Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P); 2583-7907 (E)
Vol.13 No.1, 2024, pp.11-17

www.crijournals.org
DOI: https://doi.org/10.70112/ajcst-2024.13.1.4253

Decentralized Authentication for Enhanced Security: Leveraging Blockchain Technology to Prevent Credential Theft

Sulemana Awal, Diyawu Mumin, Arnold Mashud Abukari, Abukari Aziz Danaa and Jibril Fuseini
Department of Computer Science, Tamale Technical University, Ghana
E-mail: awalsulemana3@gmail.com, mdiyawu@tatu.edu.gh, amashud@tatu.edu.gh, azizdanaa@tatu.edu.gh, fjibreel@tatu.edu.gh
(Received 24 January 2024; Revised 20 February 2024; Accepted 3 March 2024; Available online 18 March 2024)

Abstract - **Online services currently rely heavily on centralized authentication methods to manage user identification and authentication. However, these methods are vulnerable to account hacking, which can compromise user data and lead to attacks. A potential solution to this issue is the use of distributed ledger technology, such as blockchain, to decentralize credential ownership and provide a secure, immutable platform for verifying user identity. This paper aims to analyze the drawbacks of centralized authentication systems and propose an alternative that uses blockchain technology for authentication, ensuring robustness, transparency, and security. The proposed system is tested on web applications using the Ethereum testnet and an authentication provider (backend server).**
Keywords: **Ethereum, Authentication, Smart Contracts, Blockchain**

I. INTRODUCTION

The act of verifying that someone or something is who or what it claims to be is known as authentication. All online services and systems rely on authentication as their primary method of creating user identities so that users can access the system [1]. Today's world heavily relies on online communications for most of our daily activities, underscoring the critical need to prioritize authentication processes to prevent identity theft and other related risks. Despite the numerous legal challenges associated with the sharing of sensitive data attributes, concerns about individual privacy often go unnoticed or received insufficient attention. [2]. Users and devices can use decentralised ledger technologies like the blockchain for secure identification and authentication. The blockchain ledger authenticates and guarantees the legitimacy of transactions, clients, and messages. Blockchain-based authentication utilizes smart contracts that are maintained on a decentralized network.

This obviates the necessity of a third party to authenticate transactions and has the potential to enhance security and privacy while diminishing expenses. The distributed nature of the blockchain makes it more difficult for someone to attempt to compromise the authentication process [3]. In this method, the authentication provider holds the signing and decryption keys, while the encryption and verification keys are kept on the blockchain. This arrangement offers security against specific cyber-attacks, including man-in-the-middle and phishing assaults. This paper proposes a decentralised system for authentication that utilises blockchain technology. It also examines the architecture, technical specifications, and features employed in the authentication process. The suggested system consists of a browser plugin for users, a validation app for third-party services to verify user authenticity, and a Solidity contract. The solution will be deployed and validated on a single-node Ethereum blockchain. For decades now, criminals and fraudsters have exploited many ways to steal user login credentials from various systems and platforms, including banking websites and social media pages. Social engineering and other exploitation methods primarily harvest login credentials from users, leading to the introduction of multifactor authentication, also known as two-factor authentication (2FA). To some extent, 2FA has reduced the amount of user data and credentials theft; despite this, hackers and crackers still find ways to get user login credentials, mostly by attacking the Identity and Authorization Management (IAM) database itself.

Blockchain technology, as defined earlier, is a decentralised transaction and ledger system. One of the characteristics of blockchain technology is that it is almost impossible to manipulate or compromise the data, which means integrity [4]. This enables the implementation of a more secure solution to thwart the theft of user login credentials.

Therefore, this research will

1. Design an Ethereum contract to implement a decentralised authentication system that authenticates users and valid tokens for authorisation.
2. Create a user authorisation system that would be used by third parties to authenticate users to their services seamlessly.

II. REVIEW OF LITERATURE

Blockchain is a distributed mechanism for managing transactions and data that was initially designed for the Bitcoin cryptocurrency. The popularity of blockchain technology has been steadily growing since its inception in 2008. The interest in blockchain stems from its inherent characteristics which offer security, anonymity, and data integrity without the involvement of any third-party entities in transaction control. Consequently, this technology presents intriguing research opportunities, particularly in

terms of technical obstacles and constraints. The Ethereum blockchain architecture integrates computer programs into the blocks, which serve as representations of financial assets like bonds. These are referred to as smart contracts [5].

