Development of Hash-Based Multi-Factor Password Generating System

*[1]Morufu Olalere, [1]Adenrele A. Afolorunso, [2]Zainab M. Olalere, [3]Raji, A. Egigogo, [3]Dauda Buhari, and [3]Habeeb Y. Habeeb

[1]Department of Computer Science, National Open University of Nigeria, Abuja, Nigeria
[2]Department of Computer Science, Federal University of Technology, Minna, Nigeria
[3]Department of Cyber Security Science, Federal University of Technology, Minna, Nigeria

{molalere | aafolorunsho}@noun.edu.ng | {olalerezainab5 | abdullahirajiegigogo buharidauda | habeebdh1}@gmail.com

Received: 18-MAY-2023; Reviewed: 10-JUL-2023; Accepted: 21-SEP-2023
http://doi.org/10.46792/fuoyejet.v8i4.1047

ORIGINAL RESEARCH

Abstract- The use of passwords or passphrases is essential for every internet user. However, users often face a dilemma between choosing simple passwords that are easy to crack and complex passwords that are difficult to remember, leading to frequent and cumbersome password recovery processes. This paper focuses on addressing this issue by developing a multifactor, unique password-generating system using SHA-256. The system incorporates factors such as a Unique Identifier, a biometric key value, and an Android mobile phone with a Biometric scanner. To accomplish this, an algorithm is devised using JavaScript that concatenates and generates a hash value by applying the SHA-256 algorithm to the Unique Identifier and Biometric Key values. The software implementation is achieved using the JavaScript programming language, with support from predefined plugins. Once the application is created, passwords can be generated by inputting a user's fingerprint ID and an Identifier (e.g., "Facebook.com"), resulting in the generation of a 32-character unique password. This process can be applied to any identifier and can reproduce the same password when the same factors are supplied. The experimentation results demonstrate that the system is capable of generating unique passwords for different platforms and can reproduce the same password for each platform if needed. While the focus of this paper is on the development of a system for Android mobile phone operating systems, It is suggested that its functionality be expanded by developing a browser extension and versions for other operating systems to improve its usability and accessibility across multiple platforms.

Keywords- Password security, Multi Factor authentication, SHA-256 algorithm, Biometric authentication, Password management

1 Introduction

Authorization, access control, use of passwords (passphrase and code inclusive), and biometrics are four concepts that are generally used together in today's systems. They are used to ensuring high security by giving access to only those whom the intended system is meant for. Access management grants system administrators the privilege to protect sensitive information, comply with regulatory requirements, provide personalized user experiences, ensure auditing and accountability, and build user trust and confidence. A simple way to achieve these benefits is to implement a password-based authentication system. A password-based system grants access to users using a username of any form and a password. As long as the password provided is strong, the system can boast of being secure.

Nobody likes passwords. They're inconvenient. They're a prime target for attacks. Yet for years, they've been the most important layer of security for everything in our digital lives—from email to bank accounts, shopping carts to video games (Bradley, 2021). The study is expected to create complex and unique passwords, remember them, and change them frequently, but nobody likes doing that either (Martin, 2021).

In a recent Microsoft Twitter poll, one in five people reported they would rather accidentally "reply all"—which can be monumentally embarrassing—than reset a password (Jakkal, 2022). The approach taken by Microsoft to enable password-less authentication is currently restricted to its products. It relies on utilizing biometric data and email-based login verification, but these features are only accessible to Microsoft users who have signed in to their Microsoft accounts. Furthermore, this solution is limited to securing Microsoft products and cannot be extended to other platforms, applications, or websites that users may visit. Therefore, there is a need for a more universal method that can enhance password strength for users across various platforms and sites without being dependent on specific accounts or platforms.

The paper focuses mainly on mobile Android OS and requires a fingerprint scanner to work; a future paper can look at how the same functionalities can be employed on other devices and as add-ons on browsers and sites.

2 Materials and Methodology

In this paper, an algorithm was developed that relied on multiple factors and a SHA-256 hash function to generate unique passwords for users. The algorithm used a combination of a biometric factor, specifically a fingerprint, and a platform label (such as a site name, app name, or URL) as inputs. These two factors were concatenated and then hashed using an RSA hash function, resulting in a 20-character password (the length varied based on the biometric key length) represented in a 64-base format.

