# Implementation and Evaluation of a Dynamic Multi-Grid Pattern Authentication for Secure Financial Applications

Amjad Qashlan[1]*, Qamar H. Naith[2], Fatimah Alakeel[3], Malak B Alharbi[4]
Department of Cybersecurity-College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia[1]
Department of Software Engineering-College of Computer Science and Engineering,
University of Jeddah, Jeddah 21959, Saudi Arabia[2, 4]
Department of Computer Science and Engineering-College of Applied Studies,
King Saud University, Riyadh, Saudi Arabia[3]

***Abstract*—Authentication plays an important role in online banking services. Nowadays, many online services still use password-based and One-Time Password (OTP) codes as authentication methods, but this is no longer secure, especially with the increase in financial fraud. In this research, a novel pattern-based password approach that relies on the random generation of multiple numeric grids is proposed. The approach is developed and deployed as a simple mobile application to be evaluated by banking users. Performance and security tests were used to evaluate the robustness of the approach. The findings of the assessment reveal that the performance and security level of our approach are high. The assessment also shows the approach to be capable of reducing and preventing financial fraud and protecting online banking systems from different types of cyberattacks through shoulder-surfing, eavesdropping, and brute force. The findings demonstrate the robustness of the pattern password, where fraudsters will be less likely to capture and recognize the correct password or discriminate between the patterns with their different values. To the best of our knowledge, this is the first approach that proposes a solution with the use of multi-grid rather than single-grid pattern passwords.**

***Keywords*—Authentication; cyberattack; financial fraud; multiple grids; pattern-based password; testing**

## I. Introduction

With the continuous improvement in protecting the confidentiality and privacy of sensitive information for bank customers, and with the continuous implementation of fraud detection and implementation of fraud detection and innovative mitigation solutions, cyberattacks still represent a critical threat to users [1]. Given the current problems regarding safety and the ease of credit card exploitation, enhancing current authentication methods and providing optimal solutions to avoid cyberattacks is a topic that requires more investigation [2]. Research work in this area is still in its early stages, as the major security features of the different types of financial fraud and cyberattacks were not taken into consideration by most of the current state-of-the-art authentication approaches. Furthermore, many existing authentication studies focus primarily on implementation and usability aspects while providing limited formal analysis of adversarial capabilities, attack surfaces, and theoretical resistance against pattern reconstruction and brute-force attacks. Consequently, there remains a need for authentication frameworks that combine practical deployment with rigorous security evaluation.

In the literature, several authentication solutions have been proposed to tackle the issues related to the secure authentication process. According to the analysis of these developed approaches, these solutions have been classified into four categories: approaches that rely on the use of password [3]–[6], approaches that use a One-Time Password (OTP) code [7]–[12], approaches based on the use of selected pattern and images [13]–[17], and approaches that depend on the use of some user's characteristics [18]–[25]. Most of these authentication solutions have attempted to reduce fraud and cyberattack issues. However, several challenges limit its full success, such as the passcode guessability and observability from the attacker's perspective. There are some other research studies that attempted to address the limitations through the use of multi-factor authentication (MFA). However, these solutions are still unable to cover the full breadth of users' accessibility due to the fact that it is time-consuming, and there is a lack of consideration of human factor standards preventing it from providing a better user experience. Accordingly, we still need a more effective and more secure solution to successfully overcome all these limitations.

To improve on the previously suggested and implemented solutions and provide more protection against cyberattacks (e.g., fraud), this study proposes a new authentication approach that mainly relies on a Multi-Grid Pattern-Based Password (MGPP) approach that relies on the random generation of numbers on different grid layouts. The proposed approach is shown as a mobile-based application that includes four grids sized 3x3 and labelled with randomly assigned digits between 1 and 9. The arrangement of these numbers is changed dynamically, and it is possible to repeat this within the single or overall grid. The most important and distinctive feature of this approach, as compared to that of others, lies in the way of entering the predefined pattern across multiple grids' cells without clicking on the cells. In addition to the proposed authentication mechanism, this work introduces a formal threat model to characterize potential adversarial behaviors and a formal entropy-based security analysis to quantify the complexity of recovering the secret authentication pattern. These additions provide a theoretical

*Corresponding author

foundation for evaluating the security properties of the proposed framework beyond experimental testing alone.

