User Authentication Techniques Using a Dynamic SoulBound Token

Yunjae Joo and Jungwon Seo*

Department of Computer Science and Engineering, Sogang University, 915 Ricci Hall 35 Baekbeom-Ro, Mapo-gu, Seoul, South Korea
E-mail: jyj0801@sogang.ac.kr; jungwon@sogang.ac.kr
**Corresponding Author*

Received 28 January 2024; Accepted 19 June 2024

Abstract

This paper introduces a user authentication technique that utilizes a dynamic SoulBound Token (SBT) to tackle challenges associated with the oracle problem in decentralized environments. The approach uses dual smart contracts – local and global – along with blockchain tokens, removing the need for intermediary verification processes. The proposed method improves security by allowing users direct control over their authentication data, thus mitigating risks associated with centralized authorities and man-in-the-middle attacks. The feasibility and efficacy of this approach are demonstrated through a location-based prototype, indicating significant potential for application in Web 3.0 ecosystems. This paper also provides a comprehensive security analysis, underscoring the robustness of the proposed system against cyber threats.

Keywords: Dynamic SoulBound Token, decentralized authentication, oracle problem, Web 3.0 authentication.

Journal of Web Engineering, Vol. 23_5, 717–734.
doi: 10.13052/jwe1540-9589.2356

1 Introduction

Web 1.0 marked the inception of the internet, primarily focused on static text content. With the advent of Web 2.0, people witnessed the rise of platform-controlled internet services [1]. Today, people are witnessing the emergence of the Web 3.0 era. Initially, Web 3.0 was defined as an internet environment incorporating ontologies and semantic technologies [2, 3]. However, the definition of Web 3.0 has evolved and expanded to encompass a blockchain-based internet ecosystem [4].

Web 3.0, as a paradigm, endeavors to forge a user-centric internet environment by harnessing the power of blockchain technology. Its core mission is to confront the challenges posed by centralization, data monopolization, and misuse prevalent in the existing Web 2.0 landscape. To truly usher in the era of Web 3.0, substantial efforts are imperative to decentralize technologies traditionally confined to centralized environments. In particular, innovating user authentication technology within the internet environment emerges as an inevitable challenge.

Conventional authentication techniques, such as username and password combinations, fingerprints, and X.509 certification, have traditionally relied on trusted authorities to verify users. For example, when a user creates a username and password, a trusted authority stores this information along with user details such as email addresses and phone numbers. However, traditional authentication techniques that depend on centralized authorities are susceptible to the misuse of user information [5]. The incident involving Diginotar in 2011 serves as a stark reminder of the potential vulnerabilities in certificate authorities and the risks associated with man-in-the-middle attacks. In that case, a Dutch certificate authority issued a fraudulent certificate that allowed attackers to impersonate Google's services, leading to significant losses and security concerns [6].

The conventional authentication methods relying on centralized institutions do not align with the philosophy of Web 3.0, which prioritizes the protection of users' data sovereignty. Additionally, they are not well-suited for Web 3.0 environments where blockchain technology should be leveraged.

Authentication research utilizing blockchain, such as authentication tokens [7, 8] or decentralized identifiers (DIDs) [9], for Web 3.0 environments has been advancing. These innovative techniques are not dependent on centralized authorities, empowering users to retain control over their data.

Additionally, blockchain ensures the transparency and integrity of users' data in these methods.

Blockchain-based authentication technologies encounter a challenge in verifying a user's real-world existence, commonly referred to as the *oracle problem*. For instance, ensuring that the data provided by the user for authentication is indeed correct poses difficulties without centralized authority. The *oracle problem* revolves around securely and reliably acquiring external data by a blockchain network.

To tackle this challenge, research efforts have explored solutions such as linking trusted authorities to the blockchain network [10] or selecting trustworthy nodes through reputation measurements [11]. However, these approaches essentially reintroduce elements of centralization to address the *oracle problem*, reminiscent of conventional techniques reliant on centralized authorities, especially in the context of authentication.

Hence, this paper proposes a blockchain-based authentication technique tailored for the Web 3.0 environment, aimed at resolving the *oracle problem* associated with verifying user data reliability. The proposed technique uses a SoulBound Token (SBT) to directly store authentication data related to the user on the blockchain. Subsequently, the service provider authenticates the user based on this information, enabling the user to engage in self-authentication without the need for a centralized authority or intermediary agency.

