IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 March 2024, accepted 4 June 2024, date of publication 11 June 2024, date of current version 21 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3412691

RESEARCH ARTICLE

Transparent Multifactor Authentication Algorithm Based on Geolocation

CARLOS JAVIER GARCÍA-TREVIÑO, (Student Member, IEEE), JESÚS ARTURO PÉREZ-DÍAZ, (Member, IEEE), CESAR VARGAS-ROSALES, (Senior Member, IEEE), AND MAHDI ZAREEI, (Senior Member, IEEE)
School of Engineering and Sciences, Tecnologico de Monterrey, Monterrey 64849, Mexico

Corresponding author: Jesús Arturo Pérez Díaz (jesus.arturo.perez@tec.mx)

This work was supported in part by the Smart Digital Technologies and Infrastructure Research Group through the Project "Digital Technologies to Create Adaptive Smart Cities" as part of the Challenge-Based Research Program; and in part by the School of Engineering and Sciences, Tecnologico de Monterrey, providing the means to develop this collaborative work.

ABSTRACT Mexico exhibits one of the highest rates of identity theft in relation to bank accounts, with 56% of cardholders encountering fraudulent processes. Consequently, financial institutions have elevated security measures in their authentication protocols and implemented more robust procedures. To address this issue, this study advocates a multifactor authentication approach wherein an algorithm incorporates the user's location as a transparent authentication factor. Traditional multifactor authentication techniques typically necessitate intricate combinations of factors, including passwords, biometrics, external hardware, or time-based one-time passwords, to enhance security levels, albeit inadvertently introducing cumbersome steps. This research proposes a simple authentication implementation that retains the benefits of complex multifactor procedures -such as precision, accuracy, and security-, with the additional feature of utilizing an imperceptible location factor. The experimental phase encompasses the development of two functional prototypes: a software implementation on the Android environment -tested in 79 distinct locations-, and an implementation on Arduino hardware -tested in four locations-. These experiments were conducted in real-world scenarios, spanning a 21-day period, and involving data collection from different participants. The most significant result of this research is intricately linked to the runtime of each authentication process, where the average time elapsed from the user input stage to the completion of the validation section is 1.76 seconds. This time optimization is primarily attributed to the integration of native Android libraries. The findings demonstrate that it is possible to attain 100% accuracy for all secure locations with a 12-meter radius in 1.76 seconds in the case of the Android app. As for the Arduino-based security box, a 100% accuracy rate in all opening attempts is possible by employing a radius of 30 meters. The research endeavors to emphasize the versatility and applicability of the proposed solution for integration into diverse real-world scenarios where the preservation of data integrity holds utmost importance.

INDEX TERMS Accuracy, android, Arduino, location, multifactor authentication (MFA), security.

I. INTRODUCTION

In contemporary society, the verification of individuals in digital environments is a pivotal element across diverse domains, such as communication, social networks, electronic commerce, payments, and bank accounts [1]. The authentication process typically encompasses factors categorized as something *known* (e.g., passwords or responses to security questions), something *inherent* to the individual (e.g., biometrics), or something *possessed* (e.g., something you have, like a physical token, USB, or card) [2], [3]. When employed as a singular method, these factors constitute Single Factor Authentication (SFA).

Historically, these methods have epitomized the conventional approach to accessing personal information. The most ancient and widespread method involves the use

The associate editor coordinating the review of this manuscript and approving it for publication was Nurul I. Sarkar.

of a username and a password. However, the increasing prevalence of fraud, such as account takeover, email phishing, and stolen card fraud, underscores the urgent need for robust authentication processes. Despite advancements in authentication methods, the frequency of fraud has exponentially grown, leading to substantial global financial losses. A study made by Verizon's research group in 2022 showed that 80% of the hacked accounts have an SFA model [4], which translates into an opportunity to enhance authentication processes, necessitating modifications to ensure secure user identification.

Traditional authentication methods, particularly the username/password paradigm, have become vulnerable to cybercriminals because of the increments in computing power and networks bandwidths. Attempts to address this vulnerability through the implementation of complex passwords -for example, passwords with a defined length, including uppercase and lowercase letters, numbers, or special characters- have resulted in challenges for users, such as the difficulty in remembering intricate passwords. For this reason, and always keeping in mind comfort and ease of use, a certain sector of users prefers to continue using passwords that are not very secure but easy to remember and quick to type. Consequently, this has led to a compromise in security and an increase of susceptibility to cyber-criminal attacks. Researchers have responded to this challenge by exploring Multi-Factor Authentication (MFA), combining two or more authentication factors to strike a balance between security and user convenience [5].

This research acknowledges the need for a balanced MFA approach that considers both security and user comfort. It proposes and assesses the efficacy of a Context-Aware Multi-Factor Authentication (MFA) algorithm that incorporates Geolocation as an innovative authentication factor. It emphasizes the significance of harmonizing security levels and user simplicity in authentication procedures, and the distinctive contribution lies in recognizing physical location as a transparent authentication factor, offering advantages that are not guaranteed by conventional authentication algorithms. The motivation behind the study is rooted in addressing identity theft issues, particularly prevalent in countries like Mexico, Brazil, and the United States [7]. The research endeavors to contribute to the field by proposing a Context-Aware Authentication algorithm as a solution, seeking a balance between the heightened security offered by complex MFAs and the simplicity of Single Factor Authentication.

Now, in this paper we are proposing a Multi-Factor Authentication algorithm with two authentication factors: one as part of the *know* category and the other one as the **transparent** factor related to the location of the user while trying to login into his accounts. With this research, we intend to find a solution to the problem of having a weak algorithm for authentication -as an SFA- or an extremely complex one -such as an MFA with four or five factors- by mixing the benefits from both: the simplicity from SFA and the warranty of having different security levels as in a Multifactor procedure [8].

The authentication algorithm proposed in this research can be used in any application where login is commonly performed in a defined location, or when it is desired to exchange the use of a mandatory security factor for a security factor that is imperceptible to the user -in other words, a transparent factor-, with the purpose of increasing the levels of comfort for the end user by having an imperceptible factor, while maintaining the security levels of the process. For example, in implementations where the user is usually located in a known safe space -such as at home when accessing banking information or making a bank transaction, or a bank branch with a physical safe deposit box- they will be able to access their information using their username and password while using the location factor in a transparent way. In many countries, short kidnappings are common in order to extract money from bank cards or mobile applications; by considering the location factor from our proposed solution, we could prevent the theft of the money from the bank accounts.

Similarly, if it is desired to eliminate the use of an active security factor (such as a TOTP), and exchange it for a security factor imperceptible to the user -such as in the case of a university student who accesses his school platform using username and password plus a token generator with ''NetIQ Authenticator'', or a company worker who accesses his information from a defined physical location- the authentication algorithm proposed in this research can be used to increase the levels of comfort for the end user, as long as they can demonstrate that they are in the physical space they claim to be.

Given the design of our MFA system, a user can access their account not only from a single location, but also from a list of safe locations that they manage themselves. Depending on the desired security levels, this list can contain anywhere from one to ten safe locations per user. In this way, depending on the real-world application being addressed (bank login scenario or university platform, for example), the user has sufficient flexibility when accessing their information -as long as they are in a space that they have defined as safe-.

Among the contributions of this research, the primary achievements lie in:

1) The design of a fast and transparent Multifactor Authentication system for users incorporating two security factors (2FA), and a meticulous selection of an optimal radius to differentiate the implementations.
2) The implementation of a Multifactor Authentication system in two fully functional prototypes.
3) The execution of experiments in a minimally controlled real-world environment, and a comprehensive analysis to validate 100% accuracy.

In the next Section (II), we will concentrate on the Related Work. This will be followed by Section III, where the Theoretical Framework is included. In Section IV, the Design and Implementation will be described. The next section

will be Section V, where the Experimentation and Results Analysis will take place. At the end of the research, the Future Work and Conclusions will be addressed in Section VI.

II. RELATED WORK

In recent years, numerous endeavors have been made to integrate two or more authentication factors, exhibiting a broad spectrum ranging from simplistic combinations, such as a password paired with a physical token, to more intricate arrangements involving a password, fingerprints, and a TOTP -denoted as an acronym for Time-based One-Time Password- generated randomly by a computer algorithm and subject to periodic changes (typically every 30 seconds). Shukla et al. [9] propose a method that establishes a communication channel between the TOTP authenticator algorithm and the device attempting to log in, employing a ''Client and Server'' methodology. This process involves the exchange of key-value pairs and relevant information to generate and validate the Time-based One-Time Password.

In the work of Bartlomiejczyk et al. [10], a case study was presented, involving the integration of three authentication factors: an initial stage requiring a common username/password input (something *known*), followed by a Time-based One-Time Password entry (something *possessed*), and culminating in the biometric analysis of the user's fingerprint (something *inherent* to the user). The study's findings indicate that this specific combination achieves a satisfactory level of security, albeit at the expense of processing speed. As outlined in the paper, the overall time required for authentication quadruple in comparison to our proposed solution. It is noteworthy to mention that this method, while enhancing security, introduces additional and more intricate steps, potentially diminishing user convenience.

