RadChain Connect: Integration Between Blockchain and FreeRADIUS for Secure Authentication in Wi-Fi Networks/Web Environment

1st Eduardo Freitas Hoffmann
dept. of Computer Science and Statistics
Federal University of Santa Catarina
Santa Catarina, Brazil
eduardo.hoffmann@posgrad.ufsc.br

2rd Caciano Machado
Data Processing Center
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
caciano.machado@ufrgs.br

3nd Carla Merkle Westphall
dept. of Computer Science and Statistics
Federal University of Santa Catarina
Santa Catarina, Brazil
carla.merkle.westphall@ufsc.br

***Abstract*—User authentication in web environments and Wi-Fi networks is crucial due to the increase in fraud and invasions. Traditional methods, such as WPA2-PSK, have vulnerabilities, making the use of more secure solutions necessary. RadChain Connect is a proposal that combines blockchain with FreeRADIUS and self-sovereign identity, aiming to provide an efficient authentication system. Initially, it focuses on web authentication in universities, enhancing security at login. In the future, there are plans to expand the solution to Wi-Fi networks, leveraging the versatility of FreeRADIUS and the security of the EAP protocol.**

***Index Terms*—authentication, blockchain, eap, radius, decentralized identity, self-sovereign identity**

I. Introduction

The fallibility of traditional authentication methods, such as WPA2-PSK, highlights the need for more reliable solutions. RadChain Connect is an innovative approach that combines blockchain technology with FreeRADIUS and self-sovereign identity, aiming for a secure and efficient authentication system. The Internet of Things (IoT) represents a new technological phase, with an increasing reliance on connected devices such as tablets, smartphones, and Bluetooth devices [1]. The rise of smart devices interconnected to wireless local area networks (WLAN) and the diversity of web applications reinforce this dependency [2].

Wireless local area networks (WLAN) are widely used for communication but are often considered insecure due to their public nature, necessitating greater security concern [1]. There is a clear need for improvements in authentication methods to prevent unauthorized access and ensure that only authorized users have access [2].

Self-Sovereign Identity (SSI) can improve authentication methods. SSI is a model of decentralized digital identity management that allows users full control over their personal information, enabling its management, sharing,and verification in a secure and private manner [13] [17]. SSI is based on using decentralized identifiers (DIDs), verifiable credentials (VCs) and blockchains.

This study proposes the development of a system that integrates blockchain technology with the FreeRADIUS server and self-sovereign identity. The goal is to create a practical solution to enhance security and trust in authentication across various technological systems by recording and storing credentials on private blockchains and managing user accounts through FreeRADIUS.

A proof of concept of the proposed architecture has been developed through a simplified web system that simulates a university webpage. On this platform, students can generate their verifiable credentials and access online content, using a login system based on self-sovereign identity.

The remainder of the article is organized as follows: In Section 2, we provide the general context. Next, Section 3 discusses related works and the techniques addressed. In Section 4, we present an overview of the proposed architecture. Then, in Section 5, we delve into detailed tests and analyses of these findings. Finally, in Section 6, we discuss the conclusions drawn from this study and explore possible directions for future research.

II. Background

A. Blockchain

Blockchain technology is renowned for its security and immutability, functioning as a ledger that records data and transactions in chronologically organized blocks [3]. Each block has a unique hash, acting as a fingerprint [4]. The blockchain provides a reliable history due to the immutability of its records, facilitating the detection of suspicious activities and enhancing network security.

B. Decentralized Identity

Decentralized Identifiers (DIDs) are a new type of identifier that enables verifiable and decentralized digital identity. A DID refers to any subject, such as a person, organization, object, data model, or abstract entity [10].

Blockchain technology enables the creation of secure and self-sufficient identity systems, ensuring user privacy and reducing the risks of data breaches. This user-centered approach provides a high level of security and privacy, in compliance with regulations such as the GDPR [12].

979-8-3315-0694-0/25/$31.00 ©2025 IEEE

C. *Verifiable Credential*

A Verifiable Credential (VC) is a digital document that contains cryptographically secure and verifiable claims, issued by an entity to represent attributes of an individual, and is stored in a digital wallet [13]. VCs combine Decentralized Identities (DIDs) and blockchain, promoting decentralization. In the context of verifiable credentials and decentralized identity, there are three fundamental roles that play distinct and crucial functions in the ecosystem [14]:

- **Issuer:** Entity or organization that issues verifiable credentials.
- **Verifier:** Any entity or service that needs to verify the credentials presented by the identity holder.
- **Holder:** Individual or entity that possesses and controls their own identity and the associated credentials. [15].

