National College of Ireland

Leveraging X.509 Certificates and OAuth for optimized use of DIDs and VCs in Constrained IoT Devices

MSc Research Project
MSc in Cybersecurity

Ketki Shekhar Jakatdar
Student ID: x22152229

School of Computing
National College of Ireland

Supervisor: Prof. Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Ketki Shekhar Jakatdar
Student ID:	x22152229
Programme:	MSc in Cybersecurity
Year:	2024
Module:	MSc Research Project
Supervisor:	Prof. Vikas Sahni
Submission Due Date:	12/08/2024
Project Title:	Leveraging X.509 Certificates and OAuth for optimized use of DIDs and VCs in Constrained IoT Devices
Word Count:	5870
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Ketki Shekhar Jakatdar
Date:	12th August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Leveraging X.509 Certificates and OAuth for optimized use of DIDs and VCs in Constrained IoT Devices

Ketki Shekhar Jakatdar
x22152229

Abstract

Due to the rapid growth of IoT devices worldwide, unauthorized access to them they has become one of the biggest concerns. Verifiable credentials (VCs) and Decentralised identifiers (DIDs) based on blockchain technology, provide a strong alternative to centralised authorization. However, constrained IoT devices have low computing capability which makes them unable to process VCs and DIDs. To overcome this, the approach of delegating the DID and VC processing to an OAuth server is adopted. This research presents a novel approach to incorporate authorization proofs into X.509 certificates, reducing the redundant calls in the TLS v1.3 handshake. The implementation integrates Hyperledger Aries-Cloudagent-Python and modified ACE-OAuth server to handle the processing of DIDs and VCs. Key components include X.509 certificates with authorization proofs embedded as custom extensions and Proof of Possession (PoP) tokens. In order to simulate a real-world situation, the architecture illustrates a university (Faber) and a lecturer (Alice,) trying to access a constrained IoT printer. The results indicate reduction in the total steps from 6 to 4 and Round Trip Time (RTT) from approximately 2RTT to 1RTT.The achieved RTT for the TLS handshake is approximately 3.488 ms. Security testing included verifying the PoP token signature, preventing replay attacks, and detecting data tampering. Thus, the solution presented demonstrated both increased efficiency and security.

KeyWords: Decentralized Identifier (DID), Verifiable Credential (VC), ACE-OAuth, Hyperledger Aries-Cloudagent-Python, X509 certificates, Rount Trip Time (RTT).

1 Introduction

The market for Internet of Things (IoT) devices is significantly expanding by each passing day. This trend is only going to grow as businesses worldwide are increasingly adopting IoT technology. According to a study by Statista [1], the number of IoT devices is projected to rise from 15.1 billion in 2019 to 29 billion by 2030. These devices will be used in various fields like agriculture, manufacturing, IT, consumer goods, toys, and communications.

Although these devices have wide ranging benefits, a significant number of them are constrained by design. This is restated by Bormann (2014) in his work that, there

[1]Statista: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

are multiple limitations to majority of the IoT devices present today, mainly CPU processing power, memory (RAM), storage (ROM), software updates, security patches, and interoperability. Such devices due to their limited resources are susceptible to cyberattacks. Unauthorized access to these devices poses a significant threat, as it can lead to various types of cyberattacks, including passive sniffing, eavesdropping, replay attacks, Man-in-the-Middle (MiTM) attacks, and Denial of Service (DoS) attacks. These attacks compromise the privacy and integrity of sensitive data. More IoT malware attacks have been reported in the past few years, indicating that unauthorized access is still a serious problem and that improved security measures are desperately needed [2].

These devices might not contain highly sensitive and private data, but their exploitation could open a backdoor that could give access to the controllers or other devices on the network. Therefore, this consequential issue of unauthorized access to IoT devices needs to be resolved urgently. Strong Identity and Access Management (IAM) policy must be implemented to protect them and the networks that connect them. Robust authentication and authorization protocols as recommended by frameworks like NIST, will lower the risks of unauthorized access and prevent different cyberattacks. These mechanisms ensure only authorized users interact with these devices, which in turn minimizes vulnerabilities and aids in preventing and investigation of breaches. Centralized and Federated authentication methods are often utilized, but this approach has several disadvantages such as a single failure point in the network. Alternatively, Verifiable Credentials (VC) and Decentralized Identity (DID) based on Blockchain technology offer more privacy and sovereignty. However, it is almost impossible to execute DIDs or VCs directly on constrained IoT devices because they cannot process them on the device itself due to their limitations. To solve this problem, researchers suggested to delegate the authorization process to an OAuth Authorization Server (AS). However, this approach has redundant back-and-forth communication calls, resulting in increased latency and performance issues. Therefor, the flow can be simplified by using X.509 certificates in TLS communication with embedded proofs.

The above research problem motivates the following Research Question (RQ):

> **Does incorporating authorization proofs, such as AS authorization and device identity, into the X.509 certificates, used during TLS communication, optimize the OAuth-based authorization process for verifying decentralized identities and verifiable credentials in constrained IoT devices?**

This research focuses on finding out the answer to above research question. A secure and decentralized authentication system was developed by integrating Hyperledger Aries-Cloudagent-Python with an OAuth server utilizing X509 certificate with custom extensions and Proof-of-Possession (PoP) access tokens. The objective was to propose an alternative method to optimize the flow of the delegation of DIDs and VCs to an OAuth server. The system aims to maintain security by issuing a PoP access tokens to the legitimate user. The rest of the report is structured as follows: Section 2 reviews Related Work, highlighting existing solutions and identifying gaps. Section 3 details the Methodology, describing the technical approach and tools used. Section 4 presents the Design

[2]Zscaler: https://www.zscaler.com/blogs/security-research/2023-threatlabz-report-indicates-400-growth-iot-malware-attacks

Specification, outlining the system architecture and components. The Implementation of the proposed system is described in Section 5, followed by an Evaluation in Section 6, which assesses the system's performance and security. Finally, Section 7 concludes the research and suggests directions for Future Work, suggesting improvements to the proposed system.

2 Related Work

Various researchers have proposed several methodologies for authentication and authorization of IoT devices. A traditional approach involves centralized systems, where a central server performs the verification and access control for IoT devices. This method is straightforward but introduces some challenges, especially for constrained IoT devices. Centralized systems can become single points of failure and are prone to attacks such as Distributed Denial of Service (DDoS). Constrained devices, due to their limited computational capabilities, cannot manage the overhead of secure communications with central servers (Abomhara and Køien; 2014). Federated identity management is another approach. This allows multiple entities to trust each other's identity verification processes and reduces the burden of authentication with each entity separately. However, it still has the centralization issue and might get complex to manage in large-scale IoT environments More et al. (2023).

Blockchain technology offers a decentralised solution to the above mentioned issues. Without depending on a central authority, IoT devices can authenticate and authorize users using Blockchain. This technology ensures transparency and security by using an immutable ledger, making it difficult for malicious attackers to tamper with the data. Studies have indicated that VCs and DIDs can significantly improve IoT security. For example, Fedrecheski et al. (2020) examined the application of Self-Sovereign Identity (SSI) frameworks through DIDs and VCs, to IoT devices for identity management. This reduces dependency on centralised authorities and mitigates the risks associated with traditional identity management systems. In the same manner, Kuperberg (2020)'s case studies and real-world applications indicate the potential benefits of Blockchain-based identity management systems in different industries.