Furthermore, a related investigation was conducted by El Fray et al. [12], where the authors explore the incorporation of a location factor as an alternative to conventional Multifactor Authentication (MFA) methods. Notably, their methodology is positioned for potential future enhancements, as the current proposition is focused on utilizing location as a Single Factor Authentication (SFA) method. It is essential to note that the authors conducted the evaluation process in a physical and controlled environment. They acknowledge the possibility of using this location-based method as a supplementary component to other Multifactor methods, which aligns with our exploration.

The authors of the aforementioned document propose a solution utilizing the unique IP address of each device, serving as a distinct identifier for cellular phones, to measure the distance between the devices and the internet network (e.g., WiFi network or cellular network). During the initial login attempt, they record the distance information and employ it to define a ''safe zone,'' subsequently verified by other smartphones with the same application installed. The algorithm utilizes this collective information to either accept or reject the login request.

An innovative proposition is found in the article by Khattri and Singh [13], where they introduced a Multifactor Authentication (MFA) method that incorporates a Personal Identification Number (PIN), a One-Time Password (OTP), and the utilization of the Global Positioning System (GPS) to elevate security requirements for authentication. The distinctive feature of this proposal lies in the integration of GPS, wherein the distance between the login device and a user's pre-defined mobile device serves as a constraint. If the distance between these components is minimal, signifying the user's presence in a secure environment, access is granted; conversely, if the devices are distanced, the request for credentials is rejected. While this proposal represents a novel approach to addressing the initial problem, it warrants further enhancements, given that the experimentation was conducted in a virtual environment.

Similar to Khattri et al., the research of Alabdulatif et al. [15] addresses the distance between a device previously defined as secure and its location as part of an authentication factor. This work proposes an additional authentication factor to the standard ATM cash withdrawal procedures, which consists of registering an external electronic device that is designated as secure by the user. With this proposal, the ATM cash withdrawal procedure is slightly modified by linking it to the location of an electronic device: when the user wants to withdraw money, they must have their cell phone nearby in order to authorize the transaction; if they do not have their cell phone nearby, the withdrawal will not be allowed. It is important to mention that this method, despite having a transparent authentication factor, does require some external hardware -in this case, a cell phone- in order to complete the authentication process for its users. In comparison with the solution proposed in our research, we can highlight the fact that our solution is not attached to one specific external hardware or device, that the tests were conducted in a real environment, and better results were achieved in terms of both accuracy and speed of use.

The extent of the contributions of our MFA system, compared to the State of the Art, focuses on three categories: results, test quality, as well as user flexibility and convenience. Regarding the results, our system was able to achieve 100% accuracy in user authentication in just 1.76 seconds, whereas in other research, lower accuracy levels were achieved in longer times (around 95% in 3.7 seconds or more). Regarding test quality, what sets our research apart from others is experimentation in minimally controlled real-world environments, whereas in other research, experimentation was merely conducted in simulations. Similarly, another differentiator was the quantity of tests, which in our case surpasses that of any other research (more than 1900 cases compared to 200 cases). Finally, with our MFA system, user convenience takes on greater importance, as it is the user who has control over how many and which locations are considered secure (up to a maximum of 10 locations), providing greater flexibility to the applications where the MFA system could be implemented.

TABLE 1. Comparison between the proposed solution and the state of the art.

Reference	Description	Authentication Factors	Free of External Hardware	Real-World Testing	Number of Experiments	Accuracy	Time Consumption	Required Memory
Shukla et al. [9]	This research proposes a new way to generate TOTPs using a client-server approach. The main goal is to improve security by frequently changing the secret keys shared between the client and the server. This makes it harder for attackers to steal these keys and gain unauthorized access.	1: TOTP	Yes	No Testing, Just Theory	✗	✗	✗	✗
Maciej et al. [10]	This research introduces a MFA with three factors. This algorithm is applied to a web application that employs a mobile app on the Android platform to verify user identities.	1: User Credentials 2: TOTP 3: Fingerprints	No, it requires a Smartphone	Testing performed in a Simulation	200	✗	6.8 s	130 MB
Imed et al. [12]	This paper presents a SFA algorithm tailored to authenticate users near a Point-of-Sale using a location factor. When a user seeks identification at the POS terminal, their Mobile Application monitors their location. Simultaneously, a Witness Application assesses the credibility of the location data to either approve or deny the authentication request.	1: Location	No, it require 2 Smartphones: one for the MFA and another as the Witness.	Testing performed in a Physical & Controlled Environment	200	✗	4.5 s	✗
Khattri et al. [13]	MFA algorithm with three factors implemented for Online Transactions. Users register a secure device for transaction purposes. The approach involves approving or rejecting transactions based on the proximity between the registered safe device and the device used for the transaction (e.g., computer logged into an online store).	1: PIN 2: OTP 3: Location	No, it requires a Smartphone	Testing performed in a Virtual Environment	100	98.55%	30 min	✗
Samarasinghe et al. [15]	MFA algorithm with two factors. This research proposes a new MFA method for securing ATM transactions using geolocation technology. In this method, the user's smartphone serves as the second authentication factor, and its location is compared with the geographical location of the ATM being utilized. If the locations match, the transaction is authorized; otherwise, it is flagged as suspicious.	1: PIN 2: Location	No, it requires a Smartphone	Testing performed in a Simulation	20	95%	3.7 s	✗
Proposed Solution	MFA algorithm with two authentication factors. The proposed solution presents two functional prototypes: one as a software implementation on the Android platform, and another as an implementation on Arduino hardware. With the proposed solution, access is granted to the user only if the device used for logging in is near a location previously designated by the user as secure.	1: User Credentials 2: Location	Yes	Yes	+1900	100%	1.76 s	9.38MB & 503KB

For a more comprehensive understanding of the research outlined in the State of the Art, Table 1 provides a summary of the information pertaining to the related works (✗: information not available).

In summary, our investigation revealed that the fundamental authentication method, namely the use of a username and password, remains the most established, albeit having undergone enhancements to bolster security through the incorporation of supplementary methods. However, a common thread across the examined articles is the reduced consideration for user convenience, coupled with the notable limitation that most experiments were conducted in a simulated environment. While the primary focus is understandably on security, the overarching concern should also encompass user comfort. A process perceived as cumbersome or intricate may lead individuals to opt for simpler, albeit less secure, methods to circumvent complexity. Thus, striking a balance between robust security measures and user-friendly authentication procedures emerges as a crucial area for exploration.

III. THEORETICAL FRAMEWORK

A Two Factor Authentication Method for an MFA algorithm is proposed, where the main factor will be part of a subcategory from something *possessed*: the *location*. This new type of authentication is mainly related to Geolocation, which is the term used to refer to the identification of a user by its geographical coordinates. The most common way to retrieve geographical information is by a GPS -Global Positioning System- device, which is the main idea behind this project.

The GPS works by establishing communication with three or more of the satellites that are orbiting the Earth. Once the connection has been established each satellite and the GPS exchange a small signal, and depending on the amount of time that the signal traveled and its quality, it is possible for the GPS module to inform the user of the physical coordinates from where the signal exchange took place. The GPS must connect to at least three satellites, since with this amount of connections the information of latitude, longitude, and altitude (three-dimensional coordinates) will be more precise -the more satellites connected, the more precise the information-. An example of a device that uses this type of connection is the GPS modules for academic projects since their ease of integration and price.

Another way of retrieving coordinates information is by using DGPS -Differential Global Positioning System-. This type of connection is established with a similar procedure as with the standard GPS, but instead of using satellites from the outer world, they use ground-based reference stations. These reference stations are giant antennas that work in combination with satellites from around the world to correct and improve the coordinates received by the GPS module. Examples of devices that use this type of connection are modern cellphones and tablets that can connect to the cellular network.

With this in mind, we proposed two *physical* implementations for this project, which are going to be further explained in the upcoming section. The first one will consist of a mobile application for the Android Environment, which is going to use DGPS to retrieve information related to the geolocation of a user -in addition to other *know* factors- to guarantee or deny the credentials to log in to a personal account. The second implementation -using traditional GPS-, will consist of an Arduino implementation that will represent a physical safe box that opens when inside a safe location delimited by a defined radius.

IV. DESIGN AND IMPLEMENTATION

The proposed Multifactor Authentication method comprises two primary identification factors— the first falling under the category of something *known* and the second belonging to the subcategory of *location*. A third backup factor is included, which may or may not be utilized depending on the specific application; in any case, this factor falls within the something *possessed* category. The central concept of this MFA is to authenticate users during login attempts on their personal accounts only when they are in proximity to a location they have defined as secure.