D. *IEEE 802.1x Protocol*

The IEEE 802.11i standard defines the security architecture for WLAN networks, establishing a flexible key hierarchy and the exchange of these keys between client devices, such as routers, and authentication servers [2]. It utilizes the IEEE 802.1x protocol, which provides a reliable access framework in a client/server configuration [20]. Additionally, the standard employs the Extensible Authentication Protocol (EAP) to ensure a robust and reliable message exchange [2].

E. *RADIUS Protocol*

The RADIUS (Remote Access Dial-In User Service) access control protocol authenticates users through a challenge/response system and is frequently used in corporate environments that require centralized authentication, regulated authorization, and detailed accounting [21].

III. Related Research

Several studies investigate the application of blockchain in authentication systems, highlighting its ability to ensure security and transparency. Examples of this application include the authentication of IoT devices and electronic voting systems. In the context of Wi-Fi networks, the integration of blockchain with RADIUS servers and passkeys emerges as an innovative approach to enhance security and manage user credentials.

A survey conducted with keywords such as ”blockchain,” ”Wi-Fi,” ”authentication,” ”RADIUS,” ”decentralized identifiers,” and ”self-sovereign identity” in relevant sources revealed that there are works that superficially mention the intersection between authentication and blockchain. However, comparing the effectiveness of this scheme with others remains challenging, as research in this area is in its early stages, and there are few available models. Despite this, some studies stand out for their specific proposals.

Several authors have proposed various access control schemes based on blockchain (BC). [5] explore the use of DIDs, VCs, and BC that support SSI, proposing an access control model for identity management based on self-sovereign identity and attributes. An OpenID provider combined with SSI was implemented by [16] as a proof of concept, also proposing a decentralized public key infrastructure that utilizes VCs and BC, which offers a direct and verifiable way to recover digital certificates. In the work of [6], a distributed authentication system called DecAuth is proposed, utilizing blockchain technology from the Ethereum platform, demonstrating that the authentication of IoT devices can be conducted in a decentralized manner while maintaining security against known attacks.

[11] present the DT-SSIM, a decentralized and trustworthy identity management framework that adopts the concept of splitting VCs into multiple non-relatable identity shares. The goal is to prevent identity credentials from being tampered with or misused.

Continuing with the research, referring to authentication with the EAP protocol, [20] improved this authentication method by proposing a protocol called KEAPII, which aims to address three issues: dictionary attacks, Man-in-the-Middle attacks, and the problem of data accuracy received by the RADIUS server, by adding public-private key attributes and hashes, tickets, and a combination of table data with random numbers.

[7] designed and implemented an access authentication scheme that provides anonymity using Bitcoin blockchain and Intel SGX, without introducing trusted third parties, demonstrating that the proposed scheme is highly effective and practical for access control systems for public Wi-Fi hotspots.

A new approach for authenticating servers with two-factor authentication (2FA) on the blockchain platform using smart contracts is discussed in the work of [19], demonstrating that the mechanism is completely free and can be easily deployed in a private infrastructure.

Table 1 compares our proposal with various related works in the area of SSI applied to authentication via blockchain. Each row represents a characteristic addressed in each presented project.

It is important to note that none of these studies address specific characteristics related to user authentication using a FreeRADIUS server and blockchain as a ledger. Furthermore, it is observed that few research efforts have been dedicated to similar contexts, making it difficult to make comparisons and enhance this context.