However, Blockchain nodes require more computational and storage resources than traditional approaches (Novo; 2018). This poses the biggest challenge. Bartolomeu et al. (2019) emphasized the need for lightweight cryptographic protocols and efficient task delegation to more capable devices or servers. They suggested using OAuth servers to process VCs and DIDs, which would reduce the computational load on devices with limited resources. This delegation approach ensures that even devices with limited resources can benefit from advanced security measures.

2.1 DIDs and VCs for constrained IoT devices

While above surveys focused primarily on the advantages and limitations of SSI, the research by Novo (2018) introduced a fully decentralized access control system using management hub nodes. This research also demonstrated the need of intermediary nodes due to the limited resources of IoT devices. Additional research by Panda et al. (2019) presented an authentication mechanism utilizing Ethereum and Gateway nodes to connect IoT devices to the Blockchain. Kortesniemi et al. (2019) in their study have focused on the use of Decentralized Identifiers (DIDs) as privacy-enhancing solutions, employing

both direct implementations on devices and proxy-based methods. They also emphasized the crucial role of secure key management in protecting user privacy. Across these three researches, the Ethereum blockchain platform was a common element. Clearly Ethereum showcases its potential in enhancing IoT security. However, intermediary nodes introducing complexity, negative impact on performance due to high overhead are some of the common limitations identified collectively.

Inspired from above approaches, Fedrecheski et al. (2021) proposed a DID-based model for smart home systems using Ethereum as the blockchain platform. Key tools such as Web3.js and Ganache were used to simulate the Ethereum environment. However, the research highlights the need to address scalability challenges. Similar to this Zhao et al. (2023) focused on SSI in constrained IoT networks, leveraging DIDs and DID Documents (DDos) implemented using SwarmLib and Python. The study emphasizes the use of CBOR-based DID Documents for IoT (CBOR-DI), achieving a 75% reduction in DDo sizes and significant overhead reduction with DIoTComm. The research, however, requires further real-world validation. Siris et al. (2019) on the other hand, integrated the OAuth 2.0 framework with blockchain and smart contracts for IoT authorization. It compared two models: one linking authorization grants to blockchain payments and another using smart contracts for handling requests. The second model, despite offering robust security features, incurred over three times the gas cost (366,277 units) compared to the first model (102,476 units). Again, Ethereum is common across these studies. These studies highlight the trade-offs required between security, cost, and scalability in implementing blockchain solutions for IoT.

Building upon these insights following researche further explores various methodologies utilizing frameworks like OAuth 2.0 and ACE-OAuth. Mahalle and Shinde (2021) integrated OAuth 2.0 and ACE-OAuth with DIDs and VCs to enhance security and privacy in smart homes. They delegated DID and VC processing to an Authorization Server (AS). This approach reduces computational load on constrained devices. This setup uses the Constrained Application Protocol (CoAP) protocol for communication and Proof-of-Possession (PoP) access tokens. However, it faces challenges such as potential device tampering. Grande and Beltrán (2020) made use of an edge-centric model, using edge devices as intermediaries between cloud services and IoT devices. It demonstrated scalability, handling up to 1100 devices with low memory usage (310 bytes per device context) and response times of 800ms to 1.25s, but also highlighted vulnerabilities to physical tampering and potential DoS attacks. A novel access control solution using a .NET Core web application for OAuth 2.0 credential requests and DPoP access requests was introduced in (Fotiou et al.; 2022). The system achieved high efficiency, with VC issuance taking under 0.1 ms and verification times ranging from 10.01 ms on a Raspberry Pi to 160 ms on an ESP32. However, it requires physical access for configuration changes and lacks selective disclosure support. In contrast, Lagutin et al. (2019) leveraged Hyperledger Indy for decentralized identity management, focusing on the delegation of DID and VC processing to an OAuth Authorization Server. This is demonstrated through a printing service use case. While the solution enhances privacy by reducing information disclosure, it mandates network setup and manual configuration which limits its compatibility with existing federated identity systems. Moreover, the proposed research in this report extends the work of their work, focusing on optimizing the authorization flow.

2.2 DIDs and VCs for non-constrained IoT devices

After exploring use of DIDs and VCs for constrained IoT devices, this section discusses various technologies for non-constrained IoT devices. Kang and Seo (2023) proposed an OAuth-based access control framework that leverages DIDs and smart contracts. The setup utilized Hyperledger Indy and Ethereum for managing DIDs and implementing smart contracts, respectively. Although the proposed approach demonstrated improvements in data confidentiality and integrity, it faces challenges in evaluating the complexity of the experimental setups. Additionally, Dixit et al. (2022) presented a distributed-ledger-based M2M identity framework using Ethereum and Hyperledger Indy. The Inter-Planetary File System (IPFS) provided decentralised storage, and Solidity was utilized to develop smart contracts. Their primary objective was to determine Ethereum gas costs and verification time, as well as efficient handling of transactions in Indy. The scalability and transaction costs of Ethereum were identified as limitations. In contrast, Lucking et al. (2020) focused on the W3C DID specification and the Masked Authenticated Messaging (MAM) protocol for secure communication, introducing an Identity Management System (IdMS) utilising the IOTA Tangle. Due to publicly available data and potential Sybil attacks, this approach presented privacy concerns despite offering high scalability and low transaction costs. Even though the focus of each study was on a different aspect of decentralised identity management, they all encountered similar issues with privacy, scalability, and the real-world constraints of their respective blockchain platforms.

Furthermore, Zhang et al. (2019) also presented a decentralized access control framework leveraging Ethereum smart contracts. To manage access policies, they also included Judge Contracts (JC), Register Contracts (RC), and Access Control Contracts (ACCs). They used Solidity and web3.js for contract development and blockchain interaction and their gas consumption for deployment ranged from 1,380,781 to 2,543,479 units. With an average contract execution time of less than 30 seconds, the system proved its functionality yet, real-world testing could be a concern. Fan et al. (2020) introduced DIAM-IoT, an IAM framework using DIDs and VCs for user-centric data sharing. This approach was supported by the IoTeX blockchain and AWS cloud services. The Proof of Concept (PoC) implementation showcased secure data sharing. Mohanta et al. (2019) focused on DecAuth, a decentralized authentication system using the Ethereum platform to ensure secure, password-free IoT device authentication. It demonstrated robustness against common cyberattacks, utilizing Web3.js, MetaMask, and Ganache, though transaction speed and architectural complexity pose limitations. Fotiou et al. (2019) again described an Ethereum-based architecture for IoT device management, implementing token-based access control with ERC20 tokens and event-driven smart contracts. The use of Mozilla's Thing Gateway and MetaMask highlighted the feasibility of secure device management. However, potential issues related to transaction delays, fluctuating costs, and privacy were observed. Clearly, Ethereum appears to be a popular choice but all these studies share limitations such as transaction latency due to Ethereum's block times, cost fluctuations, and challenges in integration and scalability.