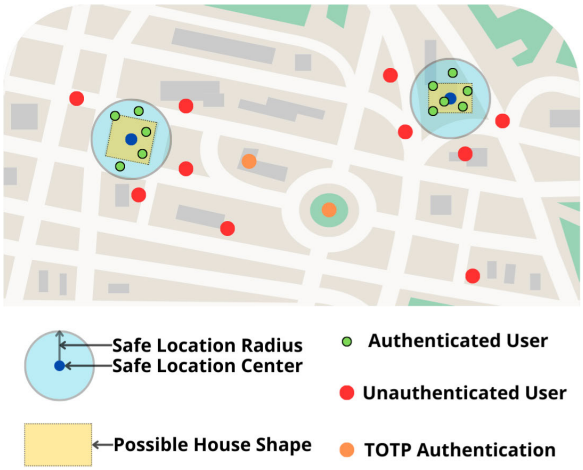

FIGURE 1. Multiple login attempts around two trusted locations.

Depending on the implementation context, each user may have a maximum of ten secure locations. These locations possess two essential properties for this proposal: their center and their radius. The former property defines the central coordinates of the secure space, while the latter confines the area of that secure space. With this information, we can summarize that secure locations for this research are circular in shape.

To counteract the possible use of location spoofing by cyber-criminals, our main advantage lies in the fact that the user must first know the locations previously defined as secure. In other words, it is not enough to know how to use a location spoofer; you must also know beforehand what the user's secure locations are, as well as their exact coordinates. Considering that the secure locations are encrypted in the database, this gives users a higher level of security against possible hacks. In addition, it is important to mention that security can be future enhanced to completely avoid any spoofing attack by linking the IMEI or SIM of the mobile phone as a third authentication factor, however this modification would rest flexibility to our solution, since a specific external hardware would be required.

During a login attempt, the algorithm retrieves the user's latitude and longitude coordinates through the device's GPS. If these coordinates fall within any previously defined secure space, the user authenticates successfully. Conversely, if the coordinates lie outside the secure spaces, the user fails to authenticate correctly. Now, in the event that the real user has issues with their authentication (either caused by the signal of their cell phone, GPS problems, or simply being outside their safe locations), they will still be able to access their account as long as they provide the necessary information for the backup factor -third security factor, which in this case is in the form of a TOTP-.

In Figure 1, an example of login attempts by a user in two secure locations is illustrated:

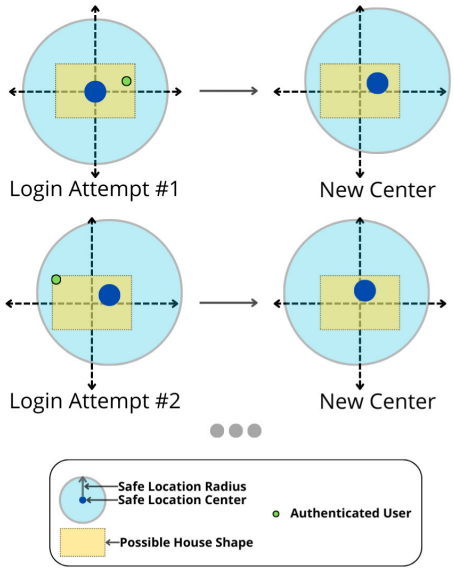

FIGURE 2. Update of the circle center given various login attempts.

It is crucial to highlight that the proposed solution has the capability to learn from user behavior and their electronic devices by consistently updating the center of each of their secure locations. During a new login attempt, the algorithm takes into account the latitude and longitude coordinates of the circle's center, and in conjunction with the current latitude and longitude coordinates of the mobile device, a weighted average of the information is calculated. Consequently, the center of the secure space is continuously adjusted to more accurately represent the user's chosen secure space. This behavior is illustrated in Figure 2:

With this design, two functional physical prototypes were developed for this research, the first being coded as an Android mobile application, while the second was developed in the Arduino programming environment. For both implementations, the location factor was used to authenticate users successfully when they are within a recognized safe space.

A. ANDROID IMPLEMENTATION

The Multifactor Authentication algorithm used in this implementation consists of three security factors: the first -from the category something *known*- in the form of user credentials (username and password, used in each login attempt), the second -from the subcategory of *location*- in the form of latitude and longitude coordinates (used in each login attempt), and finally a third factor -from the category of something *possessed*- in the form of a TOTP code that is used only in cases where the user is not within a previously registered safe location. This last factor is considered a backup factor, as it is not used constantly to authenticate the user.

The functional prototype for this implementation was developed using Android Studio, in conjunction with a remote database located on the Firebase servers, with the purpose of creating an application for Android mobile devices. The radius for this implementation was initially selected as **30 meters**.

1) MOBILE APPLICATION

The mobile application was developed for the Android environment, given its diverse low-level and freely available libraries, which provide various tools that can be integrated into data collection through the mobile phone's GPS module. Alternative approaches employing similar tools, such as React libraries, necessitate a more extensive computational time for execution. Furthermore, the Android platform offers the capability to incorporate plug-in programs or APIs for connection to a remote database.

The mobile app uses the following libraries:

1) **Navigation Graph** (version 2.5.2): The Navigation Graph library for Android simplifies screen navigation by centralizing all navigation-related information in an XML resource file. This file defines the various destinations (screens) in the app and the connections (actions) between them, providing a clear and concise overview of the app's navigation flow [17]. This library was primarily utilized to refine the visual appearance of the graphical user interface in the application.
2) **Google Maps** (version 18.1.0): The Google Maps SDK for Android is a powerful tool that allows to integrate maps features into the Android apps. It provides access to a wealth of features, including map views, marker placement, and place search [18]. This library was mainly used to graphically integrate a Google Map screen in the application.
3) **Google Play Services** (version 20.0.0): Google Play Services is a comprehensive collection of APIs and services that enhance the Android app's functionality, user engagement, and overall quality. By integrating Google Play Services, we can significantly improve the app's capabilities and offer a richer and more user-friendly experience [19]. This library was primarily used to retrieve the latitude and longitude coordinates from the cellular devices.
4) **Firebase** (version 30.4.1): The Firebase Authentication library for Android provides tools to easily integrate user authentication into apps. It supports email/password login, and social login with providers like Google, Facebook, and Twitter. Additionally, it offers features like real-time user presence, anonymous authentication, and token management, making it a comprehensive and versatile solution for secure user management in Android applications [20]. This library was mainly utilized as an online database integration.

2) FIREBASE DATABASE

The decision was made to utilize a connection to the Firebase database within the Android program. This database stores all the information of the participating users in the implementation, along with their secure locations. It is

noteworthy that the user passwords are internally encrypted by Firebase, ensuring that only the account owner has access to them.

3) GENERAL ARCHITECTURE

The Configuration and Authentication process for this implementation are illustrated in Figure 3 and 4. The Configuration Process shows the flow chart for creating a new account with this MFA algorithm, while the Authentication Process shows the flow chart for using the MFA algorithm once the user has already created an account.

4) CONFIGURATION PROCESS

The first step in utilizing the proposed authentication method is the creation of an account. For the Android ecosystem implementation, it was decided to use a username and password as the first security factor. These credentials are entered by the user on the initial screen of the application, as depicted in Figure 5 (Left). Subsequently, the user must configure the backup security factor in the form of a TOTP code. Upon completing this configuration, the information is encrypted and transmitted via the Firebase library to the remote Firebase console. At this point, the user has successfully created his account and can capture his first secure location.

When capturing a new secure location (which could be the first or any subsequent one), the user is presented with a screen as shown in Figure 5 (Right). In this step, the user can view their location on an interactive map in real-time. This is intended to visually confirm that the location detected by the Google Play Services library is indeed the actual location of the device. Once the user is satisfied with the accuracy of their new location, they must assign it a nickname for subsequent management.

When finishing these steps, each time the user wishes to access his account, he will be able to do so seamlessly — meaning, by entering only his username and password— at his secure location(s).

5) AUTHENTICATION PROCESS

This subsection will elaborate in greater detail on the internal process of our Multifactor Authentication (MFA) system. Initially, the user is required to input their access credentials (first security factor) with which they registered. This can be seen in Figure 6 (Left). Using this information, the algorithm reads the Firebase server database, searching for a user with the entered information. If no user is found, it indicates that the user does not have an account and therefore cannot utilize the Multifactor Authentication algorithm.

On the contrary, if the user has an account and at least one registered location, the information of their secure locations (IV-A2) will be compared against the location at the time of the login request. To perform this distance calculation, the Haversine formula from Purnomo et al. [21] is utilized. This formula is commonly employed to compute the distance between two sets of latitude and longitude coordinates.

If the distance between any secure location and the user's current location is less than the maximum allowed radius -30 meters (the starting value considered in this implementation while we tune the minimum radius to achieve 100% of accuracy)-, user access will be granted, as shown in Fig. 6 (Right).