TABLE I
Comparison of Other Architectures with the Proposed Architecture

Characteristics	**Proposed Scheme**	**[25]**	**[27]**	**[28]**	**[29]**	**[11]**	**[34]**	**[35]**
Platform	Ethereum Ganache	No	Indy Aries	Indy	Ethereum Ganache	Ethereum Ganache	Bitcoin	Ethereum Ganache
Authentication	DID	EAP token	SSI	SSI	AuthKey	SSI	PKI	OTP token
Application	Wi-Fi	Wi-Fi	Enterprise	IoT PKI	IoT	IoT	Public Wi-Fi	Datacenter
Smart Contract	Yes	No	No	No	Yes	Yes	No	Yes
Integration	NodeJS Express Web3 - HTML	No	VON Network	node.js OIDC Client OpenID	JavaScript API Web3	Web3	Mix SGX	OpenSSH
freeRADIUS	Yes	Yes	No	No	No	No	No	No

IV. System Overview

The proposed system consists of three main components, which aim to develop an innovative and secure authentication solution for wireless networks:

- **Blockchain (Ganache):** Stores user credentials in a secure and immutable manner within a smart contract. This practice ensures that user information is protected against unauthorized changes and fraud.
- **RadChain Connect:** An API that acts as an interface between the blockchain and the RADIUS server, allowing for an efficient flow of data between the system components.
- **RADIUS Server (freeRADIUS):** Performs user authentication in a decentralized manner, based on the information validated by the blockchain. This eliminates a single point of failure and provides a scalable solution that can accommodate a growing number of users without compromising security or performance. Can be as many RADIUS servers as needed and the failure of some of them does not compromise the service.

The objective of the system is to create a secure and efficient authentication solution for Wi-Fi networks and web environments using blockchain and freeRADIUS. The combination of these technologies aims to ensure that only authorized users can access the network and systems, providing greater security and reliability. The blockchain is used to store credentials in an immutable and transparent manner, while the RADIUS server manages the authentication process centrally.

A. *Definition of Tools and Technologies*

To enable and ensure the successful implementation of the authentication system, various tools and technologies were used in conjunction, with each playing a fundamental role in the functioning of the system. In the first phase, we employed the following tools: blockchain, self-sovereign identity, the programming language Solidity, and Web3. For the next stage, we also plan to incorporate FreeRADIUS, Node.js Express, and the implementation of the EAP protocol.

The blockchain used for testing was the Ganache platform, which is a personal development platform for decentralized application (DApp) development that simulates an Ethereum network in a secure and deterministic environment [9]. Although this blockchain was used for the development of this project, it is important to emphasize that it is not the ideal choice for production systems. Due to being a tool designed for simulation and development, Ganache has limitations in terms of scalability and security.

To develop the smart contract responsible for user registration and data retrieval management on the blockchain, we used the Solidity programming language [8].

A web application in HTML was developed to allow the generation of a Wallet and a Verifiable Credential associated with the user, utilizing JavaScript to implement the application logic, external libraries such as Web3.js, js-sha256 for hashing, jQuery, and JSON, the format used to structure the data retrieved from the smart contract, and can be seen in Fig. 1 and Fig. 2.

To code and manage the project, we used the Visual Studio Code (VSCode) IDE, which is a text editor available for Windows, macOS, and Linux, providing a lightweight yet powerful source code environment that runs on the desktop. Additionally, it comes with built-in support for JavaScript, TypeScript, and Node.js [18].

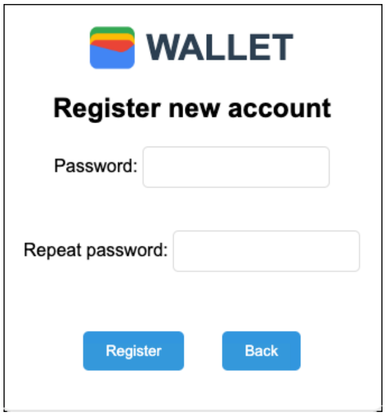
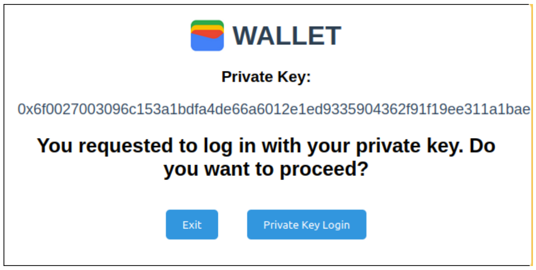


Fig. 1. User registration screen.



Private Key:

0x6f0027003096c153a1bdfa4de66a6012e1ed9335904362f91f19ee311a1bae

You requested to log in with your private key. Do you want to proceed?

Fig. 2. Wallet/private key.

B. *Use Case*

In the context of the proposed usage scenario in the first phase, the architecture aims to establish a system for authentication and management of verifiable credentials and self-sovereign identity using the private key of a wallet in an interactive web interface.

