Journal of Multimedia Information System VOL. 9, NO. 4, December 2022 (pp. 275-286): ISSN 2383-7632 (Online)
https://doi.org/10.33851/JMIS.2022.9.4.275

Ethereum-Based User Authentication Model Using EOG Data for a Service Provider

Ki Hyeon Hong[1], Byung Mun Lee[2*]

Abstract

Unlike general passwords, user authentication technology using biometric data cannot be lost or forgotten, and it has the advantage of being impossible to forge or falsify by attackers. Since this biometric data contains sensitive information, a safe storage method is needed. However, if biometric data is managed on a central server, it is limited by being vulnerable to integrity infringement attacks by system attackers and persistent infringement attacks on the authentication service. To solve this problem, a model using blockchain-based information security technology is proposed in this paper. The aim is to safely manage the user biometric data by dispersing the storage of the biometric data feature information for user authentication in smart contracts and IPFS using Ethereum. We have verified that it takes an average of 1,472.733 ms and 217.829 ms, to register and authenticate a user in the proposed model and we expect that it will provide a reliable authentication service to the users compared to the existing authentication method in which the biometric data is managed by a single server.

Key Words: Blockchain Network, EOG, Personal Identification, Smart Contract.

I. INTRODUCTION

User authentication is a means of verifying the access authority of a user in the information security field. Among several means, user authentication technology utilizing biometric data uses data measured directly from the user such as electroencephalography (EEG), fingerprints, iris recognition, and electrocardiogram (ECG) [1-2]. Among them, electrooculography (EOG) is brainwave data measured near the forehead when the user blinks naturally. User authentication utilizing such biometric data is very secure due to the fact that it cannot be lost, since it uses the physical characteristics of the user, and that an attacker cannot arbitrarily forge or falsify it [3-4].

Biometric data must be protected safely from being seized by attackers since it includes health or physical information of the user [5-6]. However, there were cases where it was seized easily by attackers because security for managing the stored biometric data was inadequate. For example, 27.8 million cases of biometric data were leaked from the database of Suprema [7].

The limitation of a biometric data storage method like this result from the characteristics of personal identification using biometrical data. For example, Article 6 of the Standard for Protection of Personal Information recommends hashing the identification information for user authentication before storing it so that it is not decoded [8]. However, since biometric data is measured differently each time, similarity between measured data must be evaluated for personal identification using biometric data [9-10]. Therefore, biometric data for user authentication is very difficult to hash. In previous studies, an asymmetric encryption method that encrypts biometric data with a public key and decrypts it with a private key was used [11]. However, there is a risk of private keys being leaked by the administrator, and storing and managing biometric data on a central server has the limitation of being very vulnerable to attempted forging or falsifying by attackers or Denial of Service (DoS) attacks.

To resolve these limitations, this paper proposes a model that can store the biometric data of users safely, in a distributed manner, and authenticate users by utilizing Ethereum. The proposed model registers the feature information for identifying individuals in the user biometric data as NFT for user authentication in Ethereum. The original biometrical data is distributed and stored in the InterPlanetary File System (IPFS). In order to provide an authentication service to users, the service environment and the biometric data storage environment are separated with a user EOG

Manuscript received November 09, 2022; Revised November 30, 2022; Accepted December 08, 2022. (ID No. JMIS-22M-11-043)
Corresponding Author (*): Byung Mun Lee, +82-31-750-4756, bmlee@gachon.ac.kr
[1]Department of IT Convergence, Gachon University, Seongnam, Korea, ghdrlgus96@naver.com
[2]Department of Computer Engineering, Gachon University, Seongnam, Korea, bmlee@gachon.ac.kr

authentication server. As a result, attackers cannot forge or falsify the biometric data that has been stored in a distributed manner, and the authentication server provides continuous authentication services to users by utilizing biometric data feature information from the Ethereum network.

Eye-blinking EOG data and the Ethereum blockchain are examined in Section 2 of this paper. In Section 3, an eye-blinking EOG user authentication model utilizing Ethereum is proposed to resolve the limitations that were presented. In Section 4, experiments to evaluate the efficiency of the model, the integrity of the biometric data, and the durability of the authentication service are designed and analyzed. Lastly, this paper is concluded in Section 5.

II. RELATED RESEARCH

2.1. Personal Identification Technology Based on Eye-Blinking EOG

Biometric data are physical and behavioral information that are measured from a specific person, and various data can be obtained depending on the measurement criteria or position [12-13]. In particular, biometric data cannot be forgotten or lost since physical data is used, and they are very secure since they cannot be forged or falsified easily by attackers [1-2]. Among them, electrooculography (EOG) data, which occurs when the eye blinks, measures the unique potential value (μV) that is dependent on the size of the eyeball or the speed at which the eye blinks, and it has the characteristics that are shown in Fig. 1 [14].

When a person blinks, the position of the eyeball moves up and down naturally. The cornea of the eye has a positive potential and the retina has a negative potential [15]. Therefore, the potential of EOG displays a change from a positive potential to a negative potential. In previous studies, 10 unique pieces of information were extracted from the EOG, as shown in Fig. 1, and used as features for personal identification [14].

For example, in Fig. 1, the duration of the positive potential is HPL, and the duration of the negative potential is defined as LPL. In addition, the maximum value of the positive potential is HP, and the minimum value of the negative potential is defined as LP. When the similarity acceptance criterion was set to ± 1 μV in 10 feature information, 93% of personal identification accuracy was confirmed for 10 users [14].

EOG data does not require a separate user authentication method from the users in healthcare fields like sleep care. For example, brain waves are measured near the forehead to measure the sleep stages in a personalized service that provides sleep-inducing sound. The user may blink naturally during this process, and may not pay attention to the user authentication process that uses the EOG data [16].