The objective was to create an application with customizable inputs where one input was fixed: the

*Corresponding Author

Section B- ELECTRICAL/COMPUTER ENGINEERING & COMPUTING SCIENCES
Can be cited as:
Olalere M., Afolorunso A. A., Olalere Z. M., Egigogo R. A., Buhari D., and Habeeb H. Y. (2023). Development of Hash-Based Multi-Factor Password Generating System, FUOYE Journal of Engineering and Technology (FUOYEJET), 8(4), 429-432. http://doi.org/10.46792/fuoyejet.v8i4.1047

fingerprint key of the Android device user. The other input was a field specifying the site name, app name, URL, or any other identifier for the target platform. Once these inputs were provided, the system would initiate a function that compared the recently supplied biometric value with the one already registered and stored on the device chip. If the values were determined to be a match, an algorithm would be invoked to retrieve the key associated with the biometric value. The two values, biometric and key, would then be concatenated and processed using a 256-bit SHA algorithm, resulting in a password comprising a combination of alphanumeric and special characters with a length ranging from 16 to 32 digits. Users could easily copy this password and paste it as needed for registration or login purposes on the desired app or site.

Importantly, the entire process and the resulting password were not stored anywhere on the phone, app, or email. They were simply discarded after use. This algorithm-dependent system was capable of reproducing the same password when the two required inputs were provided again. This section focuses on presenting the framework and necessary interpretation, the SHA_256-based algorithm, and the system flowchart.

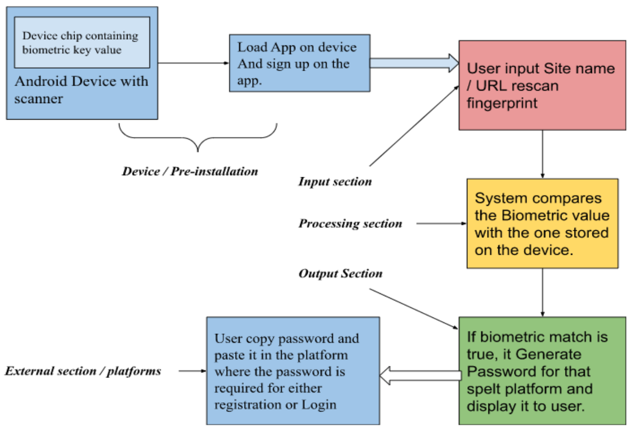

Device chip containing biometric key value
Android Device with scanner
Load App on device And sign up on the app.
User input Site name / URL rescan fingerprint
Device / Pre-installation
Input section
Processing section
System compares the Biometric value with the one stored on the device.
Output Section
External section / platforms
User copy password and paste it in the platform where the password is required for either registration or Login
If biometric match is true, it Generate Password for that spelt platform and display it to user.

Fig. 1: Developed biometric password generating software framework

2.1 Proposed Framework

Figure 1 shows the framework of the system, and the keynotes are explained below.

Device/Pre-installation: This is the beginning of the whole process where the software is installed on a mobile phone device with a scanner for fingerprints. This device must have an Android OS to be able to work properly. From the diagram, it has a chip on the device board that is used to store the biometric key value that is generated when a user activates the fingerprint function on the device. This value is going to be used for both registration and usage of the app.

Input Section: After the software has been installed, to operate and use it, it would first require you to input the platform name, unique identifier, or app name (your choice). This would be used to differentiate one password from another, and the password generated is entirely dependent on the identifier one provides. NB: This section is very important as it is both case-sensitive and unique. It implies that the password generated using the keyword facebook is different from Facebook and also from facebook.com.

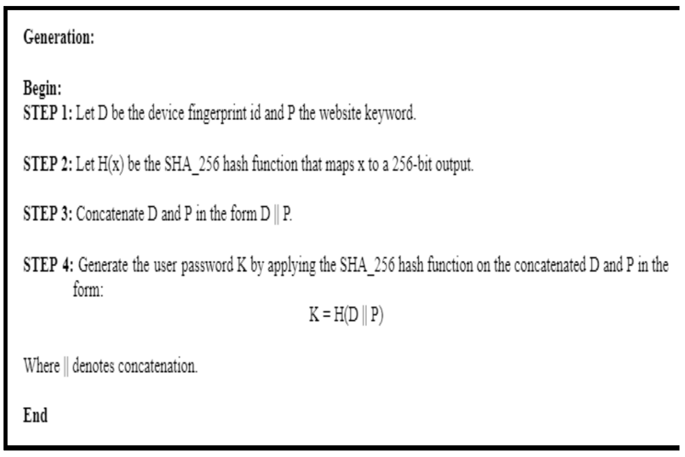
Processing Section: Once you finish inputting the identifier, you click on generate password, and the system would require you to reconfirm your fingerprint and compare the just supplied biometric value with the one stored on the device if both are the same. Once it's found to be the same, concatenates the identifier and the biometric key value, hashes the concatenated value using the SHA-256 hashing algorithm, and generates a unique 16- to 32-character alphanumeric password for the particular platform you specify.

