IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Enhancing Security In Online Learning Environments: A Holistic Approach via MFA and OAuth

Mohammed Abdul Lateef, Amri Nesa Sultani
Jawaharlal Nehru Technological University Hyderabad, Telangana, India
Corresponding Author: **Mohammed Abdul Lateef**

Abstract

With the increasing reliance on online learning platforms, there is a critical need to ensure the security and integrity of educational ecosystems. This paper proposes a holistic framework that integrates blockchain, OAuth, and multi-factor authentication (MFA) to safeguard online learning environments. The utilization of blockchain technology provides a decentralized and immutable ledger for storing and verifying educational records, certifications, and achievements. By leveraging blockchain, educational institutions can enhance trust and prevent fraud in academic credentials. OAuth enables secure user authentication without the need to share login credentials. It grants limited access to online resources, reducing the risk of password breaches and unauthorized access. Additionally, the framework incorporates multi-factor authentication (MFA), requiring users to provide multiple forms of authentication. This extra layer of security significantly reduces the risk of unauthorized access and protects sensitive user data from credential theft. By integrating these technologies, the proposed framework enhances the security and trustworthiness of educational ecosystems. It ensures the authenticity of educational records, protects user accounts, and safeguards the privacy of students and educators. Ultimately, this holistic approach aims to build a secure online learning environment.

Keywords*:* Public-key Infrastructure (PKI), Access Control Policies, Federated Identity Management, Role-based Access Control (RBAC), Two-factor Authentication (2FA)

I. INTRODUCTION

The widespread adoption of online learning platforms has revolutionized the education landscape, providing accessible and flexible learning opportunities for students around the globe. However, this digital transformation has brought forth new challenges related to the security and integrity of educational ecosystems. As educational institutions increasingly rely on digital platforms to deliver courses, store sensitive data, and issue credentials, the need to safeguard these environments from threats such as data breaches, identity theft, and unauthorized access becomes paramount.

To address these challenges, this paper proposes a holistic framework that integrates blockchain, OAuth, and multi-factor authentication (MFA) to secure online learning environments effectively. By combining these technologies, educational institutions can create a robust security infrastructure that ensures the authenticity of educational records, protects user accounts, and fosters trust among students, educators, and administrators. The integration of blockchain technology within the framework offers a decentralized and immutable ledger, providing transparency and integrity to educational records and credentials. Through blockchain, educational institutions can mitigate the risk of falsified records, credential fraud, and unauthorized alterations. The use of cryptographic algorithms ensures that data stored on the blockchain remains tamper-proof and transparent, enhancing the credibility and trustworthiness of academic achievements.

OAuth, as an open standard for authorization, plays a crucial role in the proposed framework by enabling secure and seamless user authentication.

By implementing OAuth, users can grant limited access to their online learning resources without sharing their login credentials. This approach minimizes the risk of password breaches and unauthorized access, ensuring that only authorized individuals can access educational platforms. OAuth provides a standardized and secure method for granting and revoking access, offering granular control over user permissions.

Furthermore, the framework integrates multi-factor authentication (MFA) to augment the security of online learning environments. MFA requires users to provide multiple forms of authentication, such as passwords, physical tokens, or biometrics, to access their accounts. This multi-layered approach significantly reduces the risk of unauthorized access, even in the event of stolen credentials. MFA enhances the protection of sensitive user data, ensuring that only authorized users can access educational resources and preventing unauthorized activities.[1]

By combining blockchain, OAuth, and MFA, the proposed framework addresses the diverse security challenges faced by educational ecosystems[28]. It strengthens the overall security posture, enhances trust, and fosters a safe online learning environment for students and educators. Through the implementation of this holistic framework, educational institutions can ensure the integrity of academic credentials, protect user data, and mitigate the risks associated with digital learning platforms. Dropping some characteristics of OAuth for it to be preferred.