Since its inception, blockchain has expanded its applications and has a multitude of potential uses in the future, particularly with the emergence of Ethereum and smart contracts. Blockchain technology has been implemented in areas including healthcare delivery, finance (probably the largest area of application), security, e-governance, accounting and auditing, supply chain and so on [6]. In recent years, the rise of credential theft has emerged as a notable issue for individuals, organisations, and institutions. Traditional centralised authentication systems are vulnerable to attacks such as phishing, social engineering, and password breaches.

To address these security challenges, researchers and practitioners have explored the use of decentralised authentication systems, specifically employing blockchain technology and Identity and Access Management (IAM) principles. This literature review aims to examine existing studies and research papers that focus on preventing credential theft through the implementation of decentralised authentication systems.

A. Authentication and Credential Theft

Authentication is a fundamental security measure that ensures the integrity and confidentiality of digital resources. It verifies the identity of users, devices, or entities seeking access to a system. Authentication involves the verifying user credentials to establish their identity. As highlighted by Sadhu *et al.,* [7], strong authentication mechanisms play a pivotal role in mitigating the risks associated with credential theft, such as phishing attacks, password cracking, social engineering, etc. According to Baig and Eskeland [8], authentication is essential for safeguarding sensitive information, thwarting unlawful entry, and upholding the integrity of the system. It acts as the first line of defence against cyber threats.

Credential theft often leads to data breaches, exposing sensitive information to unauthorised individuals [9]. These security breaches can lead to monetary losses, harm to one's reputation, and legal ramifications for both individuals and companies. Stolen credentials enable identity fraud, where attackers impersonate victims to carry out malicious activities [10]. This can lead to financial losses, damage to personal reputations, and disruption of individuals' lives. Further, credential theft can result in unauthorised access to financial accounts and payment systems, leading to direct financial losses for individuals and organisations [11]. Attackers exploit stolen credentials to conduct fraudulent transactions or gain unauthorised access to funds.

Authentication plays a critical role in safeguarding digital resources against credential theft. Understanding the various forms of credential theft and their consequences is essential for implementing robust authentication mechanisms. By adopting strong authentication methods and promoting user awareness, individuals and organisations can mitigate the risks associated with credential theft and enhance overall cybersecurity.

B. Decentralised Authentication and Blockchain Technology

Decentralised authentication systems offer a novel approach to enhancing security and privacy in digital systems. This review examines the shift towards decentralised authentication and its foundation on blockchain technology, which has garnered attention for its potential to revolutionise traditional authentication mechanisms. Decentralised authentication systems distribute the authentication process across multiple entities instead of relying on a central authority. As highlighted by Khashan *et al.,* [12], this approach eliminates single points of failure and enhances resilience against attacks.

Decentralised authentication systems, as described by Al Hwaitat *et al.,* [13], eliminate the reliance on a single server or entity, making them more resilient to attacks and reducing the risk of system-wide breaches. Chen *et al.,* [14] emphasise that decentralised authentication leverages cryptography and distributed consensus to enhance security. The secure storage of user credentials mitigates the risk of unauthorised access or credential theft. Decentralised authentication empowers users with greater control over their identities and personal information [15]. Self-sovereign identities enable users to manage their credentials directly, promoting privacy and data ownership.

Blockchain is a distributed and decentralised ledger that ensures secure and transparent record-keeping. As explained by Swan [16], blockchain's fundamental concepts include distributed consensus, immutability, and cryptographic hashing. Blockchain, a distributed and tamper-resistant ledger, provides a viable method to improve authentication security. Several studies have investigated how blockchain technology is used in authentication procedures. For example, Pustokhina [17] proposed a blockchain-based authentication system that enables users to control and manage their digital identities securely. By leveraging the transparency and immutability of blockchain, this approach reduces the risk of identity theft and unauthorised access.

Blockchain networks achieve consensus without a central authority, ensuring the integrity of transactions [18]. Once data is stored on the blockchain, it becomes virtually unalterable, thereby preventing unauthorised modifications. It allows for the safe and unchangeable storage of user IDs and login credentials. [19]. This reduces the risk of identity theft and unauthorised modifications. As highlighted by Ingle *et al.,* [20], blockchain allows the creation of trust networks without centralised authorities. The blockchain's authentication events provide transparency and an auditable trail for compliance and forensic analysis. Blockchain-based authentication can enhance MFA by combining

cryptographic keys or digital signatures with traditional methods [21]. This adds an extra layer of security, reducing the likelihood of successful credential theft. Implementing blockchain-based authentication systems presents challenges such as scalability, energy consumption, and user experience [22].