Output Section: Once the password is generated, it displays it for the user to see, copy, and either paste on the desired app, site, or platform or type the password as displayed on the app. In addition, a copy button is added to make this process simpler.

External Section/Platforms: This section indicates every other place outside the system where the generated password can be used. An app, website, desktop app, database, etc.

2.2 The SHA-256-Based Algorithm

Listing 1 shows the mathematical representation of the SHA-256 hash-based password-generating system.



Generation:

Begin:
STEP 1: Let D be the device fingerprint id and P the website keyword.

STEP 2: Let H(x) be the SHA_256 hash function that maps x to a 256-bit output.

STEP 3: Concatenate D and P in the form D || P.

STEP 4: Generate the user password K by applying the SHA_256 hash function on the concatenated D and P in the form:

$$K = H(D \| P)$$

Where || denotes concatenation.

End

Listing 1: The SHA-256-Based Algorithm

2.3 System Flowcharts

The flowchart in Figure 2 shows the flow of data throughout the operation of the system. As seen on the diagram, the program starts by opening the application, asking the user to scan his or her fingerprint, and comparing it with the device's Fingerprint chip copy, If found to be true, the resulting fingerprint key is concatenated with the identifier supplied, hashed with the SHA_256 Hashing algorithm, and generated into a unique password.

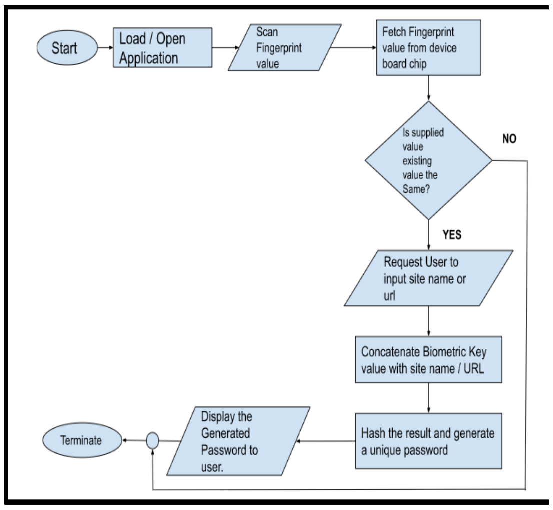

Start
Load / Open Application
Scan Fingerprint value
Fetch Fingerprint value from device board chip
Is supplied value existing value the Same?
NO
YES
Request User to input site name or url
Concatenate Biometric Key value with site name / URL
Hash the result and generate a unique password
Display the Generated Password to user.
Terminate

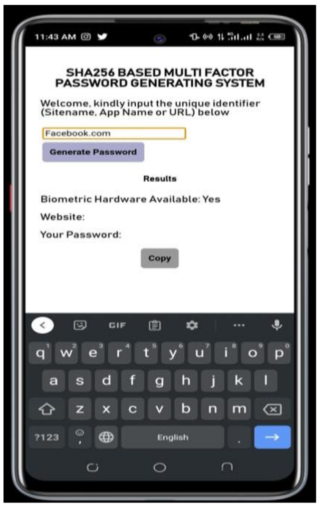
Fig. 2: Flowchart of biometric password-generating system

3 Results and Evaluation

The SHA-256 Biometric password-generating system's development is discussed here. Included here is the code section, which runs the software with the same entries to demonstrate the operation routine and effectiveness of the software. The model shows, among other things, how it's able to generate a unique password for an identifier and recreate such a password with the same identifier.

3.1 System Showcases

The software is developed using the JavaScript programming language, and the entire functionality is supported with some predefined plugins. Among the plugins is a function that calls and reads the biometric value, the hashing code, etc. Figures 3, 4, 5, and 6 show the result of the development.



