Research of Authentication Methods in Mobile Applications

Bohdan Oliinyk[1], Yurii Shcheblanin[1], Oleg Kurchenko[1], Oleksandr Toroshanko[1], and Nataliia Korshun[2]

[1]Taras Shevchenko National University of Kyiv, 60 Volodymyrska str., Kyiv, 01601, Ukraine
[2]Borys Grinchenko Kyiv University, 18/2 Bulvarno-Kudriavska str., Kyiv, 04053, Ukraine

Abstract

Today, most web applications use authentication methods that often rely on social media authentication or simple registration by entering a unique ID and password. When it comes to the use of social media authentication methods in large companies, these solutions are driven by their relative simplicity, speed, and reliability. It is commonly believed that their encryption algorithms and methods of transmitting authentication data are encapsulated and protected from hacking. However, the analysis shows that each method has its advantages and disadvantages. The paper considers somewhat non-standard and, in some cases, faster methods of user authentication in web applications by using the capabilities of the Android operating system and its server.

Keywords

Access control, authentication, biometrics, continuous authentication, cryptography, cybersecurity, cyber threats, data privacy, identity management.

1. Introduction

If we talk about authenticating a conditional user in any system, from highly secure password storage software (Bitwarden, 1Password, etc.) [1] to a simple image editor [2], then usually, when downloading, we see a window with the possibility of choosing several authentication methods: authenticate through a social network, or using a phone number, email and, if necessary, entering a password.

In the first case, the user must have an account in the selected social network through which authentication will be performed [3].

In the second case, the user needs to go from entering his or her email to entering a password twice.

At first glance, this is not a complicated procedure, but if we imagine a situation in which several users want to have access to a conditional password manager or even a photo editor library, from a simple family that wants to share a password to a platform where they study together, or a company that uses a shared account in a service that a large number of employees should have access to, this causes certain technical difficulties and risks [4–7].

2. Formulation of the Research Task

In the cases considered, it would be nice if the authentication system could quickly and with minimal user action provide access to an existing account. Some methods will use only the capabilities of the Android operating system, while other authentication methods will be performed exclusively on the server.

The server is a computer configured to respond to messages from the application. It stores data, processes it, searches for it in databases if necessary, and issues it as a response to the application's request.

Only the result of their execution with data that the system can use to directly authenticate the user will be returned to the phone.

Consider the following authentication methods [8]:

- User authentication by IP address.

CPITS 2023: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2023, Kyiv, Ukraine
EMAIL: bogdasin321@gmail.com (B. Oliinyk); sheblanin@ukr.net (Y. Shcheblanin); kurol@ukr.net (O. Kurchenko); toroshanko@gmail.com (O. Toroshanko); n.korshun@kubg.edu.ua (N. Korshun)
ORCID: 0000-0001-6494-9483 (B. Oliinyk); 0000-0002-3231-6750 (Y. Shcheblanin); 0000-0001-7671-8287 (O. Kurchenko); 0000-0002-2354-0187 (O. Toroshanko); 0000-0003-2908-970X (N. Korshun)
© 2023 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3421/short14.pdf

- User authentication using a QR code.
- User authentication using a connection code.
- User authentication using Network Service Discovery (NSD) technology.

3. Presentation of the Main Material

For all authentication methods, we have the following prerequisite:

An account has been created in the system and there is at least one user who is already authenticated to this account.

The task of the authentication system is to ensure that another user successfully authenticates to an existing account.

1. Authentication by IP address

This type of authentication can only be performed on one device and is used to quickly re-enter an account if the user has previously logged out. It is performed mainly with the help of a server that will act as a kind of hub and will issue authentication permissions based on the IP address of the device and the encrypted token. The device will only need to contact the server to get permission to re-authenticate.

If the user re-opens the application using the same IP address within a certain period and the server confirms his/her token, the device is successfully authenticated. A possible scheme of this process is shown in Fig. 1 [9]. Note: in the diagram, one device is shown as two for ease of perception.