In order to assess the effectiveness of the MGPP authentication mechanism, two questions were addressed:

Q1: How efficient is the Multi-Grid Pattern-Based Password authentication approach?

Q2: To what extent is a Multi-Grid Pattern-Based Password authentication approach effective in minimizing financial fraud?

To investigate these two research questions, two experimental studies have been conducted; one for evaluating the performance aspects, and another for evaluating the security aspect. Performance analysis shows that the proposed approach has a high level of performance and the capability of accepting and handling more users while also maintaining their authentication connection under different conditions. Security analysis indicates that the MGPP approach is highly effective in minimizing financial fraud. Moreover, the formal entropy analysis demonstrates that the proposed framework substantially enlarges the authentication search space compared with conventional single-grid pattern-based authentication systems, thereby reducing the probability of successful brute-force and pattern-guessing attacks. It has been proved that the login security success rates will be higher than those of single- grid pattern-based password methods. It also revealed that the password patterns across users are more varied and difficult to have identical pattern-based passwords for different users. This research contributes toward filling the gap in the literature regarding authentication approaches and related security issues. To the best of the authors' knowledge, this research is the first to:

- Provide a novel Multi-Grid Pattern-Based Password (MGPP) authentication framework that utilizes multiple dynamically randomized grids for secure user authentication;
- Develop a novel authentication algorithm that increases numerical randomization and pattern-space diversity to improve resistance against observation-based attacks and password guessing attempts;
- Introduce a formal threat model that characterizes adversarial capabilities and attack surfaces relevant to pattern-based authentication systems;
- Provide a formal entropy-based security analysis that quantifies the authentication search space and evaluates theoretical resistance against brute-force attacks; and
- Experimentally validate the proposed framework through comprehensive performance and security evaluations in a mobile banking authentication environment.

The structure of this study is organized as follows: In Section II, the authors provide an overview of the current literature on different authentication approaches. Section III presents the proposed MGPP authentication framework, including the registration process, authentication mechanism, cryptographic protection model, and formal threat model. A clear description of the development of the MGPP authentication approach is presented in Section IV. Section V describes, in detail, the two experimental studies conducted to evaluate the MGPP approach. Afterwards, Section VI presents the performance evaluation, security evaluation, and formal entropy analysis together with a discussion of the main findings. The study is then concluded in Section VII.

## II. Related Work

In the literature, several authentication approaches and methods have been developed and implemented in different sectors including banking, healthcare and educational sectors. In addition, based on the literature, user authentication approaches can be categorized, as shown in Fig. 1 [26]–[28]. As illustrated in Fig. 1, existing authentication approaches typically involve a trade-off between security and convenience. While biometric-based authentication methods generally provide strong protection against unauthorized access, they often require specialized hardware and may introduce privacy concerns. Conversely, knowledge-based approaches such as passwords and graphical patterns offer greater accessibility and ease of deployment but remain vulnerable to observation, guessing, and credential theft attacks. Biometric authentication systems have strong properties to protect against misuse of special hardware; however, there are also privacy problems that arise when using such systems. Knowledge-based systems, such as password authentication and graphical patterns, are very easy to use and are therefore very often implemented. However, such systems have several flaws, for instance, observation, guessing, etc. We thus have to cope with the problems of enhancing the security of graphical authentication systems without degrading usability, and thus, new graphical authentication systems have to be implemented.