III. PROPOSED METHOD

Currently, digital authentication requires a significant level of reliance on third parties. Users must trust that service providers or websites will safeguard their authentication data. This is due to the possibility of service providers or websites unintentionally obtaining personal information for the purposes of data mining, profiling, and exploitation. To enhance existing systems, we developed an authentication solution with the following capabilities:

1. Secure Authentication: With a blockchain-based system, any user's authentication must be verified at any time by multiple participating nodes, not just one. It is possible to authenticate without using a centralised authentication service.

2. Multiple Login Options: A user can send any sequence of text; it can be a hash from a fingerprint device or a password generator. The key is hashed again and sent to the contract. Developers can use keystore keys/passwords, fingerprint devices, or any form of authentication that generates a hash to identify the user on the proposed system.

3. Unified Logins: Developers can use this proposed system as a login service for their applications. One account will allow users to authenticate themselves on any application or service that uses this proposed system as an authentication service provider. The controller script would encapsulate and abstract the Solidity contract methods, allowing developers and system administrators to authenticate users via an endpoint. Because it is a blockchain project, there is transparency, and the developer can view all of the system's logs on a chosen blockchain explorer.

4. No Need for Centralised Database for Credentials Storage: the dangers of data loss and data infringement are irrelevant. The chain will only retain records of authentication transactions.

5. Immutable: The blockchain is updated using the hashed copies of the transaction logs that were produced via node consensus.

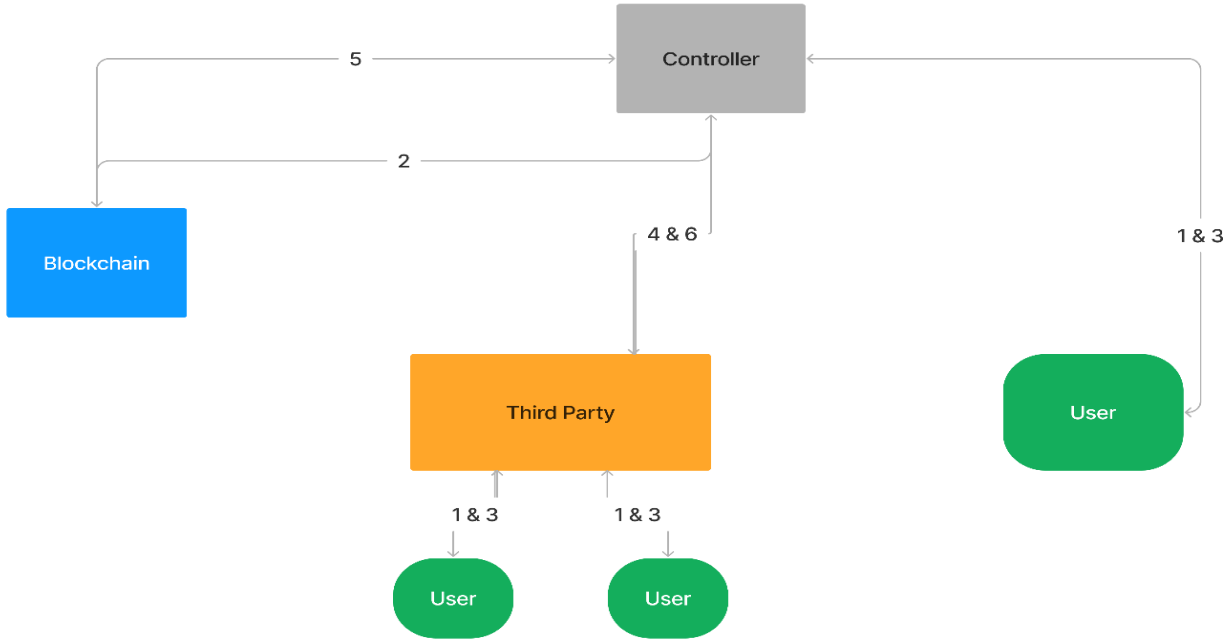


Fig. 1 Authentication Method

Data that has been encrypted cannot be changed. It is nearly impossible to reconstruct the Merkle tree to make any changes to these logs.