B. ARDUINO IMPLEMENTATION

The Multifactor Authentication algorithm employed in this implementation comprises two security factors. The first, categorized as something *known*, takes the form of a numerical password of variable length, utilized in each login attempt. The second, falling under the subcategory of *location*, is represented by latitude and longitude coordinates, employed in each login endeavor. Due to the specific nature of this application, the inclusion of a backup factor for user authentication was not feasible. The implementation functions as a secure box, akin to those used in banking, which can only be opened in close proximity to the location where it was previously closed.

The secure box initiates its process when it is open, and upon closing, the coordinates of the secure location are configured. To reopen the box, it is necessary to be in proximity to the location where it was closed. Once the box is open again, a new configuration process for the secure location commences. In other words, we can state that the secure space is redefined with each closure of the secure box, as shown in Fig. 7. It is crucial to note that, when in the context of location #2, the secure area of location #1 is no longer valid, as this space is redefined with each closure.

The functional prototype for this implementation was developed using the Arduino Integrated Development Environment (IDE) in conjunction with a local database stored on a microSD card. The purpose of this development was to code a microcontroller capable of authenticating users. The radius for this implementation was initially selected as **30 meters**.

1) LIBRARIES AND DOCUMENTATION

The algorithm for integration with a safe box was developed for the Arduino microcontroller, as it has various public libraries that are constantly developed by the community and that have the tools necessary to be incorporated into data collection through an external GPS module.

The Arduino code uses the following libraries:

1) **Keypad** (version 3.1.1): The Keypad library for Arduino simplifies interfacing with matrix-style keypads. It abstracts away the complexities of pin management, scan routines, and debouncing, allowing you to easily read key presses with clear and concise code [22]. This library was primarily used to process the activation of the buttons on the matrix keypad.
2) **LiquidCrystal_I2C** (version 1.1.4): facilitates controlling I2C-connected LCD displays. It offers functions

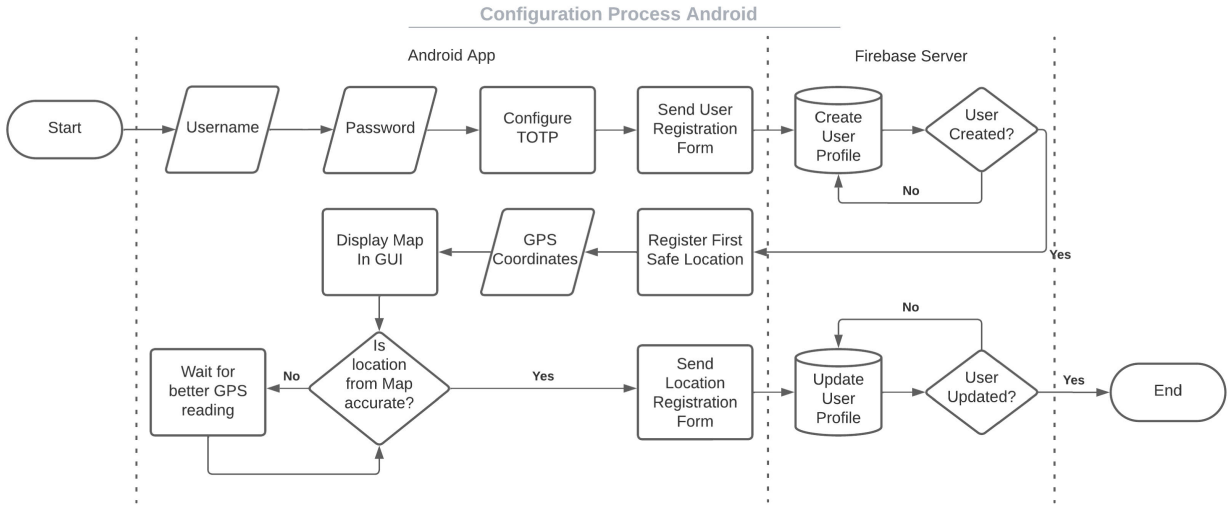

FIGURE 3. Flow chart for configuration process in android.

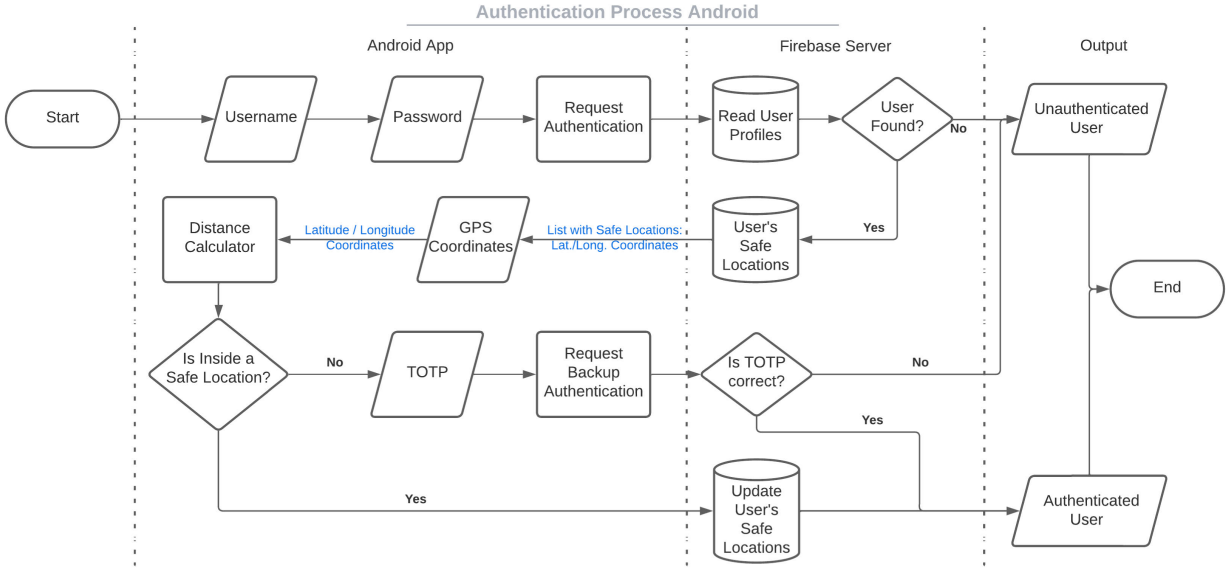

FIGURE 4. Flow chart for authentication process in android.

analogous to the standard LiquidCrystal library, allowing you to easily display text, create custom characters, and manage various display options [23]. This library was mainly used to graphically represent on the LCD the information that the user was entering for the authentication process.

3) **SD** (version 1.2.4): The SD library for Arduino provides a simple and efficient way to read and write data to SD cards. This library made it easy to store data and load files from an SD card [24].
4) **TinyGPS++** (version 1.0.3): compact and resilient library for Arduino that parses the most common NMEA sentences used by GPS devices. These sentences contain information like latitude, longitude, altitude, time, and speed. By parsing these sentences, TinyGPSPlus allows you to easily extract and use this information [25].

2) LOCAL DATABASE

It was decided to use a local database, so a physical connection with a microSD memory card was implemented within the Arduino program. All information about the participant's login attempts is stored in this component.

3) ELECTRIC CIRCUIT

Given the ever-evolving nature of this programming environment, finding the necessary electronic components for the Arduino implementation was a straightforward task. The components were chosen based on their ease of use, available documentation (IV-B1), and cost-effectiveness.

The Electronic Circuit employs the elements from Fig. 8, which are:

1) **Arduino UNO** (x2): versatile and beginner-friendly microcontroller board designed for electronics

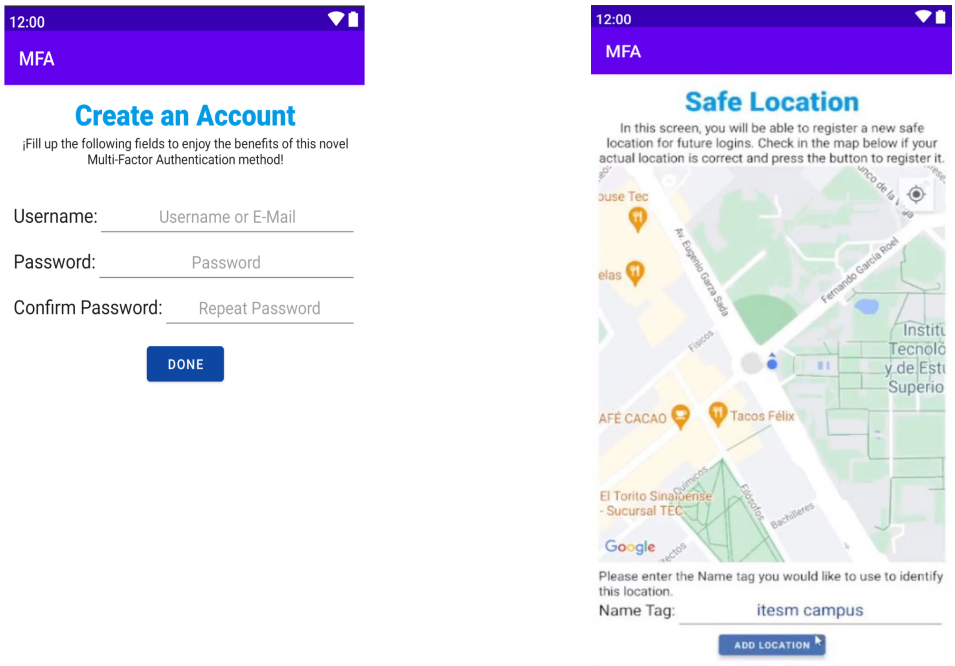

FIGURE 5. Initial configuration: Create an account (left) & register new location (right).