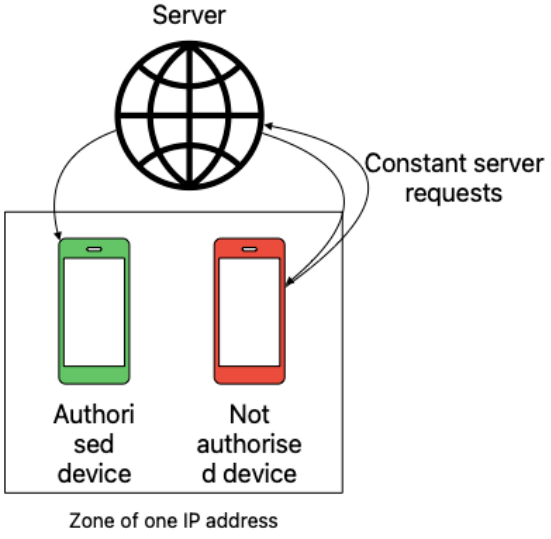

Figure 1: Pairing by IP scheme

The advantages of an authentication system using an IP address are as follows:

- Essentially, on the device that we want to authenticate in the system, we do not need to force the user to take any action, the application will poll the server and when it receives confirmation that the device is authenticated, we can give access to the account.
- This system is characterized by the high speed of the authentication process.
- Since it is possible to implement such a system only using your server, you can note the high level of reliability of this method.

The disadvantages of this authentication system are as follows:

- In some cases, the implementation of this logic can be difficult.
- You need to have your server.
- Because the IP address of the device is analyzed, there is a possibility to authenticate the wrong device if the method is implemented incorrectly.
- It cannot be said that this method is 100% secure.
- Can only be used on a device that has already been authenticated before.
- The authenticated device must be located in the same IP address area.

2. User authentication using a QR code

To implement this authentication method, the user must scan the provided code with any available QR code scanner and be either near the device on which it is generated or can scan it [10].

The technical implementation of this authentication method involves the implementation of a QR code generation procedure.

The Android Studio IDE software [11], which is used to develop applications for Android, provides tools for generating a QR code, in the context of this study we will not consider how to create an application for generating a QR code, this technology will be considered in the next study.

Let us consider how the system on which this QR code is scanned understands that it is necessary to authenticate the user and allow him to enter the application [12].

So, the system, when generating a QR code, starts a procedure that accesses the server and passes it certain parameters that will be required to authenticate another client in this account. After receiving these parameters, the server must return a so-called Deeplink in its response [13].

A Deeplink is a hyperlink that redirects the user to a specific section of an application or website. This feature reduces the number of intermediate user actions and helps the user get to the desired page in a minimum number of steps.

Deeplink can [14]:

- Redirect the user to an application on a smartphone rather than to a page in a browser, where he or she will not be able to take any action.
- Collect statistics on conversions and visitors.
- Redirect the user directly to the desired section on the site, not to the main page.
- Be indexed in search engines.
- Be easily embedded in QR codes.

Fig. 2 shows a scheme that can be used for authentication using a QR code.

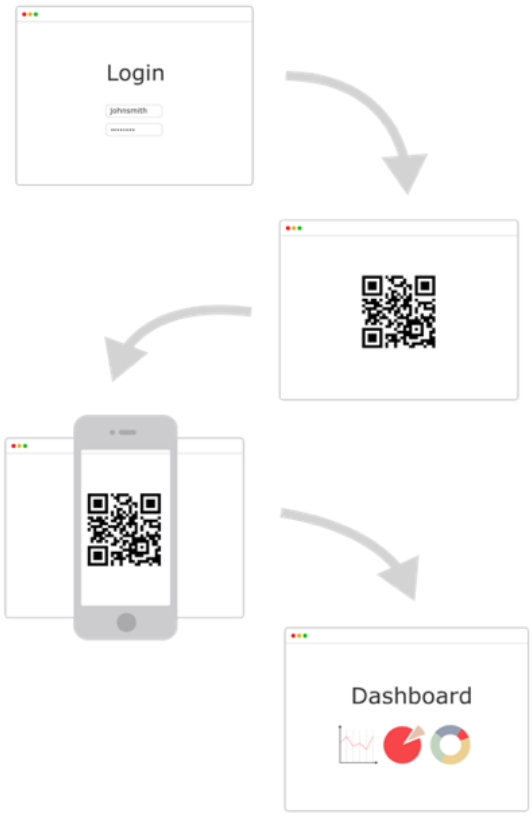

Figure 2: QR-code authentication.

It is important to remember that the implementation of Deeplink requires the use of a special service to generate such a link and make it work correctly, such as Google's Firebase service [15].

Thus, we have a ready-made QR code with a built-in Deeplink. On another client, you need to scan the QR code, which will result in a certain identifier by which we can contact the server and, if it confirms the correctness of this identifier, successfully authenticate the user to the account [16].