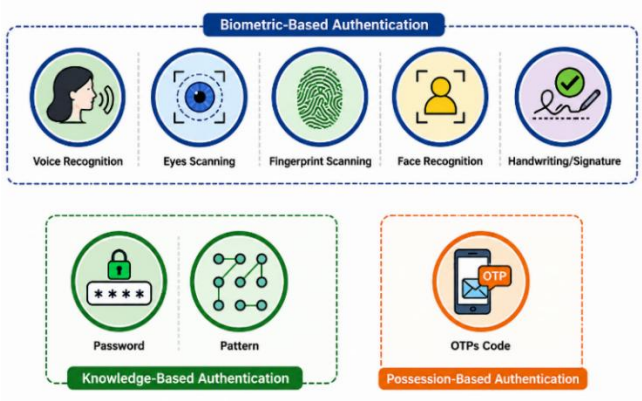


Fig. 1. Comparison of user authentication methods based on "security and convenience" parameters.

### A. Knowledge-Based Authentication

Knowledge-Based Authentication (KBA) is a password or pattern that identifies and memorizes end users. To date, many online services are still using "password-based authentication", which focuses on the use of username/password and cryptographic authentication [3]–[6]. The authors in [13] proposed a password-based authentication method that relies on the concept of password generators as a recommended system. This allows users to select and use one of the lists of presented passwords. To increase the robustness of used passwords and to remove the issue of re-changing passwords multiple times, the

authors in [13], [14] proposed a new approach that combined both pattern-based and password-based authentication. A new pattern-based password authentication method was developed by researchers in [15] that uses a Picture-Based Pattern (PBP) technique as a password to secure the transaction. Unlike [15], an enhancement for pattern-based password authentication has been proposed by researchers in [16] by analyzing the input patterns of mobile banking users based on three main aspects: the normal finger pressure levels; the time the user takes to input data; and the physical touch dimensions when using a touch screen. The authors in [17] proposed a grid-based (graphical-based) pattern password method as an alternative pattern-based password solution. This method relies on drawing the pattern from the displayed alphanumeric characters as a form of the password rather than selecting the image.

Despite these improvements, most existing pattern-based authentication approaches continue to rely on a single-grid structure and provide limited protection against repeated observation attacks, pattern reconstruction attempts, and systematic password guessing. Furthermore, the majority of these studies focus on implementation and usability aspects while providing little formal analysis of the theoretical complexity of recovering the underlying authentication pattern.

### *B. Possession-Based Authentication*

(PBA) relies on a One-Time Password (OTP), which is a randomly generated unique code sent via SMS, Email, or authenticator app to the user in each login attempt. The code expiry is time-based (usually between 15 and 60 seconds) for critical systems, and the user should enter the code to continue the login authentication process. The authors in [7] proposed an enhancement to the OTP algorithms by proposing an offline framework that can be implemented in smartphones to increase the security of online banking transactions. Very similar research studies have been conducted by researchers in [8], [9], [11], where they developed several OTP algorithms with the incorporation of various encryption and hashing techniques such as the Advanced Encryption Standard (AES) [29] and Blowfish [30] to encrypt or hash the verification code. Unlike previous works, the authors in [10], [12] proposed a new OTP algorithm with the incorporation of a QR code feature, where the user receives an OTP code in the form of a QR code that appears on an external device, and they must scan it using their mobile device.

### *C. Biometrics-Based Authentication*

Biometrics-Based Authentication (BBA) is a unique characteristic and attribute for each user that can be used as a form of personal identification (user identity). For example, for a fingerprint authentication process, the authors in [19] have proposed an authentication method to enhance the security of online banking, using an external device for fingerprint authentication. Unlike [19], the authors in [18] proposed a new approach based on the fingerprint authentication process through sensors on the user's mobile device rather than using an external device. The research conducted by [20] and [21] used fingerprints as an authentication factor for bank card systems for enhancing ATM services. The authors in [20] described an algorithm for enhancing the phases of fingerprint image extraction and matching. An alignment-based algorithm was used for matching two templates. While the researchers in [21] proposed a method that totally relies on the elimination of cards and the use of fingerprint scanning instead, as it can serve as both a password and access card. In regard to the "eyes authentication process", only a few research studies have been found. The authors in [24], [25] proposed an eyes-based authentication approach using eye- scanning integrated with iris-scan authentication. The concept of this approach is to analyze the user's eyes and their iris pattern, which is previously scanned and stored during the registration process and account creation. The pattern of eyes and iris is analyzed and converted into binary co-registration, which is then processed and stored in the bank database, which will be required prior to completing the online bank transaction process.