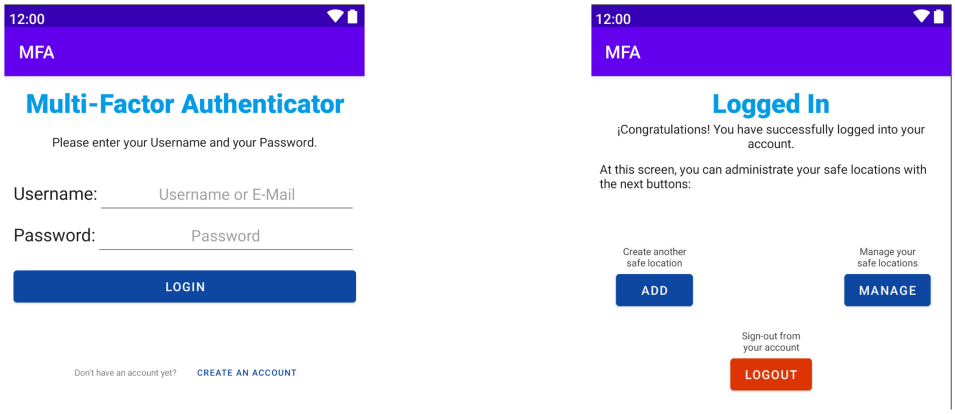

FIGURE 6. Account activity: Login into an account (left) & account management (right).

prototyping. It has built-in pins that can connect to sensors, LEDs, and other components, allowing to control them through code.

2) 4 × 4 **Matrix Keypad**: compact and versatile input device. It uses a clever matrix arrangement to simplify wiring, which results in a reduction in the number of pins required on the microcontroller.
3) **LCD1602 with I2C Module**: compact liquid crystal display that allows to display text and numbers from the microcontroller board. The I2C module simplifies wiring and frees up precious pins from the board.
4) **GPS NEO-6M Module**: tiny navigation module for Arduino projects. It uses satellite signals to pinpoint your location anywhere on Earth.
5) **MicroSD Card Reader**: module for expanding the project's memory, acting like a tiny, portable hard drive.

For this implementation, it was necessary to use two Arduino boards. The reason for this decision is related to the microcontroller's SRAM: as it constantly collects information from the GPS module, there may come a point where the dynamic memory of the board -which is only 2KB per Arduino for the models used in this research- is exhausted. To avoid this inconvenience, the decision was made to use two microcontrollers simultaneously. One of them acts as the primary, receiving information from the matrix keypad, displaying data on the LCD, and storing login attempts in the local database. Meanwhile, the secondary is responsible for collecting data received from the GPS module.

4) GENERAL ARCHITECTURE

The Authentication process for this implementation is illustrated in Figure 9. This figure represents the flow diagram

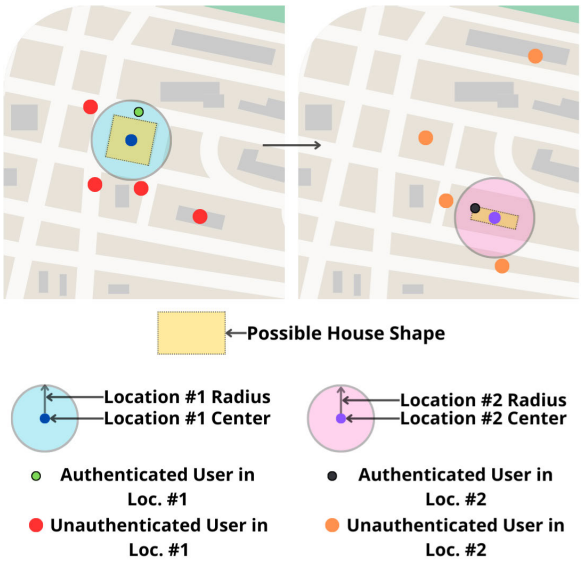

FIGURE 7. Multiple login attempts, where the trusted location is redefined.

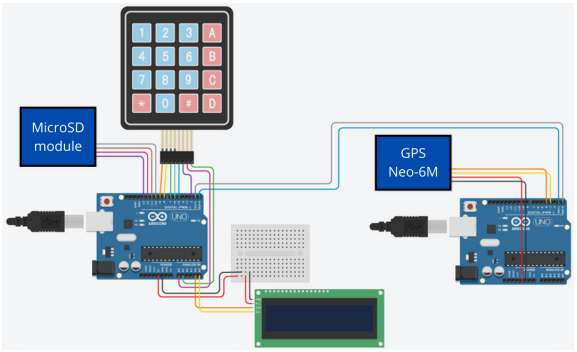

FIGURE 8. Electric circuit for authentication process in arduino. Primary circuit on the left, Secondary on the right.

for opening a safe box, provided that it is located in the vicinity of the coordinates when it was closed.

5) AUTHENTICATION PROCESS

As an initial condition to employ the proposed authentication method, we must assume that the secure box is in an open state —meaning, objects can be inserted and removed-. When the desire arises to close it, the first security factor (something *known*) must be utilized in the form of a freely defined numerical password. This authentication credential is entered on the matrix keypad and simultaneously displayed on the LCD screen, which serves as a visual aid for the user at each step of the process. After the user has selected the password, they must press the confirmation button on the matrix keypad, triggering the operation of the second security factor (the *location* subcategory). Subsequently, information is collected through the GPS module for storage in the database (Section IV-B2). At this point, we can be certain that the secure box has been closed successfully.

Now, the opening process mirrors the closing process. The user must enter the password defined earlier during the closing stage. If the password is incorrect, access to the secure box will be denied. If the password is correct, the algorithm will verify, through the transparent second location factor, that the secure box is physically within a radius of no more than 30 meters from the point where it was closed -this calculation is performed using the Haversine formula [21]-. If the distance is less, access to the contents of the box will be granted. However, if the distance is greater, the box will remain closed.

It is noteworthy that this algorithm is iterative, and the final opening moment leads to a new closure of the secure box in a cyclical manner.

V. EXPERIMENTATION AND RESULT ANALYSIS

To test the proposed solution, two functional prototypes were deployed in physical format implemented in minimally controlled real environments. This section will discuss the requirements, the test environments, as well as the results obtained from both implementations.

A. ANDROID IMPLEMENTATION

1) STORAGE REQUIREMENTS

The Android Package Kit (APK) plays a crucial role in the Android ecosystem by facilitating app development and distribution, ultimately enhancing user experience. It serves as the fundamental distribution and installation format for applications on Android devices. Functionally, it encapsulates all necessary files for an app, including code, resources, and configuration data, enabling a standardized installation process. Advantages of APKs include self-containment for easy sharing and installation, as well as compression for efficient storage utilization.

For this reason, the application was distributed to participants in this implementation in APK format, which only required 9.38MB of system storage space.

2) TEST ENVIRONMENT

The experimentation process was carried out in a real testing environment with users, where each had the freedom to use their trusted mobile device in geographical spaces of their preference. As a primary guideline, it was essential to follow the configuration and account creation process described in Section IV-A4. Subsequently, participants were instructed to perform the highest number of login attempts in as many locations as possible, as detailed in Section IV-A5.

For this experiment, there were **20 participants**, among whom **79 different locations** were registered, resulting in **1165 successful login attempts**.

3) RADIUS AND TIME REDUCTION

Once the testing phase was completed, the information from all participants was collected for result analysis. As an initial indicator, a descriptive statistics analysis was performed to understand the distribution of the information.