Such EOG data provide a more secure authentication service against integrity infringement attacks from attackers by integrating record-based authentication like passwords [17]. For example, an attacker cannot seize personal information if EOG data authentication fails even if he/she has acquired the password [18].

2.2. Previous User Authentication Services and their Limitations

User authentication is the process of identifying a user to obtain access authorization for personal information. While information for user authentication is diverse, such as physical access keys, passwords, and biometric data, they basically have an operating structure like the one shown in Fig. 2 [19-20]. $User_1$ in Fig. 2 is a password user. $User_1$ first presents the ID and the password to the user authentication system. The authentication system hashes and stores the password. If $User_1$ requests the authentication system to verify the ID and the password, the authentication system compares whether the hashed password matches and returns the result.

$User_2$ in Fig. 2 is a biometric data user. $User_2$ presents the ID and the biometric data to the authentication system. Since the authentication system must evaluate the similarity of the biometric data, it cannot hash and store it. If $User_2$ requests verification in the future, the similarity of the biometric data is evaluated, and the authentication result is returned. That is, the user authentication service can be divided into registration and verification stages, and the registered information is managed by the authentication system.

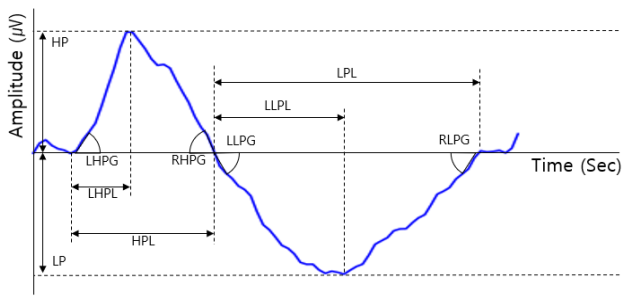

Fig. 1. Features from EOG data.

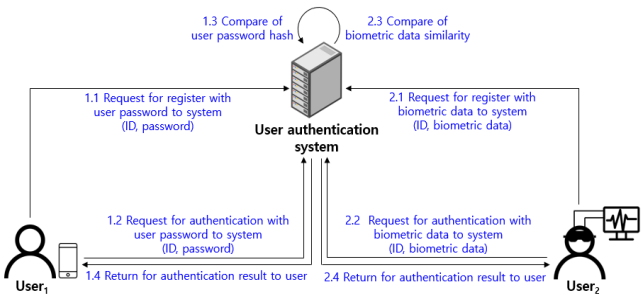

Fig. 2. Scenario for user authentication.

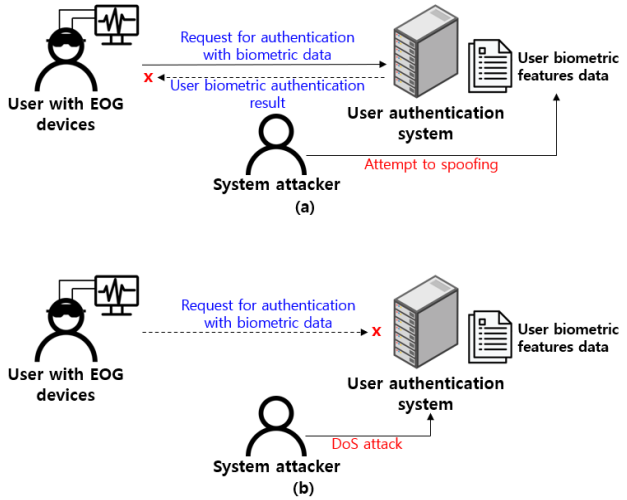

Fig. 3. Threats for user authentication, (a) Spoofing by attacker, (b) DoS attack by attacker.

If an attacker forges or falsifies the biometric data stored in the authentication system or paralyzes the authentication system, the users cannot receive the normal authentication service. For example, Fig. 3(a) shows a situation in which an attacker attempts an integrity infringement attack and forges or falsifies the biometric data feature file that is stored in the authentication system. Since the attacker does not directly own the user's biometric data, user authentication is not possible. However, the user biometric data feature information that is registered in the authentication system can be falsified with the attacker's or arbitrary biometric data feature information. In this case, the user will not be able to access the user authentication system by using the existing biometric data, and the attacker may become the one who succeeds in user authentication with the forged or falsified biometric data.

Fig. 3(b) shows a situation in which an attacker attempts a persistent infringement attack and paralyzes the authentication system. For example, the authentication system that has received a Denial of Service (DoS) attack cannot process the user's authentication service request. To resolve these limitations, it is necessary to ensure not only the confidentiality of the biometric data that is stored in the authentication system but also, it is an integrity, and the continuity of the authentication service provided by the authentication system must be maintained.

2.3. Information Security of Blockchain and the Ethereum Network

Blockchain is a database that connects the information in the blockchain nodes participating in the blockchain network into a chain, and it is a kind of information management technology [21-22]. Data is linked in chain form and registered as blocks on the blockchain to ensure integrity, and the blocks are generated according to the consensus algorithm to verify the integrity of the blocks. For example, in the proof of work (PoW) method that is used in the Bitcoin network, the nonce value is added when the hash value is calculated so that the calculated hash value is below a certain number. The hash value is calculated with the transaction information, the previous block hash value, and the nonce value. Therefore, if the transaction information of the previous block or the transaction information of the current block is forged or falsified, the hash value and the nonce value must be recalculated. Since this requires an enormous amount of computation, forgery or falsification of the blockchain is impossible.

In addition, since the blockchain is periodically synchronized within the blockchain network, all blockchain nodes provide the same service. Therefore, even if an attacker paralyzes a specific blockchain node with a DoS attack, the user will be guaranteed continuity of service through another blockchain node [23].