The proposed architecture is simple but very secure and flexible. Public users, developers, third-party web services authenticating through this solution, and the authentication blockchain are the stakeholders in my solution. The figures and diagrams below illustrate the proposed architecture. The steps involved in completing authentication on the proposed system are as follows.

1. The user registers on the system through the controller endpoint. In return, the user gets a hash token.
2. All user data is encrypted using the KECCAK256 encryption algorithm and stored on-chain.

3. The token from registration is used as a public key. This token is required for the user's login or authentication.
4. Third-party online services can also authenticate users via various means, including scanning the user token in the form of a QR code or passing the token to the controller Application Programming Interface (API).
5. The verification is done by checking the submitted token on-chain. If the user data exists on-chain then login is successful or else login fails.
6. Any service or individual requesting the login or authentication service would then receive the verification response.

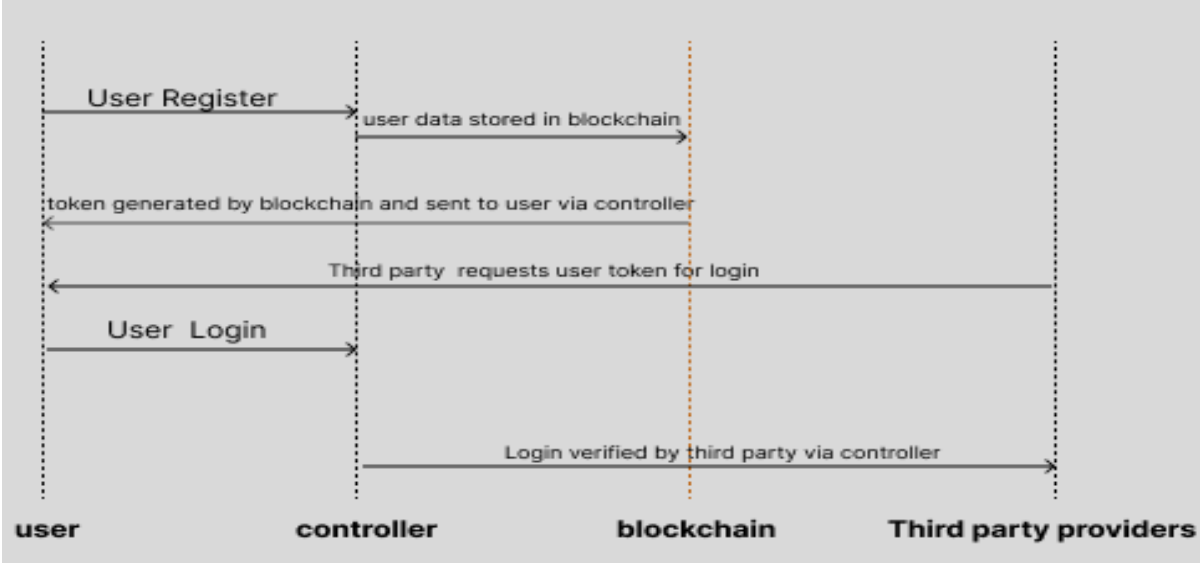


Fig. 2 Data Flow Chart

The diagram in Fig. 1 below illustrates that a single direct user can access the controller through API calls, particularly if they are developers or use a command line interface (CLI) for self-authentication. Third parties can also utilize the proposed system by authenticating users through the authentication system. It is important to note that the service would be more useful to third parties than single users, but it is still necessary to highlight the flexibility of the system.

The next diagram (Fig. 2) illustrates the flow of data from registration to user verification by a third party or a single direct user request.

A. The KECCAK256 Algorithm

Keccak-256 is a cryptographic hash function that belongs to the Keccak family of cryptographic sponge functions. It's based on the sponge construction, which is a generalisation of the Merkle-Damgård construction used in many cryptographic hash functions. In 2012, the NIST hash function competition selected Keccak as the winner, with the aim of selecting a new cryptographic hash standard, but it did not ultimately become the SHA-3 standard. Keccak-256 is mostly utilised exclusively by the Ethereum platform. It is clear that hashing techniques are not exclusive to central systems, decentralised systems, including the blockchain, employ hashing techniques. This is essential given that the blockchain, similar to any other centralised server or computer, contains sensitive data.

