthikeyan, M., Vijaykumar,P, Tamesh,C., & Karunakaran, P. (2022). Secure IoT Healthcare Architecture with Deep Learning-Based Access Control System. Journal of Nanomaterials, 2022.

[6] Ngan, M. (2021) Modern token authentication in node with express, Okta Developer. Available at: https://developer.okta.com/blog/2019/02/14/modern-token-authentication-in-node-with-express (Accessed: 25 November 2023).

[7] Chanda, S., Luhach, A. K., Alnumay, W., Sengupta, I., & Roy, D. S. (2022). A lightweight device-level Public Key Infrastructure with DRAM based Physical Unclonable Function (PUF) for secure cyber physical systems. Computer Communications, 190, 87-98.

[8] Aivaliotis, V., Tsantikidou, K., & Sklavos, N. (2022). IoT-Based Multi-Sensor Healthcare Architectures and a Lightweight-Based Privacy Scheme. Sensors, 22(11), 4269.

[9] Ragothaman, K., Wang, Y., Rimal, B., & Lawrence, M. (2023). Access control for IoT: A survey of existing research, dynamic policies, and future directions. Sensors, 23(4), 1805.

[10] Fadhel, A. B., Bianculli, D., & Briand, L. (2015). A comprehensive modelling framework for role-based access control policies. Journal of Systems and Software, 107, 110-126.

[11] de Carvalho Junior, M. A., & Bandiera-Paiva, P. (2018). Health information system role-based access control current security trends and challenges. Journal of Healthcare Engineering, 20(1), 1-15.

[12] Wu, L., & Du, J. (2024). Designing novel proxy-based access control scheme for implantable medical devices. Computer Standards & Interfaces, 87, 103754.

[13] Mounika, A., Babu, C. S., Babu, B. A., Kumar, U. L., & Kumar, K. J. (2023). Android Mobile Devices Context Based Access Control Systems. International Journal of Innovative Research in Engineering & Management, 10(2), 125-129.

[14] Xiao, S., Ye, Y., Kanwal, N., Newe, T., & Lee, B. (2022). SoK: Context and risk aware access control for zero trust systems. Security and Communication Networks, 2022, 1-15.

[15] Huda, M. N., Sonehara, N., & Yamada, S. (2009). A privacy management architecture for patient-controlled personal health record system. Journal of Engineering Science and Technology, 4(2), 154-170.

[16] Zouina, M., & Outtai, B. (2019, April). Towards a distributed token-based payment system using blockchain technology. In 2019 international conference on advanced communication technologies and networking (commnet) (pp. 1-10). IEEE.

[17] Cai, T., Cai, H. J., Wang, H., Cheng, X., & Wang, L. (2019). Analysis of blockchain system with token-based bookkeeping method. IEEE Access, 7, 50823-50832.

[18] Triartono, Z., & Negara, R. M. (2019, September). Implementation of Role-Based Access Control on OAuth 2.0 as Authentication and Authorization System. In 2019 6th international conference on electrical engineering, computer science and informatics (EECSI) (pp. 259-263). IEEE.

[19] Ismail, L., & Buyya, R. (2023). Metaverse: A Vision, Architectural Elements, and Future Directions for Scalable and Realtime Virtual Worlds. arXiv preprint arXiv:2308.10559. Available at: https://arxiv.org/abs/2308.10559 (Accessed: 17 November 2023).

[20] Zhang, Q., Du, J., Zheng, P., Zhang, L., Zhang, Y., Xu, J., Wei, Z., & Chen, X. (2022, November). Blockchain and Distributed Ledger Technology Standardization in ITU-T: Diversification, Globalization and Future. In 2022 5th International Conference on Hot Information-Centric Networking (HotICN) (pp. 48-54). IEEE.

[21] Laghari, A. A., Khan, A. A., Alkanhel, R., Elmannai, H., & Bourouis, S. (2023). Lightweight-biov: blockchain distributed ledger technology (bdlt) for internet of vehicles (iovs). Electronics, 12(3), 677.

[22] Carelli, A., Palmieri, A., Vilei, A., Castanier, F., & Vesco, A. (2022). Enabling secure data exchange through the iota tangle for iot constrained devices. Sensors, 22(4), 1384.

[23] Stafford, V. A. (2020). Zero trust architecture. NIST special publication, 800, 207.

[24] Staff, I. T. (2023, October 23). Implementing a Zero trust security model at Microsoft. Inside Track Blog. https://www.microsoft.com/insidetrack/blog/implementing-a-zero-trust-security-model-at-microsoft/

[25] Malik, A. A., Anwar, H., & Shibli, M. A. (2015, December). Federated identity management (FIM): Challenges and opportunities. In 2015 Conference on Information Assurance and Cyber Security (CIACS) (pp. 75-82). IEEE.