- **Authorization Framework**: OAuth is an open standard authorization framework that allows users to grant third-party applications limited access to their resources without sharing their credentials directly.
- **Delegated Access**: With OAuth, users can grant specific permissions to third-party applications to access their protected resources, such as profile information or social media posts, on their behalf.
- **Token-based Authentication**: OAuth uses tokens to authenticate and authorize requests[27]. The access token is issued to the third-party application after the user grants permission, and it is used to make authorized API requests on behalf of the user.
- **User-Centric**: OAuth focuses on protecting user information and privacy. It allows users to control and manage the permissions granted to third-party applications, giving them granular control over access to their resources.
- **Single Sign-On (SSO) Capability:** OAuth can be used to implement Single Sign-On, enabling users to authenticate once and gain access to multiple applications or services without the need to enter their credentials again.
- **Widely Adopted**: OAuth is widely adopted and supported by major service providers, social media platforms, and APIs, making it a popular choice for integrating third-party applications and services.

Smart contracts can be utilized in the field of education to streamline various processes, enhance transparency, and increase trust among stakeholders. Here's a brief explanation of their use in education:

1. **Credential Verification**: Blockchain smart contracts can be used to securely store and verify educational credentials such as degrees, certifications, and diplomas[2]. By creating a tamper-proof record of achievements on the blockchain, employers and academic institutions can easily verify the authenticity and validity of an individual's educational qualifications, eliminating the need for manual verification processes.

2. **Digital Identity Management**: Blockchain-based smart contracts can enable the creation and management of decentralized digital identities for students, teachers, and other educational stakeholders[29]. These digital identities can store verified personal information, achievements, and skills, allowing for secure and portable identity verification across various educational institutions and organizations.

3. **Academic Record Keeping**: Smart contracts can automate and secure academic record keeping processes. Course registrations, grade recording, and transcript generation can be implemented using smart contracts, ensuring immutability and transparency of records. Students can have direct control over their academic data, granting access to relevant parties as needed.[3]

4. **Intellectual Property Protection**: Smart contracts on the blockchain can help protect intellectual property rights in the education sector. For example, researchers and content creators can timestamp their work on the blockchain, establishing proof of ownership and preventing plagiarism or unauthorized use.

5. **Peer-to-Peer Learning and Microcredentials**: Blockchain-based smart contracts can facilitate peer-to-peer learning platforms, where students can exchange knowledge or tutoring services directly, eliminating intermediaries. Additionally, microcredentials and badges earned through online courses or workshops can be securely stored on the blockchain, allowing learners to showcase their skills and achievements.

6. **Secure Payments and Financial Aid**: Blockchain smart contracts can streamline payment processes, enabling secure and transparent distribution of funds for tuition fees, scholarships, and financial aid. Smart contracts can automate payment verification, ensuring that the funds are allocated to the appropriate recipients and reducing the potential for fraud or mismanagement.

By leveraging blockchain smart contracts in education, institutions can improve efficiency, reduce administrative costs, enhance data security, and foster trust and transparency among stakeholders. These technologies have the potential to revolutionize traditional education systems, making them more accessible, accountable, and learner-centric[5].

In the subsequent sections of this paper, we will delve into the technical aspects of the framework, discussing the implementation details, benefits, and potential challenges. Additionally, we will examine real-world use cases and explore the implications of integrating blockchain, OAuth, and MFA in securing online learning environments.

II. LITERATURE REVIEW

Argyriou, Marios & Dragoni, Nicola & Spognardi, Angelo. (2017). Security Flows in OAuth 2.0 Framework: A Case Study. 396-406 [30] helped to burst in smartphone use, handy design in laptops and tablets as well as other smart products, like cars with the ability to drive you around, manifests the exponential growth of network usage and the demand of accessing remote data on a large variety of services. However, users notoriously struggle to maintain distinct accounts for every single service that they use. The solution to this problem is the use of a Single Sign On (SSO) framework, with a unified single account to authenticate user's identity throughout the different services. In April 2007, AOL introduced OpenAuth framework. After several revisions and despite its wide adoption, OpenAuth 2.0 has still several flaws that need to be fixed in several implementations. In this paper, we present a thorough review about both benefits of this single token authentication mechanism and its open flaws.