Storing raw data on a blockchain would allow for universal data access. This suggests that the data may not have undergone complete security preservation. Alternatively, it is preferable to save data in an encrypted format, which will result in only a hash being seen. The appearance of the content cannot be ascertained from this hash. Since you are not part of the group, this hash is irrelevant to you.

While algorithms like SHA-256 are commonly employed for password storage and data integrity verification, Keccak-256 is exclusively utilised by the Ethereum blockchain. This proposed solution can be used to authenticate users from various public websites. The flexibility and security combined make the proposed method impossible to break, but very simple to use as a solution for authentication for applications and other client services. The algorithm used for authentication and the creation of users on the proposed solution is as follows:

Algorithm 2: Login

1. If a third party requests login, it provides the user with a public key and requests user's private key for login. If it is an individual login, he provides both a salt and a private key.
2. The public key Kpl and the private key Kpr are passed into the encryption function KECCAK256(Kpr+Kpl).
3. The resulting key is then used to compare with the key onchain.
4. If the keys match, then the user is granted access.
5. Else, login fails.

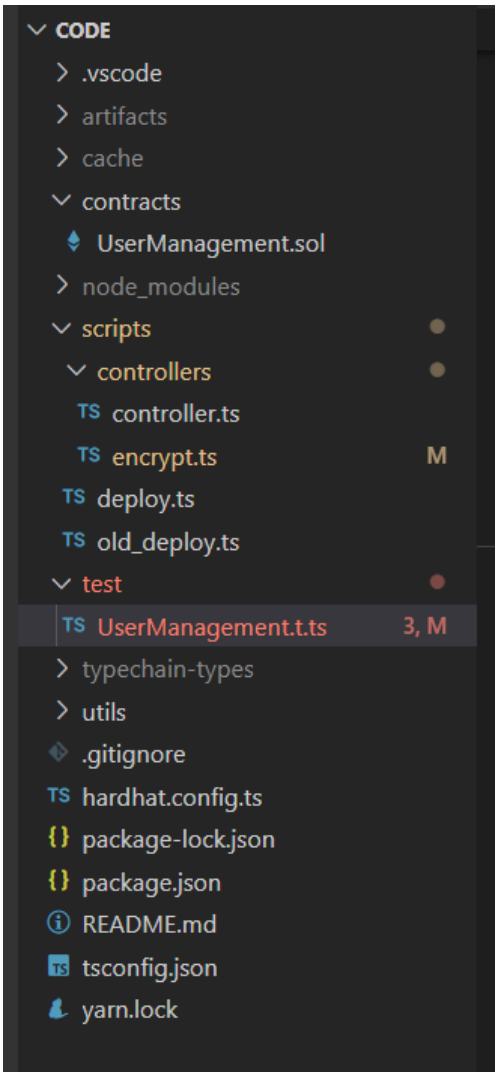


Fig. 3 Screenshots of codes

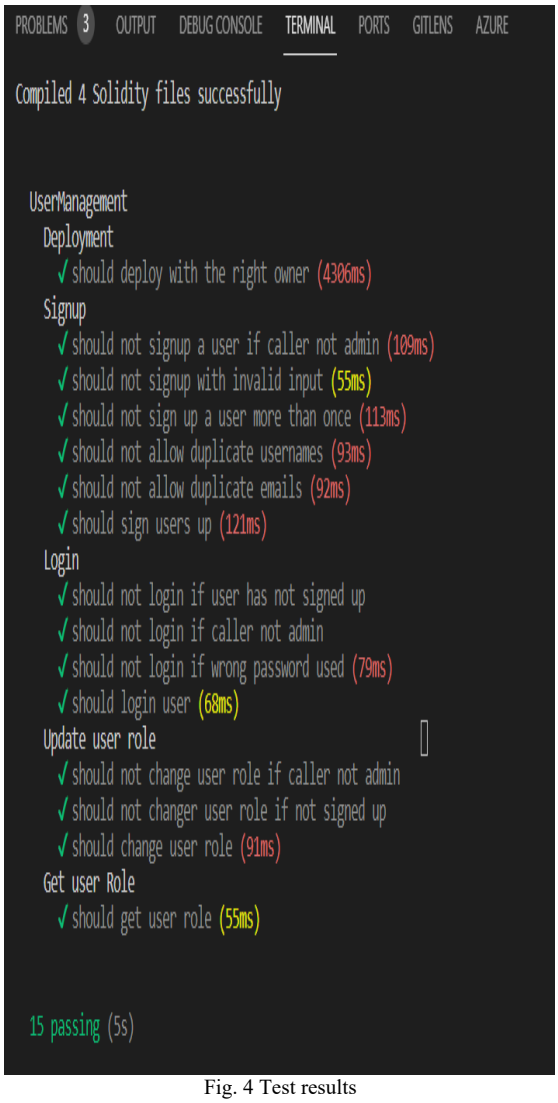
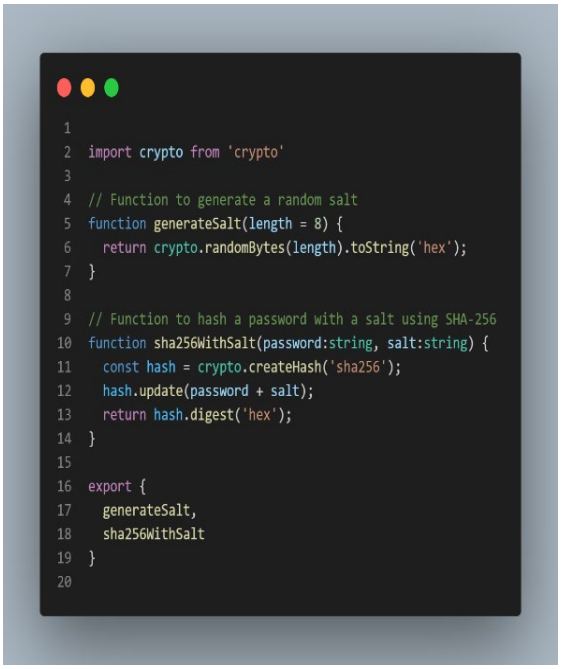