The application allows for the generation and verification of Verifiable Credentials, which are used to access the institution's website. The idea is for the issuing institution to store information (such as the course, DID, and signature) directly on the blockchain to authenticate the student, as seen in Fig. 3 and Fig. 4, respectively.

Typically, user authentication occurs through a username and a previously registered password, making credentials and network access more vulnerable to invasion attacks.

When creating an account in the University Wallet, a private key is automatically generated, which must be securely kept by the student. To log in to the University student grades page, the user will be required to log in with their private key to gain access to the system, which automatically redirects the user to their Wallet. The user must then enter the initially registered password to access their private key and perform the login.

In the usage scenario represented in Fig. 3, the intention is to present the RadChain Connect architecture, the main focus of our study, which will facilitate the integration between the FreeRADIUS server and the blockchain.

The following steps describe the sequence of the proposed architecture:

Step 1:

Registration:

1. The user accesses the university system in order to check their semester grades. They must log in with a private key stored in their Wallet. If they do not have an account, they will need to create one by entering a password, after which they will possess a private key.
2. The web interface includes, in its code configuration, among other functions, the ABI (Application Binary Interface), which is the standard way in which the Ethereum ecosystem interacts with contracts, along with the unique address of the smart contract. The received data generates a verifiable credential, which is signed and sent to the blockchain. The private key is generated and made available within the Wallet for the user.

Login:

3. The user logs into their student environment by accessing their Wallet using the password initially registered.
4. The login to the Wallet is then completed, and a message is displayed to the user: ”You requested to log in with your private key. Do you wish to proceed?” If the user clicks on ”Login with private key,” the authentication is performed using the private key recorded on the blockchain.

Step 2:

5. In the Edge Device layer, devices request authorization for secure connection to the network through routers, using the received credentials.
6. The routers connect to the RADIUS server, sending the provided credentials in order to verify their usability.
7. Since freeRADIUS cannot connect directly to the blockchain, it calls the specific functions implemented in RadChain Connect to interact with Ganache.
8. RadChain Connect, which also has the ABI and the address of the smart contract on the configured blockchain, calls the function on the blockchain that verifies whether the user is valid or invalid.
9. The Ganache blockchain sends the response to the API.
10. RadChain Connect bridges the blockchain and freeRADIUS by sending the response to the server.
11. The RADIUS server returns the request to the router, either allowing or denying the device’s access.
12. The router either authorizes or denies the device access to the network.
13. Finally, if authorized, the user gains access to the network.

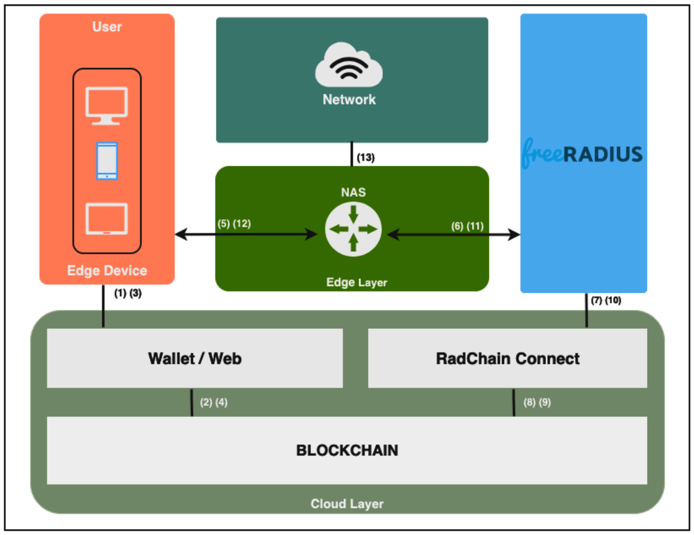


Fig. 3. Overview of the proposed architecture.

In the sequence diagram of the proposed architecture, represented in Fig. 4, it is possible to observe the interactions between the different parts of the architecture, highlighting the communications between the entities in all phases and within the two suggested steps. All entities involved in the proposed framework can communicate with the smart contract solely through the interface provided to them.

All users have a unique password to access their personal Wallet and exclusive private key, serving as an identifier.