SHA256 BASED MULTI FACTOR PASSWORD GENERATING SYSTEM

Welcome, kindly input the unique identifier (Sitename, App Name or URL) below

Facebook.com

Generate Password

Results

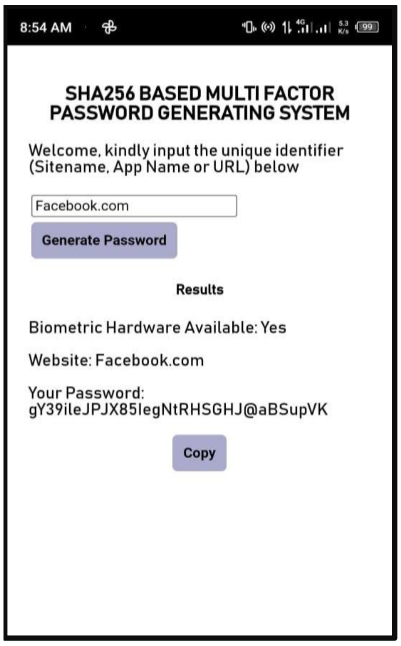
Biometric Hardware Available: Yes

Website:

Your Password:

Copy

Fig. 3: Password and the unique website ID sample



SHA256 BASED MULTI FACTOR PASSWORD GENERATING SYSTEM

Welcome, kindly input the unique identifier (Sitename, App Name or URL) below

Facebook.com

Generate Password

Results

Biometric Hardware Available: Yes

Website: Facebook.com

Your Password: gY39ileJPJX85legNtRHSGHJ@aBSupVK

Copy

Fig. 4: Hash-concatenated value

3.2 System Evaluation

Length: The SHA-512 algorithm produces a fixed-length output of 512 bits (64 bytes), regardless of the input size. This length provides a robust password that exceeds the recommended minimum length for secure passwords, which is typically around 8–12 characters. Thus, the app generates passwords of sufficient size to enhance security.

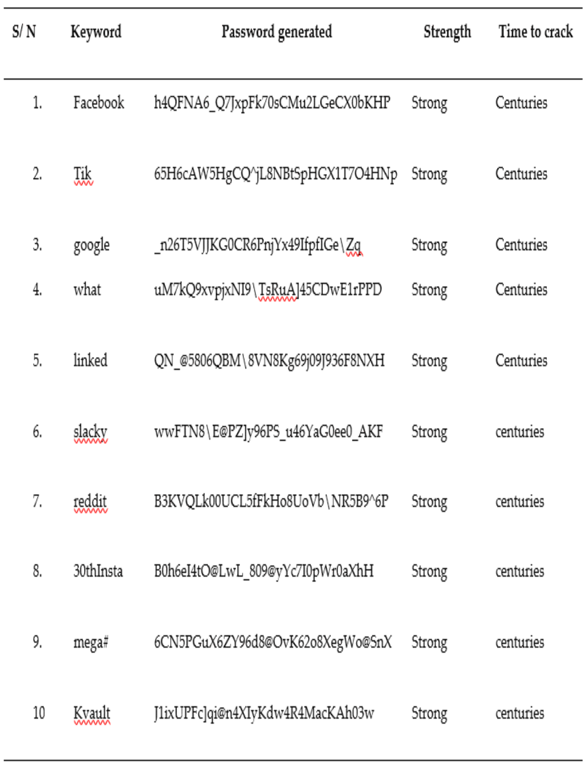
Complexity: By incorporating both an Android device biometric key and a keyword formed by the user, the app introduces complexity to the password generation process. Biometric keys typically consist of a combination of alphanumeric characters and symbols, adding another layer of complexity to the password. Combining this with the user word increases the overall complexity, incorporating both upper and lowercase letters, possibly special characters, and multiple words. This complexity enhances the resistance of the generated passwords against various attacks, such as brute force or dictionary-based attacks.

Randomness: The SHA-512 algorithm is designed to produce a highly unpredictable and random output, given any input. Although the app incorporates deterministic factors like the biometric key and the user keyword, the password generation process still maintains a sufficient level of randomness due to the cryptographic properties of SHA-512. This randomness ensures that the generated passwords are not easily guessable or susceptible to pattern-based attacks.