The tests on Android yielded positive results related to the research proposal of utilizing location as a transparent

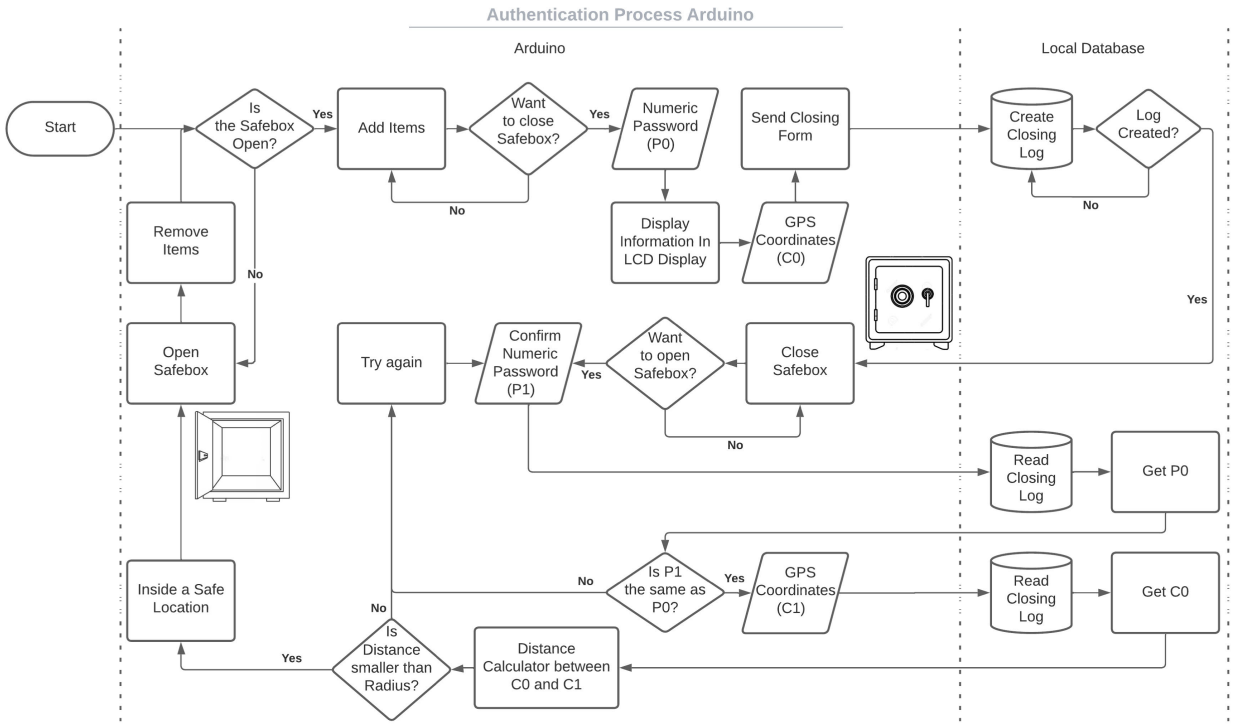

FIGURE 9. Flow chart for authentication process in arduino.

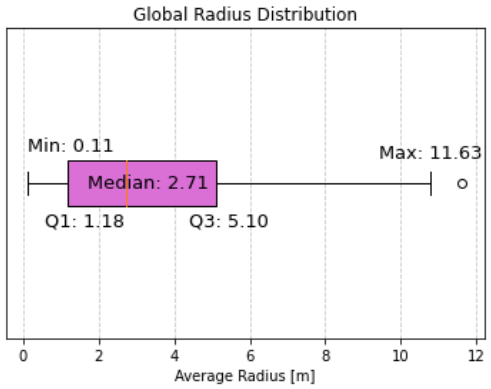

FIGURE 10. Radius descriptive statistics with android results.

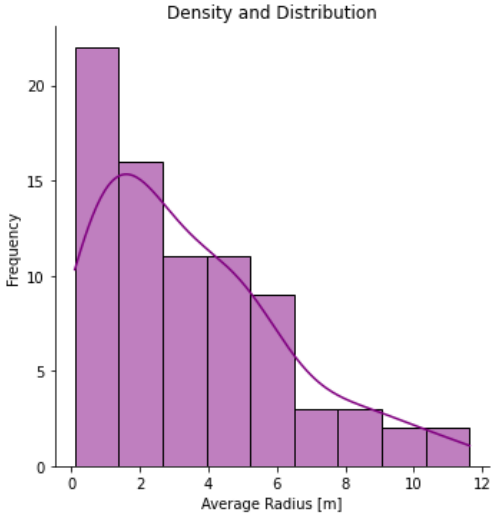

FIGURE 11. Radius density and distribution with android results.

security factor. Modern mobile devices have sufficient technology to extensively leverage location data collection through DGPS. As observed in Figure 10, the maximum radius among the 79 locations was 11.63 meters. Regarding quartiles, the first quartile is delimited by a radius of 1.18 meters, while the third quartile has a radius of 5.10 meters. Similarly, an average radius of 3.5 meters was obtained, indicating that the originally defined 30-meter radius has ample room for improvement.

A decision was made to reduce the radius for the Android implementation to better delimit each user's secure space. As shown in Figure 11, the average radius has a distribution centered at 2.71 meters, and for radii greater than 12 meters, there are hardly any cases where users have successfully attempted login. In other words, for larger radii, the frequency of successful login cases stagnates at zero.

With this information, a radius reduction analysis was conducted for this implementation. The calculations made to determine the accuracy of this implementation are solely related to the reduction of the initial radius of 30 meters. Taking into account the 79 registered locations, the radius was reduced to different values - for example, 8 meters, 9.5 meters, 10 meters - and it was recorded how many locations would continue to be accurately represented in full. For a location to continue to be fully accounted for, it is

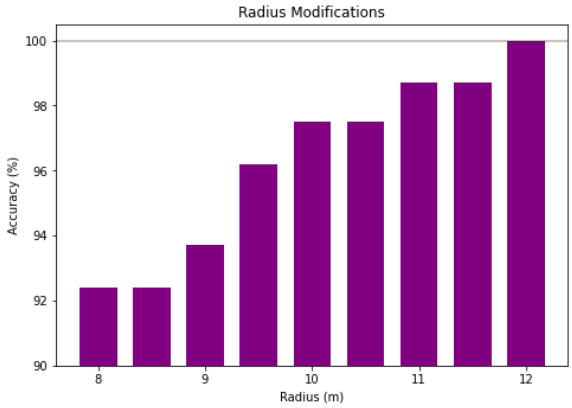

FIGURE 12. Final radius selection by accuracy.

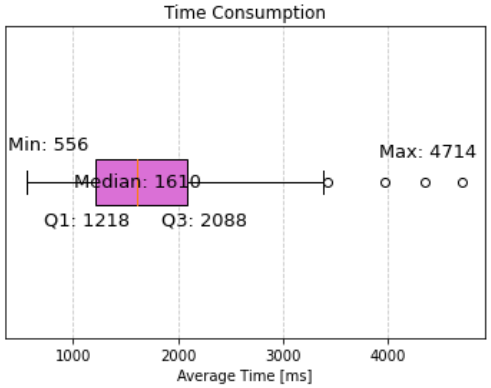

FIGURE 13. Time reduction with android results.

necessary that its average radius is smaller than the reduction made; if a safe location could no longer be fully represented due to this reduction, then the accuracy would decrease.

Figure 12 shows that if the radius were reduced to 8 meters —similar to the third quartile value in Figure 10— it would accurately represent 92.4% of the total experiment cases. Alternatively, considering a bit more margin with a 12-meter radius would accurately represent 100% of the total experiment cases. In other words, if the radius were reduced to 12 meters for this implementation, this modification would have *no negative impact* on the experimentation conducted to date, while *positively impacting user security* by better defining a secure space.

To analyze the execution time of the proposed solution, a box plot was chosen (Fig. 13). It can be observed that the time required, from the moment the user finishes entering their access credentials (the first security factor: username and password) and presses the login button until the user is inside the account, is at least 0.6 seconds in the best-case scenario, and no more than 4.7 seconds in the worst-case.

Similarly, the results showed that the average validation time for each user is 1.76 seconds, which represents a significant reduction in time compared to the research presented in the State of the Art.

It is important to note that, from the user's perspective, they are only entering their access credentials, making it seem like a Single-Factor Authentication (SFA) algorithm. However,

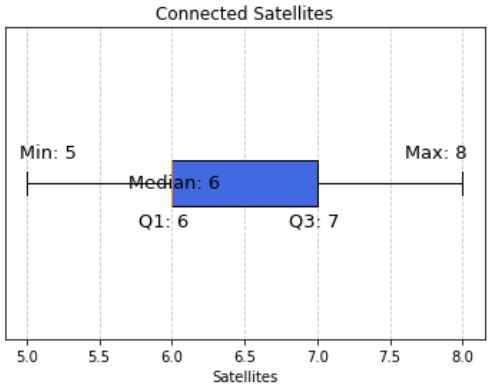

FIGURE 14. Connected satellites with arduino implementation.

due to the existence of the *location* subcategory factor that utilizes the user's location, the proposed solution is, in reality, a Multifactor Authentication (MFA).

B. ARDUINO IMPLEMENTATION

1) STORAGE REQUIREMENTS

The code distribution for Arduino necessitates flexibility and varied formats. First, the widely-used plain text *INO* format, which allows for easy sharing and code review. Second, the *ZIP* libraries, which provide reusable functionalities.

