A IMPLEMENTATION OF USING PASSWORDLESS ON OPENID CONNECT PROTOCOL

Piti SUWANNAKOM[1] and Utharn BURANASAKSEE[1]

[1] Faculty of Science and Technology, Rajamangala University of Technology Suvarnabhumi, Thailand; 164480322004-st@rmutsb.ac.th (P. S.); utharn.b@rmutsb.ac.th (U. B.)

ARTICLE HISTORY
Received: 19 April 2024 **Revised:** 3 May 2024 **Published:** 17 May 2024

ABSTRACT

The increasing proliferation of applications and their concurrent or multi-provider development complicates users' digital experience, necessitating the memorization of numerous digital IDs to access various web applications. This leads to frequent issues with forgotten usernames or passwords. Despite the development of Single Sign-On (SSO) protocols that allow access to multiple applications with a single ID, the need to remember passwords remains a significant challenge. This research aims to design and develop a passwordless authentication system based on the OpenID Connect protocol to enhance security and user convenience in accessing information systems. Additionally, the study explores the operational impact of this system to evaluate its effectiveness and implications for information system access. This research presents a solution to password-related challenges and enhances authentication security by enabling access to multiple applications with a single ID, using private keys and authenticating via QR code scans through a mobile application. This approach elevates authentication reliability to Assurance Level 3 (AAL3) and enhances user convenience, representing an advancement over the existing OpenID Connect protocol standards for information system access.

Keywords: Authentication, Password-less, Single Sign-on Protocol

CITATION INFORMATION: Suwannakom, P., & Buranasaksee, U. (2024). A Implementation of Using Passwordless on Openid Connect Protocol. *Procedia of Multidisciplinary Research, 2*(5), 19.

INTRODUCTION

In the contemporary digital landscape, a vast array of applications is continuously emerging, often developed asynchronously or by multiple service providers. This diversity necessitates users to memorize a plethora of digital IDs to access different web applications. A digital ID, fundamentally comprising a username and password, serves as a credential set enabling users to authenticate themselves to service providers. The multiplicity of digital ID accounts leads to frequent issues with forgotten usernames or passwords. However, authentication through these means remains fraught with risks, such as password breaches. Even though two-factor authentication (2FA) methods, like one-time passwords (OTPs) sent via SMS or email, have been introduced to bolster security, they often introduce operational complexities and maintain the inherent requirement of remembering usernames and passwords.

To address these challenges, protocols like Single Sign-On (SSO) by Fett & Schmitz (2017) have been developed, allowing users to access multiple applications with a single digital ID, thus mitigating the need to remember numerous usernames and passwords. Protocols such as OAuth 2.0 (Hardt, 2012), OpenID Connect 1.0 (Sakimura et al., 2014), and SAML 2.0 (Wilson & Kingnokar, 2022) facilitate authentication rights to various services using a single digital ID from an external account. While these protocols alleviate the burden of remembering multiple credentials, they still require users to remember their username and password, along with any additional authentication factors if stringent authentication measures are enforced.

Moreover, with the prevalence of SSO services offered by various providers, users tend to have multiple digital IDs, each potentially accessing several web applications. This proliferation exacerbates the issue of password forgetfulness. Thus, there's a pressing need for innovative authentication solutions that can transcend the limitations and security risks associated with traditional username and password systems, propelling towards more secure, user-friendly, and efficient authentication methods in the digital domain.

Considering these challenges, this research aims to develop an authentication system that uses private keys for identity verification without requiring passwords. The designed protocol allows for private key-based login, with the condition that the private key resides on a device like a computer or mobile phone, but the login attempt is made from a different device without the need for browser extensions or copying the private key. This research introduces a method enabling easy mobile phone access, significantly leveraging the phone's camera feature. The method involves scanning a QR code and removing the code specified by the browser session, then transmitting this code to the mobile application to sign transactions, thereby indicating that the browser session will be authenticated on behalf of the user.

LITERATURE REVIEWS

This leads to the concept of enhancing authentication methods to a higher level, such as using private keys for system access. The advantage of utilizing private keys lies in elevating security to its utmost level. Many systems already employ private key authentication, for instance, system administrators accessing servers via the SSH Protocol (Ylonen & Lonvick, 2006), or software developers uploading source code to repositories like Bitbucket (Atlassian Inc., 2023), GitHub (GitHub Inc., 2023), and GitLab (GitLab Inc., 2023), where private keys are commonly used for authentication. While IT professionals and software developers are accustomed to using private keys for authentication, the general user base remains largely unaware of their benefits in various systems.