For the "face recognition authentication methods", the authors in [31] used face scanning as an authentication factor to enhance the security of ATM services, instead of the current static key (PIN) for ATMs. Limited research studies have been conducted regarding the voice authentication process [22], [23], [32]. Such methods focus on training the system to recognize an individual's unique voice characteristics and matching it with the data stored in the bank's database to determine the identity of the user. The authors in [22], [23] used voiceprint and other vital signs to assist people with visual impairments. In such an authentication process, the account will be validated by entering the user's voiceprint; this will then allow further transactions via voice recognition that implements Mel Frequency Cepstral Coefficient (MFCC) [33], Discrete Wavelet Transform (DWT) [34], and Vector Quantization (VQ) [35] methods to continue. Similar to [32], the authors designed a system that recognizes the tone of voice, and it is found that the developed voice recognition software has successfully activated the door opening mechanism using a voice command that ONLY works for the authenticated individual.

### *D. Multi-Factor Authentication*

(MFA), many research studies [31], [36]–[40] have provided multi-factor authentication solutions by incorporating the password with the OTP together, or the OTP and the password with a combination of more biometric factors such as fingerprints, facial recognition, retinal scan, IRIS recognition, or voice recognition. The two-factor biometric authentication method is proposed based on a combination of face ID and fingerprint [39], [40]. Unlike [36], [39], [40], the authors in [37] integrated the OTP with the fingerprints of users to authenticate online transactions. Slightly similar to [37], the researchers in [38] enhanced the security of the authentication process by proposing a three-factor authentication method, which includes passwords, familiar random pictures, and fingerprint scanning to make user authentication more secure (not only a combination of biometric factors). Although MFA approaches significantly enhance authentication security, they often increase system complexity, authentication time, and user interaction requirements. In mobile banking environments, excessive authentication steps may negatively affect usability and user acceptance. Consequently, there remains a need for lightweight authentication mechanisms capable of achieving stronger security guarantees without introducing additional authentication factors [41].

Existing graphical authentication methods such as password-based, pattern-based, biometric, possession-based methods, and multi-factor authentication methods have their own merits and have achieved a lot. Nevertheless, many open issues still need to be tackled [42]. The majority of graphical authentication methods use a single-grid-based approach; thus, they are susceptible to various types of observation-based attacks, i.e., various types of shoulder-surfing attacks and even authentication pattern reconstruction attacks where users use their authentication patterns as credentials. In addition, most existing graphical password authentication schemes have only tackled the issues for implementation and usability testing, without any formal adversary models and corresponding attack surfaces. There are only a handful of graphical password authentication schemes that can theoretically quantify the secret authentication patterns' resistance to brute-force attacks and guessing attacks by using entropy to measure the authentication passwords' complexity [43,44].

This study proposes a Multi-Grid Pattern-Based Password (MGPP) authentication, in which graphical authentication is supported by a Multiple of Graphical Passwords, each entry corresponding to a multiple number of dynamically generated grids for a single position in a pattern that is verified in a protected manner in order to verify a user's login credentials. After defining a formal threat model which details the possible ways in which an adversary can behave, we follow a formal entropy-based security analysis, i.e., a theoretical assessment of the resistance of the system for graphical password-based authentication to brute-force, as well as to guessing attacks. The results from the theoretical assessment are supported by performance, as well as by a security evaluation.

## III. Proposed MGPP Authentication Framework

The proposed Multi-Grid Pattern-Based Password (MGPP) authentication framework is designed to improve the security of graphical password systems against observation-based attacks while maintaining usability for online authentication services. Unlike conventional single-grid graphical authentication schemes, the proposed framework employs multiple interconnected grids containing dynamically randomized numeric values. The framework combines pattern-based authentication with randomized numerical representations to increase ambiguity during login sessions and reduce the likelihood of successful pattern reconstruction by adversaries.