The paper makes several significant contributions:

- It introduces a dynamic SBT technique as a solution to the oracle problem, providing a novel approach to address a critical challenge in blockchain authentication.
- The paper presents a user-centric authentication technique that eliminates the need for third-party intermediaries, enhancing security and control for both service users and providers.
- The practical applicability of the proposed approach is demonstrated through the implementation of a location-based prototpye, confirming its feasibility and effectiveness.

Following the introduction in Section 1, Section 2 describes the background knowledge to understand the proposed approach, and Section 3 introduces related works that include papers addressing the blockchain oracle problem and those utilizing blockchain tokens. Section 4 introduces

the proposed approach, Section 5 presents the experiments, and Section 6 concludes.

2 Preliminaries

This section provides an overview of the SoulBound Token (SBT) and the smart contract utilized within the proposed approach.

2.1 SoulBound Token (SBT)

The SBT draws inspiration from the concept of *SoulBound* items found in the MMORPG game World of Warcraft, initially conceived by Ethereum founder Vitalik Buterin [12]. In the realm of World of Warcraft, a *SoulBound* item is intrinsically linked to a game character, rendering it untradeable and non-transferable. Vitalik adapted this concept to propose SBT as a token that could embody the unique characteristics of blockchain users [13]. While there is currently no established Ethereum standard for SBT, discussions suggest that its form will resemble that of a non-fungible token (NFT) [14]. Unlike NFT which is permissioned and transferable, SBT is not. Essentially, SBT is envisioned as a means of recording and encapsulating human experiences and activities [15].

2.2 Smart Contract

Smart contracts, a groundbreaking concept initially proposed by Nick Szabo [16], have played a pivotal role in the blockchain network, particularly since Ethereum's adoption of this technology [17]. These smart contracts essentially comprise code that runs on a blockchain and can execute a wide array of services within the blockchain network.

Solidity, a well-established programming language, serves as the preferred tool for crafting smart contracts. Within the smart contract, Solidity offers fundamental data access control capabilities. For instance, Solidity introduces crucial keywords such as *modifier, require*, and *msg.sender* to regulate access. The *modifier* keyword defines a function that users can use both before and after executing a smart contract. It acts as a means to alter the behavior of Solidity's function, enabling users to assess whether specific conditions are met in advance. The *require* keyword, on the other hand, bolsters the integrity of smart contracts by validating input values and triggering an error if they fail to satisfy predetermined criteria. Additionally, the *msg.sender* keyword reveals the address of the account responsible for

invoking the smart contract, allowing users to ascertain the identity of the transaction executor. This feature can be used to determine if the contracting parties possess the necessary authorization.

3 Related Work

This section introduces related research aimed at addressing the *oracle problem* [18, 19], as well as studies focusing on blockchain-based authentication [20–22].

Beger and Huber introduced the OracleLink technique as a potential solution to the *oracle problem* in blockchain [18]. OracleLink focuses on the election of specific nodes responsible for retrieving external data through smart contracts, and these elected nodes are entrusted with the task of fetching external data. While OracleLink aimed to enhance the reliability of oracle nodes through smart contracts, it is important to note that there is a limitation associated with the potential for data manipulation due to malicious actions by nodes.

Nelaturu and Keerth introduced a study aimed at addressing the *oracle problem* through the implementation of a decentralized oracle network [19]. In this research, Nelaturu's approach seeks to enhance the trustworthiness of oracles by establishing a network comprised of specific oracle nodes responsible for delivering trusted data to the blockchain network through the consensus of these nodes. However, it's worth noting that this study has a limitation as it may not completely eliminate trust-related issues that could arise during the election process of distributed oracle nodes.

Kim and Ryou presented a study addressing user authentication by leveraging the user's DID and SBT, with the aim of mitigating the limitations associated with blockchain-based authentication techniques reliant on cryptocurrency wallets [20]. In their paper, they proposed an approach wherein the user is endowed with an identity authentication entity through the SBT issuer via DID authentication. While this study introduces a technique utilizing SBT for identity authentication, there remains a potential limitation concerning the security of the user's identity information in the presence of an SBT issuer.