Among blockchains, Ethereum is a blockchain service that supports the implementation of a smart contract function [24]. In order to operate a distributed application, a smart contract links with the user's web or app to create, read, update, and delete (CRUD) the desired information, if the conditions are met. In the Ethereum blockchain network in Fig. 4, the blockchain is periodically synchronized, and the Ethereum Virtual Machine (EVM) byte code is registered and managed in the transaction of the Ethereum blockchain. This EVM byte code is a smart contract, and it owns the state, and changes and manages the state according to transactions generated by the user. If a user registers information such as his or her copyright or personal information, it is a unique smart contract state, and it cannot be replaced since it can be distinguished from other smart contract states.

The state of such an irreplaceable smart contract can be defined as a non-fungible token (NFT). For example, a user can create a smart contract and register the metadata of the

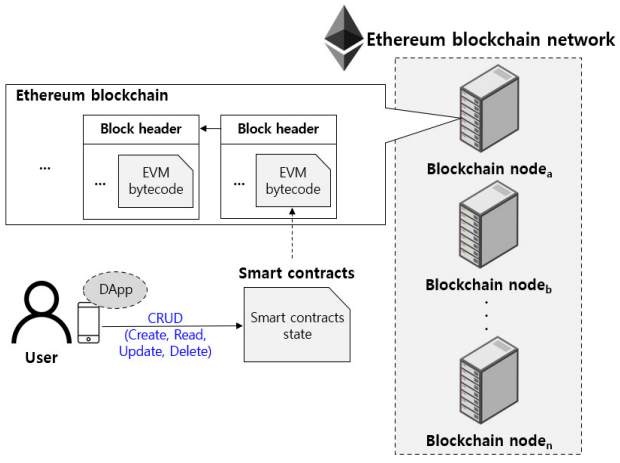

Fig. 4. Structure of Ethereum and smart contracts.

photos or videos he or she owns as an NFT. Since smart contracts do not provide the storage for large amounts of data such as photos and videos, the actual data is stored in the IPFS system in a distributed manner and the address value for access is stored in the smart contract state. The smart contract created like this is unique, and it cannot be forged or falsified because it is signed with the Ethereum public key of the owner.

Therefore, we would like to propose a method to safely manage EOG data for user authentication in this study. To this end, we propose a model that configures an eye blinking EOG user authentication model using Ethereum, registers the EOG features NFT, and authenticates users.

In the proposed model, EOG data is used as feature information for user authentication. In addition, we would like to propose a model that can be expanded since the smart contract can register diverse biometric data according to the design.

III. EYE-BLINKING EOG USER AUTHENTICATION MODEL USING ETHEREUM

3.1. Eye-Blinking EOG User Authentication Model Using Ethereum

The eye-blinking EOG user authentication model using the smart contract of Ethereum is a model that was proposed so that eye-blinking EOG features information can be managed as an NFT for user authentication, as shown in Fig. 5. In general, user authentication is essential when content needs to be provided only to members. In this paper, we propose a model that registers the EOG features NFT for user authentication and authenticates users through it.

It is provided by separating the Ethereum blockchain node that manages EOG data and the user EOG authentication server that provides the user authentication service, as shown in Fig. 5. The functions of the authentication server are largely divided into three types. They are the NFT registration process, the user authentication process, and the NFT forgery or falsification inspection and recovery process. First, while providing the eye-blinking EOG data to the user authentication server, a request is made to register it as an EOG features NFT (①). In the registration module of the server, the EOG data is safely stored as a file in IPFS (②). This is to securely protect the user's EOG data from attackers. Finally, an EOG features NFT is generated by extracting the EOG feature information, and is registered in a smart contract (③).

In order to be provided with contents, the user makes a request to the content provider for membership registration (④). The content provider makes a request for user authentication to the user authentication server in order to provide contents only to the member (⑤). The authentication module of the server receives the returned EOG features NFT that was registered by the user (⑥). This is to compare it with the EOG data that was presented to the content provider by the user to receive the contents. If the comparison result matches, the content provider authenticates the user as a member (⑦).

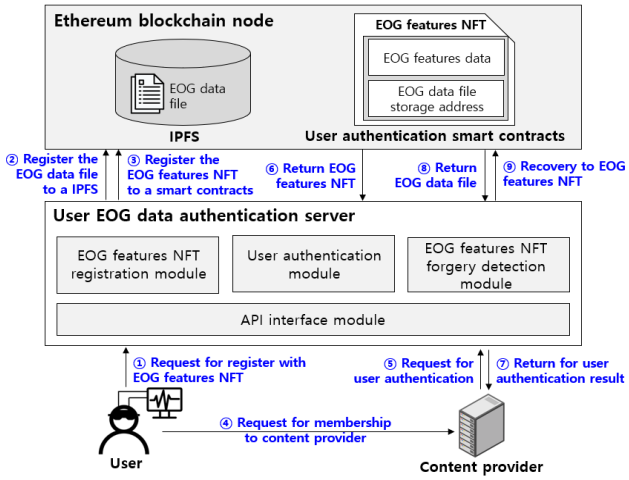

Fig. 5. Structure of user authentication model for EOG data using Ethereum.

When user authentication fails, as in Fig. 5 ⑦, it can be seen as one of two situations. The first is the case where a non-member user has made a request for user authentication. The second is the case where the EOG features NFT has been forged or falsified by an attacker. In the second situation, the user who owns the EOG features NFT cannot access the user authentication service. Therefore, the forged or falsified EOG features NFT must be recovered. The user's EOG data is needed to recover the forged or falsified EOG features NFT, and this can be done by using the EOG file of IPFS that was stored in Fig. 5 ②.

The forgery and falsification recovery module receives the returned EOG file of IPFS (⑧). If the EOG feature information is extracted from the EOG file, the extracted feature information will be identical to the EOG features NFT before it was forged or falsified. Therefore, the forged or falsified EOG features NFT is recovered (⑨).