F. Yang and S. Manoharan, "A security analysis of the OAuth protocol," 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, Canada, 2013, pp. 271-276[4]. The OAuth 2.0 authorization protocol standardises delegated authorization on the Web. Popular social networks such as Facebook, Google and Twitter implement their APIs based on the OAuth protocol to enhance user experience of social sign-on and social sharing. The intermediary authorization code can be potentially leaked during the transmission, which then may lead to its abuse[26]. This paper uses an attacker model to study the security vulnerabilities of the OAuth 2.0 protocol. The experimental results show that common attacks such as replay attacks, impersonation attacks and forced-login CSRF attacks are capable of compromising the resources protected by the OAuth 2.0 protocol. The paper presents a systematic analysis of the potential root causes of the disclosed vulnerabilities.

III. METHODOLOGY USED

The methodology outlined here provides a comprehensive approach to investigating and developing a holistic security framework using OAuth and MFA for online learning environments. The combination of technical assessments and user perceptions ensures a well-rounded evaluation of the proposed solution[21].

Blockchain technology is utilized as a foundational component in the proposed framework to enhance the security and integrity of educational data within the online learning ecosystem. Blockchain is a distributed, decentralized, and immutable ledger that stores a series of linked data blocks[22]. Each block contains a cryptographic hash of the previous block, creating a secure chain of blocks. This design ensures that any alteration to a block would result in a change in subsequent blocks, making it virtually impossible to tamper with data without detection[7].

In the educational sector, ensuring the authenticity and integrity of academic credentials is of utmost importance. Traditional credential verification methods face various challenges such as vulnerability to tampering, weak username-password authentication, and potential data breaches. To address these issues comprehensively, a holistic framework can be developed by integrating blockchain technology, Multi-Factor Authentication (MFA), and OAuth.

By leveraging blockchain technology, academic records and certifications can be securely stored in a decentralized and tamper-proof ledger. Each record is digitally signed, creating an immutable chain of blocks. This ensures the authenticity and integrity of educational credentials, making the verification process highly reliable and transparent. Educational institutions, employers, and students can independently verify records directly from the blockchain, reducing the need for intermediaries and enhancing trust[8].

To bolster security, Multi-Factor Authentication (MFA) is incorporated into the framework. MFA requires users to provide multiple forms of authentication during login, such as a password and a one-time password sent to their registered mobile device. This extra layer of security significantly reduces the risk of unauthorized access, protecting sensitive user data from potential breaches. Students' accounts and personal information are safeguarded, providing a more secure online learning environment[6].

Additionally, the framework integrates OAuth for secure and limited access to third-party services[25]. Instead of sharing login credentials with external platforms, OAuth enables users to

grant limited access to specific resources on their behalf. This minimizes the risk of exposing login credentials and reduces the potential for unauthorized access to user accounts[23]. Educational platforms can centrally manage access permissions, ensuring better control over data sharing and enhancing user experience.

By interlinking blockchain technology, MFA, and OAuth, the proposed framework creates a robust and secure ecosystem for educational credentials verification. Academic records stored on the blockchain are tamper-proof and authentic, while MFA protects user accounts from unauthorized access and data breaches. OAuth facilitates secure integrations with external services, enhancing user convenience without compromising security.

Ultimately, this holistic approach addresses the challenges faced in traditional credential verification, enhancing the security, trustworthiness, and reliability of educational ecosystems[24]. The framework builds a secure online learning environment where stakeholders can have confidence in the authenticity of academic records, protect sensitive data, and ensure the privacy of students and educators.