Zhao and Si introduced a study focused on authenticating user identities using NFT [21]. They highlighted the challenge of adequately verifying the true owner of an NFT during the NFT transaction process. To address this, they proposed a technique that links NFTs to social network services (SNS) to establish ownership of NFT. However, it's worth noting that Zhao's research

may have limitations, as using SNS accounts for owner verification could potentially lead to unintended exposure of the owner's identity information.

In another study proposed by Fotiou [22], users request an access token from an authentication server using the authentication authorization they receive from the resource owner. The authenticator sends a token in the form of ERC-721 to both the user and the blockchain server along with the authentication authorization from the resource owner. The user then presents the received token to the resource server in order to access the resource, and the resource server makes a determination on authentication by matching the token with the token stored in the blockchain. However, Fotiou's research has a drawback: if the certifier is compromised during the issuance of authentication tokens or faces internal issues, it may lead to problems where tokens are issued to unauthorized individuals, potentially compromising the security of the server.

4 Approach

This section introduces a novel authentication approach aimed at resolving the *oracle problem* and eliminating the need for intermediary and centralized authority. The overall process of the proposed approach is depicted in Figure 1.

The proposed approach comprises three phases: the initialization phase, aimed at preparing for authentication; the pre-authentication phase, focused on storing data for authentication; and the authentication phase, dedicated

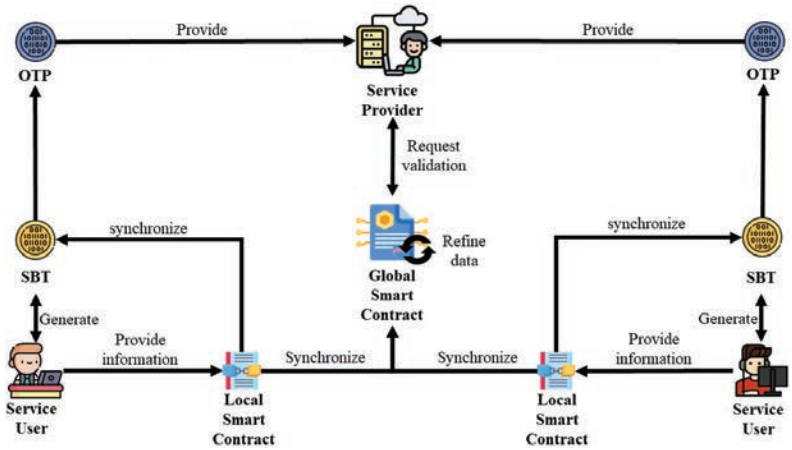


Figure 1 Overview of the proposed approach.

Table 1 Notations table

Notation	Description
su	Service user
sp	Service provider
$\mathbb{G}$	Global smart contract
$\mathbb{L}$	Local smart contract
ad	Blockchain address
SBT_{id}	SBT identifier
OTP_{id}	OTP identifier
$user_{info}$	User's information

to performing authentication. Figure 1 illustrates the proposed approach and Table 1 provides the notations used in this approach. It is assumed that both the service user and service provider possess knowledge of each other's global smart contract and local smart contract's blockchain addresses, their respective blockchain addresses, and each other's signature values. Additionally, there is a mutual agreement within each smart contract regarding the data to be stored.

The entities involved and utilized in the proposed approach are as follows:

- Service user (su): The service user, also known as the end-user, is the entity responsible for issuing the SBT and local smart contracts. It maintains control over its user information, which can be stored in the local smart contract whenever the user deems it necessary. When requesting authentication from a service provider, an OTP token linked to the SBT is generated and provided to the service provider.
- Service provider (sp): The service provider is responsible for delivering services to the service user and creating the global smart contract. Upon receiving an OTP token from the user, the service provider verifies the information contained in the OTP token using the global smart contract. If the OTP token is validated as correct, the service provider proceeds to deliver the requested service to the service user.
- Global smart contract ($\mathbb{G}$): The global smart contract is a smart contract generated by their service provider, tasked with overseeing the data maintained by the local smart contracts according to predefined rules established by the service provider. When the service provider initiates OTP verification, the global smart contract assesses whether the information within the OTP aligns with the data stored within the global smart contract. It's worth noting that a global smart contract can be linked to multiple local smart contracts.