In Fig. 5, it can be seen that personal information for user authentication is not stored in the authentication server. However, the personal information is distributed and stored in the Ethereum blockchain network and IPFS. Therefore, the user's personal information cannot be acquired even if the attacker attacks the authentication server. In addition, even if an attacker attempts a DoS attack, other authentication servers provide the user authentication service to the

users and content providers by using the EOG features NFT that has been stored in a distributed manner.

During this process, if the EOG data file is seized by the system attacker, the attacker can carry out the registration and authentication with the user's EOG data. Therefore, the user presents a private key together with the EOG data to encrypt the EOG features NFT, and the authentication server registers the encrypted EOG data and the EOG features NFT.

3.2. EOG Features NFT Registration Module

In order for a user to be provided with the user authentication service, the user must first register the EOG features NFT in a smart contract. This is as shown in Fig. 6. The user makes a request for registration to the authentication server after measuring the EOG data. The measured EOG data is the potential data that was measured when the user blinked, as shown in Fig. 1, and it is an array of real numbers. In the authentication server, the EOG data is distributed and stored in IPFS to securely protect it. The result stored in IPFS is the address with which the EOG file can be accessed from IPFS. This address is identical to the hash value of the stored EOG data. If an attacker attempts to forge or falsify an EOG file stored in IPFS, the hash value and address value will not match, making it impossible to change the EOG file. The authentication server extracts the 10 feature information that were defined in Fig. 1 from the EOG data to generate the EOG features NFT. In addition, the storage address of the EOG file is recorded together in the NFT so that it can be used in the future to confirm that the EOG feature information and the EOG file that is stored in IPFS match.

The user authentication smart contract, which registered the EOG features NFT, is electronically signed with the Ethereum public key that was presented by the user for registration. Therefore, the owner of the EOG features NFT is the user who provided the Ethereum public key. When a smart contract is registered in the Ethereum blockchain, it returns the address of the smart contract. The user can verify the EOG features NFT that he or she had registered by presenting a smart contract address to the Ethereum blockchain.

3.3. User Authentication Module

In order to provide contents only to members, the content provider must register the users as members. During this process, it is necessary to confirm whether the user is someone who can be trusted by verifying the EOG data that was presented by the user. The EOG data can be verified by confirming that the feature information of EOG data that was presented by the user and the feature information that is stored in the EOG features NFT are similar. To this end, the authentication server provides a service for authenticating a user by receiving the EOG data and the smart contract address. This is as shown in Fig. 7.

First, in order to be provided with contents, the user makes a request to the content provider for membership registration by presenting the EOG data and the smart contract. The content provider makes a request for verification of the EOG data to the authentication server. Since user authentication evaluates similarity, an evaluation criterion is needed. Since the EOG features NFT is registered in the smart contract address, it receives the returned EOG feature information for the similarity evaluation.

The authentication server extracts the feature information from the EOG data that was presented by the user to evaluate the similarity with the EOG feature information and returns True if it is judged to be the same user, and False if it does not match. As the result, the content provider receives the returned user authentication result through the EOG data and the smart contract address and stores the smart contract address for the case where the user can be trusted.

Like this, the content provider stores and manages only the smart contract address and does not directly manage the

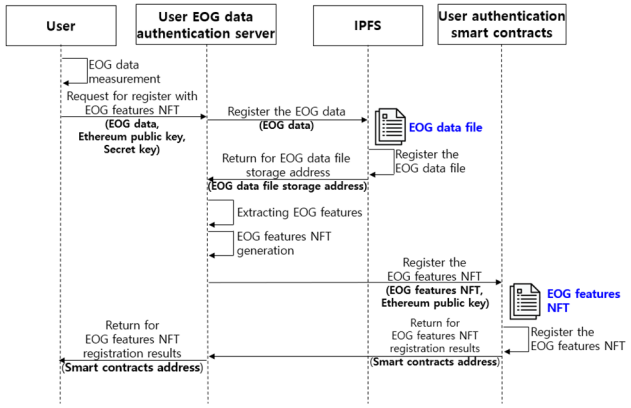

Fig. 6. Process of EOG features NFT registration.

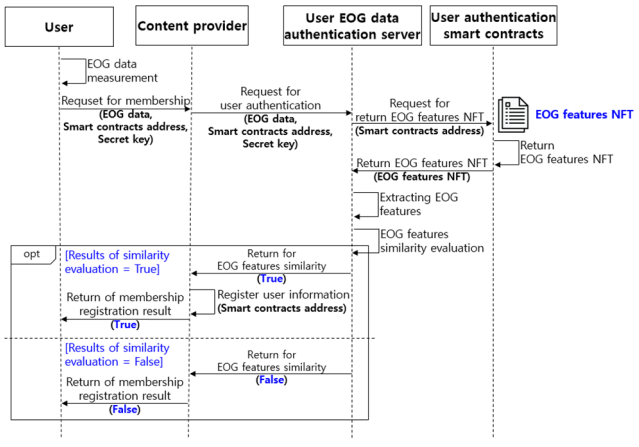

Fig. 7. Process of user authentication.

user's EOG data. User authentication is performed through the authentication service of the authentication server. In addition, while the authentication service is provided from the authentication server, the actual EOG data and EOG feature information are distributed and stored in the Ethereum blockchain and IPFS. As a result, an attacker cannot acquire the biometric information even if he or she seizes information from the content provider or the authentication server.

3.4. EOG Features NFT Forgery and Falsification Recovery Module

If user authentication fails as in Fig. 7, it will be due to two major causes. The first is the case where a user who does not possess an EOG features NFT requests membership registration. In this case, authentication is refused because the user's EOG and NFT do not match. The second is the case where the NFT is forged or falsified. For such a case, authentication may be refused even if the user presents a normal EOG, because the NFT does not match. If user authentication is refused when a normal user requests authentication, the NFT has been forged or falsified.