Two INO files were used (one for the primary Arduino and another for the secondary), along with four libraries in ZIP format. The code used for the primary microcontroller required 8.1KB, the code for the secondary Arduino 6.79KB, the *Keypad* library 71KB, the *LiquidCrystal_I2C* library 17.3KB, the *SD* library 85KB, and the *TinyGPS++* library 315KB. This results in a total of 503.19KB of information for the complete implementation.

2) TEST ENVIRONMENT

The experimentation process was carried out in a real test environment with one user, where he had the freedom to perform these experiments in the geographical spaces of his choice. As the only guideline, it was essential to follow the authentication process described in Section IV-B5.

For this experiment, there was **1 participant**, who registered **739 successful opening and closing processes** of the safe box, distributed among **4 different locations**.

3) RADIUS SELECTION

The results obtained with the Arduino experimentation phase were favorable. As mentioned above, the NEO-6M module used in this implementation uses a traditional GPS connection that establishes communication with the satellites orbiting the earth; the more satellites connected, the better the accuracy of the location coordinates. As can be seen from Figure 14, the module was able to connect with between 5 and 8 satellites, with an median of 6.

This implementation was tested in four different locations, and as can be seen from Figure 15, in all of them the average radius was between 13 and 17 meters, which means that

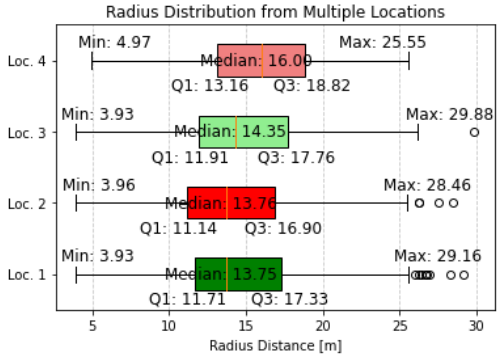

FIGURE 15. Radius descriptive statistics with arduino results.

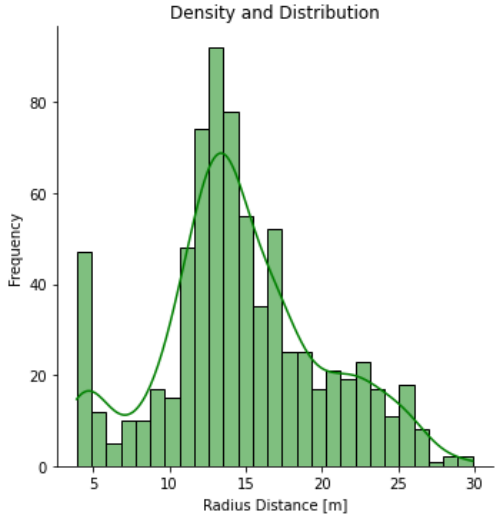

FIGURE 16. Radius density and distribution with arduino results.

the module used was able to make a constant connection with the satellites closest to it. It is worth noting that among the 739 successful opening and closing processes that were registered, an average radius of 14.7 meters was obtained. Given the difference that existed between this value and the initial radius of 30 meters, the possibility of reducing it was evaluated, as occurred in Section V-A3.

Given the distribution of the radius in each opening and closing process shown in Figure 16, it can be observed that a larger space was necessary as a limitation in the second factor of authentication. The distribution shows login attempts with radii ranging from 4 to 30 meters, so the idea of a reduction as significant as that of the first implementation was discarded; however, a small modification, at the expense of sacrificing certain attempts, is possible.

The calculations that were made to obtain the accuracy of this implementation considered the 4 registered locations. As in section V-A3, the radius was reduced to different values and it was counted how many locations continued to be represented correctly in full. Figure 17 shows that if the radius were reduced to 25 meters -being this a value similar to the maximum obtained from location 4 of Fig. 15-, it would be possible to represent 95.8% of the total cases of the experiment; while, if the initial value of 30 meters

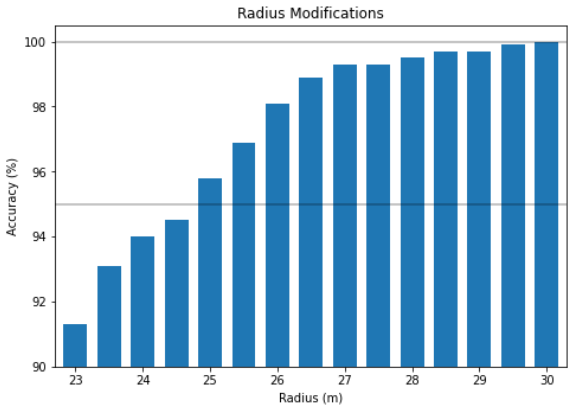

FIGURE 17. Final radius selection by accuracy.

of radius were taken into account, it would be possible to represent 100% of the total cases of the experiment. In other words, making a small reduction in radius for this implementation would have a negative impact -albeit slight- on the experimentation conducted to date.

C. SUMMARY

Given the aforementioned values, both implementations yielded positive results related to the research proposal of utilizing location as a transparent security factor. In comparison with the State of the Art, the solution proposed here stands out in various areas such as execution time, ease of use, memory savings, and user convenience. Now, if we consider that the average time a person takes to perform a single-factor authentication process (whether it be for user credentials, PINs, TOTPs, etc.) is between 1 and 4 seconds, the advantages of the proposed solution are highlighted. With this solution, users obtain the convenience and speed of a Single Factor Authentication (SFA), along with the security provided by Multi-Factor Authentication (MFA) when using a transparent second factor.

VI. CONCLUSION AND FUTURE WORK

This research work presented the design, implementation, and results of the location-based authentication algorithm developed for the Android and Arduino environments. The test results related to the implementation feasibility, execution time, and the selection of an optimal radius for each implementation were also included.

The proposed solution achieved favorable results for both physical prototypes. In the case of the Android mobile application, it is possible to achieve 100% accuracy for all secure locations with just a 12-meter radius in only 1.76 seconds. In the case of the security box implemented on an Arduino circuit, it is possible to achieve 100% accuracy in all opening and closing attempts with a 30-meter radius.

In comparison with the research presented in the State of the Art (Table 1), the proposed solution stands out due to various factors. For example, the experimentation was conducted in a minimally controlled real-world environment.

Additionally, precision levels (100% accuracy) were obtained that surpass any described in related works, highlighting the fact that this was achieved with a smaller radius and a significant decrease in the required time (1.76 seconds). Our solution is faster than all solutions presented in the State of the Art.

It is worth noting that the proposed solution lacks external hardware required for its operation —which commonly falls into the category of something you *have*-. From the user's perspective, they are only entering their access credentials while transparently utilizing a second security factor — from the *location* subcategory— to authenticate individuals. These two reasons contribute to user satisfaction by offering a simple and fast authentication process without external elements that could cause dissatisfaction.

Thanks to the results of this research, we can conclude that the ideal scenario for the use of this MFA system is the mobile banking application. Given the insecurity that many countries in Latin America face [7], the population is prone to suffer from short kidnappings, so they are forced to empty their bank accounts from their phone applications or physical ATM cards. We believe that our MFA system prevents these scenarios by denying access to the user's account if they are not located in a safe location. A conventional user performs their bank transactions from less than ten different locations (for example, their school, office, work, or home); our system allows the registration of up to ten safe spaces, with which all the locations that the user needs can be registered, and in this way the user could be protected by our second factor of fast and transparent authentication. It is important to note that, in the case that a user wants to make a transfer from a temporary location, they will only have to register that specific location as safe, so our solution provides flexibility to users.

Some limitations of the proposed research are related to the quantity and quality of the connection between GPS modules and satellites and/or reference stations, as well as the physical representation of secure locations, as these were implemented in the form of two-dimensional circles with an adaptable center and variable radius. Although the experimentation was carried out in a real-world environment, the connection between the GPS and the satellites could be compromised in cases where participants were located in physical places with multiple floors or poor signal reception (e.g., apartment buildings or office towers).

Future work in this research will assess the viability and effects of including altitude as a third parameter in secure locations, transforming the circles into spheres or cylinders with a defined height. By implementing these modifications, it will be possible to better address user authentication regardless of whether they are physically located in single-story locations or in office or residential towers.

A future improvement that will also be evaluated is the inclusion of a third authentication factor from the category of something you *have*, with the aim of strengthening the MFA algorithm. With this third factor, it is planned to link only one electronic device -by registering its Internal Mobile Equipment Identity (IMEI), its SIM cellular number, or both, in the database- per user account, as currently the user can access his account with any device as long as it is in the vicinity of a safe location. The objective of this new factor will be to combat the possibility of cyber-criminals using location spoofers, since in addition to needing to know the user's safe locations and their coordinates, they would also need to obtain the user's physical device. We are aware that adding a factor requiring external hardware may reduce user convenience, so we have decided to leave this factor out of the scope of this research and propose it as a possible future improvement.