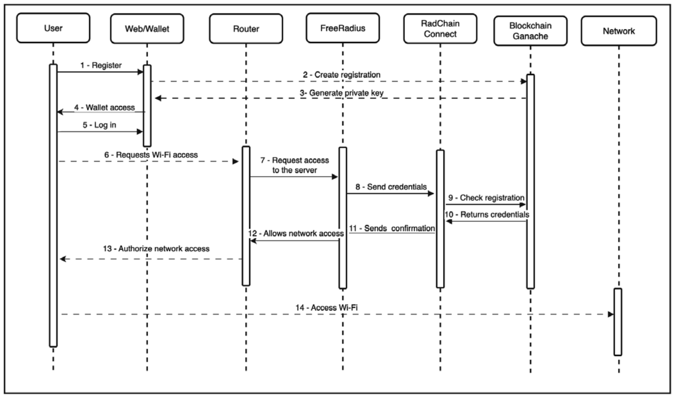


Fig. 4. Sequence diagram of the proposed architecture.

Scenario I - Web Login:

1) *User* → *Wallet/Web*: **Registration**
The user registers by accessing the web interface, in this case represented by the Wallet.
2) *Wallet/Web* → *Blockchain Ganache*: **Creates account**

The Web API sends the user's data to be registered on the blockchain.

3) *Blockchain Ganache* → *Wallet/Web*: **Generates Private Key**
Informs the user of the created private key, stored in the Wallet.

4) *Wallet/Web* → *User*: **Wallet Access**
Logs into the Wallet and provides access to the private key.

5) *User* → *Wallet/Web*: **Performs login**
The user logs into the university's web environment using their Wallet.

Scenario II - Wi-Fi Login:

6) *User* → *Router*: **Requests Wi-Fi access**
Requests connection to the network, providing the access credentials.

7) *Router* → *freeRADIUS*: **Requests access to the server**
Sends a request message to the server for access.

8) *freeRADIUS* → *RadChain Connect*: **Sends the credentials for verification on the blockchain.**
Authorizes or denies access to the network.

9 - 10 *RadChain Connect* ↔ *Blockchain Ganache*: **Verification and return of credentials**
Carries out the registration search and returns the requested data.

11) *RadChain Connect* → *FreeRADIUS*: **Sends response**
Sends the search result indicating whether the user is valid or invalid.

12) *FreeRADIUS* → *Router*: **"Permission to access the Wi-Fi network**
Returns to the router whether the user and the requested credentials have access to the network.

13) *Router* → *User*: **Network Access Authorization**
Access permission for the user.

14) *User* → *Network*: **Network access**
User accesses the network

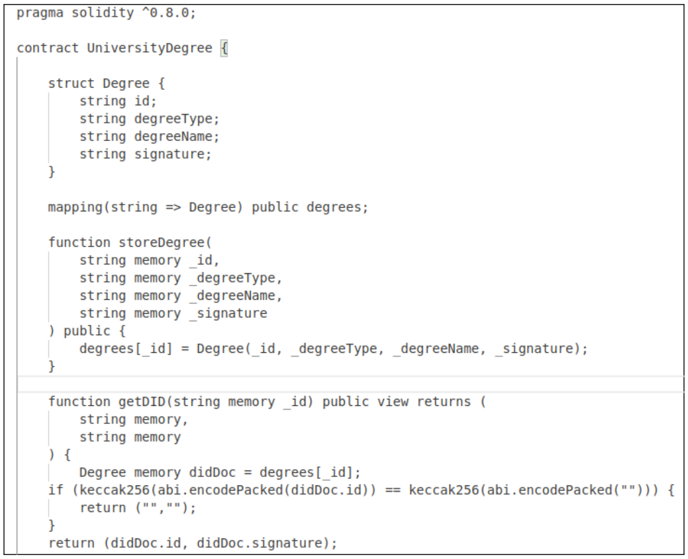
V. Experiments, Results, and Analyses

In this section, we present the results of the simulation tests and performance analyses conducted to evaluate the efficiency of our architecture.

A. Implementation:

We configured the development environment of the proposed framework on the Ethereum blockchain. During the project development process, testing was conducted on Ganache, a local testing platform that simulates an Ethereum network. The Ethereum platform is a public blockchain that also offers resources for deploying a private blockchain. There is no reason to deploy our solution on a public chain, as this would incur costs and have a considerably slower execution than on a private chain.