The same authors Fotiou et al. (2020) presented another study that makes use of the similar setup as the previous ones. The setup involved a blockchain-based OAuth 2.0 authorization system utilizing ERC-721 tokens, Ethereum blockchain, Solidity for smart contract development, and tools like web3.js and MetaMask. This system demonstrated efficient token operations but faced limitations such as privacy concerns and potential security risks from compromised authorization server keys. Yin et al. (2022) presen-

ted SmartDID, a privacy-preserving distributed identity management system for IoT, as a solution to this problem. Bulletproofs and Pedersen commitments were combined to create a dual-credential model. It utilized the Fisco Bcos blockchain and Java 1.8, achieving efficient identity verification with setup times averaging 0.4 seconds. However, the complexity of cryptographic verifications and the PBFT consensus algorithm's communication overhead posed scalability challenges. Unlike all the above studies, a completely new terminology was introduced by Diego et al. (2021). They discussed the IoT-as-a-Service (IoTaaS) model, implementing an identity management system based on SSI. It used Hyperledger Indy, Aries Framework, and JavaScript. Performance testing on Raspberry Pi 2 B showed efficient credential operations with minimal resource usage, with yet again scalability issues.

Compared to traditional methods for constrained IoT devices, all the above systems offer enhanced security and flexibility. However, these systems introduce communication overhead due to their cryptographic operations.

2.3 Research Niche

Table 1: Comparison of previous work

Related Work	**Blockchain Platform**	**Proposed Solution**	**Limitations**
Novo (2018)	Management hub nodes	Introduced a fully decentralized access control system.	Increases architectural complexity due to need intermediary nodes.
Panda et al. (2019)	Ethereum	Authentication mechanism utilizing Ethereum and Gateway nodes to connect IoT devices to the Blockchain.	Introduces complexity and high overhead, negatively impacting performance.
Kortesniemi et al. (2019)	Ethereum	Focused on the use of DIDs using both directly on devices and proxy-based methods.	Emphasized the need for secure key management.
Fedrecheski et al. (2021))	Ethereum	DID-based model for smart home systems using Web3.js and Ganache.	Scalability challenges need to be addressed.
Zhao et al. (2023)	SwarmLib	Focused on SSI in constrained IoT networks, leveraging DIDs and DID Documents (DDos).	Requires real-world validation.
Siris et al. (2019)	Ethereum	Integrated OAuth 2.0 with blockchain and smart contracts for IoT authorization.	Over 3 timer higher gas costs with the smart contract model.

Mahalle and Shinde (2021)	Not specified	Constrained Application Protocol (CoAP) and PoP access tokens.	No Proof of Concept (PoC), device tampering, challenges in device configuration.
Grande and Beltrán (2020)	Edge or Fog computing	Edge-centric model with edge devices as intermediaries between cloud services and IoT devices.	Physical tampering and DoS attacks.
Fotiou et al. (2022)	VC Issuer (github repo)	Introduced a .NET Core web application for OAuth 2.0 credential requests and DPoP access requests.	Lacks selective disclosure support.
Lagutin et al. (2019)	Hyperledger Indy	Delegation of DID and VC processing to an ACE-OAuth server.	Redundant calls in TLS handshake, requires network setup and manual configuration.

3 Methodology

3.1 Research Methods

A comprehensive research methodology was adopted to successfully implement and evaluate the project. The research began with an in-depth analysis of Hyperledger Aries-Cloudagent-Python and OAuth server through their GitHub repositories to thoroughly understand the code. Additionally, Decentralized Identifiers (DID) and Verifiable Credentials (VC) were explored from various World Wide Web Consortium (W3C) resources, such as W3C:DID [3] and W3C:VC [4]. For testing tools known from earlier projects were chosen because they were well-known and suitable for beginners. Hyperledger Aries-Cloudagent-Python provided detailed information through its wiki pages, while the OAuth server offered guidance via a detailed cookbook. Basic setup tasks, such as downloading Apache, PHP, Visual Studio Code, Docker Desktop, and setting up a Node.js Express TypeScript server, were done by following various online blogs and publicly available YouTube tutorials.

Throughout the implementation, each step was carefully documented along with the screenshots to simplify future reference and to assist others in recreating the research. All the common errors encountered during the implementation process were also documented. All the necessary screenshots, error logs, and technical stack details will be included in the configuration manual submitted with this report. Furthermore, assistance from professors, supervisors, and the Hyperledger community was seeked to resolve the issues encountered during the initial setup of Hyperledger Indy-SDK. To finalize the evaluation metrics, industry standards such as NIST, W3C, and Internet Engineering Task Force (IETF)'s RFC 8446 were referred to ensure the quality of the proposed solution.

[3]W3C DID: https://www.w3.org/TR/did-core/

[4]W3C VC: https://www.w3.org/TR/vc-data-model-2.0/

3.2 Hyperledger vs Ethereum

It was also observed from the related work section above, that Ethereum is a popular choice, but a study by Ucbas et al. (2023) highlighted Hyperledger's superiority over Ethereum for IoT applications due to its better performance in latency and throughput. Hyperledger's permissioned network structure and efficient consensus mechanism result in lower computational overhead and more reliable, timely data processing. Also, Hyperledger is a permissioned and private blockchain, whereas Ethereum is a permissionless and public network. That's why Hyperledger offers better security and confidentiality unlike Ethereum (Alamri et al.; 2022). This feature ensures that only authorized participants can access and interact with the Hyperledger system. These features make Hyperledger an ideal choice for resource-constrained IoT environments.

3.3 ACE-OAuth Framework

ACE-OAuth stands for "*Authentication and Authorization in Constrained Environments (ACE) OAuth*". It was introduced in August 2022 as RFC 9200 (Seitz et al.; 2022). It is essentially an extension of the OAuth 2.0 framework that is designed specifically for constrained IoT devices. Because constrained IoT devices have limited memory, processing power, and connectivity, they have special challenges that the standard OAuth protocol is not prepared to handle. This resulted in the creation of ACE-OAuth. IoT devices with limited resources can be supported by ACE-OAuth, which guarantees effective and secure communication without compromising functionality. It offers fine-grained access control, making it a popular choice for industries deploying large-scale IoT systems (Seitz et al.; 2022). It also provides a scalable and secure solution. Furthermore, ACE-OAuth uses a Proof-of-Possession token, which is the standard access token bound to a private key (Arnaboldi and Tschofenig; 2019). This makes ACE-OAuth the best choice for this research as well.

4 Design Specification

This section describes the architecture of the proposed solution and the authentication flow.

4.1 Architecture

The architecture diagram of the scenario used to implement the proof of concept for the proposed solution is shown in Figure 1. The key difference between this diagram and the research reported in Lagutin et al. (2019) is the optimization of the authentication flow with the use of authorization proofs embedded in X509 certificates. Faber is the university that acts as a trusted authority and can issue DIDs and VCs to the lecturer i.e. Alice in this case. These credentials are required to confirm that Alice is authorized to use the printing services. Alice initiates the printing requests by communicating with the Authorisation Server (AS).

The AS is essentially the ACE-OAuth server which plays a central role in the system. It handles authentication and authorization processes. Upon receiving a request, the AS verifies Alice's credentials and checks her rights to print. The '*degree*' attribute from Alice's DID is the VC, i.e. the '*right to print*'. Once authenticated, the AS generates

a Proof of Possession (PoP) access token, which includes the necessary claims and the document to be printed. The document to be printed is assumed to be sent by Alice to the Resource Server (RS). For this PoC, Alice sends just the PoP access token, to which the RS after successful verification, responds with a "Access granted" message. This is to replicate the process of printing.