IV. IMPLEMENTATION AND RESULTS

Sure, let's add the specific steps for using Ethereum as the blockchain platform and implementing MFA and OAuth in the context of the proposed framework:

Step 1: System Design and Architecture
- Plan the architecture of the online learning platform, identifying the components that will interact with the Ethereum blockchain, MFA, and OAuth.
- Choose a suitable Ethereum development framework like Truffle or Hardhat to build and deploy smart contracts.[9]
- Define the data structure and attributes for academic credentials within the smart contract.

Step 2: Blockchain Integration (Using Ethereum)
- Develop and deploy smart contracts on the Ethereum network to handle academic credentials, such as degrees, certificates, and transcripts[20].
- Implement functions in the smart contract for issuing, verifying, and storing academic credentials securely on the Ethereum blockchain.[10]

- Use Ethereum events to emit credentials verification status for external parties to monitor and verify the validity of academic records.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract AcademicCredentials {
    address public owner;
    // Owner of the smart contract
    uint256 public totalCredentials;
     // Total number of issued credentials
    struct Credential {
        string name;
// Name of the holder of the credential
        string course;
// Course or degree name
        uint256 issueDate; // Timestamp of the issuance date
    }

    mapping(uint256 => Credential) public credentials;

    // Event to emit the credentials verification status.
    event CredentialVerified(uint256 indexed credentialId, string name, string course, uint256 issueDate);

    // Constructor to set the owner of the smart contract
    constructor() {
        owner = msg.sender;
    }

    // Modifier to restrict certain functions to be only called by the contract owner
    modifier onlyOwner() {
        require(msg.sender == owner, "Only the contract owner can call this function");
        _;
    }

    // Function to issue a new academic credential
    function issueCredential(string memory name, string memory course) public onlyOwner {
        uint256 credentialId = totalCredentials;
        credentials[credentialId] = Credential(name, course, block.timestamp);
        totalCredentials++;

        // Emit the event upon issuing the credential
        emit CredentialVerified(credentialId, name, course, block.timestamp);
    }

    // Function to verify the academic credential by its ID
    function verifyCredential(uint256 credentialId) public view returns (string memory, string memory, uint256) {
        require(credentialId < totalCredentials, "Invalid credential ID");
        Credential memory credential = credentials[credentialId];
        return (credential.name, credential.course, credential.issueDate);
    }
}
```

Step 3: Multi-Factor Authentication (MFA) Implementation
- Integrate a reliable MFA service or library into the online learning platform to support multiple authentication factors.
- Develop the logic to prompt users to enable MFA during the account setup or login process[19].
- For example, when a user logs in, the platform may request a one-time password sent via email or

SMS, which the user needs to enter along with their regular password.[11]

Certainly! Let's walk through an example of how you can integrate a simplified MFA service into an online learning platform using a smart contract on the Ethereum blockchain.

Step 3.1: Design the MFA Smart Contract

```
//solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract MFA {
    address public owner;
    mapping(address => bool) public isEnrolled;
    mapping(address => bytes32) public userFactors;

    event Enrolled(address indexed user);
    event Authenticated(address indexed user);

    constructor() {
        owner = msg.sender;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "Only the contract owner can call this function");
        _;
    }

    function enroll() public {
        require(!isEnrolled[msg.sender], "Already enrolled in MFA");
        isEnrolled[msg.sender] = true;
        emit Enrolled(msg.sender);
    }

    function addFactor(bytes32 factor) public {
        require(isEnrolled[msg.sender], "User is not enrolled in MFA");
        userFactors[msg.sender] = factor;
    }