Currently, new authentication protocols like WebAuthn (Raman, 2023) and FIDO2 (Bindel et al., 2023) developed by the Fast Identity Online Alliance with contributions from leading tech companies such as Apple, Google, and Microsoft, aim to standardize the use of private keys for authentication without the need for passwords. These protocols facilitate transactions without uploading private keys to the system, leveraging browser extensions to select the private key for system login. However, this method poses challenges, such as being limited to computers or laptops and encountering issues with mobile phones where browser extensions cannot be installed, or when accessing from a non-personal device, like in public libraries or internet cafes, where copying the private key for authentication can be problematic.

Previous research by Konheim (2016) discussed integrating location with one-time passwords using public-key cryptography for authentication, specifically in the context of ATM machines and banking applications. However, this approach was limited to bank-specific applications, and obtaining a private key from a bank server for encryption presented security risks and practical difficulties, as private keys are typically generated and stored on the user's mobile application and not transferred or exported.

RESEARCH METHODOLOGY

The research study focused on developing a passwordless authentication system using the OpenID Connect protocol proceeded through the following stages:

Stage 1: Initial Literature Review and Data Collection

The researcher began by conducting a thorough review of existing literature and gathering relevant data. This foundational step aimed to gain a comprehensive understanding of the current state of passwordless authentication mechanisms within the OpenID Connect protocol framework, thereby informing the design of the proposed system.

Stage 2: System Analysis and Design

In this phase, the researcher analyzed the feasibility and reliability of implementing a passwordless authentication system within the OpenID Connect protocol. This involved exploring potential solutions to address the challenges identified during the literature review, ensuring the proposed system's design was both innovative and viable.

Stage 3: System Development Using PHP

The development phase saw the researcher implementing the designed passwordless authentication system using PHP as the programming language. This step was crucial in translating the theoretical design into a tangible system that could be tested and evaluated.

Stage 4: Testing and Evaluation in a Sandbox Environment

Upon completing the system development, the researcher deployed the system within a controlled sandbox environment for testing. This allowed for a direct comparison of the developed system's features against those of existing methods outlined in prior research. The evaluation focused on assessing the quality of the system by examining its advantages and limitations relative to previously proposed solutions.

The Implementation Design

In the research process, the investigation began with an in-depth study of relevant literature and data collection related to authentication challenges. The aim was to research, explore, and construct a passwordless authentication system on the OpenID Connect protocol. Notably, this system employs a pair of keys for authentication without storing the private key on the system. To avoid the complexities of managing and using key pairs, users are not required to generate or carry a copy of the key pair. Instead, a mobile application is utilized to generate and store the key pair, and to sign transactions for authentication purposes.

Public-key cryptography, a widely adopted method for securing data during network transmission, employs a pair of keys for enhanced security compared to conventional encryption methods. However, this does not necessarily mean it is the best encryption method for all scenarios, as this depends on the specific requirements of each organization or individual. Public-key encryption involves two types of keys always used in tandem for encryption and decryption: the public key, which can be shared openly, and the private key, which is kept secret by the creator.

Public-key cryptography can be applied to both encryption and authentication. The encryption process involves: 1) Each user generating their own key pair for encryption and decryption, 2) The public key being distributed to others while the private key remains with the individual, 3) Encrypting data with the recipient's public key before transmission, and 4) The recipient using their private key, which matches the public key, to decrypt the data correctly.