The advantages of this authentication method are:

- The relative simplicity of implementation, this method is quite common, so there is a lot of documentation on its implementation [17].
- Ability to use Deeplink, which greatly simplifies the implementation of this method.

At the same time, the method has the following disadvantages:

- Only those devices that have the technical ability to scan a QR code and analyze it can authenticate to the system.
- The authentication process requires the use of third-party servers, albeit reliable ones. If, for example, Google Firebase is unavailable or has some other errors, the entire authentication.
- Logic will stop working and nothing can be done about it.
- Requires additional actions from the user, which is a disadvantage, although not a significant one, as some other methods will make this process much easier.

3. Authentication using the connection code

This authentication method is implemented as follows [18].

A code is generated on the server that can consist of numbers, letters, or a combination of both. The code can also have an arbitrary length and, optionally, an expiration date [19].

Device A: initiates a request to the server. If we describe the algorithm in simple terms, it will look like this: "Generate a code and send it to me in response to this request." After that, it is possible to display this code on the device screen and ask the user to enter it on another device.

Device B performs the following actions: enters the code in the appropriate window, then encrypts it with the most convenient and possible encryption method (for example, SHA-256 using salt) and sends its hash to the server [20].

The server, in turn, processes the encrypted code sent to it, decodes it, checks whether it matches the code generated by the same server earlier, and if it does, it sends a response to Device B.

After receiving confirmation from the server, we can successfully authenticate the user (Device B) to the account.

A diagram of the implementation of this authentication method is shown in Fig. 3 [21].

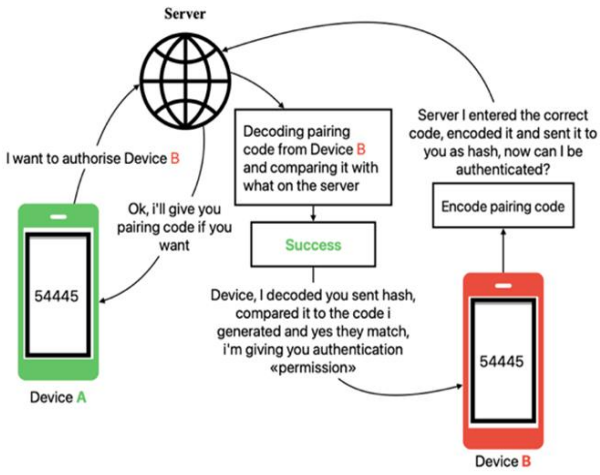

Figure 3: Authentication by pairing code scheme

The advantages of this method are [22]:

- High convenience. The authentication code can be sent to any device, it will be valid at any distance and does not require the device we want to authenticate to be nearby [23].
- A sufficiently high level of security and fault tolerance of the authentication system is achieved by using an in-house developed and administered authentication server [24].
- The flexibility of implementation, the logic can be edited to suit your needs, for example, the same code parameters, its appearance, validity, length, etc., which makes it possible to use this authentication method in different systems with different security requirements [25].

The disadvantages of the method are that:

- The implementation of the method requires the use and administration of its server.
- Physically, the user has to go through more steps in the authentication process. Compared to other methods, the user needs to: send the code if there is no physical access to the device on which it was generated, and enter the code through the input device. Also, the possibility of an error in the process of entering the authentication code by the user cannot be ruled out. At the same time, this method is the most convenient and secure if there is a large distance between the devices [26].

4. Authentication using Network Service Discovery (NSD) technology

First, you need to analyze the NSD technology itself, its features, and how it works.

So, the Network Service Discovery (NSD) function provides the application with access to services provided by other devices on the local network. Devices that support the NSD function include printers, webcams, HTTPS servers, and other mobile devices [27].

NSD implements a mechanism for searching for services based on DNS-SD.

DNS-SD allows your application to request services by specifying the type of service and the ID of the device that provides the desired type of service.

DNS-SD is supported on both Android and other mobile platforms.

To start the implementation on an authenticated device, we will need to register a special service on the network using an IP address and port. As an example, we can use the following method of use: over this network, we can transmit the so-called JSON [28] in which we can transfer any data to another device, in this case, for authentication, we can transmit an encrypted key. The other device that we need to authenticate will need to detect this service, determine the data we have transmitted and decrypt the key. After all the necessary steps, we can successfully authenticate the user. The scheme of this method is shown in Fig. 4.