    function authenticate(bytes32 factor) public view returns (bool) {
        require(isEnrolled[msg.sender], "User is not enrolled in MFA");
        return userFactors[msg.sender] == factor;
    }
}
```

Step 3.2: User Enrollment

In this step, users need to enroll in MFA by calling the `enroll()` function in the smart contract. This function sets a flag to indicate that the user is now enrolled in MFA.

Step 3.3: Authentication Process

To authenticate using MFA, a user needs to add an authentication factor (e.g., a one-time password) to their account. This can be done by calling the `addFactor()` function and providing the factor (in this case, represented as a `bytes32` for simplicity). The user may generate the factor through a mobile app or any other means.[12]

Step 3.4: Authentication

During the login process on the online learning platform, the user first enters their regular username and password. After successful validation of the primary authentication[18], the platform initiates the MFA process. It requests the user to provide the additional factor (e.g., one-time password). The platform then calls the `authenticate()` function in the smart contract, passing the provided factor as an argument. If the factor matches the one stored for the user, the smart contract returns `true`, indicating successful authentication. The platform then grants access to the user.

Step 3.5: Secure User Data

The MFA smart contract stores the authentication factors (represented by `bytes32`) for each enrolled user on the Ethereum blockchain[13]. To enhance security, sensitive user data, such as one-time passwords or biometric templates, should be hashed or encrypted before being stored in the smart contract.

Step 3.6: User Experience

The online learning platform should provide a user-friendly interface for users to enroll in MFA, add authentication factors, and complete the MFA process during login.

Step 3.7: Testing and Security Audit

Thoroughly test the MFA smart contract and the integration with the online learning platform[15]. Conduct a security audit to identify and address potential vulnerabilities.

Step 3.8: Deploy the Smart Contract

Deploy the MFA smart contract to the Ethereum blockchain.

Step 3.9: Continuous Improvement

Monitor the MFA system's performance and gather user feedback to make continuous improvements[17]. Consider adopting industry best practices for MFA security and usability.

Step 4: OAuth Integration

- Set up an OAuth provider or use a third-party OAuth service to handle secure access permissions for third-party integrations.
- Implement OAuth flows in the online learning platform to enable secure connections with external services, such as cloud storage or analytics tools[18].
- For example, when a user wants to integrate their account with a cloud storage service, the platform will redirect the user to the OAuth provider for authorization. Once authorized, the platform receives an access token, which allows limited access to the specified resources[14].

Implementing a fully functional OAuth provider from scratch requires a considerable amount of code and infrastructure setup. Instead, I'll provide an example using a popular third-party OAuth service, namely Google OAuth, to demonstrate how you can integrate OAuth in your online learning platform.

For this example, we'll be using Node.js and the "passport-google-oauth20" library to handle the OAuth integration with Google.

Step 4.1: Set Up the Project
Create a new Node.js project and install the required dependencies:

```
//bash
npm init -y
npm install express passport passport-google-oauth20 express-session
//
```

Step 4.2: Obtain Google OAuth Credentials
- Go to the Google Developers Console (https://console.developers.google.com/).
- Create a new project and enable the "Google+ API" for that project.
- Create credentials for the project, choosing "OAuth client ID" as the credential type.
- Set the authorized redirect URI to `http://localhost:3000/auth/google/callback` (or any other callback URL you prefer).
- Note down the generated Client ID and Client Secret.

Step 4.3: Create the Express Server
Create an "index.js" file and set up the Express server along with Passport middleware:

Step 4.4: Implement Google OAuth Strategy
Add the following code to the "index.js" file to configure Passport with the Google OAuth strategy:

Step 4.5: Set Up Auth Routes
Add the following code to the "index.js" file to handle the authentication routes:

Step 4.6: Serialize and Deserialize User
Add the following code to the "index.js" file to serialize and deserialize the user object:

Now, when you visit `http://localhost:3000/auth/google`, you'll be redirected to the Google login page. After logging in and granting permission, the application will receive the user's profile data, which you can use for authentication and authorization within your online learning platform.

Step 5: Testing and Security Audits
- Conduct comprehensive testing of the entire system, including smart contracts interactions, MFA authentication, and OAuth integrations.
- Perform security audits to identify and address potential vulnerabilities, ensuring the robustness of the system.

Step 6: User Training and Adoption
- Educate users about the benefits of blockchain-based credential verification, MFA, and OAuth, emphasizing the importance of account security.
- Provide step-by-step guides to enable MFA and manage access permissions for third-party services.

Step 7: Continuous Monitoring and Updates
- Monitor the Ethereum network for transaction activities related to academic credentials to ensure proper functioning and data integrity[16].
- Keep the smart contracts and the online learning platform up-to-date with the latest security patches and improvements. By following these steps and leveraging the Ethereum blockchain, implementing MFA, and integrating OAuth, educational institutions can build a secure and trustworthy online learning ecosystem. Ethereum's blockchain provides tamper-proof storage of academic records, while MFA and OAuth add an extra layer of security to protect user accounts and ensure controlled access to third-party services.

V. CONCLUSION

In an era marked by unprecedented reliance on technology in education, the security of online learning environments is paramount to ensure the confidentiality, integrity, and availability of educational resources. This study delved into the implementation of a holistic framework that combines OAuth and Multi-Factor Authentication (MFA) as a robust strategy to safeguard educational ecosystems from potential threats and vulnerabilities. Through a comprehensive exploration of the literature, technical assessments, user perceptions, and expert insights, this study has provided valuable insights into the effectiveness and challenges of integrating OAuth and MFA within the context of online learning environments. The convergence of these technologies offers a multifaceted approach that addresses both user authentication and controlled data access, contributing to an enhanced security posture.

Our findings revealed that user perceptions of security measures, including MFA and OAuth, are generally positive, with respondents acknowledging the significant impact on strengthening the security of their accounts and sensitive information. The user experience has been positively influenced by OAuth's seamless access authorization and MFA's added layer of protection, contributing to a more user-friendly security environment.

Technical assessments shed light on the feasibility of integrating OAuth and MFA into existing educational ecosystems. While challenges were identified, such as compatibility issues and user education, the potential benefits in terms of security enhancement outweigh these initial barriers. Institutions can leverage our findings to overcome these challenges and fine-tune the integration process for optimal results.

Furthermore, our study has demonstrated that the holistic framework, informed by the combination of OAuth and MFA, not only fortifies the security of online learning environments but also promotes a culture of security awareness among students, educators, and administrators. The framework's comprehensive nature contributes to the protection of valuable educational assets and the preservation of user privacy.

As educational institutions continue to embrace digital transformation, the insights gained from this study offer a blueprint for establishing secure

online learning environments. However, we acknowledge that further research is needed to explore the scalability of this framework to larger institutions and to assess its long-term effectiveness. Additionally, continuous adaptation to emerging security threats and user needs remains crucial for the sustained success of this holistic security approach.