Let the authentication environment consist of $G$ independent sub-grids, where each sub-grid contains $C$ cells. The total number of cells available in the authentication interface is therefore defined as [see Eq. (1)]:

$$N = G \times C \tag{1}$$

where, $N$ is the total number of cells in the authentication interface, $G$= number of sub-grids, $C$= number of cells per sub-grid. In the proposed implementation, four $3 \times 3$ grids are employed. Therefore, $G = 4$ and $C = 9$, which yields $N = 36$. The authentication process relies on a user-defined pattern consisting of a sequence of selected cells distributed across the multiple grids. Let the pattern length be represented by $L$, where $L = 8$, and let the pattern be expressed as [see Eq. (2)]:

$$P = \{p_1, p_2, \dots, p_L\} \tag{2}$$

where, $p_i$ denotes the position of the $i^{th}$ selected cell within the overall authentication space.

Each selected cell is represented by a coordinate pair [see Eq. (3)]:

$$p_i = (x_i, y_i) \tag{3}$$

where, $x_i$ denotes the column index, $y_i$ denotes the row index, $i \in \{1,2, \dots, L\}$. The complete authentication pattern can therefore be represented as [see Eq. (4)]:

$$P = \{(x_1, y_1), (x_2, y_2), \dots, (x_L, y_L)\} \tag{4}$$

which uniquely identifies the user's secret graphical password. To enhance resistance against shoulder-surfing and observation attacks, each authentication session generates a randomized numerical assignment for all cells. Let [see Eq. (5)]:

$$R_t = \{r_1, r_2, \dots, r_N\} \tag{5}$$

represent the set of numbers assigned to the cells during login session $t$, where $r_i$ denotes the numeric value displayed in cell $i$. The mapping between cell positions and displayed numbers changes dynamically for every authentication session according to [see Eq. (6)]:

$$R_t \neq R_{t+1} \tag{6}$$

which ensures that the same graphical pattern produces different visible numeric sequences across consecutive login attempts.

During authentication, the user does not directly redraw the graphical pattern. Instead, the user observes the randomized values displayed inside the selected pattern cells and enters the corresponding numeric sequence through a text input field. Let the observed login sequence be represented as [see Eq. (7)]:

$$S_t = \{s_1, s_2, \dots, s_L\} \tag{7}$$

where, $s_i$ denotes the numeric value currently displayed at position $p_i$ during session $t$.

In the server authentication phase, all numbers are transferred to the corresponding cell positions, and the generated pattern from step 3 is compared to the pattern already stored in the database which was generated in the registration phase. Visible information for authentication and secret pattern for storage and user's authentication information are completely separated but remain consistent.

The MGPP system has two stages of operation, namely the registration stage and the login authentication stage. During the registration stage a user generates a secret multi-grid pattern. The secret multi-grid pattern is then transformed into a secure form and stored in the system for future reference by the user. In the login authentication stage, a user is presented with a number of displayed values, i.e., numeric values that are assigned in a totally random fashion, for example. The user is able to recreate his secret multi-grid pattern with the aid of the displayed values. The processes of registration and login authentication are detailed in Section III-B and Section III-C, respectively.

### A. MGPP System Architecture

In this study, we present a new type of authentication framework called Multi-Grid Pattern-Based Password (MGPP) system, which utilizes client-server architecture to deliver high quality secure user authentication experience by thwarting the observation-based attacks on classic password-based authentication systems. The framework is composed of three main entities, i.e., user's device, authentication server, and credential repository. User's device is used to display multiple grids to end user, to capture users' input and to send login requests to authentication servers. In addition, authentication servers are utilized to generate multiple grids for end users, to verify the generated patterns by end users, to manage users' sessions and to encrypt/decrypt all communications with end users and their corresponding devices. Last but not least, credential repositories are used to store the protected pattern representations that were generated for end users during their registration phase.