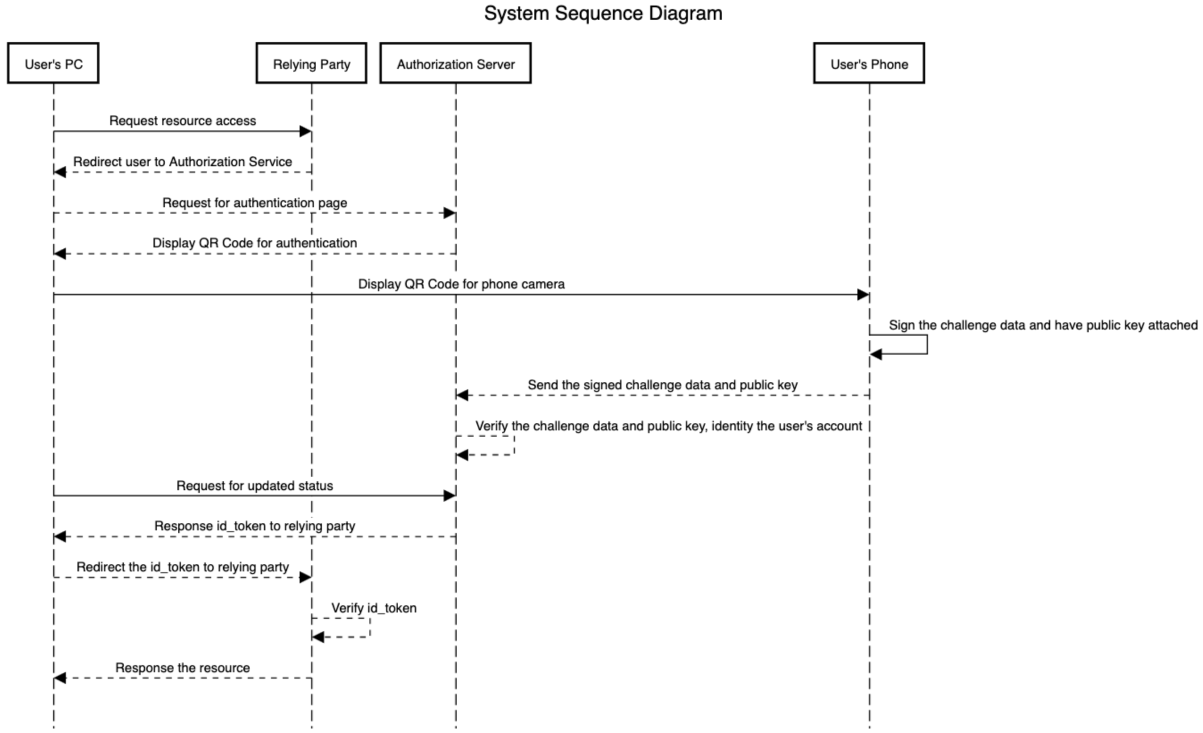

System Sequence Diagram
User's PC
Relying Party
Authorization Server
User's Phone
Request resource access
Redirect user to Authorization Service
Request for authentication page
Display QR Code for authentication
Display QR Code for phone camera
Sign the challenge data and have public key attached
Send the signed challenge data and public key
Verify the challenge data and public key, identity the user's account
Request for updated status
Response id_token to relying party
Redirect the id_token to relying party
Verify id_token
Response the resource

Figure 1 System Sequence diagram

Digital Signature authentication uses the principles of public-key cryptography for authentication. The digital signature system involves: 1) Data being processed through a mathematical function called a "hash function" to produce a message digest, 2) The data being encrypted with a private key, indicating the sender's digital signature and consent for verification by the recipient using the sender's public key, and 3) The recipient verifying the data's origin by recalculating the message digest and decrypting the digital signature with the sender's public key. If the decryption is successful and the message digests match, the data is confirmed to originate from the sender.

Since OAuth 2.0 serves as an Authorization Framework for authentication and authorization, allowing third-party applications to access various services securely, as outlined in the standard RFC6749 by the IETF. OpenID Connect or OIDC utilizes OAuth 2.0 meaning authentication through OAuth 2.0 as the OpenID Connect protocol is built upon the OAuth 2.0 protocol, providing a foundation for clients to verify the identity of end-users based on authentications performed by an Authorization Server, which includes both Relying Parties (RP) and Identity Providers (idP) for simplicity. The identity of the end-users is constructed in the form of JSON Web Token (JWT) with the signature signed by the Identity Provider allowing any relying parties to verify and trust information passed by the user.

The research analyzed and designed the system to explore the feasibility of passwordless authentication on the OpenID Connect protocol, focusing on the authentication process using key pairs. As illustrated in Figure 1, the process begins with the user requesting access to resources. The authenticator then directs the user to an authentication request, displaying a QR code for authentication. The user scans this code using a mobile application to sign the authentication data, which then returns a signature and public key to the authenticator for verification. Upon successful verification, an access token is issued to the user for system access, allowing the relying party to verify the access token with the authenticator, completing the authentication process. After that, the Authorization Server returns JWT token called id_token which contains user's information signed by the Authorization Server.