Fig. 4 Test results

In conclusion, there are several features and reasons for implementing this solution with the above algorithm. We have conducted tests and found that the above implementation significantly reduces the potential for stealing user credentials. Even if there is a man-in-the-middle attack on the client or third-party service alone, it would not be enough to get the needed information to hack or gain unauthorised access.

IV. IMPLEMENTATION

The project prototype consists of the Ethereum blockchain and a script that allows for user validation by third parties. The files in the solution follow the structure shown in Fig. 3, which was created using the hardhat library. User Management sol and User Managements are the principal files that house the blockchain and JS code, respectively. Third parties often integrate the suggested solution into larger systems, leaving the prototype without a distinct user interface; however, the test.js scripts can run all tests. Fig. 3 and Fig. 4 are screenshots of the code and test results used for the prototype:



```
1
2  import crypto from 'crypto'
3
4  // Function to generate a random salt
5  function generateSalt(length = 8) {
6    return crypto.randomBytes(length).toString('hex');
7  }
8
9  // Function to hash a password with a salt using SHA-256
10 function sha256WithSalt(password:string, salt:string) {
11   const hash = crypto.createHash('sha256');
12   hash.update(password + salt);
13   return hash.digest('hex');
14 }
15
16 export {
17   generateSalt,
18   sha256WithSalt
19 }
20
```