With the aim of achieving the necessary functionalities, we developed our own smart contract in the Solidity programming language (Fig. 5), with specific functions for registration and information retrieval, which can be called via DApp. Once deployed, the smart contract generates a unique contract address that is configured in the DApp and the interaction API.



```
pragma solidity ^0.8.0;

contract UniversityDegree {

    struct Degree {
        string id;
        string degreeType;
        string degreeName;
        string signature;
    }

    mapping(string => Degree) public degrees;

    function storeDegree(
        string memory _id,
        string memory _degreeType,
        string memory _degreeName,
        string memory _signature
    ) public {
        degrees[_id] = Degree(_id, _degreeType, _degreeName, _signature);
    }

    function getDID(string memory _id) public view returns (
        string memory,
        string memory
    ) {
        Degree memory didDoc = degrees[_id];
    if (keccak256(abi.encodePacked(didDoc.id)) == keccak256(abi.encodePacked(""))) {
        return ("","");
    }
    return (didDoc.id, didDoc.signature);
```

Fig. 5. Smart contract in solidity.

The Visual Studio Code IDE was used to develop the contract and deploy it on the blockchain network. An API using Node.js Express was developed to establish the connection between the blockchain and the RADIUS server. The unique address of the smart contract was configured in this API, along with the specific functions to perform user authentication.

To enable user registration, we developed a DApp using HTML and JavaScript to provide a user-friendly interface for the system administrator.

Finally, a router with WPA2 Enterprise technology support was configured to enable user authentication. The configuration was performed on the TPLink AX1500 router, defining the IP address of the RADIUS server and a password in the AP system. In freeRADIUS, the IP address of the router and a password were configured to enable communication between these two devices.

freeRADIUS is a highly configurable authentication and authorization server solution that can be integrated with various technologies, such as databases (MySQL, LDAP, etc.), plain text files, Active Directory, OpenVPN, RESTful API, among others.

As freeRADIUS does not have a native module for direct communication with the blockchain, it was necessary to install and configure the REST module so that it could interact with the data registered on the blockchain, enabling communication with RadChain Connect, which, in turn, would query the information on the blockchain, facilitating integration between systems and technologies. The implementation codes of the project can be found in the following repository: https://github.com/efhoffmann/RadChainConnectV1.git.

B. Results and Analysis:

To carry out the tests, we utilized a single computer system equipped with an Intel Core i5-2450M processor, 16 GB of RAM, running the Linux Ubuntu 20.04 LTS operating system. The system was implemented and tested in a controlled environment.

The tests included user registration, generation of verifiable credentials, hash calculations for authenticating existing users, and verification of credentials on the blockchain. The results of this first phase showed that integrating self-sovereign identity and blockchain for user authentication in web services offers enhanced security over traditional methods by using an encrypted private key instead of common passwords.

C. Testing and Verification:

- **User Registration:** Users were successfully registered on the blockchain through the integrated RadChain Connect system.
- **User Authentication:** Users were successfully authenticated using the credentials stored on the blockchain.
- **Security:** The immutability of the blockchain ensures that user credentials cannot be altered without authorization; once a transaction is recorded in the blocks, it cannot be modified. It is practically impossible to break the hash of a private key, which is why it is crucial to keep it stored securely.

VI. Conclusion

The integration of blockchain with self-sovereign identity offers an innovative solution for user authentication, combining the immutability and security of blockchain with the flexibility and robustness of traditional AAA systems. Although traditional authentication methods have their advantages, they possess several vulnerabilities that can be exploited.

In comparison with related works, which mostly addressed innovative authentication methods for IoT devices but did not utilize smart contracts and FreeRADIUS as tools, this project demonstrated in its first stage a practical implementation of a more secure and modern authentication method, using blockchain-based authentication and decentralized identity, providing a viable example for future expansions and enhancements in real environments.

The proposed approach can be applied in various scenarios, such as businesses and universities, improving security and trust in user authentication. As future work, we intend to continue expanding the project and utilize this authentication method in Wi-Fi networks.

Acknowledgment

This work was conducted with the support of the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) – Funding Code 001.

References

[1] Awaneesh Kumar Yadav et al. "Secure and User Efficient EAP-based Authentication Protocol for IEEE 802.11 Wireless LANs". Em: 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). 2020, pp. 576–584.