In Step 3 of the research, the development of a passwordless authentication system within the OpenID Connect protocol was undertaken, utilizing PHP as the programming language. This innovative system eschews traditional password-based authentication in favor of a transaction signing mechanism using a pair of keys facilitated through a mobile application. Upon a user attempting to log in via a computer, the system generates a session-specific random message and a callback URL, which are then used to create a QR code. Users scan this QR code with their mobile devices, which have been previously equipped with a designated application for QR code scanning. The application signs the authentication data (Signature) and returns it to the authentication system along with the public key (Public Key), serving as the user's identifier.

The authentication process unfolds by decrypting the signature with the public key. If the decryption is successful, indicating the authenticity of the signature, the system proceeds to generate an access token (Access Token). This token, representing confirmed user authentication, is stored within the system. Subsequently, user data is relayed back to the system via the OpenID Connect protocol, enabling user access to the system. This method not only streamlines the authentication process but also enhances security by eliminating the need for password-based login mechanisms.

In the last step, the system was deployed in a sandbox environment for integration with real applications.

RESEARCH RESULTS & DISCUSSION

In the design of the database, the researcher employed the standards of the OpenID Connect 1.0 Core specification to structure the recording of user data, as illustrated in Figure 2. The database consists of several key tables:

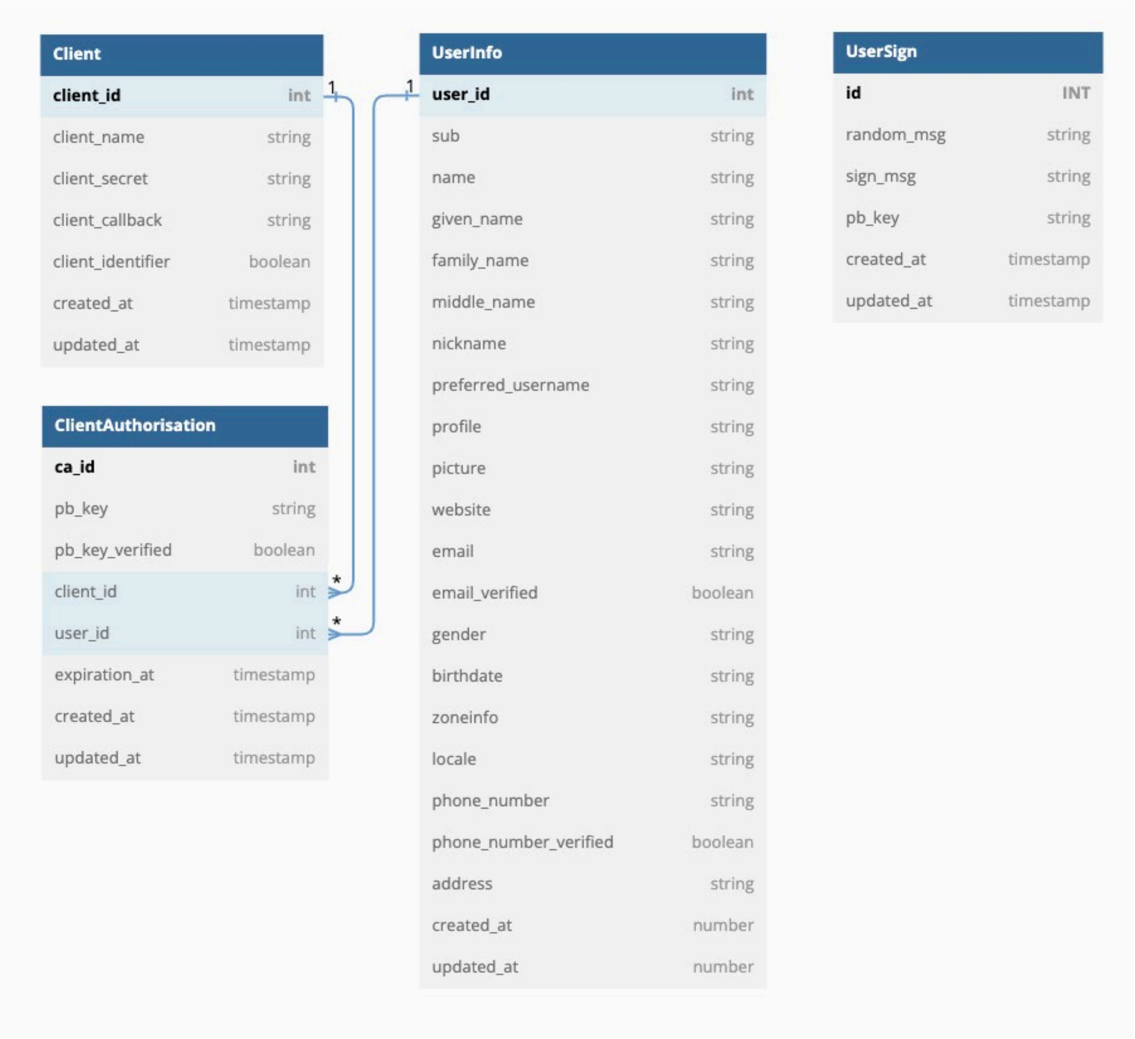
1) The `Client` table stores information about the Relying Parties that are authorized to interact with the Authorization Server. This table serves as a registry for client applications, detailing each one's credentials and authorized scopes.