- Local smart contract ($\mathbb{L}$): The local smart contract is a smart contract responsible for directly storing information utilized for authentication by the service user. When the user inputs specific data, this information is synchronized with both the global smart contract and the SBT. It's important to note that a local smart contract can be associated with multiple SBTs.
- SBT: The SBT or SoulBound Token, is a blockchain token generated by the service user. The user has the ability to update their stored information within the SBT by inputting their preferred authentication information into the local smart contract.
- OTP: The OTP, or one-time password, is a blockchain token generated by the service user when initiating an authentication request with the service provider. It is linked to the SBT and takes a specific format associated with it.

4.1 Initialization Phase

This section outlines the process of creating a $\mathbb{G}$, $\mathbb{L}$, and an SBT for authentication. It should be designed in a way that ensures only the sp can access the information stored within it. Additionally, it should be configured to allow only $\mathbb{L}$ to serve as intermediaries for data storage in $\mathbb{G}$. The $\mathbb{G}$ manages $\mathbb{L}$ using a map structure. Algorithm 1 provides a pseudocode representation of the design for $\mathbb{G}$.

$\mathbb{L}$ should be configured to permit only the su to input data while preventing any read access to the stored data. Furthermore, it should be responsible for syncing data with SBT and $\mathbb{G}$ with the user input specific data. Within $\mathbb{L}$, the SBT is managed using the SBT_{id}, which serves as the identifier for SBT. Algorithm 4.1 provides a pseudocode representation of the design for the $\mathbb{L}$.

The creation of the SBT can occur simultaneously as the service user generates an SBT identifier, SBT_{id}, through a smart contract. The SBT_{id} is generated by combining a random value (R_l) supplied by the local smart contract with the user's blockchain address (su_{ad}) using the XOR operation ($SBT_{id} = R_l \oplus su_{sig}$). Algorithm 4.1 provides a pseudocode representation of the design for an SBT.

4.2 Pre-authentication Phase

This section delineates the procedure by which a service user (su) stores data in a local smart contract ($\mathbb{L}$) and generates a one-time password (OTP) token for authentication purposes. The pre-authentication phase is depicted in Figure 2.

Algorithm 1 Global smart contract design

[Deploy]
input: *msg.sender*, $\mathbb{L}_{ad}$
owner ← [ ]
L_manager ← {address, string}
Constructor(*msg.sender*, $\mathbb{L}_{ad}$**)** {
    owner ← *msg.sender*
    L_manager ← {L_data, null}
}
[Read]
Input: *msg.sender*, $\mathbb{L}_{ad}$
require (owner == *msg.sender*)
allow to read L_manager[$\mathbb{L}_{ad}$]
if not
    deny
[Write]
input: $\mathbb{L}_{ad}$, data
require (L_manager[L_ad] is exist)
require ($\mathbb{L}_{ad}$ == *msg.sender*)
allow to write L_manager[$\mathbb{L}_{ad}$] ← data
if not
    deny

Algorithm 2 Local smart contract design

1: **[Deploy]**
2: **input:** *msg.sender*
3: owner ← [ ]
4: SBT_manager ← {string, string}
5: **constructor (*msg.sender*)**{
6: owner ← *msg.sender*;
7: }
8: **Write**
9: **input:** *msg.sender*, data, SBT_{id}
10: **require** (owner == *msg.sender*)
11: allow to write data to SBT_{id}
12: SBT_manager ← {SBT_{id}, data}
13: synchronize (SBT_{id}, $\mathbb{G}_{ad}$)

The su accesses their own $\mathbb{L}$ using the blockchain address associated with their user account. Access to the $\mathbb{L}$ can be restricted if the blockchain address attempting access is not properly registered within the $\mathbb{L}$, a verification process that occurs upon deployment.