Advantages of this method [29]:

- Ease of implementation, the NSD library is bundled with the Android Studio IDE software, and there is also official, well-described documentation on the website.
- Speed, if we have two devices in the same local network, we can automatically authenticate the user, no additional actions are required from the user.
- The ability to authenticate without the participation of the server.
- The ability to authenticate without access to the Internet.

The disadvantages include:

- The presence of a local network, makes it impossible to use this method for devices that are outside of it.
- Reliability is not guaranteed, messages may be lost and, as a result, authentication will not take place.
- The use of multicast traffic, which may lead to a greater discharge of mobile device batteries.
- The possibility of unauthorized access to authentication data that may be locally cached.

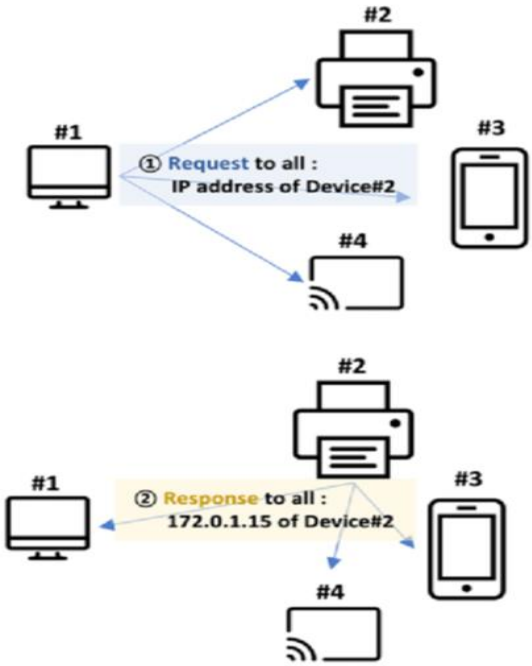

Figure 4: Multicast DNS Architecture.

4. Conclusions

The study has shown that there are many non-standard, interesting, and effective methods of user (device) authentication, in addition to those that are currently most common. The tendency to use social networks as an authentication method and enter a password and identifier is dictated by large companies that have large computing resources and can determine which authentication method the user is most likely to use to access the application [23]. In most cases, this method of authentication can be considered the most reliable due to the high security of the company itself [24].

However, such solutions may not always ensure the confidentiality, integrity, and availability of user data. It is worth recalling the process of registration in such popular networks as Instagram, Facebook, etc., where two-factor authentication technology was out of the question a couple of years ago.

In this paper, we have considered the methods of user authentication by IP address, using a QR code, using a connection code, and Network service discovery technology. Each of these methods is unique and has its specifics of implementation and the corresponding level of security, so it is not appropriate to say that one is better and one is worse.

The method of user authentication should be chosen taking into account the technology of operation and security requirements of the system in which the user (device) must be authorized.

If we talk about choosing a specific authentication method, we can consider the following cases as an example. To develop an application that should work locally or devices that will be authenticated will be located nearby, it will be convenient to use NSD and a QR authorization code, depending on the security requirements, you can choose between a more secure QR code and a less secure NSD. As an example, applications such as Walkie-talkie or Zello can serve as a good example, if you want to provide similar functionality to these, these authentication methods will be the best choice.

If you are developing an application whose user can have several options for authentication and, accordingly, have different implementations, you can use IP address authentication.

In general, there are many examples of the implementation of different authentication methods, the main thing is that their use can be different, up to the fact that you can implement authentication using all methods, it all depends on the specific goals that will be set.

5. References

[1] D. Berestov, et al. Analysis of Features and Prospects of Application of Dynamic Iterative Assessment of Information Security Risks. CEUR Workshop, 2923 (2021) 329–335.

[2] Google Developers, I Want to Encrypt Data, 2021. URL: https://developers.google.com/tink/encrypt-data

[3] L. Ho, N. Katuk, Social Login with Oauth for Mobile Applications: User's View, IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2016. doi: 10.1109/ISCAIE.2016.7575043

[4] Z. Hu, et al., Authentication System by Human Brainwaves Using Machine Learning and Artificial Intelligence, in: Advances in Computer Science for Engineering and Education IV (2021) 374–388. doi: 10.1007/978-3-030-80472-5_31

[5] Z. Hu, et al., High-Speed and Secure PRNG for Cryptographic Applications, International Journal of Computer Network and Information Security 12(3) (2020) 1–10. doi: 10.5815/ijcnis.2020.03.01

[6] M. TajDini, et al., Wireless Sensors for Brain Activity—A Survey, Electronics 9(12), iss.