[2] Awaneesh Kumar Yadav et al. "An EAP-Based Mutual Authentication Protocol for WLAN-Connected IoT Devices". Em: IEEE Transactions on Industrial Informatics 19.2 (2023), pp. 1343–1355.

[3] Leonardo Rodrigues Carvalho, *Tecnologia Blockchain e as suas possíveis aplicações no processo de comunicação científica*. Brasilia, DF: Universidade de Brasilia, 2018.

[4] BLOCKCHAIN, BIG DATA E IOT: O QUE VOCÊ PRECISA SABER. [Online]. Available: https://operdata.com.br/blog/blockchain-big-data-e-iot-o-que-voce-precisa-saber/.

[5] Rafael Belchior et al. "SSIBAC: Self-Sovereign Identity Based Access Control," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 1935–1943.

[6] B. K. Mohanta, A. Sahoo, S. Patel, S. S. Panda, D. Jena and D. Gountia, "DecAuth: Decentralized Authentication Scheme for IoT Device Using Ethereum Blockchain," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 558-563.

[7] Y. Niu, L. Wei, C. Zhang, J. Liu and Y. Fang, "Towards Anonymous yet Accountable Authentication for Public Wi-Fi Hotspot Access with Permissionless Blockchains," in IEEE Transactions on Vehicular Technology, vol. 72, no. 3, pp. 3904-3913, March 2023.

[8] Solidity. [Online]. Available: https://docs.soliditylang.org/en/v0.8.26/.

[9] Truffle Suite. Ganache. [Online]. Available: https://archive.trufflesuite.com/docs/ganache/

[10] Drummond Reed, Manu Sporny, Dave Longley, Christopher Allen, Ryan Grant, and Markus Sabadello. 2020. Decentralized Identifiers (DIDs) v1.0 Core architecture, data model, and representations. DIDs v1.0. https://www.w3.org/TR/did-core/

[11] Efat Samir et al. "DT-SSIM: A Decentralized Trustworthy Self-Sovereign Identity Management Framework," in: IEEE Internet of Things Journal 9.11 (2022), pp. 7972–7988.

[12] Identidade Digital Descentralizada: o que é e como ela potencializa o mundo em rede. [Online]. Available: https://esr.rnp.br/temas-diversos/identidade-digital-descentralizada-esr/.

[13] Andrea De Salve et al. "A multi-layer trust framework for Self Sovereign Identity on blockchain". Em: Online Social Networks and Media 37-38 (2023). Cited by: 0; All Open Access, Green Open Access, Hybrid Gold Open Access. doi: 10.1016/j.osnem.2023.100265. url: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85167990287

[14] Seungjoo Lim et al. "A Subject-Centric Credential Management Method based on the Verifiable Credentials," in: 2021 International Conference on Information Networking (ICOIN). 2021, pp. 508–510.

[15] Modelo de dados de credenciais verificáveis v1.1. [Online]. Available: https://www.w3.org/TR/vc-data-model/dfn-verifier/.

[16] Zoltán András Lux et al. "Distributed-Ledger-based Authentication with Decentralized Identifiers and Verifiable Credentials," in: 2020 2nd Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS). 2020, pp. 71–78.

[17] Nuno Miguel da Conceição Fernandes Verdasca et al. "Análise da aceitação da identidade auto-soberana como método de identificação da identidade digital em Portugal". Tese de dout. 2023.

[18] Visual Studio Code. [Online] Available: https://visualstudio.microsoft.com/pt-br/.

[19] V. Amrutiya, S. Jhamb, P. Priyadarshi and A. Bhatia, "Trustless Two-Factor Authentication Using Smart Contracts in Blockchains," 2019 International Conference on Information Networking (ICOIN), Kuala Lumpur, Malaysia, 2019, pp. 66-71.

[20] Yi Ma e Hongyun Ning. "Improvement of EAP Authentication Method Based on Radius Server," in: 2018 IEEE 18th International Conference on Communication Technology (ICCT). 2018, pp. 1324–1328.

[21] RADIUS: Securing Public Access to Private Resources. [Online]. Available: https://books.google.com.ec/books?id=o5xQNbuvJ7QC

[22] Sahilpreet Singh Sidhu et al. "Trust Development for Blockchain Interoperability Using Self-sovereign Identity Integration," in: 2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). 2022, pp. 0033–0040.