In real world scenario, the Printing Service (PS) manages a network of printers, identified as IoT devices, distributed across various locations. For this PoC it is the Resource Server (RS). It is also assumed that the AS's private key is already shared with the RS and printer. These certificates ensure that only authorized printers can participate in the network, maintaining the integrity and security of the printing service.

Following are the assumptions made for the implementation of this research:

1. **Trust Relationship:** It is assumed that the trust relationships between the University, Authorization Server (AS), and Printing Service (PS) are pre-established and secure.

2. **Printer Recognition:** The printer is assumed to be part of the Printing Service (PS) network.

3. **Document Handling:** The process for sending the document to the printer is assumed to be seamless, with no issues related to file format, size, or network transmission.

4. **Printer Availability:** The printer is assumed to be available, operational, and not subject to issues such as being out of paper, offline, or malfunctioning during the printing request.

5. **No Revocation:** It is assumed that there is no need for DID and VC revocation, meaning they remain valid for the entire session.

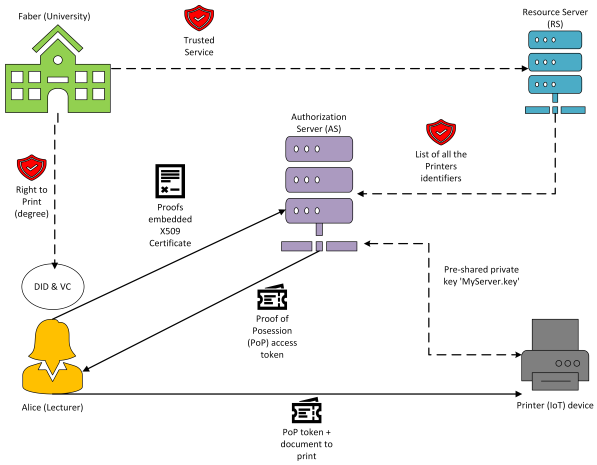


Figure 1: Architecture for University-Printer scenario

4.2 Optimized Authentication Flow

Figure 2 illustrates the optimized authentication flow which was achieved with the successful implementation of embedding the authentication proofs in X509 certificates.

Service Discovery:

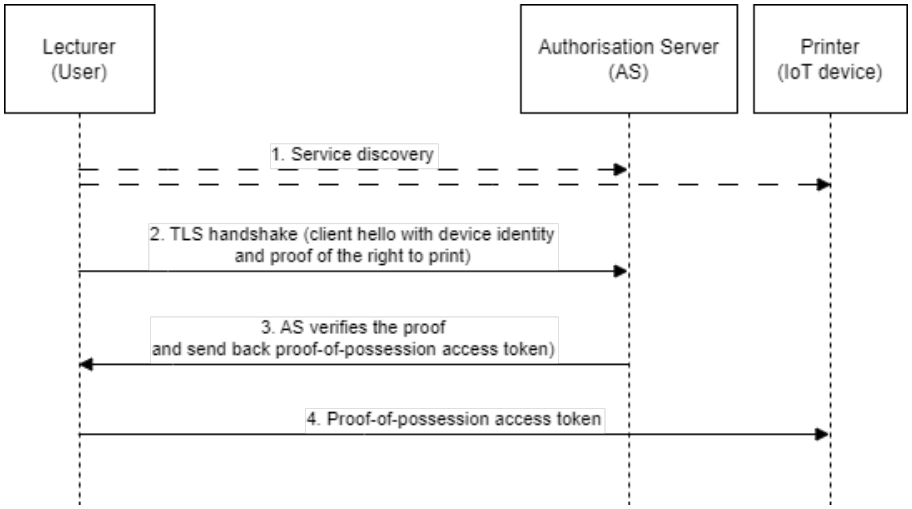


Figure 2: Optimized auth flow by embedding proofs in X509 certificates

- Alice (Lecturer) initiates service discovery to locate the printer and the Authorization Server (AS). It is assumed that Alice already has the credentials (DID and VC) issued by Faber (University) at this point.
- This step ensures that the user identifies the correct AS and printer to interact with, for the printing service.

TLS Handshake:

- The client (Postman) starts the TLS handshake by sending a "Client Hello" message as a Postman HTTPS POST request, that includes Resource Server (RS) identifier and proof of the right to print i.e. 'degree' attribute for Alice.
- The proofs are embedded within a self-signed X509 certificate, containing custom extensions like DIDs and VCs of Alice and RS identifiers.

Proof Verification by AS:

- The AS verifies the provided proofs, checking the VC, nonce, degree attribute, RS identifier along with certificate validation.
- Upon successful verification, the AS generates a Proof of Possession (PoP) token and sends it back to the client.

PoP Token Presentation:

- Alice presents the PoP token via another Postman HTTP POST request to the RS (printer), assuming with the document to print.
- The printer validates the PoP token, ensuring that it was issued by the trusted AS and that it has not been tampered with or used previously.
- Upon successful validation, the printer grants access and processes the print job, which in this case is, returns a "Access granted" message.

5 Implementation

The implementation involved setting up Hyperledger Aries based Blockchain system, an ACE-OAuth authorization server, a dummy printing server, and X509 certificates with custom extensions. Each stage of the implementation is discussed in detail further in this section.

5.1 DID & VC generation

To issue DID and VC to Alice, Aries Cloud Agent Python (ACA-Py) project was used. Two agents, one for Faber university and another for Alice, were launched using Docker Desktop. These agents were connected to a public ledger managed by the BC Government's VON team as shown in Figure 3. This ledger is useful for the visualization of transactions such as issue and verify DID registrations, schema definitions, credential issuances, etc. Both the agents's Swagger UI was enabled to provide an interactive interface to easily access the API endpoints as shown in Figure 4 and Figure 5 for Alice and Faber respectively. Initially, a secure DIDComm connection was established between the agents, which is an encrypted communication channel for interactions between the agents, including the credential issuance process.

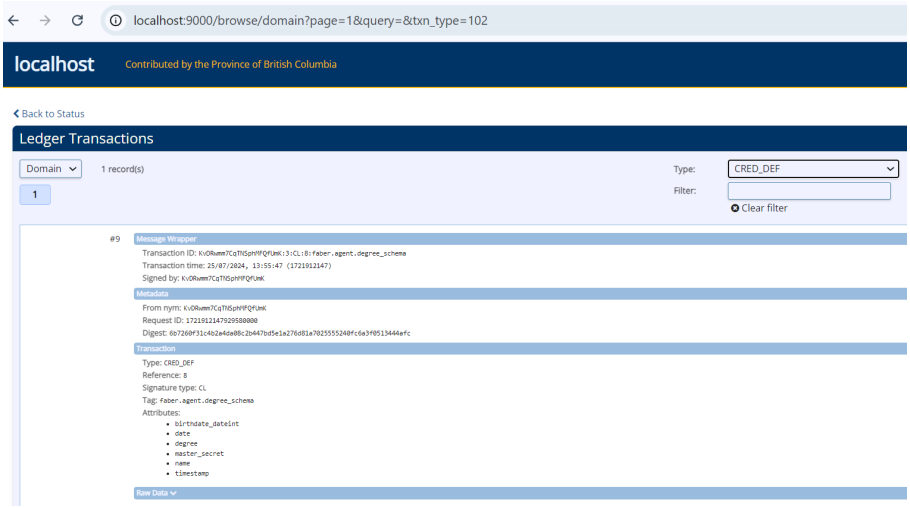


Figure 3: Public Ledger for Alice and Faber agents

A schema defines the structure and types of attributes a credential will contain. Whereas, a credential definition links this schema to a specific issuer which creates the VCs, which in this research is Faber. For this study, the schema defined attributes are as shown in Figure 6a. Faber's agent sent a credential offer to Alice's agent, involving the schema ID, credential definition ID, and specific data values. Alice's agent automatically responded with a credential request, prompting the Faber agent to issue the credential. The credential was then securely stored in Alice's digital wallet. The final step involved the Faber agent receiving an acknowledgment that the credential was successfully stored. This completed the credential issuance flow with Alice's credentials as shown in Figure 6b.