2092 (2020) 1–26. doi: 10.3390/electronics9122092
[7] M. TajDini, et al., Brainwave-based Authentication using Features Fusion, Comput. Secur. 129, no. 103198 (2023) 1–11. doi: 10.1016/j.cose.2023.103198
[8] Top 10 Web Application Security Risks, URL: https://owasp.org/www-project-top-ten/
[9] K. Kang, et al., Android Phone as Wireless USB Storage Device Through USB/IP Connection, IEEE International Conference on Consumer Electronics (ICCE), January 2011. doi: 10.1109/ICCE.2011.5722588
[10] What is Android Studio, URL: https://developer.android.com/studio/intro
[11] What is Deep Linking, URL: https://www.appsflyer.com/resources/guides/deep-linking-101/
[12] Guide to Mobile Application Deep Linking, URL: https://appinventiv.com/blog/mobile-application-deep-linking/
[13] L. Moroney, The Definitive Guide to Firebase, Build Android Apps on Google's Mobile Platform, January 2017 doi:10.1007/978-1-4842-2943-9.
[14] S. Boonkrong, Authentication and Access Control: Practical Cryptography Methods and Tools, Apress; 1st ed. Edition, December 2020. doi: 10.1007/978-1-4842-6570-3
[15] J. Picolet, Hash Crack: Password Cracking Manual (v3), January 2019.
[16] P. Vyshnavi. P. Tamil Selvi, R. Gandhiraj, Android App for Arithmetic Encoding and Decoding, International Conference on Communication and Signal Processing, 2016. doi: 10.1109/ICCSP.2016.7754241
[17] S.P. Singh, SHA Algorithms-Traditional and New secure, efficient Algorithm: Cryptography, Security, August 2021.
[18] C. Xu, W. Wei, S. Zheng, Efficient Mobile RFID Authentication Protocol for Smart Logistics Targets Tracking, IEEE Access PP, 11, (2023) 4322–4336. doi: 10.1109/ACCESS.2023.3234959
[19] A. Abd El-Aziz, A. Kannan, JSON Encryption, International Conference on Computer Communication and Informatics, January 2014. doi: 10.1109/ICCCI.2014.6921719
[20] J. Jeong, J. Park, H. Kim, Service Discovery Based On Multicast DNS in Ipv6 Mobile Ad-Hoc Networks, 57th IEEE Semiannual Vehicular Technology Conference, April 2003. doi: 10.1109/VETECS.2003.1207126
[21] S. Kulibaba, Advanced Communication Model with the Voice Control and the Increased Security Level, CEUR Workshop, 3288 (2022) 64–72.
[22] P. Sun, Privacy Protection and Data Security in Cloud Computing: A Survey, Challenges, and Solutions, IEEE Access, 7 (2019) 147420–147452. doi: 10.1109/ACCESS.2019.2946185
[23] A. Mantelero, The Future of Data Protection: Gold Standard vs. Global Standard, Comput. Law Secur. Rev. 40(105500) (2021). doi: 10.1016/j.clsr.2020.105500
[24] The White House, Executive Order 14028, Improving the Nation's Cybersecurity, 2021. URL: https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity
[25] S. Rao, et al., The AES-256 Cryptosystem Resists Quantum Attacks. Int. J. Adv. Res. Comput. Sci. 8 (2017) 404–408.
[26] A. Younis, K. Kifayat, M. Merabti, An Access Control Model for Cloud Computing, J. Inf. Secur. Appl. 19(1) (2014) 45–60. doi:10.1016/j.jisa.2014.04.003
[27] Z. Durumeric et al., Neither Snow nor Rain nor MITM…: An Empirical Analysis of Email Delivery Security, 2015 Internet Measurement Conference, New York, 2015, 27–39. doi: 10.1145/2815675.2815695
[28] J. Chen, V. Paxson, J. Jiang, Composition Kills: A Case Study of Email Sender Authentication, 2020, 18.
[29] M. Haider, H. Mohammed, A Survey of Email Service; Attacks, Security Methods and Protocols, Int. J. Comput. Appl. 162 (2017) 31–40. doi: 10.5120/ijca2017913417