An authorized user can periodically store authentication information in the $\mathbb{L}$ that they wish to use. For instance, if the su and the service provider

Algorithm 3 SBT design

[Deploy]
Input: *msg.sender*, su_{sig}
owner ← [ ]
constructor (*msg.sender*){
    owner ← *msg.sender*;
    }
$R_l \leftarrow$ generated($\mathbb{L}_{ad}$)
$SBT_{id} \leftarrow R_l \oplus su_{sig}$

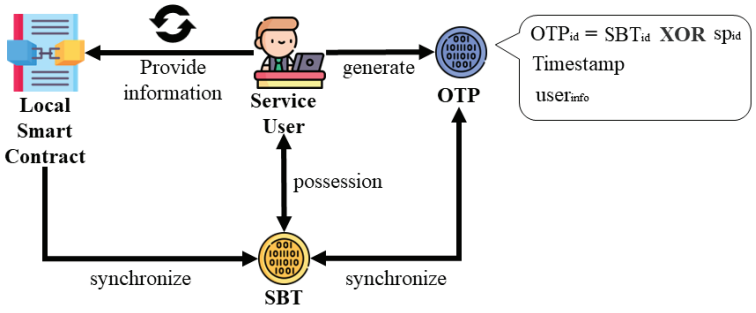


Figure 2 Process of the pre-authentication phase.

(sp) have pre-agreed that the su will authenticate using location data, the su can continually input this information into the $\mathbb{L}$.

When desired, the information that the user wishes to use for authentication is continuously stored in the $\mathbb{L}$. When authentication is needed, an OTP token is generated and made available for the sp.

The OTP token comprises OTP_{id}, $Timestamp$, $user_{info}$. ($OTP = \{OTP_{id}, Timestamp, user_{info}\}$). The OTP_{id} can be created by performing an XOR operation between the SBT_{id} and sp_{sig} ($SBT_{id} \oplus sp_{sig}$). Notably, only the sp receiving the OTP from the su can read the contained information. After a single use, all data values in the OTP are reset to zero, rendering them unusable. Additionally, once utilized, $user_{info}$ is hashed and stored in the global smart contract ($\mathbb{G}$) to prevent future reuse.

4.3 Authentication Phase

This section details the procedure by which the sp receives an OTP token from the su and proceeds with user authentication based on that token. The authentication phase is depicted in Figure 3.

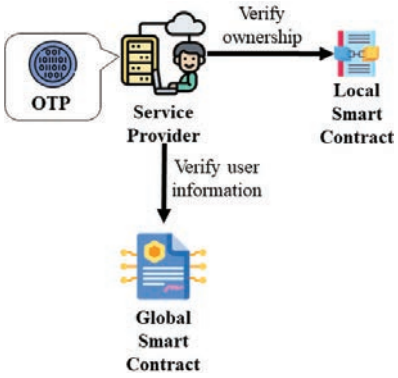


Figure 3 Process of the authentication phase.

During the authentication process, the sp verifies whether the OTP provided by the su was generated by the SBT. The sp uses the received OTP_{id} to perform an XOR operation with sp_{sig}, thereby deriving the SBT_{id} ($OTP_{id} \oplus sp_{sig}$). With the obtained SBT_{id}, the sp then conducts an XOR operation with su_{sig} to obtain R_l ($SBT_{id} \oplus su_{sig}$). This R_l value is subsequently forwarded to the $\mathbb{L}$ for validation. The $\mathbb{L}$ checks if the R_l was generated by itself and verifies the ownership address of the $\mathbb{L}$.

If it is confirmed that R_l has been generated by $\mathbb{L}$ and the ownership address is accurate, the sp deems the OTP transmission successfully completed. The sp then accesses the $\mathbb{G}$ issued by them to retrieve the $user_{info}$ provided by $\mathbb{L}$. Subsequently, the sp compares the $user_{info}$ contained in the OTP with the $user_{info}$ obtained from $\mathbb{L}$. If the two sets of information match, the sp concludes that the user information remains trustworthy and consistent, permitting the authentication process to proceed. This comparison between the OTP data and $\mathbb{G}$ records effectively safeguards against data tampering and synchronization errors during transmission.

5 Experiments

5.1 Security Analysis

This section conducts a comprehensive security analysis of the proposed approach. It is important to emphasize that this analysis excludes considerations of 51% attacks or Sybil attacks that specifically target the blockchain network. Instead, the focus lies on demonstrating the challenging nature of man-in-the-middle and brute-force attacks for potential adversaries ($\mathbb{A}$).