In order to receive normal authentication services, the user must inspect and recover the EOG features NFT that has been forged or falsified. For example, a user who has failed the authentication in Fig. 7 makes a request for recovery from forgery or falsification, as shown in Fig. 8. In order to recover the EOG features NFT that has been forged or falsified, the user's EOG data is needed. Since the user's EOG data was distributed and stored in IPFS during the EOG features NFT registration process in Fig. 6, it can be used for NFT recovery. The authentication server receives the returned EOG features NFT from the smart contract address for which recovery from forgery or falsification was requested. The EOG feature information of the user and the address where the EOG file is stored are registered in the EOG features NFT. Among them, the storage address of the EOG file is the hash value of the EOG file that is stored in IPFS, and it is the address value to access the EOG file.

The EOG feature information that was extracted from the EOG file must match the EOG features NFT. This is because the EOG data that was stored in the EOG file is the EOG data that was used to extract the features when the NFT was registered. Therefore, a comparison is made to verify whether the two features match exactly, and if they do not match, it is an EOG features NFT that has been forged or falsified. For the recovery of the forged or falsified EOG features NFT, the EOG features NFT is modified with the feature information that was extracted from the EOG file, and then the smart contract is registered. As a result, the authentication server inspects whether the EOG features NFT has been forged or falsified, and it can recover the NFT that has been forged or falsified.

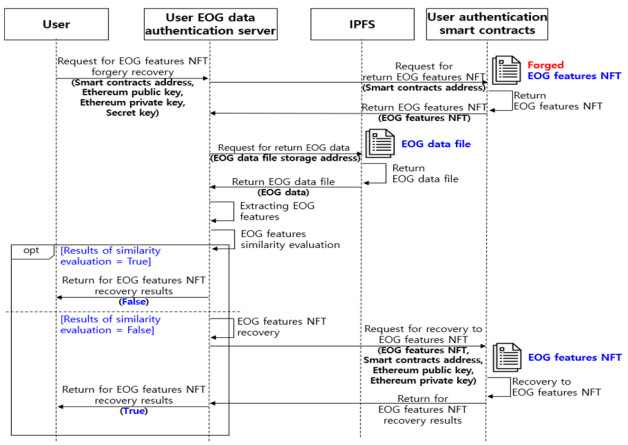

Fig. 8. Process of EOG features NFT forgery detection.

IV. EXPERIMENT AND ASSESSMENT

4.1. Experimental Environment and Overview

An experimental environment was configured and experiments were carried out to confirm whether the proposed eye-blinking EOG user authentication model is securely protected from integrity infringement attacks and persistent infringement attacks by attackers. The experimental environment was configured as shown in Fig. 9. Fig. 9(a) shows the PC environment in which the Ethereum blockchain node operates. Windows 10 was used as the operating system. For the CPU, an Intel(R) Core(TM) i7-10700 with 16 GB of RAM was used. The Ethereum blockchain network was implemented with Ganache-cli v6.12.2 and the smart contracts that will be distributed in the Ethereum network were developed with Truffle v5.5.28. IPFS was configured by using Go-ipfs v0.4.21.

The user EOG authentication servers are shown in Fig. 9(b). Two were configured (Authentication Server$_a$, Authentication Server$_b$) for the durability assessment of the authentication service. For the authentication server, a Raspberry Pi 4 model B was used, and the Raspberry Pi OS-

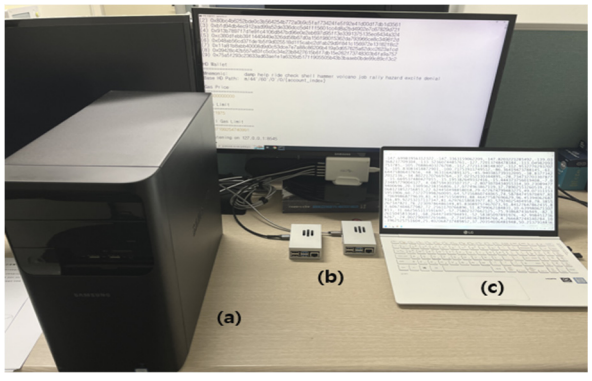

Fig. 9. Experimental environment, (a) Ethereum blockchain node, (b) User EOG data authentication servers, and (c) Log files on notebook.

based kernel version 5.10.63 was used for the operating system. The service was implemented from the Node.js v16.15.3 LTS with the Express.js v4.17.3 module. In addition, Web3.js v1.8.0 was used to make requests to the Ethereum network and IPFS.

Lastly, the user requesting the user authentication service and the content provider were implemented as shown in Fig. 9 (c). They were configured with a laptop PC, and Windows 11 was used for the operating system. For the CPU, an Intel(R) Core(TM) i5-8250U with 16GB of RAM was used. To implement the user and the content provider, EOG features NFT registration, user authentication, and forgery or falsification inspection and recovery requests were implemented from the Node.js v16.13.1 LTS with the Express.js v4.17.3 module.

For the experimental data, the EOG data of two persons (User1, User2) that had been measured with an OpenBCI Ganglion board in a previous eye-blinking EOG-based individual identification study was used [14]. The EOG data contains the data of a user who had blinked their eye naturally for 30 seconds, and it consists of about 6,000 real number EOG potential values that were measured at 200 Hz. This EOG data consists of a total of 100 eye-blinking EOG data, including 50 sample data for each user.

4.2. User Authentication Model Efficiency Assessment Experiment

The time required to register the EOG features NFT and the time required for user authentication were measured to assess the efficiency of the user authentication model. The time required for NFT registration is the time required for the user who had requested NFT registration in Fig. 6 to receive the returned smart contract address in which the EOG features NFT had been registered. An average of 1,472.733 ms, a maximum of 2,091 ms, and a minimum of 1,181 ms were confirmed, as shown in Fig. 10.