Table 1 provides compelling evidence that the passwords generated possess a high level of strength using a renowned password testing tool owned by Bitwarden, an Open-source password manager. The tool rates password strength on a scale of 'very weak', 'weak', 'good', and 'strong'. In addition, Table 2 affirms that the system efficiently produces the exact password when a specific

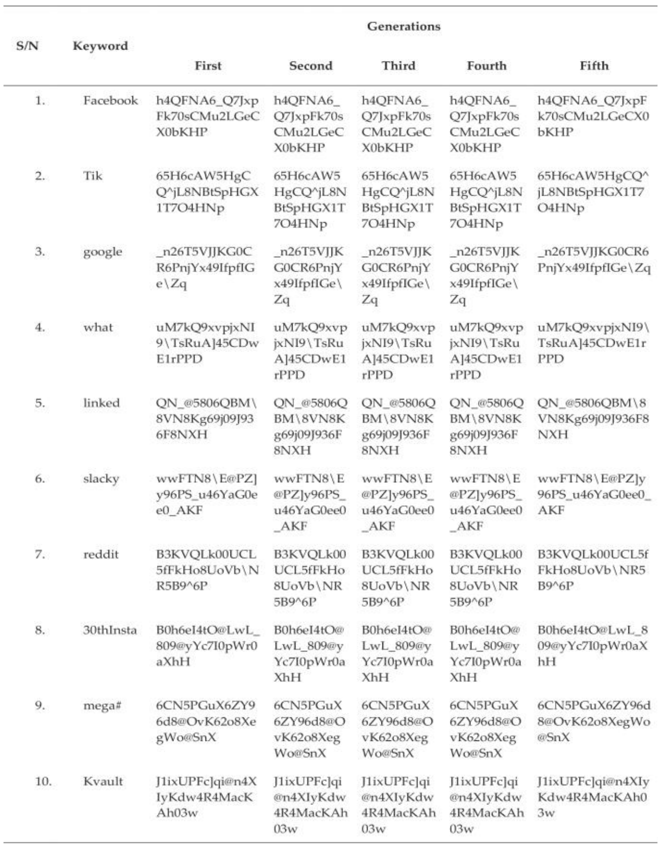
user inputs the same keyword while at the same time generating a uniquely random one when they are little changes in the parameters.
With this evaluation, the system is said to have passed the evaluation tests.

Table 1. Evaluation of Password Strength grayscale



S/ N	Keyword	Password generated	Strength	Time to crack
1.	Facebook	h4QFNA6_Q7JxpFk70sCMu2LGeCX0bKHP	Strong	Centuries
2.	Tik	65H6cAW5HgCQ^jL8NBtSpHGX1T7O4HNp	Strong	Centuries
3.	google	_n26T5VJJKG0CR6PnjYx49IfpfIGe\Zq	Strong	Centuries
4.	what	uM7kQ9xvpjxNI9\TsRuAJ45CDwE1rPPD	Strong	Centuries
5.	linked	QN_@5806QBM\8VN8Kg69j09J936F8NXH	Strong	Centuries
6.	slacky	wwFTN8\E@PZJy96PS_u46YaG0ee0_AKF	Strong	centuries
7.	reddit	B3KVQLk00UCL5fFkHo8UoVb\NR5B9^6P	Strong	centuries
8.	30thInsta	B0h6eI4tO@LwL_809@yYc7I0pWr0aXhH	Strong	centuries
9.	mega#	6CN5PGuX6ZY96d8@OvK62o8XegWo@SnX	Strong	centuries
10	Kvault	J1ixUPFc]qi@n4XIyKdw4R4MacKAh03w	Strong	centuries