A man-in-the-middle attack involves an $\mathbb{A}$ intercepting and possibly altering communications between two parties. In the proposed approach, for a man-in-the-middle attack to succeed, the $\mathbb{A}$ may attempt to compromise various communication channels, including: (1) $\mathbb{L} \rightarrow \mathbb{G}$, (2) $\mathbb{L} \rightarrow SBT$, (3) $SBT \rightarrow OTP$, (4) $OTP \rightarrow sp$.

In each of these communication phases, the $\mathbb{A}$ may attempt to interfere with the information exchange. However, it is crucial to emphasize that man-in-the-middle attacks are highly unlikely to succeed in any of these processes. The proposed approach leverages blockchain technology for all authentication and communication processes, inherently employing private key-based signature technology. As such, for an $\mathbb{A}$ to interfere, they would need to obtain the private key of either the su or the sp – a practically unfeasible endeavor. Additionally, the tokens or smart contracts containing $user_{info}$ have access control features that prevent access without compromising the associated private key.

In theory, an $\mathbb{A}$ might try to steal the private keys of the su or sp via a brute-force attack to gain access to $user_{info}$ stored in the $\mathbb{G}$, $\mathbb{L}$, or the SBT. However, the likelihood of such an attack succeeding is extremely low.

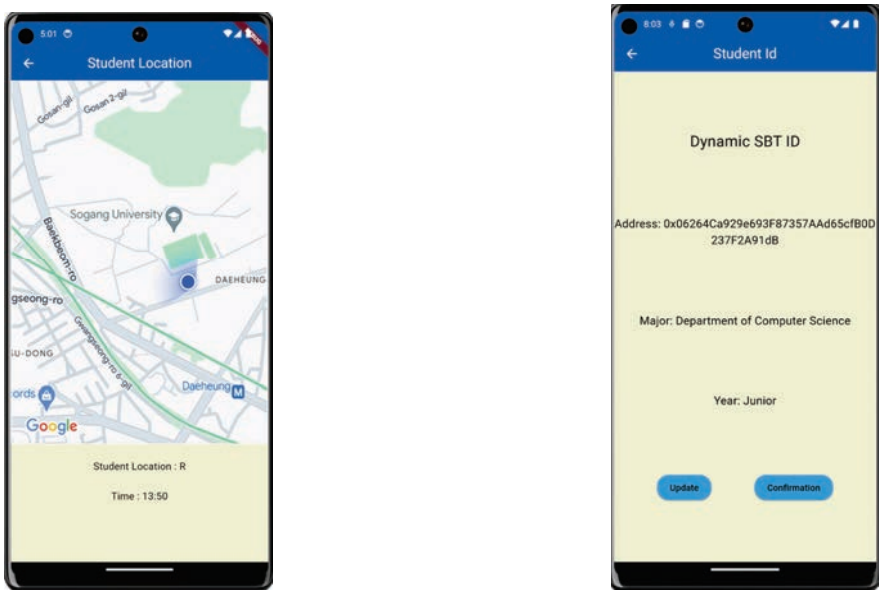
Private keys used in blockchain technology are generated based on BIP-0039 [23]. This standard involves selecting 12 to 24 words from a pool of 2048 to create a seed phrase. The combination possibilities for a 12-word phrase are approximately 5.27×10^{39}, calculated as $(2048!/(2048 - 12)!)$, which far exceeds the estimated age of our galaxy, 1.4×10^{10}.

Practically speaking, the sheer number of possible private key combinations renders brute-forcing the private keys of su and sp computationally unrealistic. Therefore, the system is effectively secure against such attacks.

5.2 Implementation

This section presents the results of experiment designed to assess the viability of the proposed approach within a decentralized Web 3.0 environment. The user interface screen of the prototype built for the experiment is shown in Figure 4. The approach emphasizes providing authentication mechanisms, such as tokens, directly from individuals to service providers without requiring an external verification process. The primary objective of this experiment is to empirically evaluate the feasibility of applying the proposed approach to a location-based authentication system.