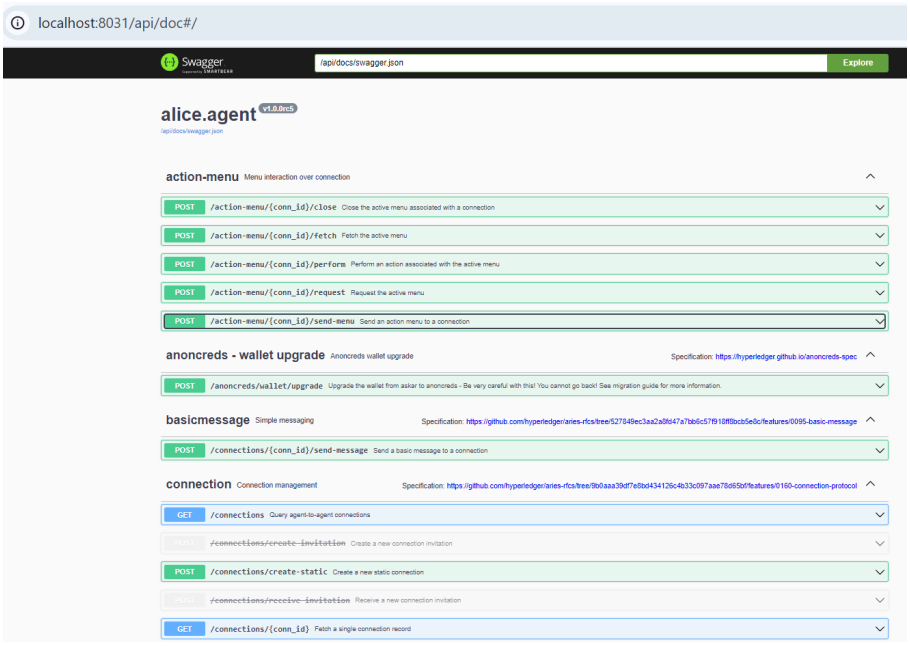


Figure 4: Alice's agent Swagger UI

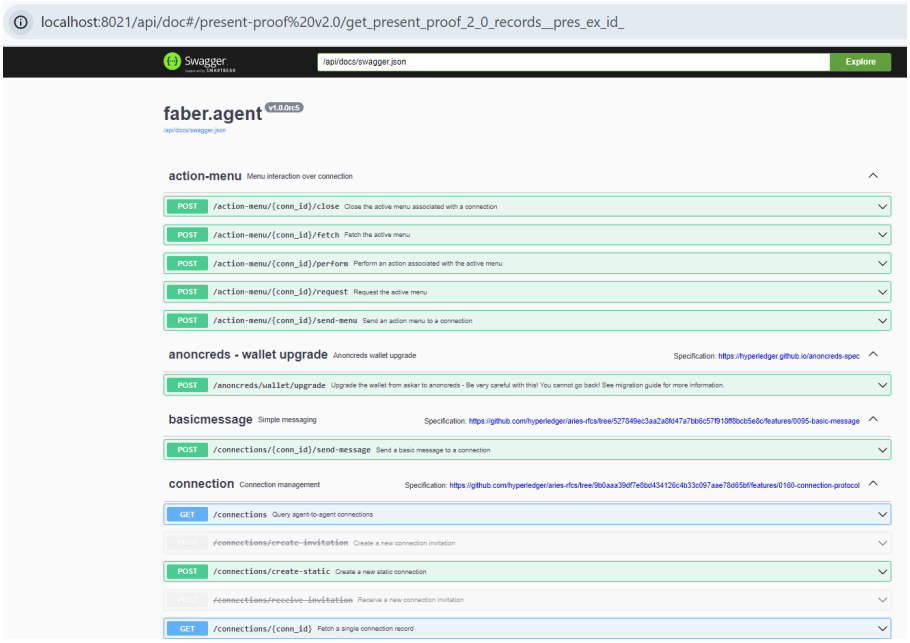
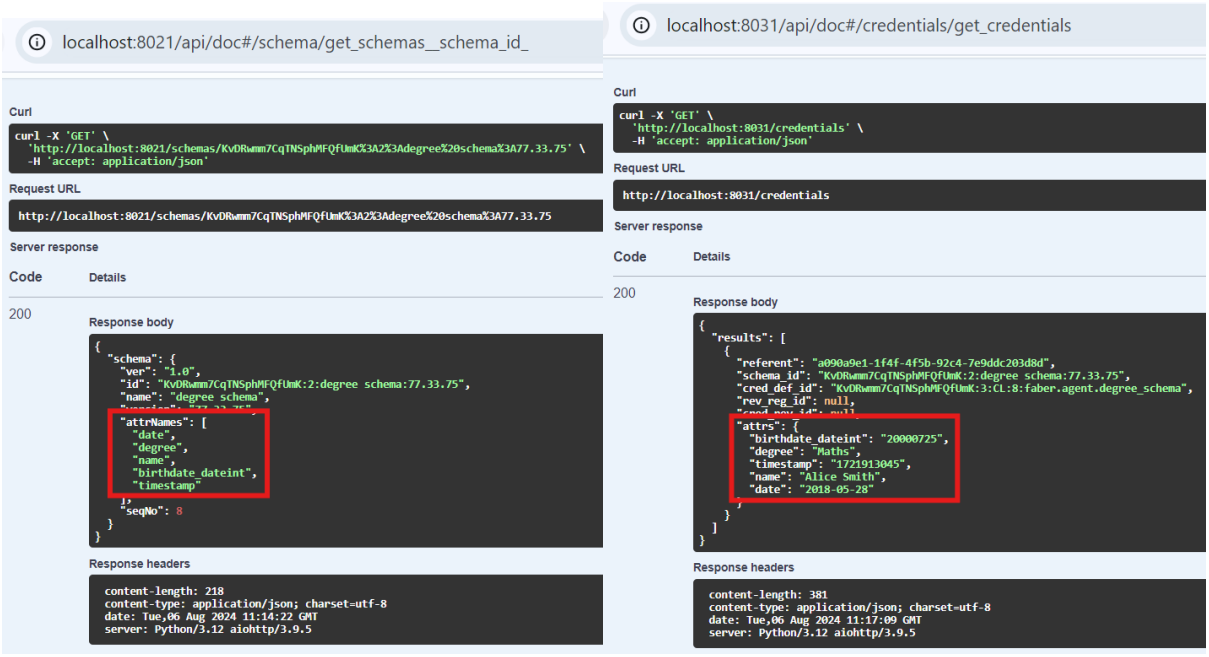