At this time, the number of registered EOG features NFTs is equal to the number of users who are registered in the authentication service. Even if the number of registered users increases, the time required to register the EOG features NFT tends to be constant, as shown in Fig. 10. In other words, we found out that the time required to register an EOG features NFT remained the same, even if the number of users using the authentication service increased.

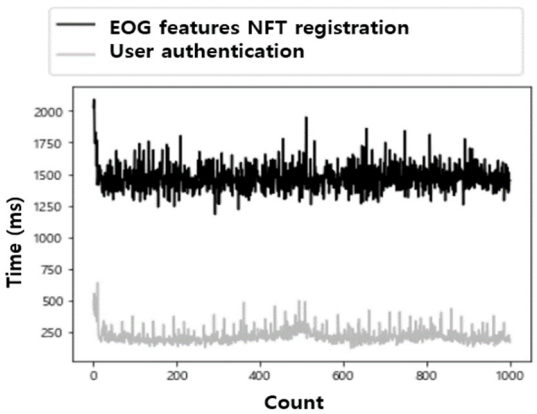

Fig. 10. Assessment of user authentication model efficiency.

Table 1. User authentication result for the EOG features NFT of $User_1$.

	$User_1$	$User_2$
True	500	0
False	0	500

The time required for user authentication is the time required for the content provider in Fig. 7 to present the EOG data and the smart contract address to the authentication server and receive the returned similarity evaluation result. An average of 217.829 ms, a maximum of 640 ms, and a minimum of 122 ms were confirmed, as shown in Fig. 10. Regarding the time required for user authentication, it was confirmed to be significantly shorter than the time required to register an NFT. A short amount of time seems to be required because IPFS and smart contracts do not generate new transactions when there is a user authentication request.

In addition, NFT registration is performed only once in the actual service environment, and the rest are all user authentication services. User authentication does not access the IPFS, it simply uses the NFT that has been registered with the smart contract. Therefore, we confirmed that efficiency is excellent for the proposed model.

In order to verify whether the model that was proposed additionally provides user authentication services accurately, authentication was requested 500 times for $User_1$ and 500 times for $User_2$ to the EOG features NFT of $User_1$. It can be seen from Table 1 that, authentication for $User_1$ was successful all 500 times while authentication for $User_2$ failed all 500 times. Therefore, we confirmed that the proposed user authentication service returned reliable results.

4.3. User Authentication Model Integrity Assessment Experiment

The time required for the recovery of a forged or falsified EOG features NFT was measured to assess the integrity of the user authentication model. The time required for recovery is the time required for the user, who has determined that the NFT has been forged or falsified, to make a recovery request and receive the returned result, as shown in Fig. 8. At this time, two situations are possible. The case where recovery is requested for an NFT that has been forged or falsified, and the case where recovery is requested for an NFT that has not been forged or falsified.

When recovery was requested for an NFT that was not forged or falsified, an average of 336.147 ms, a maximum

Table 2. Forgery or falsification inspection result according to whether or not forgery or falsification has occurred.

	Forged NFT	NFT
Forged NFT	1,000	0
NFT	0	1,000

of 610 ms, and a minimum of 230 ms were confirmed, as shown in Fig. 11. In contrast, when recovery was requested for an NFT that was forged or falsified, an average of 1,622.742 ms, a maximum of 2,030 ms, and a minimum of 1,336 ms were confirmed. NFT recovery is not carried out for an NFT that has not been forged or falsified. Therefore, it was assumed that the time required was short because the EOG features NFT that was registered in the smart contract was not recovered.

The proposed model recovers only the EOG features NFT that has been forged or falsified. Therefore, the accuracy of determining that the forged or falsified NFT has been forged or falsified is very important. According to Table 2, all 1,000 forged or falsified NFT were determined to be forged or falsified while all 1,000 NFT that have not been forged or falsified were determined not to have been forged or falsified.

As a result, we confirmed that the proposed model can

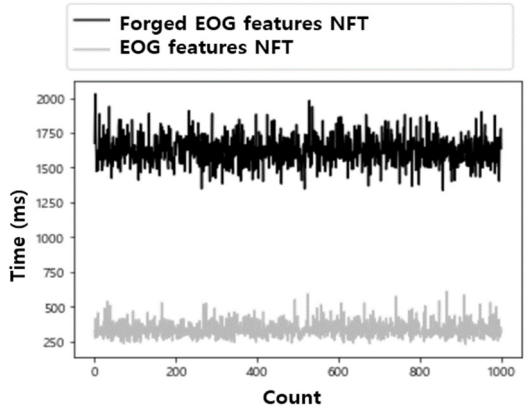

Fig. 11. Assessment of user authentication model integrity.

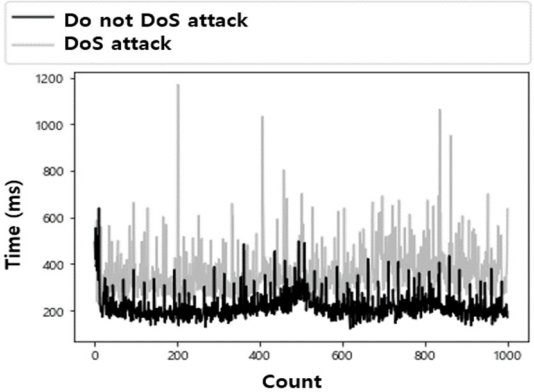

Fig. 12. Assessment of user authentication model durability.

Table 3. User authentication service durability assessment result.

	$User_1$	$User_2$
$User_1$	500	0
$User_2$	0	500

inspect forged or falsified NFT and recover from the forgery or falsification.