2) The `UserInfo` table is dedicated to holding user-related data essential for generating the `id_token`. This table includes attributes that represent user profile information, such as names, email addresses, and any other data pertinent to user identity within the system.

3) The `ClientAuthorization` table maintains records of public keys registered by users for the purpose of client authentication. It also oversees the lifecycle of these keys, including their revocation or expiration, ensuring that only valid keys are used for secure transactions.

4) The `UserSign` table contains randomly generated messages that are dispatched to the Relying Party and presented to the user. This mechanism is part of a challenge-response authentication flow, where the user is required to sign the message using a pre-installed mobile application, thus validating their identity.

This database structure is integral to supporting the passwordless authentication mechanism within the OpenID Connect framework, facilitating secure and efficient user authentication and authorization processes.



Client	
client_id	int
client_name	string
client_secret	string
client_callback	string
client_identifier	boolean
created_at	timestamp
updated_at	timestamp

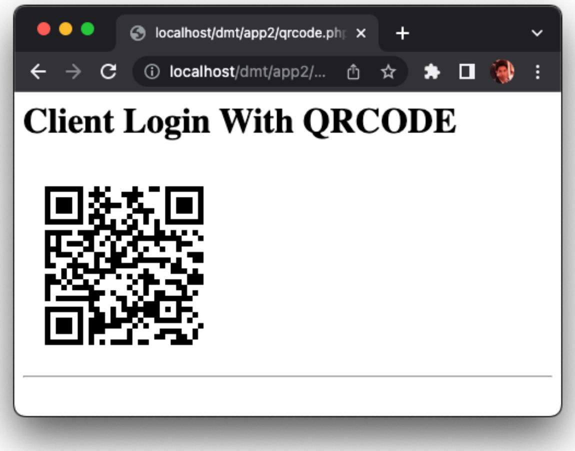
ClientAuthorisation	
ca_id	int
pb_key	string
pb_key_verified	boolean
client_id	int
user_id	int
expiration_at	timestamp
created_at	timestamp
updated_at	timestamp

UserInfo	
user_id	int
sub	string
name	string
given_name	string
family_name	string
middle_name	string
nickname	string
preferred_username	string
profile	string
picture	string
website	string
email	string
email_verified	boolean
gender	string
birthdate	string
zoneinfo	string
locale	string
phone_number	string
phone_number_verified	boolean
address	string
created_at	number
updated_at	number

UserSign	
id	INT
random_msg	string
sign_msg	string
pb_key	string
created_at	timestamp
updated_at	timestamp

Figure 2 Database diagram

The development of the system began with the modification of the login interface to incorporate a QR code login option, as depicted in Figure 3, facilitating the authentication process via the OpenID Connect protocol.



Client Login With QRCODE

Figure 3 Login screenshot

This is because the challenge information needs to be signed by the private key residing in the phone. To make the system secure, the challenge needs to be long enough random string and it is not feasible for the user to type those challenge on the phone. Figure 3 prompts the users to scan the QR code as an alternative method to authenticate the user with a mobile application for identity verification and system access.

The researcher conducted a qualitative assessment of the system's development, as outlined in Table 1. This evaluation compared the developed system's features against previous methods to highlight its advantages and limitations. The proposed method in this research, which employs a pair of keys for authentication, significantly enhances security, obviating the need for users to remember usernames and passwords. Moreover, it simplifies the process by eliminating the necessity to transfer key pairs to other computers for authentication, offering convenience, user-friendliness, and efficiency in transactions.