Figure 5: Faber's agent Swagger UI

5.2 Proof embedding in X509 Certificate

This step is the first crucial step to achieve the research's objective. Multiple files are created using OpenSSL, each serving a specific purpose. The '*ca.crt*' and '*ca.key*' are the Certificate Authority (CA)'s certificate and private key respectively. While '*alice.key*' and '*alice.crt*' are Alice's private key and public certificate respectively. The '*ext.cnf*' file, as shown in Figure 7, is particularly important because it defined custom extensions containing data such as Alice's DID, name, degree, birthdate, date & timestamp and the Resource Server identifier. The Resource Server identifier allows the AS to identify the specific device the client wants to access. These custom extensions, identified by unique



(a) Attributes defined in schema (b) Alice's Verifiable Credentials

Figure 6

Object Identifiers (OIDs), allow the certificate to carry essential credential data directly within the certificate. The '*alice.crt*' is signed with the custom extensions defined in 'ext.cnf'. Later, '*alice.pfx*' bundles '*alice.crt*' and '*alice.key*' for secure transport from Postman to the AS. The '*ca.srl*' tracks issued certificate serial numbers.

ext.cnf

```
[ req ]
default_bits = 2048
prompt = no
default_md = sha256
distinguished_name = dn
req_extensions = v3_req

[ dn ]
C = IR
ST = Dublin
L = Dublin
O = NCI
OU = Cybersecurity
CN = localhost

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment
subjectAltName = @alt_names

# Custom extensions with key-value pairs
1.3.6.1.4.1.12345.1 = ASN1:UTF8String:DID:
4zQmVxZryPRXNhpuMrbdEmjTdcHyEgkUbBnv3Ebx3TDhpK25
1.3.6.1.4.1.12345.2 = ASN1:UTF8String:Name: Alice Smith
1.3.6.1.4.1.12345.3 = ASN1:UTF8String:Degree: Maths
1.3.6.1.4.1.12345.4 = ASN1:UTF8String:Date: 2018-05-28
1.3.6.1.4.1.12345.5 = ASN1:UTF8String:Birthdate: 20000725
1.3.6.1.4.1.12345.6 = ASN1:UTF8String:Timestamp: 1721913045
1.3.6.1.4.1.12345.7 = ASN1:UTF8String:ResourceServer: Printing Server 1

[ alt_names ]
DNS.1 = localhost
```

Figure 7: 'ext.cnf' file to add custom extensions in X509 certificate

5.3 Authorization Server (ACE-OAuth Server)

Next step was to configure an ACE-OAuth server using a standard OAuth server from the provided GitHub repository [5]. Apache served as the web server and MySQL Workbench was used for test client database. A custom grant type, "*grant_type=did*", was designed to validate Alice's DID and VC. The implementation involved drafting a PHP script, '*DIDGrant.php*'. This script retrieved the client certificate from the Postman request and extracted the DID and specific credential attributes, such as the '*degree*' field, which serves as proof of Alice's right to print. The script validated the nonce, degree, checks the certificate's expiration, and parses custom extensions embedded in the X509 certificate. If the validation is successful, the server generates a Proof of Possession (PoP) token. It is basically an access token signed with the server's private key - '*MyServer.key*'. The response also included a confirmation of the Alice's public key as the value of '*cnf*' claim in JSON Web Key (JWK) format.

5.4 PoP token generation

Proof of Possession (PoP) token's generation was the second critical component of this research. Its implementation, as discussed earlier, was handled by the 'DIDGrant.php' script. The PHP logic first created a standard access token and then generated a PoP token by calling on the function '*generatePoPAccessToken()*'. This method began by extracting the public key details from the client certificate in JSON Web Key (JWK) format. The payload for the PoP token included information such as the issuer ("*OAuth Server*"), subject (*Alice's DID*), issued and expiration times, and the client public key confirmation (*cnf*) with the JWK. Additionally, it added the standard access token and the resource server identifier - '*Printing Server 1*', in the response. The payload is then signed using the server's private key with the *RS256* algorithm to produce the PoP token.

Upon decoding the encoded PoP token from response on JWT [6], fields such as '*iss*', '*sub*', '*iat*', '*exp*', '*cnf*', and '*resource_server*' are revealed. This confirmed the legitimacy and intended use of the token was met. The '*cnf*' claim with the client public key, makes the PoP access tokens superior to standard access tokens. This ensures that the token can only be used by the holder of the corresponding private key, thereby preventing misuse even if the token itself is intercepted.

5.5 Resource Server (Printing Server)

As the final step, a Resource Server (RS) was set up using Node.js. This RS basically acts as the constrained IoT device, i.e. a university printer. RS validated the incoming PoP tokens from the AS. The AS's public key was utilized to verify the token's signature and validate its claims, including checking the expiration and matching the expected resource server identifier - "*ResourceServer: Printing Server 1*". Additionally, the implementation included logic to prevent replay attacks by tracking used tokens. This ensured that each token could only be used once. Upon successful validation, the server responded with "*Access granted*" message, which basically meant that Alice could use the printer. Various error scenarios, such as invalid tokens or unauthorized resource access attempts were also

[5] OAuth server: https://github.com/bshaffer/oauth2-server-php
[6] JWT.io: https://jwt.io/

handled with appropriate error messages. Figure 8 illustrates the successful authorization scenario for Alice.

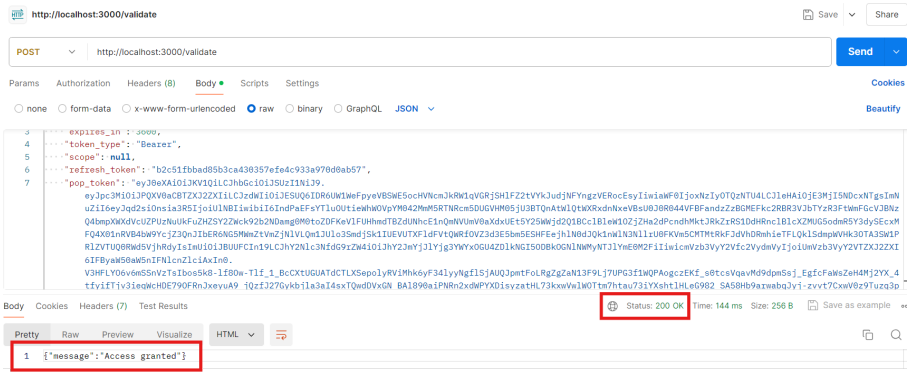


Figure 8: Access Granted to Alice

6 Evaluation

6.1 Performance Testing

6.1.1 Round Trip Time

To evaluate the performance of the system, the Round-Trip Time (RTT) for the TLS handshake between the client (Postman) and the AS was measured. According to RFC 8446, the RTT can be determined as the time difference between when the "Client Hello" message is sent and when the "Server Hello" message is received, as these represent the primary communication steps in establishing a secure connection [7] (Rescorla; 2018). For this implementation, the timestamp for the '*Client Hello*' is 0.002143 seconds, and for the '*Server Hello*' it is 0.005631 seconds. The RTT is thus calculated as:

$$RTT = Server\ Hello\ Timestamp - Client\ Hello\ Timestamp$$
$$RTT = 0.005631 - 0.002143 = 0.003488\ seconds$$
$$RTT = 3.488\ milliseconds$$

Based on typical TLS handshake times, RTT for the solution presented in Lagutin et al. (2019) can be estimated to range between 100 to 200 milliseconds. This estimation is done considering the additional steps involved in verifying the PoP token and validating credentials. The optimization, in terms of RTT, achieved in this research can be observed in the Table 2. Conclusively, this research reduced the need for multiple communication exchanges and achieved the research objective.