Fig. 5 Encryption codes

```solidity
// SPDX-License-Identifier: GPL-3.0

pragma solidity 0.8.21;

import {Ownable2Step} from "@openzeppelin/contracts/access/Ownable2Step.sol";

contract UserManagement is Ownable2Step {
    enum UserRole {
        None,
        User,
        Admin
    }

    struct User {
        bytes32 username;
        bytes32 email;
        bytes32 encryptedPassword;
        UserRole role;
    }

    mapping(address user => User userStruct) public users;
    mapping(bytes32 username => bool used) userNameExists;
    mapping(bytes32 email => bool used) public emailExists;

    event UserSignedUp(
        address indexed userAddress,
        bytes32 username,
        bytes32 email,
        UserRole role
    );
    event UserRoleUpdated(address indexed userAddress, UserRole role);

    modifier userExists(address userAddress) {
        require(
            (users[userAddress].username) != bytes32(0),
            "User does not exist"
        );
        _;
    }

    function signUp(
        bytes32 username,
        bytes32 email,
        bytes32 encryptedPassword,
        address userAddress
    ) external onlyOwner {
        require(
            users[userAddress].username == bytes32(0) &&
                users[userAddress].email == bytes32(0),
            "user already registered"
        );
        require(!emailExists[email], "Email already registered");
        require(!userNameExists[username], "username unavailable");
        require(
            username != bytes32(0) ||
                email != bytes32(0) ||
                encryptedPassword != bytes32(0),
            "Invalid input"
        );

        User memory newUser = User({
            username: username,
            email: email,
            encryptedPassword: encryptedPassword,
            role: UserRole.User
        });

        users[userAddress] = newUser;
        emailExists[email] = true;
        userNameExists[username] = true;

        emit UserSignedUp(userAddress, username, email, UserRole.User);
    }

    function login(
        bytes32 encryptedPassword,
        address userAddress
    ) external view onlyOwner userExists(userAddress) returns (bytes32) {
        User memory user = users[userAddress];

        require(
            encryptedPassword == user.encryptedPassword,
            "Incorrect password"
        );

        return user.username;
    }

    function updateUserRole(
        address userAddress,
        UserRole role
    ) external onlyOwner userExists(userAddress) {
        users[userAddress].role = role;

        emit UserRoleUpdated(userAddress, role);
    }

    function getUserRole(address userAddress) external view returns (UserRole) {
        return users[userAddress].role;
    }
}
```

Fig. 6 Solidity code

V. COMPARISON WITH CENTRALISED AUTHENTICATION SYSTEMS

A. Active Directory

Active Directory (AD) is a Microsoft-developed directory service designed specifically for Windows domain networks. It functions as a standardized and centralised system that streamlines network management of user data, security, and distributed resources. An Active Directory service performs key functions and is crucial as a centralized authentication service. Some of its functions includes centralised authentication, user and resource management, security and access control, and group policy management.

B. Custom Authentication with MySQL

We designed an authentication service using PHP as the backend language and MySQL as the database solution to store user records, with the goal of testing and verifying the usability and accuracy of the proposed solution. Table I has the values for the test on the custom setup and the proposed decentralised solution.

TABLE I COMPARISON ANALYSIS OF THE PROPOSED SOLUTION AND OTHER CURRENTLY USED SOLUTIONS

Metrics	Proposed Solution	Active Directory (Ad)	Custom Auth With Mysql
Time to authenticate	100ms .avg	2secs .avg	175ms .avg
Accessibility	Impossible to access the Ethereum blockchain as it doesn't seat on a single machine.	Can be accessed on the server that holds the authentication storage. Extra physical security is needed to secure.	Can be accessed on the hosted server. Hackers are mostly able to download or backup the database.
Cost	Initial setup costs may be lower, but management costs can vary depending on the number of authentication services.	Initial setup and management costs can be high, especially for redundancy and failover.	Initial setup and management costs can be high, especially for redundancy and failover.

VI. CONCLUSION

While the proposed solution does not aim to address all cybersecurity threats and attacks, it offers resilience against those related to user authentication and credential theft. This solution provides a secure platform for third parties to authenticate users without a single point of failure or risk of data loss. The algorithms used for encryption, the mode of transmitting data to the blockchain, and the data stored on the blockchain make it nearly impossible to hijack or steal user credentials. Despite the high level of security in the proposed

solution, the decentralized nature of blockchain raises inherent concerns that could impact its scalability, either directly or indirectly [23]. This issue could potentially hinder the adoption of such solutions, as they may not scale as efficiently as centralized authentication systems. For this solution to be highly efficient and widely adopted by third parties for user authentication, deploying it on a public blockchain, such as Ethereum, is crucial. This approach enables the solution to operate on a broader scale, as it would be more accessible to third parties compared to private hosting. Finally, it is worth noting that this proposed solution can contribute to the repository of knowledge for academic purposes, further studies, and the promotion of blockchain technology in simple, everyday systems, beyond its common association with cryptocurrencies. In conclusion, our research demonstrates the potential of blockchain technology to enhance the security of user credentials. By decentralizing and encrypting sensitive information, we have shown that blockchain can mitigate the risks associated with credential theft. The results of our study underscore the importance of exploring innovative solutions to address the growing problem of credential theft. Blockchain's immutability and distributed ledger technology have the capacity to disrupt traditional security paradigms and significantly improve protection against unauthorized access.