[26] Xiao, L., Wan, X., Lu, X., Zhang, Y., & Wu, D. (2018). IoT security techniques based on machine learning: How do IoT devices use AI to enhance security? IEEE Signal Processing Magazine, 35(5), 41-49.

[27] Machorro-Cano, I., Alor-Hernández, G., Paredes-Valverde, M. A., Rodríguez-Mazahua, L., Sánchez-Cervantes, J. L., & Olmedo-Aguirre, J. O. (2020). HEMS-IoT: A big data and machine learning-based smart home system for energy saving. Energies, 13(5), 1097.

[28] PunithaIlayarani, P., & Dominic, M. M. (2019, February). Anatomization of fog computing and edge computing. In 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-6). IEEE.

[29] Zareen, M.S., Tahir, S., Akhlaq, M., & Aslam, B. (2019). Artificial Intelligence/ Machine Learning in IoT for Authentication and Authorization of Edge Devices. 2019 International Conference on Applied and Engineering Mathematics (ICAEM), 220-224.

[30] Khanh, Q. V., Hoai, N. V., Van, A. D., & Minh, Q. N. (2023). An integrating computing framework based on edge-fog-cloud for internet of healthcare things applications. Internet of Things, 23, 100907.

[31] Amorim, I., & Costa, I. (2023). Homomorphic Encryption: An Analysis of its Applications in Searchable Encryption. arXiv preprint arXiv:2306.14407. Available at: https://arxiv.org/abs/2306.14407 (Accessed: 23 November 2023).

[32] Munjal, K., & Bhatia, R. (2023). A systematic review of homomorphic encryption and its contributions in healthcare industry. Complex & Intelligent Systems, 9(4), 3759-3786.

[33] El-Hajj, M., Mousawi, H., & Fadlallah, A. (2023). Analysis of Lightweight Cryptographic Algorithms on IoT Hardware Platform. Future Internet, 15(2), 54.

[34] Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., & Tschofenig, H. (2022). RFC 9200: Authentication and Authorization for Constrained Environments Using the OAuth 2.0 Framework (ACE-OAuth).

[35] Arnaboldi, L., & Tschofenig, H. (2019, March). A formal model for delegated authorization of IoT devices using ACE-OAuth. In OAuth Security Workshop.Available at: https://datatracker.ietf.org/doc/rfc9200/

[36] Seitz, L. (2022). RFC 9201: Additional OAuth Parameters for Authentication and Authorization for Constrained Environments (ACE). Available at: https://datatracker.ietf.org/doc/rfc9201/

[37] Gerdes, S., Bergmann, O., Bormann, C., Selander, G., & Seitz, L. (2022). RFC 9202: Datagram Transport Layer Security (DTLS) Profile for Authentication and Authorization for Constrained Environments (ACE). Available at: https://datatracker.ietf.org/doc/rfc9202/

[38] Vijayaraghavan, V., & Agarwal, R. (2017). Security and privacy across connected environments. Connected Environments for the Internet of Things: Challenges and Solutions, 19-39.

[39] Palombini, F., Seitz, L., Selander, G., & Gunnarsson, M. (2022). RFC 9203: The Object Security for Constrained RESTful Environments (OSCORE) Profile of the Authentication and Authorization for Constrained Environments (ACE) Framework. Available at: https://datatracker.ietf.org/doc/rfc9203/

[40] Gharaee, H., Shabani, F., Mohammadzadeh, N., & Mehranpoor, S. (2020). Biometric Based User Authentication Protocol in Smart Homes. International Journal of Web Research, 3(1), 29-41.

[41] Mahalle, P. N., & Shinde, G. R. (2021). OAuth-based authorization and delegation in smart home for the elderly using decentralized identifiers and verifiable credentials. Security issues and privacy threats in smart ubiquitous computing, 95-109.

[42] Aftab, M. U., Qin, Z., Hundera, N. W., Ariyo, O., Zakria, Son, N. T., & Dinh, T. V. (2019). Permission-based separation of duty in dynamic role-based access control model. Symmetry, 11(5), 669.

[43] Agrawal, M., Zhou, J., & Chang, D. (2019). A survey on lightweight authenticated encryption and challenges for securing industrial IoT. Security and Privacy Trends in the Industrial Internet of Things, 71-94.

[44] Whaiduzzaman, M., Barros, A., Chanda, M., Barman, S., Sultana, T., Rahman, M. S., Roy, S., & Fidge, C. (2022). A review of emerging technologies for IoT-based smart cities. Sensors, 22(23), 9271.

[44] Lagutin, D., Kortesniemi, Y., Fotiou, N., & Siris, V. A. (2019, February). Enabling decentralised identifiers and verifiable credentials for constrained IoT devices using OAuth-based delegation. In Proceedings of the Workshop on Decentralized IoT Systems and Security (DISS 2019), in Conjunction with the NDSS Symposium, San Diego, CA, USA (Vol. 24).