To ensure accurate acquisition of location information, this experiment utilized timetable data from students in the Department of Computer Science,



(a) Location User Interface

(b) Dynamic SBT User Interface

Figure 4 Prototype user interface.

Bioengineering, and Economics at Sogang University. This approach allowed for the creation of a dataset comprising 1200 unique timetables, representing 100 students from each department and grade level.

To implement the $\mathbb{L}$ and $\mathbb{G}$ components required for the proposed approach, Solidity version 0.8.0 was utilized. Additionally, Hyperledger Besu was used to establish the blockchain network essential for the experiment. In this experiment, students were provided with dynamics SBT student cards, enabling them to periodically record their location data in $\mathbb{L}$. When accessing a service, the data stored within the SBT is converted into an OTP, which is subsequently transmitted to the sp. The sp then compares and evaluates the location information in the OTP against the pre-learned timetable data to verify the student's major and grade level for authentication.

The results of this experiment demonstrate the stability of synchronization and data processing procedures between $\mathbb{L}$ and $\mathbb{G}$, SBT, and OTP within the proposed approach. Furthermore, this experiment used machine learning models, including XGBoost, LightGBM, and random forest, achieving an impressive accuracy rate exceeding 96% during the authentication process.

This experiment underscores the potential of personal authentication system utilizing dynamic SBTs within the context of Web 3.0 environments. By introducing a decentralized authentication technique based on location information, this experiment showcases the feasibility of implementing an efficient and secure authentication system. This research is significant as it substantiates the viability of developing a diverse authentication system leveraging the proposed techniques.

6 Conclusion

This paper introduces an authentication technique leveraging dynamic SBT to address the prevailing oracle problem in blockchain-based authentication. It presents a method for authenticating service users to service providers without intermediaries, using a combination of two smart contracts and two blockchain tokens. To validate the proposed approach, this paper provides a safety analysis and demonstrates its successful implementation through a practical prototype focused on location-based authentication.

For future research, further exploration of diverse use cases is suggested, along with the development of prototypes that harness the potential of the proposed approach across various applications.

Acknowledgment

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2023-00259099) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00397538, Development of public opinion polling technology based on web3 that ensures fairness, anonymity, and transparency, 50%).

References

[1] Seneviratne, O., McGuinness, D. 2023. Web 3.0 Meets Web3: Exploring the Convergence of Semantic Web and Blockchain Technologies. *CEUR Workshop Proceedings*. Available online: https://ceur-ws.org/Vol-3443/ESWC_2023_TrusDeKW_paper_247.pdf.

[2] Nath, K., Dhar, S., Basishtha, S. 2014. Web 1.0 to Web 3.0 - Evolution of the Web and its various challenges. *2014 International Conference on Reliability Optimization and Information Technology*. doi:10.1109/ICROIT.2014.6798297.

[3] Aghaei, S., Nematbakhsh, M.A., Farsani, H.K. 2012. Evolution of the World Wide Web: From Web 1.0 to Web 4.0. *International Journal of Web & Semantic Technology*. doi:10.5121/ijwest.2012.3101.

[4] Guan, C., Ding, D., Guo, J. 2022. Web3.0: A Review and Research Agenda. *2022 RIVF International Conference on Computing and Communication Technologies*. doi:10.1109/RIVF55975.2022.10013794.

[5] Zarrin, J., Phang, H.W., Saheer, L.B., Zarrin, B. 2021. Blockchain for decentralization of internet: prospects, trends, and challenges. *Cluster Computing*. doi:10.1007/s10586-021-03301-8.

[6] Rao, C., Lin, Z. 2021. VAPKI: A Blockchain-Based Identification System with Validation and Authentication. *2021 7th International Conference on Computer and Communications*. doi:10.1109/ICCC54389.2021.9674554.

[7] Putri, M.C.I., Sukarno, P., Wardana, A.A. 2020. Two factor authentication framework based on ethereum blockchain with dApp as token generation system instead of third-party on web application. *Register*. doi:10.26594/register.v6i2.1932.

[8] Kamboj, P., Khare, S., Pal, S. 2021. User authentication using Blockchain based smart contract in role-based access control. *Peer-to-Peer Networking and Application*. doi:/10.1007/s12083-021-01150-1.