REFERENCES

[1] A. R. Chowdhury, T. Chatterjee, and S. DasBit, "LOCHA: A light-weight one-way cryptographic hash algorithm for wireless sensor network," *Procedia Computer Science*, vol. 32, pp. 497-504, 2014.

[2] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, 2016, pp. 25-30.

[3] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in Bitcoin P2P network," in *Proc. 2014 ACM SIGSAC Conf. Computer and Communications Security*, 2014, pp. 15-29.

[4] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, 2014, pp. 486-504.

[5] Z. Gao, "When deep learning meets smart contracts," in *Proc. 35th IEEE/ACM Int. Conf. Automated Software Engineering*, 2020, pp. 1400-1402.

[6] E. Borgsten and O. Jiang, "Authentication using smart contracts in a blockchain," 2018.

[7] P. K. Sadhu, V. P. Yanambaka, and A. Abdelgawad, "Internet of Things: Security and solutions survey," *Sensors*, vol. 22, no. 20, pp. 7433, 2022.

[8] A. F. Baig and S. Eskeland, "Security, privacy, and usability in continuous authentication: A survey," *Sensors*, vol. 21, no. 17, pp. 5967, 2021.

[9] E. Bertino, H. Lee, M. Huang, C. Katsis, Z. Shen, B. Ribeiro, *et al.*, "A pro-active defense framework for IoT systems," in *2023 IEEE 9th International Conference on Collaboration and Internet Computing (CIC)*, 2023, pp. 125-132.

[10] C. Wang, *Anti-fraud engineering for digital finance: Behavioral modeling paradigm*. Springer Nature, 2023.

[11] A. Haslebacher, J. Onaolapo, and G. Stringhini, "All your cards have belonged to us: Understanding online carding forums," in *2017 APWG Symposium on Electronic Crime Research (eCrime)*, 2016, pp. 41-51.

[12] O.-A. Khashan, S. Alamri, W. Alomoush, M.-K. Alsmadi, S. Atawneh, and U. Mir, "Blockchain-based decentralized authentication model for IoT-based e-learning and educational environments," *Computers, Materials & Continua*, vol. 75, no. 3, pp. 3133-3158, 2023.

[13] A. K. Al Hwaitat, M. A. Almaiah, A. Ali, S. Al-Otaibi, R. Shishakly, A. Lutfi, *et al.*, "A new blockchain-based authentication framework for secure IoT networks," *Electronics*, vol. 12, no. 18, pp. 3618, 2023.

[14] Z. Chen, Y. Jiang, X. Song, and L. Chen, "A survey on zero-knowledge authentication for Internet of Things," *Electronics*, vol. 12, no. 6, pp. 1145, 2023.

[15] U. Khalid, M. Asim, T. Baker, P. C. Hung, M. A. Tariq, and L. Rafferty, "A decentralized lightweight blockchain-based authentication mechanism for IoT systems," *Cluster Computing*, vol. 23, no. 3, pp. 2067-2087, 2020.

[16] M. Swan, *Blockchain: Blueprint for a new economy*. O'Reilly Media, Inc., 2015.

[17] I. V. Pustokhina, "Blockchain technology in the international supply chains," *Int. J. Wireless and Ad Hoc Communication*, vol. 1, no. 1, pp. 16-25, 2021.

[18] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[19] F. Toutara and G. Spathoulas, "A distributed biometric authentication scheme based on blockchain," in *2020 IEEE Int. Conf. Blockchain (Blockchain)*, 2020, pp. 470-475.

[20] C. Ingle, A. Samudre, P. Bhavsar, and P. Vidap, "Audit and compliance in service management using blockchain," in *2019 IEEE 16th India Council International Conference (INDICON)*, 2019, pp. 1-4.

[21] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of trust: A decentralized blockchain-based authentication system for IoT," *Computers & Security*, vol. 78, pp. 126-142, 2018.

[22] L. Yu, M. He, H. Liang, L. Xiong, and Y. Liu, "A blockchain-based authentication and authorization scheme for distributed mobile cloud computing services," *Sensors*, vol. 23, no. 3, pp. 1264, 2023.

[23] G. Almashaqbeh and R. Solomon, "Sok: Privacy-preserving computing in the blockchain era," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, 2022, pp. 124-139.