Table 2. Evaluation of Password Retrieval Capability



S/N	Keyword	Generations				
		First	Second	Third	Fourth	Fifth
1.	Facebook	h4QFNA6_Q7JxpFk70sCMu2LGeCX0bKHP	h4QFNA6_Q7JxpFk70sCMu2LGeCX0bKHP	h4QFNA6_Q7JxpFk70sCMu2LGeCX0bKHP	h4QFNA6_Q7JxpFk70sCMu2LGeCX0bKHP	h4QFNA6_Q7JxpFk70sCMu2LGeCX0bKHP
2.	Tik	65H6cAW5HgCQ^jL8NBtSpHGX1T7O4HNp	65H6cAW5HgCQ^jL8NBtSpHGX1T7O4HNp	65H6cAW5HgCQ^jL8NBtSpHGX1T7O4HNp	65H6cAW5HgCQ^jL8NBtSpHGX1T7O4HNp	65H6cAW5HgCQ^jL8NBtSpHGX1T7O4HNp
3.	google	_n26T5VJJKG0CR6PnjYx49IfpfIGe\Zq	_n26T5VJJKG0CR6PnjYx49IfpfIGe\Zq	_n26T5VJJKG0CR6PnjYx49IfpfIGe\Zq	_n26T5VJJKG0CR6PnjYx49IfpfIGe\Zq	_n26T5VJJKG0CR6PnjYx49IfpfIGe\Zq
4.	what	uM7kQ9xvpjxNI9\TsRuAJ45CDwE1rPPD	uM7kQ9xvpjxNI9\TsRuAJ45CDwE1rPPD	uM7kQ9xvpjxNI9\TsRuAJ45CDwE1rPPD	uM7kQ9xvpjxNI9\TsRuAJ45CDwE1rPPD	uM7kQ9xvpjxNI9\TsRuAJ45CDwE1rPPD
5.	linked	QN_@5806QBM\8VN8Kg69j09J936F8NXH	QN_@5806QBM\8VN8Kg69j09J936F8NXH	QN_@5806QBM\8VN8Kg69j09J936F8NXH	QN_@5806QBM\8VN8Kg69j09J936F8NXH	QN_@5806QBM\8VN8Kg69j09J936F8NXH
6.	slacky	wwFTN8\E@PZ]y96PS_u46YaG0ee0_AKF	wwFTN8\E@PZ]y96PS_u46YaG0ee0_AKF	wwFTN8\E@PZ]y96PS_u46YaG0ee0_AKF	wwFTN8\E@PZ]y96PS_u46YaG0ee0_AKF	wwFTN8\E@PZ]y96PS_u46YaG0ee0_AKF
7.	reddit	B3KVQLk00UCL5fFkHo8UoVb\NR5B9^6P	B3KVQLk00UCL5fFkHo8UoVb\NR5B9^6P	B3KVQLk00UCL5fFkHo8UoVb\NR5B9^6P	B3KVQLk00UCL5fFkHo8UoVb\NR5B9^6P	B3KVQLk00UCL5fFkHo8UoVb\NR5B9^6P
8.	30thInsta	B0h6eI4tO@LwL_809@yYc7I0pWr0aXhH	B0h6eI4tO@LwL_809@yYc7I0pWr0aXhH	B0h6eI4tO@LwL_809@yYc7I0pWr0aXhH	B0h6eI4tO@LwL_809@yYc7I0pWr0aXhH	B0h6eI4tO@LwL_809@yYc7I0pWr0aXhH
9.	mega#	6CN5PGuX6ZY96d8@OvK62o8XegWo@SnX	6CN5PGuX6ZY96d8@OvK62o8XegWo@SnX	6CN5PGuX6ZY96d8@OvK62o8XegWo@SnX	6CN5PGuX6ZY96d8@OvK62o8XegWo@SnX	6CN5PGuX6ZY96d8@OvK62o8XegWo@SnX
10.	Kvault	J1ixUPFc]qi@n4XIyKdw4R4MacKAh03w	J1ixUPFc]qi@n4XIyKdw4R4MacKAh03w	J1ixUPFc]qi@n4XIyKdw4R4MacKAh03w	J1ixUPFc]qi@n4XIyKdw4R4MacKAh03w	J1ixUPFc]qi@n4XIyKdw4R4MacKAh03w

4 Conclusion

The SHA-256 multifactor generator utilizes three factors: biometric data, a unique identifier, and a phone equipped with a biometric scanner. This comprehensive system offers a distinct and practical solution for generating unique, non-memorable passwords for each online platform and application without requiring users to remember them. Importantly, this approach does not rely on a password database, instead employing an algorithm to generate the same password if the unique factors remain unchanged. Consequently, the likelihood of password attacks on users' online platforms is significantly reduced. Future work could involve expanding its functionality as a browser extension, an iOS-compatible tool, or even a freely available application on app stores.

References

Bradley, T. (2021). Microsoft takes first steps to finally kill the password. Forbes. Retrieved April 26, 2023, from https://www.forbes.com/sites/tonybradley/2021/09/20/microsoft-takes-first-steps-to-finally-kill-the-password/?sh=1dd7fb8f46d1

Jakkal, V. (2022). The passwordless future with Microsoft. Microsoft Security Blog. Retrieved April 26, 2023, from https://www.microsoft.com/en-us/security/blog/2021/09/15/the-passwordless-future-is-here-for-your-microsoft-account/

Martin, A. (2021). New Blog Post: The Passwordless Future is here for your Microsoft account. techcommunity.microsoft.com. Retrieved April 26, 2023, from https://techcommunity.microsoft.com/t5/security-compliance-and-identity/new-blog-post-the-passwordless-future-is-here-for-your-microsoft/m-p/2756606