REFERENCES

[1] *Commission Implementing Regulation European Union 2015/1502 of 8 September 2015 on Setting out Minimum Technical Specifications and Procedures for Assurance Levels for Electronic Identification Means Pursuant to Article 8(3) of Regulation (EU) No 910/2014 of the European Parliament and to the Council on Electronic Identification and Trust Services for Electronic Transactions in the Internal Market*, Eur. Commission Parliament, Brussels, Belgium, 2015.

[2] D. Dasgupta, A. Roy, and A. Nag, "Multi-factor authentication," in *Advances in User Authentication*. Cham, Switzerland: Springer, 2017, pp. 185–233, doi: 10.1007/978-3-319-58808-7_5.

[3] A. Titterington. (2023). *Qué Es La Autenticación De Múltiples Factores?* [Online]. Available: https://latam.kaspersky.com/blog/what-is-two-factor-authentication/26390/

[4] Verizon. (2022). *Verizon 2022 Data Breach Investigations Report*. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/

[5] T. Petsas, G. Tsirantonakis, E. Athanasopoulos, and S. Ioannidis, "Two-factor authentication: Is the world ready? Quantifying 2FA adoption," in *Proc. 8th Eur. Workshop Syst. Secur.* Bordeaux, France: Association for Computing Machinery, Apr. 2015, pp. 1–12, doi: 10.1145/2751323.2751327.

[6] A. Acar, W. Liu, R. Beyah, K. Akkaya, and A. S. Uluagac, "A privacy-preserving multifactor authentication system," *Secur. Privacy*, vol. 2, no. 5, pp. 1–5, Sep. 2019, doi: 10.1002/spy2.88.

[7] J. S. Kiernan. (2022). *Credit Card & Debit Card Fraud Statistics*. [Online]. Available: https://wallethub.com/edu/cc/credit-debit-card-fraud-statistics/25725

[8] H. Diazgranados. (2021). *Política De Parches Y Uso De Contraseñas Reducen Hasta Un 60% El Riesgo De Ciberataques a Empresas*. [Online]. Available: https://latam.kaspersky.com/blog/politicas-de-parches-y-uso-de-contrasenas-reducen-hasta-un-60-el-riesgo-de-ciberataques-a-empresas/22812/

[9] V. Shukla, A. Chaturvedi, and N. Srivastava, "A new one time password mechanism for client-server applications," *J. Discrete Math. Sci. Cryptogr.*, vol. 22, no. 8, pp. 1393–1406, Nov. 2019, doi: 10.1080/09720529.2019.1692447.

[10] M. Bartlomiejczyk, E. F. Imed, and M. Kurkowski, "Multifactor authentication protocol in a mobile environment," *IEEE Access*, vol. 7, pp. 157185–157199, 2019, doi: 10.1109/ACCESS.2019.2948922.

[11] R. Dedenok. (2022). *Suplantación Bancaria Y Robo De Identidad*. [Online]. Available: https://latam.kaspersky.com/blog/wells-fargo-phishing-identity-theft/24789/

[12] I. El Fray, M. Bartłomiejczyk, M. Kurkowski, S. Szymoniak, and O. Siedlecka-Lamch, "User authentication protocol based on the location factor for a mobile environment," *IEEE Access*, vol. 10, pp. 16439–16455, 2022, doi: 10.1109/ACCESS.2022.3148537.

[13] V. Khattri and D. K. Singh, "Implementation of an additional factor for secure authentication in online transactions," *J. Organizational Comput. Electron. Commerce*, vol. 29, no. 4, pp. 258–273, Oct. 2019, doi: 10.1080/10919392.2019.1633123.

[14] A. Titterington. (2023). *Tipos De Autenticación De Dos Factores: Pros Y Contras*. [Online]. Available: https://latam.kaspersky.com/blog/types-of-two-factor-authentication/26453/

[15] A. Alabdulatif, R. Samarasinghe, and N. N. Thilakarathne, "A novel robust geolocation-based multi-factor authentication method for securing ATM payment transactions," *Appl. Sci.*, vol. 13, no. 19, p. 10743, Sep. 2023, doi: 10.3390/app131910743.
[16] Kaspersky. (2023). *El Estado De Uso Y Seguridad De Pagos Digitales En América Latina*. [Online]. Available: https://latam.kaspersky.com/blog/seguridad-en-pagos-digitales-latam/26516/
[17] Android. (2023). *Navigation*. Android Developers. [Online]. Available: https://developer.android.com/guide/navigation
[18] Google. (2023). *Maps SDK for Android*. Google Maps Platform. [Online]. Available: https://developers.google.com/maps/documentation/android-sdk?hl=es-419
[19] Google. (2023). *Set Up Google Play Services*. Google Play services. [Online]. Available: https://developers.google.com/android/guides/setup
[20] Firebase. (2023). *Firebase Authentication*. [Online]. Available: https://firebase.google.com/docs/auth?authuser=0#multi-factor
[21] R. Purnomo, T. D. Putra, H. Kusmara, W. Priatna, and F. Mukharom, "Haversine formula to find the nearest PetShop," *Jurnal Teknik Informatika Sistem Informasi*, vol. 9, no. 3, pp. 2205–2221, Oct. 2022, doi: 10.35957/jatisi.v9i3.2434.
[22] Arduino. (2023). *Keypad*. Arduino Reference. [Online]. Available: https://www.arduino.cc/reference/en/libraries/keypad/
[23] Arduino. (2023). *LiquidCrystal I2C*. Arduino Reference. [Online]. Available: https://reference.arduino.cc/reference/en/libraries/liquidcrystal-i2c/
[24] Arduino. (2023). *Arduino Reference*. AD. [Online]. Available: https://www.arduino.cc/reference/en/libraries/sd/
[25] Arduino. (2023). *TinyGPSPlus*. Arduino Reference. [Online]. Available: https://www.arduino.cc/reference/en/libraries/tinygpsplus/

CARLOS JAVIER GARCÍA-TREVIÑO (Student Member, IEEE) was born in Monterrey, Mexico, in September 1998. He received the Mechatronics Engineering degree from Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey Campus, in December 2021. He is currently pursuing the degree in computer science with Tecnológico de Monterrey, Monterrey Campus. His research interests include software development, machine learning and deep learning models for predictive approaches, and convolutional neural networks. He received the Academic Honorific Award for being part of the 10% highest grade point average of the generation.

JESÚS ARTURO PÉREZ-DÍAZ (Member, IEEE) received the B.Sc. degree in computer science from the Autonomous University of Aguascalientes, in 1995, and the Ph.D. degree in new advances in computer science systems from Universidad de Oviedo, in 2000. He was a Full Associate Professor with University de Oviedo, from 2000 to 2002. Currently, he is a Researcher and a Professor with Tecnológico de Monterrey—Campus Guadalajara, Mexico, and a member of Mexican National Researchers Systems. His research interests include cyber-security in SDN and multifactor authentication, where he has supervised several master's and Ph.D. theses and published several articles in international journals. He was recognized by the COIMBRA Group as one of the Best Young Latin-American Researcher, in 2006, and received a research stay with Louvain Le Nouveau University, Belgium. He has been awarded by CIGRE and Intel for the development of innovative systems. He received the Best Student Award from the Autonomous University of Aguascalientes, for his B.Sc. degree.

CESAR VARGAS-ROSALES (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering and communications and signal processing from Louisiana State University. He is the coauthor of the book *Position Location Techniques and Applications* (Academic Press/Elsevier). His research interests include personal communications, 5G/6G, cognitive radio, MIMO systems, intrusion/anomaly detection in networks, localization, interference, network and channel coding, and optimum receiver design. He is a member of Mexican National Researchers System (SNI), Mexican Academy of Science (AMC), and the Academy of Engineering of Mexico. He is an Associate Editor of IEEE Access and *International Journal of Distributed Sensor Networks*. He is a Distinguished Lecturer of the IEEE Communications Society, from 2021 to 2022, the IEEE Communications Society Monterrey Chapter Chair, and the Faculty Advisor of the IEEE-HKN Lambda-Rho Chapter with Tecnologico de Monterrey. He was the Technical Program Chair of the IEEE Wireless Communications and Networking Conference (IEEE WCNC).

MAHDI ZAREEI (Senior Member, IEEE) received the M.Sc. degree in computer networks from the University of Science, Malaysia, in 2011, and the Ph.D. degree from the Malaysia-Japan International Institute of Technology, University of Technology, Malaysia, in 2016. In 2017, he joined the School of Engineering and Sciences, Tecnológico de Monterrey, as a Postdoctoral Fellow, where he was a Research Professor, in 2019. His research interests include wireless sensor and ad hoc networks, information security, and the applied machine learning and natural language processing. He is a Distinguished Researcher and a Professor. He is a Level I Member of Mexican National Researchers System and serves as an Associate Editor for IEEE Access, *PLOS One*, and *Ad Hoc and Sensor Wireless Networks* journals.