Table 1 Comparison of Proposed Authentication Methods

Feature	**Username & Password**	**Username, Password & OTP Authentication**	**Passwordless Authentication**
Password Forgetting Issue	Must remember username and password	Must remember username and password	No need for passwords
User Convenience	Must enter username and password	Must enter username, password, and OTP	Use mobile phone to scan QR code
AAL Level	Level 1	Level 2	Level 3
OTP Costs	No cost	Incurs cost	No cost
Transaction Speed	Immediate access if username and password are correct	Must wait for OTP after entering correct username and password	Immediate access after scanning QR code with mobile phone

Our results revealed that the system could develop an authentication system that uses private keys for identity verification without requiring passwords. As well new authentication protocols like WebAuthn (Raman, 2023) and FIDO2 (Bindel et al., 2023) developed by the Fast Identity Online Alliance with contributions from leading tech companies such as Apple, Google, and Microsoft, aim to standardize the use of private keys for authentication without the need for passwords. These protocols facilitate transactions without uploading private keys to the system, leveraging browser extensions to select the private key for system login. However, this method poses challenges, such as being limited to computers or laptops and encountering issues with mobile phones where browser extensions cannot be installed, or when accessing from a non-personal device, like in public libraries or internet cafes, where copying the private key for authentication can be problematic. In addition, Konheim (2016) discussed integrating location with one-time passwords using public-key cryptography for authentication, specifically in the context of ATM machines and banking applications. However, this approach was limited to bank-specific applications, and obtaining a private key from a bank server for encryption presented security risks and practical difficulties, as private keys are typically generated and stored on the user's mobile application and not transferred or exported.

CONCLUSION

Addressing the common issue of users forgetting their usernames or passwords, this research introduces a solution through password-less authentication, mitigating the security risks associated with password theft, phishing attacks, and brute-force attempts. While the development of Single Sign-On (SSO) protocols has facilitated access to multiple applications with a single digital ID, users are still burdened with remembering passwords. This study proposes the use of private key encryption to sign authentication data, elevating the reliability of authentication to Assurance Level 3 (AAL3).

Given the challenges users face in managing private keys, the research innovates by employing a mobile application to generate private keys and sign authentication data using QR codes. This approach eliminates the need for users to transfer private keys to different computers, enhancing usability and security. This research builds upon the OpenID Connect protocol, a standard that allows integration with existing information systems.

Future research will focus on examining the security robustness of the developed system to ensure its safe application in real-world environments. This includes identifying and rectifying potential vulnerabilities to uphold the system's integrity, especially concerning secure and AAL3- compliant authentication methods. The study aims to understand related challenges thoroughly and develop appropriate solutions to address them effectively.

REFERENCES

Atlassian Inc. (2023). *Set up an SSH key*. Retrieved from https://support.atlassian.com/bitbucket-cloud/docs/set-up-an-ssh-key.

Bindel, N., Cremers, C., & Zhao, M. (2023, May). FIDO2, CTAP 2.1, and WebAuthn 2: Provable security and post-quantum instantiation. In *2023 IEEE Symposium on Security and Privacy (SP)* (pp. 1471-1490). IEEE.

Fett, D., Küsters, R., & Schmitz, G. (2017, August). The web SSO standard OpenID Connect: In-depth formal security analysis and security guidelines. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)* (pp. 189-202). IEEE.

GitHub Inc. (2023). *Connecting to GitHub with SSH*. Retrieved from https://docs.github.com/en/authentication/connecting-to-github-with-ssh.

GitLab Inc. (2023) . *Use SSH keys to communicate with GitLab*. Retrieved from https://docs.gitlab.com/ee/user/ssh.html.

Jones, M., & Hardt, D. (2012). *The oauth 2.0 authorization framework: Bearer token usage*. (No. rfc6750).

Konheim, A. G. (2016). Automated teller machines: their history and authentication protocols. *Journal of Cryptographic Engineering, 6*(1), 1-29.

Raman, S. (2023). *WebAuthn*. Retrieved from https://webauthn.guide.

Sakimura, N., Bradley, J., Jones, M., De Medeiros, B., & Mortimore, C. (2014). OpenID Connect Core 1.0 incorporating errata set 1. *The OpenID Foundation, specification, 335*.

Wilson, Y., & Hingnikar, A. (2022). SAML 2. In *Solving Identity Management in Modern Applications: Demystifying OAuth 2, OpenID Connect, and SAML 2* (pp. 127-141). Berkeley, CA: Apress.

Ylonen, T., & Lonvick, C. (2006). *The secure shell (SSH) protocol architecture* (No. rfc4251).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright: © 2024 by the authors. This is a fully open-access article distributed under the terms of the Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0).