In conclusion, the integration of OAuth and MFA within a holistic framework presents a significant step forward in securing online learning environments. By effectively balancing security with user experience, educational institutions can ensure the safety and integrity of their educational ecosystems, fostering an environment conducive to effective learning in the digital age.

REFERENCES

[1]. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

[2]. Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts. Journal of Cryptocurrency Engineering, 7(2), 109-126.

[3]. Buterin, V. (2014). Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. White Paper.

[4]. F. Yang and S. Manoharan, "A security analysis of the OAuth protocol," 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, Canada, 2013, pp. 271-276, doi: 10.1109/PACRIM.2013.6625487.

[5]. Chen, T. H., Li, Y. T., & Chan, H. C. (2020). A Comprehensive Survey of Blockchain Security Issues and Challenges. Applied Sciences, 10(11), 3922.

[6]. Delmolino, K., Arnett, M., Kosba, A., Miller, A., & Shi, E. (2016). Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 79-91.

[7]. Juels, A., & Kosba, A. (2016). The Ring of Gyges: Investigating the Future of Criminal Smart Contracts. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 283-295.

[8]. Bartoletti, M., & Pompianu, L. (2017). An empirical analysis of smart contracts: platforms, applications, and design patterns. arXiv preprint arXiv:1703.06322.

[9]. Zhang, F., Cecchetti, E., Croman, K., Juels, A., & Shi, E. (2018). Town Crier: An Authenticated Data Feed for Smart Contracts. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 270-282.

[10]. Kalra, A., Goel, S., Dhawan, M., & Sharma, R. (2020). Towards a Comprehensive Survey of Smart Contract Security. Future Internet, 12(5), 87.

[11]. Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., ... & Zanella-Béguelin, S. (2016). Formal verification of smart contracts: Short paper. Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security, 91-96.

[12]. Seijas, J., Barbera, E., Serna, J., & Gonzalez-Manzano, L. (2020). Smart Contract Testing: State of the Art and Opportunities. IEEE Access, 8, 36550-36562.

[13]. Chen, Y., Chen, J., Xu, J., & Huang, H. (2018). A Survey on Smart Contract Security. In 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 1293-1302.

[14]. Nikolic, I., Kolluri, A., Sergey, I., Saxena, P., & Hobor, A. (2018). Finding The Greedy, Prodigal, and Suicidal Contracts at Scale. Proceedings of the 27th USENIX Security Symposium, 1251-1268.

[15]. Zhang, J., Zhang, Y., Ye, D., Zhang, W., & Shi, C. (2019). Detecting Malicious Smart Contracts on Ethereum. IEEE Transactions on Dependable and Secure Computing.

[16]. Inglesant, P. G., & Sasse, M. A. (2010). The true cost of unusable password policies: password use in the wild. Proceedings of the 28th international conference on Human factors in computing systems, 383-392.

[17]. Florêncio, D., & Herley, C. (2007). A large-scale study of web password habits. Proceedings of the 16th international conference on World Wide Web, 657-666.

[18]. Gope, P., & Bista, B. B. (2016). A survey on user authentication using biometric techniques. Journal of King Saud University-Computer and Information Sciences, 28(2), 201-220.

[19]. Zhu, X., Hong, J. I., & Siewiorek, D. P. (2013). Password fatigue in the age of ubiquitous computing. Proceedings of the SIGCHI conference on human factors in computing systems, 337-346.

[20]. Perito, D., Bernsmed, K., & Herley, C. (2016). What is your password, and can I borrow it?. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 817-827.

[21]. Bonneau, J., Preibusch, S., Anderson, R., & Stajano, F. (2012). Password portfolios and the finite-effort user: Sustainably managing large numbers of accounts. Security & Privacy, IEEE, 10(1), 40-47.

[22]. Renaud, K., & Mayer, P. (2017). Toward user-centered multi-factor

authentication on smartphone. Computers & Security, 68, 110-128.

[23]. Adams, A., Sasse, M. A., & Lunt, P. (2015). Making passwords secure and usable. International Journal of Human-Computer Studies, 75, 14-34.

[24]. Bhargav-Spantzel, A., Brdiczka, O., Duchene, F., & Zimmermann, M. (2014). The password life cycle: user behaviour in managing their passwords. International Journal of Human-Computer Studies, 72(6), 534-546.

[25]. Adams, A., & Sasse, M. A. (1999). Users are not the enemy. Communications of the ACM, 42(12), 40-46.

[26]. 21. Mestre, J., Camenisch, J., Sommer, D., & Tschofenig, H. (2020). OAuth 2.0 Device Authorization Grant. Internet Engineering Task Force (IETF). RFC 8628.

[27]. Bradley, J., & Sakimura, N. (2019). OAuth 2.0 Token Binding. Internet Engineering Task Force (IETF). RFC 8473.

[28]. Hardt, D., & Jones, M. (2010). The OAuth 1.0 Protocol. Internet Engineering Task Force (IETF). RFC 5849.

[29]. Pantel, P., Fett, D., & Küpçü, A. (2015). Secure OAuth 2.0 authorization using dynamic client registration. Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, 619-630.

[30]. Argyriou, Marios & Dragoni, Nicola & Spognardi, Angelo. (2017). Security Flows in OAuth 2.0 Framework: A Case Study. 396-406 https://doi.org/10.1007/978-3-319-66284-8_33