[9] Sporny, M., Longley, D., Sabadello, M., Reed, D., Steele, O., Allen, C. 2022. Decentralized Identifiers (DIDs) v1.0. *W3c*. Available online: https://www.w3.org/TR/did-core/.

[10] Han, X., Yuan, Y., Wang, F.Y. 2019. A Blockchain-based Framework for Central Bank Digital Currency. *2019 IEEE International Conference on Service Operations and Logistics, and Informatics*. doi:10.1109/SOLI48380.2019.8955032.

[11] Adler, J., Berryhill, R., Veneris, A., Poulos, Z., Veira, N., Kastania, A., 2018. Astraea: A Decentralized Blockchain Oracle. *2018 IEEE International Conference on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data*. doi:10.1109/Cybermatics_2018.2018.00207.

[12] Buterin, V. 2022. Soulbound. *Vitalik Buterin's Website*. Available online: https://vitalik.eth.limo/general/2022/01/26/soulbound.html.

[13] Ohlhaver, P., Weyl, E.G., Buterin, V. 2022. Decentralized Society: Finding Web3's Soul. *SSRN*. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4105763.

[14] Cabot-Nadal, M.A., Playfor, B., Payeras-Capella, M.M., Gerske, S., Mut-Puigserve, M., Pericas-Gornals, R. 2023. Private Identity-Related Attribute Verification Protocol Using SoulBound Tokens and Zero-Knowledge Proofs. *2023 7th Cryber Security in Networking Conference*. doi:10.1109/CSNet59123.2023.10339754.

[15] Tejashwin, U., Kennith, S.J., Manivel, R., Shruthi, K.C., Nirmala, M. 2023. Decentralized Society: Student's Soul Using Soulbound Tokens. *2023 International Conference for Advancement in Technology*. doi:10.1109/ICONAT57137.2023.10080658.

[16] Szabo, N. 1997. The Idea of Smart Contract. *Nick Szabo's Papers and Concise Tutorials*. Available online: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html.

[17] Hegedus, P. 2018. Towards Analyzing the Complexity Landscape of Solidity Based Ethereum Smart Contracts. *2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*. Available online: https://ieeexplore.ieee.org/document/8445056.

[18] Berger, B., Huber, S., Pfeifhofer, S. 2020. OraclesLink: An architecture for secure oracle usage. *2020 Second International Conference on Blockchain Computing and Applications*. doi:10.1109/BCCA50787.2020.9274455.

[19] Nelaturu, K., Adler, J., Merlini, M., Berryhill, R., Veira, N., Poulos, Z., Veneris, A. 2020. On Public Crowdsource-Based Mechanisms for a Decentralized Blockchain Oracle. *IEEE Transactions on Engineering Management*. doi:10.1109/TEM.2020.2993673.

[20] Kim, G., Ryou, J. 2023. Digital Authentication System in Avatar Using DID and SBT. *Mathematics*. doi:10.3390/math11204387.

[21] Bellagarda, J., Abu-Mahfouz, A.M. 2022. Connect2NFT: A Web-Based, Blockchain Enabled NFT Application with the Aim of Reducing Fraud and Ensuring Authenticated Social, Non-Human Verified Digital Identity. *Mathematics*. doi:10.3390/math10213934.

[22] Fotiou, N., Pittaras, I., Siris, V.A., Voulgaris, S., Polyzos, G.C. 2020. OAuth 2.0 authorization using blockchain-based tokens. *arxiv*. doi:10.48550/arXiv.2001.10461.
[23] Bip-0039. *gits*. Available online: https://github.com/bitcoin/bips/blob/master/bip-0039/bip-0039-wordlists.md.

Biographies

Yunjae Joo is currently enrolled in the master's degree program in Software Engineering & Blockchain at Sogang University. Additionally, he graduated from Sangmyung University with a major in Software Engineering.

Jungwon Seo holds a Ph.D. in Software Engineering & Blockchain from Sogang University. Additionally, he earned a Master's degree in Computer Science & Engineering from Sogang University in March 2020, specializing in Software Engineering & Blockchain. He also graduated with a Bachelor's degree in Management Information Systems from the State University of New York at Buffalo's Business Department in May 2016.