6.2 Security Testing

6.2.1 Replay Attack

RFC 8446 recognizes that including data in certificates to reduce RTT can expose systems to replay attacks (Rescorla; 2018). To test this, the PoP token received from the AS was

[7] IETF: https://www.ietf.org/archive/id/draft-song-ippm-inband-e2e-rtt-measurement-02.html

Table 2: Optimization achieved in this research for authentication flow

Aspect	Lagutin et al. (2019)	Proposed Project
Total number of steps in auth flow	6	4
Round Trip Time (RTT)	Approximately 2RTT	Approximately 1RTT
Efficiency Gain	-	Reduction by 1RTT, minimizing communication overhead

pasted in the POST request sent to the RS via Postman. On the first attempt, the server responded with "*Access granted*" confirming valid token. However, when the same request was resent, the server returned "*Token already used*" message. This demonstrated, as shown in Figure 9, that the tokens cannot be reused maliciously, ensuring effective detection and prevention of replay attacks.

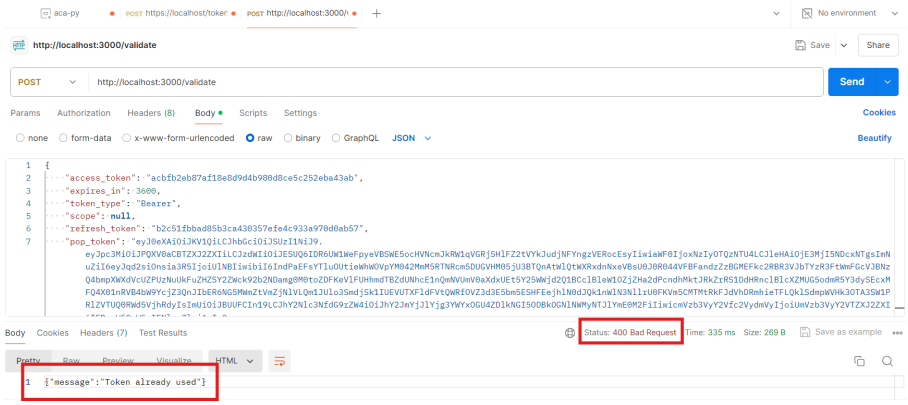


Figure 9: Replay attack prevention

6.2.2 Data Tampering

To test the server's response against data tampering of the PoP token, Burpsuite was used with Postman to intercept and modify the PoP token before it was sent to the RS. The Postman was configured to use Burpsuite as a proxy to intercept requests. With interception turned on in Burpsuite, a POST request containing a valid PoP token was sent from Postman. The encoded PoP token was then tampered with in Burpsuite before forwarding the request to the RS. As a result, the RS responded with "*Access denied: Invalid token signature*" as shown in Figure 10. This outcome confirms that any modification to the encoded PoP token changes its signature, making it invalid and ultimately preventing unauthorized access.

6.3 Discussion

Using DIDs and VCs is advantageous for constrained IoT devices as they offer enhanced privacy and sovereignty compared to traditional authorization systems. Unlike centralized and federated methods, DIDs and VCs provide decentralized control over identity management, reducing the major risk of a single point of failure. This decentralization

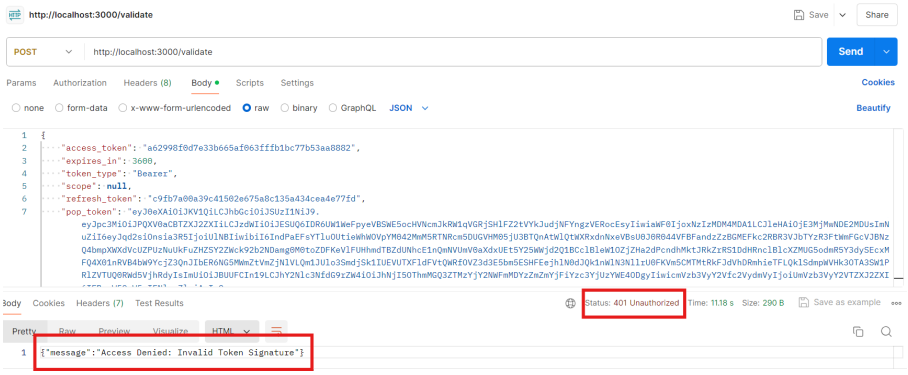


Figure 10: Data tampering prevention

is especially beneficial for constrained IoT devices as it minimizes the need for constant communication with a central authority, thereby reducing latency and improving performance. The integration of ACE-OAuth for constrained IoT devices is additionally beneficial as it allows the delegation of complex cryptographic operations to the Authorization Server (AS). This process transfers the computational burden from the IoT devices to the AS.

The proposed solution met the research objective of optimizing the delegated OAuth-based authorization process for verifying DIDs and VCs in constrained IoT devices, by incorporating authorization proofs into the X.509 certificates. This approach significantly reduces the Round Trip Time (RTT). The optimization is achieved by reducing the authentication flow from 6 steps to 4 steps. The security testing confirmed that robust mechanisms against potential vulnerabilities is present. The PoP token's signature was verified successfully on jwt.io which confirmed its authenticity and integrity. Replay attack prevention was demonstrated by ensuring that tokens cannot be reused and finally data tampering prevention was validated using Burpsuite, where any modification to the PoP token resulted in unauthorized access being effectively blocked. The presented solution can also be applied applied in various real-life scenarios like below:

1. **Healthcare Devices:** To ensure integrity of patient's private data.
2. **Industrial IoT (IIoT) and Operation Technology (OT):** Ensure Business Continuity (BC) by preventing Denial of Service (DoS) attacks.
3. **Smart Home Devices:** Preventing unauthorized access to burglars and thieves.

However, despite these strengths, the approach has some limitations. Key management could become complex, especially when dealing with multiple clients and ensuring the security of private keys. Another possible limitation is the need to generate a new certificate for each client with different attributes in custom extensions, which could be tiring and inefficient.

7 Conclusion and Future Work

The purpose of the research was to determine whether including authorisation proofs, such as AS authorization and device identity into the X.509 certificates used during TLS communication, could optimize the OAuth-based authorization process for verifying

DIDs and VCs in constrained IoT devices. The research successfully met this objective by developing an integrated system using Hyperledger Aries, an ACE-OAuth server, and X.509 certificates with custom extensions. The implementation demonstrated efficiency improvements by reducing the authentication flow from 6 steps to 4 steps and achieving an Round-Trip Time (RTT) of approximately 3.488 milliseconds. It is significantly lower than the estimated 100-200 milliseconds RTT as compared to earlier research. Security testing confirmed robust defenses against replay attacks and data tampering. As compared to previous research, this study provided a more detailed and quantifiable analysis of performance and security.