4.4. User Authentication Model Durability Assessment Experiment

The time required to receive the returned user authentication result from Authentication $Server_b$, after requesting user authentication from the paralyzed Authentication $Server_a$ and not being provided with the service, was measured to assess the durability of the user authentication model. For comparison, the time required for user authentication in Fig. 10 was compared together in Fig. 12. As a result, when the user first requested user authentication from Authentication $Server_a$ that had received a DoS attack, an average of 359.716 ms, a maximum of 1,170 ms, and a minimum of 217 ms were confirmed. This is a slightly increased result compared to the 217.829 ms that was required to receive the returned authentication result from Authentication $Server_b$ that had not received a DoS attack.

In addition, in order to check whether other authentication servers (Authentication $Server_a$ and Authentication $Server_b$) provide the same user authentication service, EOG features NFT was registered in Authentication $Server_a$ and user authentication was requested from Authentication $Server_b$.

As a result, when the NFT of $User_1$ was registered 500 times in Authentication $Server_a$, they were authenticated as $User_1$ all 500 times from Authentication $Server_b$. In addition, when the NFT of $User_2$ was registered 500 times in Authentication $Server_a$, they were authenticated as $User_2$ all 500 times from Authentication $Server_b$.

Therefore, we confirmed that even if one authentication server cannot provide service due to a DoS attack, the same authentication service can be provided by another authentication server in the proposed model.

As a result, it was confirmed through the experiments that the proposed authentication model can provide authentication services to users and content providers with 1,472.733 ms to register an EOG features NFT and 217.829 ms to authenticate a user. In addition, it was shown that the proposed user authentication model can securely protect the EOG data from integrity infringement attacks and persistent infringement attacks by attackers by confirming that it takes 1,622.742 ms to inspect and recover a forged or falsified EOG features NFT and 359.716 ms to receive the same authentication service from another authentication server.

V. CONCLUSION

The existing user authentication models provide user authentication services by managing the biometric data of the users on a single central server. However, such a storage method cannot be securely protected from mistakes made by managers and attacks by attackers. In addition, there is a limitation that the users cannot be provided with an authentication service if attackers attempt to forge or falsify the EOG data or carry out DoS attacks on the central server.

Therefore, a user authentication model that registers EOG data as an NFT to authenticate users by utilizing Ethereum blockchain technology was proposed in this study. The proposed model operates smart contracts and IPFS on one of the Ethereum blockchain nodes, and manages the EOG data and the EOG features NFT. In addition, an authentication service was provided to the users by receiving the returned EOG features NFT that had been distributed by installing an authentication server for the user EOG.

Since biometric data is used as a password in existing user authentication services, there is a risk that biometric data may be seized easily by an attacker. However, the smart contract address that can access biometric data is used as a password in the proposed authentication model, and the biometric data is used as a kind of one-time authentication key to verify the smart contract address. Therefore, it is not necessary to manage biometric data directly, and there is no risk of being seized by an attacker.

In order to check whether the proposed authentication model resolves the limitations of the existing authentication models, an integrity assessment experiment, in which the EOG features NFT is forged or falsified and then recovered, and a durability assessment experiment, which verifies whether an equivalent authentication service is provided when one of the authentication servers have become paralyzed due to DoS attacks, were conducted. For the integrity assessment experiment, it took an average of 1,622.742 ms to recover a forged or falsified NFT. For the durability assessment experiment for the authentication service, it was possible to receive an authentication service in 359.716 ms. There is ample room to improve the time required like this by improving the communication method between the biometric data measurement device and the authentication server [25-26]. Therefore, if the proposed authentication model is used, it will be possible to provide users with a user authentication service based on biometric data that is more stable to the users by improving the limitations of the user authentication service that is provided from a single central server.

If user authentication model that was additionally proposed is utilized, it is expected that it will be possible to protect the user EOG data safely when providing personalized services in healthcare fields like sleep care. In addition, it will be possible to provide safe authentication services by expanding to biometric data like fingerprints, iris, and voice that can be used for user authentication.

However, SPOF (Single Point of Failure) that can occur in an IPFS system was not considered in the proposed model. Since all participating nodes own DHT (Distributed Hash tables) and communicate based on the Kademlia protocol in an IPFS, the risk of an SPOF attack is low. However, there is sufficient possibility that a SPOF can occur in an actual environment due to the limitation that addresses of all participating nodes are not owned.

In addition, there is the need to cope with the cost of maintaining an IPFS system. As one of the ways to solve this problem, content providers that provide authentication services can manage the DHT for a secure IPFS. In addition, it will be possible for content providers to pay the cost for maintaining the IPFS system rather than the cost of managing the biometric data directly.

ACKNOWLEDGMENT

This work was supported by the Technology development Program funded by the Ministry of SMEs and Startups (MSS, Korea) (Grants No. S2957039, S3229617).

REFERENCES

[1] D. K. Tak, "Image based fingerprint sensor applied convergence authentication method for smartphone," *The Society of Convergence Knowledge Transactions*, vol. 9, no. 4, pp. 1-11, 2021.

[2] J. S. Kim, S. H. Kim, and S. B. Pan, "Electrocardiogram signal based personal identification performance analysis using pre-trained network model," *Journal of KIIT*, vol. 18, no. 1, pp. 107-114, 2020.

[3] W. C. Lim and K. C. Kwak, "A multilinear LDA method of tensor representation for ECG signal based individual identification," *Smart Media Journal*, vol. 7, no. 4, pp. 90-98, 2018.

[4] J. S. Han and E. G. Im, "Implementation of individual gait recognition using RNN," *KIISE Transactions on Computing Practices*, vol. 24, no. 7, pp. 358-362, 2018.

[5] S. L. Nita, M. I. Mihailescu, and V. C. Pau, "Security and cryptographic challenges for authentication based on biometrics data," *Cryptography*, vol. 2, no. 4, 2018.

[6] J. Kang and Y. Woo, "Data management plan for clinical trials in a blockchain environment," *Journal of Korean Institute of Information Technology*, vol. 19, no. 10, pp. 137-143, 2021.

[7] E. Haasnoot, L. J. Spreeuwers, and R. N. Veldhuis, "Presentation attack detection and biometric recognition in a challenge-response formalism," *EURASIP Journal on Information Security*, vol. 2022, no. 1, pp. 1-15, 2022.

[8] Korean Law Information Center, Nov. 2022. https://www.law.go.kr/admRulLsInfoP.do?admRulSeq=2100000019404.

[9] Y. Lee, Y. Ku, and T. Kwon, "A study of user perception on features used in behavior-based authentication," *Journal of the Korea Institute of Information Security & Cryptology*, vol. 29, no. 1, pp. 127-137, 2019.

[10] D. H. Lim, "Personal authentication system using multimodal biometric algorithm," *Journal of Korean Institute of Information Technology*, vol. 15, no. 12, pp. 147-156, 2017.

[11] S. L. Nita, M. I. Mihailescu, and V. C. Pau, "Security and cryptographic challenges for authentication based on biometrics data," *Cryptography*, vol. 2, no. 4, pp. 39, 2018.

[12] A. Das, C. Galdi, H. Han, R. Ramachandra, J. L. Dugelay, and A. Dantcheva, "Recent advances in biometric technology for mobile devices," in *Proceedings of the 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Redondo Beach, CA, 2018, pp. 1-11.

[13] G. H. Choi, H. M. Moon, and S. B. Pan, "Biometrics system technology trends based on biosignal," *Journal of Digital Convergence*, vol. 15, no. 1, pp. 381-391, 2017.

[14] K. H. Hong, B. M. Lee, and Y. J. Park, "Realtime individual identification based on EOG algorithm for customized sleep care service," *Journal of Convergence for Information Technology*, vol. 9, no. 12, pp. 8-16, 2019.

[15] A. Borowicz, "Using a multichannel Wiener filter to remove eye-blink artifacts from EEG data," *Biomedical Signal Processing and Control*, vol. 45, pp. 246-255, 2018.

[16] H. S. Wi and B. M. Lee, "Customized realtime control of sleep induction sound based on brain wave data," *Journal of Korea Multimedia Society*, vol. 23, no. 2, pp. 204-215, 2020.

[17] S. Luo, A. Nguyen, C. Song, F. Lin, W. Xu, and Z. Yan, "OcuLock: Exploring human visual system for authentication in virtual reality head-mounted display," in *Proceedings of the 2020 Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 2020.

[18] M. S. Hossain, K. Huda, S. S. Rahman, and M. Ahmad, "Implementation of an EOG based security system by analyzing eye movement patterns," in *Proceedings of the 2015 International Conference on Advances in Electrical Engineering (ICAEE)*, Bangladesh, 2015, pp. 149-152.

[19] Y. K. Kim, C. J. Chae, and H. J. Cho, "User authentication method using EEG signal in FIDO system," *Journal of the Korea Convergence Society*, vol. 9, no. 1, pp. 465-471, 2018.

[20] S. R. Kim, "Design of the personalized user authentication systems," *Journal of Convergence for Information Technology*, vol. 8, no. 6, pp. 143-148, 2018.

[21] D. Kim and K. Seo, "PGP certification system in blockchain environments," *Journal of Korea Multimedia Society*, vol. 23, no. 5, pp. 658-666, 2020.

[22] S. H. Jung, J. H. Kim, and C. B. Sim, "Implementation of university point distributed system based on public blockchain," *Journal of Korea Multimedia Society*, vol. 24, no. 2, pp. 255-266, 2021.

[23] H. Kim, "A study on the blockchain based knowledge sharing platform," *The Journal of Society for e-Business Studies*, vol. 27, no. 1, pp. 95-109, 2022.

[24] W. H. Nam and H. Y. Kil, "ATL model checking for analysis of ethereum smart contracts," *The Transactions of the Korean Institute of Electrical Engineers*, vol. 70, no. 12, pp. 2006-2014, 2021.

[25] R. Maharaj, V. Balyan, and M. T. Khan, "Optimising data visualisation in the process control and IIoT environments," *International Journal on Smart Sensing and Intelligent Systems*, vol. 15, no. 1, pp. 1-14, 2022.

[26] R. Maharaj, V. Balyan, and M.T. Kahn, "Design of IIoT device to parse data directly to scada systems using LoRa physical layer," *International Journal on Smart Sensing and Intelligent Systems*, vol. 15, no. 1, pp. 1-13, 2022.

Journal of Multimedia Information System VOL. 9, NO. 4, December 2022 (pp. 275-286): ISSN 2383-7632 (Online)
https://doi.org/10.33851/JMIS.2022.9.4.275

AUTHORS

Ki Hyeon Hong is currently fourth year M.S. student in Department of IT Convergence degree at Gachon University in Korea. He received a B.S. degree in 2021 from Gachon University, Korea. He is working on projects for Untact Smart Access Security System based on AIoT, and is conducting research on password security, IoT, and blockchain. His research interests include AIoT Smart Service and blockchain.

Byung Mun Lee received a B.S. degree in 1988 from Dongguk University, Seoul, Korea and a M.S. degree from Sogang University and a Ph.D. degree from University of Incheon Korea, in 1990 and 2007. He had worked for LG Electronics for 7 years. He is currently a professor in the department of Computer Engineering, Gachon University, South Korea. He had been at California State University Sacramento, USA from 2013 to 2014 as a visiting scholar. His research interests are IoT for healthcare, AIoT Smart Service, network protocols, blockchain, NFT, DID, smart services, etc.