7.1 Future Work

Future work includes to further enhance the security and functionality of the system. Integrating Elliptic Curve Integrated Encryption Scheme (ENSI) with X.509 certificates will help prevent eavesdropping and passive sniffing attacks. To offer more adaptability in identity management, alternative DID methods such as "*did:key*" for simplicity and "*did:web*" for its potential in web-based scenarios can be explored. Implementing OpenID for Verifiable Credential (VC) issuance will allow users to receive a '*proof of right VC*' streamlining the credential issuance process. Additionally, implementing OpenID for VC Presentation will enable users to obtain an access token for further optimizing the authorization flow. These enhancements might improve security and in turn broaden the application of DIDs and VCs in a variety of real-world scenarios. They might also pave the way for more efficient and secure identity management of constrained IoT devices.

References

Abomhara, M. and Køien, G. M. (2014). Security and privacy in the internet of things: Current status and open issues, *2014 International Conference on Privacy and Security in Mobile Systems (PRISMS)*, pp. 1–8.

Alamri, B., Crowley, K. and Richardson, I. (2022). Blockchain-based identity management systems in health iot: A systematic review, *IEEE Access* **10**: 59612–59629.

Arnaboldi, L. and Tschofenig, H. (2019). A formal model for delegated authorization of iot devices using ace-oauth. 4th OAuth Security Workshop 2019, OSW 2019 ; Conference date: 20-03-2019 Through 22-03-2019.
URL: *https://osw2019.sec.uni-stuttgart.de/*

Bartolomeu, P. C., Vieira, E., Hosseini, S. M. and Ferreira, J. (2019). Self-sovereign identity: Use-cases, technologies, and challenges for industrial iot, *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1173–1180.

Bormann, C. (2014). RFC 7252: The Constrained Application Protocol (CoAP) — datatracker.ietf.org, `https://datatracker.ietf.org/doc/html/rfc7252`. [Accessed 16-06-2024].

Diego, S., Regueiro, C. and Maciá-Fernández, G. (2021). Enabling identity for the iot-as-a-service business model, *IEEE Access* .

Dixit, A., Smith-Creasey, M. and Rajarajan, M. (2022). A decentralized iiot identity framework based on self-sovereign identity using blockchain, *IEEE Conference on Local Computer Networks* .

Fan, X., Chai, Q., Xu, L. and Guo, D. (2020). Diam-iot: A decentralized identity and access management framework for internet of things, *International Symposium on Blockchain and Secure Critical Infrastructure* .

Fedrecheski, G., Costa, L., Afzal, S., Rabaey, J., Lopes, R. D. and Zuffo, M. (2021). A low-overhead approach for self-sovereign identity in iot, *Global Internet of Things Summit* .

Fedrecheski, G., Rabaey, J. M., Costa, L. C. P., Calcina Ccori, P. C., Pereira, W. T. and Zuffo, M. K. (2020). Self-sovereign identity for iot environments: A perspective, *2020 Global Internet of Things Summit (GIoTS)*, pp. 1–6.

Fotiou, N., Pittaras, I., Siris, V. A., Voulgaris, S. and Polyzos, G. C. (2020). Oauth 2.0 authorization using blockchain-based tokens, *Workshop on Decentralized IoT Systems and Security (DISS) 2020* .

Fotiou, N., Pittaras, I., Siris, V. A., Voulgaris, S., Voulgaris, S. and Polyzos, G. C. (2019). Secure iot access at scale using blockchains and smart contracts, *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)* .

Fotiou, N., Siris, V. A., Polyzos, G. C., Kortesniemi, Y. and Lagutin, D. (2022). Capabilities-based access control for iot devices using verifiable credentials, *2022 IEEE Security and Privacy Workshops (SPW)* .

Grande, E. and Beltrán, M. (2020). Edge-centric delegation of authorization for constrained devices in the internet of things, *Computer Communications* .

Kang, J. and Seo, M. (2023). Enhanced authentication for decentralized iot access control architecture, *Cryptography* .

Kortesniemi, Y., Lagutin, D., Elo, T. and Fotiou, N. (2019). Improving the privacy of iot with decentralised identifiers (dids), *Journal of Computer Networks and Communications* .

Kuperberg, M. (2020). Blockchain-based identity management: A survey from the enterprise and ecosystem perspective, *IEEE Transactions on Engineering Management* **67**(4): 1008–1027.

Lagutin, D., Kortesniemi, Y. and Fotiou, N. (2019). Enabling decentralised identifiers and verifiable credentials for constrained iot devices using oauth-based delegation, *Proceedings 2019 Workshop on Decentralized IoT Systems and Security* .

Lucking, M., Luecking, M., Fries, C., Lamberti, R. and Stork, W. (2020). Decentralized identity and trust management framework for internet of things, *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* .

Mahalle, P. N. and Shinde, G. R. (2021). Oauth-based authorization and delegation in smart home for the elderly using decentralized identifiers and verifiable credentials, *Mahalle, P. N., Shinde, G. R., Dey, N., and Hassanien, A. E. (eds.) Security issues and privacy threats in smart ubiquitous computing* .

Mohanta, B. K., Sahoo, A., Patel, S., Panda, S. S., Jena, D. and Gountia, D. (2019). Decauth: Decentralized authentication scheme for iot device using ethereum blockchain, *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)* .

More, P. D., Sakhare, S. R. and Mahalle, P. (2023). Identity management in the internet of things: A survey of the state of the art, *IEEE Systems, Man, and Cybernetics Magazine* **9**(4): 13–19.

Novo, O. (2018). Blockchain meets iot: An architecture for scalable access management in iot, *IEEE Internet of Things Journal* .

Panda, S. S., Satapathy, U., Mohanta, B. K., Jena, D. and Gountia, D. (2019). A blockchain based decentralized authentication framework for resource constrained iot devices, *International Conference on Computing Communication and Networking Technologies* .

Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3, RFC 8446.
URL: *https://www.rfc-editor.org/info/rfc8446*

Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S. and Tschofenig, H. (2022). Authentication and Authorization for Constrained Environments Using the OAuth 2.0 Framework (ACE-OAuth), RFC 9200.
URL: *https://www.rfc-editor.org/info/rfc9200*

Siris, V. A., Dimopoulos, D., Fotiou, N., Voulgaris, S. and Polyzos, G. C. (2019). Oauth 2.0 meets blockchain for authorization in constrained iot environments, *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* .

Ucbas, Y., Eleyan, A., Hammoudeh, M. and Alohaly, M. (2023). Performance and scalability analysis of ethereum and hyperledger fabric, *IEEE Access* **11**: 67156–67167.

Yin, J., Xiao, Y., Pei, Q., Ju, Y., Liu, L., Xiao, M. and Wu, C. (2022). Smartdid: A novel privacy-preserving identity based on blockchain for iot, *IEEE Internet of Things Journal* .

Zhang, Y., Kasahara, S., Shen, Y., Jiang, X. and Wan, J. (2019). Smart contract-based access control for the internet of things, *IEEE Internet of Things Journal* .

Zhao, X., Zhong, B. and Cui, Z. (2023). Design of a decentralized identifier-based authentication and access control model for smart homes, *Electronics* .