The authentication interface is composed of multiple sub-grids which are all displayed to the user on one large multi-grid workspace. Let the set of sub-grids be represented as [see Eq. (8)]:

$$\mathcal{G} = \{g_1, g_2, \dots, g_G\} \tag{8}$$

where, $\mathcal{G}$ denotes the set of sub-grids and $G$ denotes the total number of grids. In the proposed implementation, $G = 4$, and each grid contains $C = 9$ cells organized in a $3 \times 3$ structure. Consequently, the complete authentication workspace contains $N = 36$ distinct cell positions.

Each cell is uniquely identified through its coordinate location and grid membership. The position of a cell is represented by $c_{i,j,k}$ where $i$ denotes the grid identifier, $j$ denotes the row index, and $k$ denotes the column index. The complete authentication space is therefore represented as [see Eq. (9)]:

$$\mathcal{C} = \{c_{i,j,k} \mid 1 \le i \le G, 1 \le j \le 3, 1 \le k \le 3\} \tag{9}$$

where, $\mathcal{C}$ denotes the set of all available cells within the MGPP environment.

To prevent the formation of static visual patterns, the system dynamically assigns numeric values to the cells at the beginning of each authentication session. Let the session-specific numeric assignment be represented by [see Eq. (10)]:

$$M_t: \mathcal{C} \to \mathcal{D} \tag{10}$$

where, $M_t$ denotes the session-specific mapping function, $\mathcal{C}$ is a set of cells, and $\mathcal{D}$ is a set of displayed digits by [see Eq. (11)]:

$$d = M_t(c_{i,j,k}) \tag{11}$$

where, $d$ represents the displayed numeric value. MGPP does not follow traditional graphical password schemes in which users are required to re-draw their password every time they login. In MGPP, the secret pattern structure is decoupled from the authentication information revealed to the user in each session. Users need only to remember the positions of the cells they chose for their secret pattern. The numbers revealed in these cells in each session are random and therefore different from session to session. Thus, the same secret pattern is always revealed as a different sequence of input for every login session.

The authentication workflow then continues on the server side where the session ID and a random grid is generated and sent to the client as a configuration for a grid which is then displayed within the application's GUI. On the client side, the user now selects all positions where he had registered a pattern before and enters the displayed digits in the correct order and submits the resulting sequence of digits to the authentication server. There the entered sequence of digits is changed back to its positional representation and is then compared with the protected pattern which had been stored during registration. Fig. 2 shows an overview of the MGPP's operational workflow for registration, authentication, for the encryption of the user's credentials and for the verification of stored credentials which are stored on the user's client.

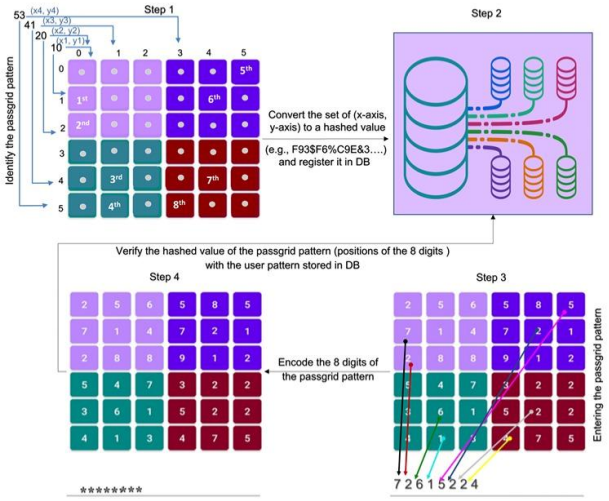


Fig. 2. The flow of the proposed approach working mechanism.

Let the authentication function be represented as $A(U, S_t)$, where $U$ denotes the registered user identifier and $S_t$ denotes the submitted authentication sequence. The authentication decision is expressed as [see Eq. (12)]:

$$A(U, S_t) = \begin{cases} 1, & \text{if the reconstructed pattern matches the stored pattern} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where, a value of 1 indicates successful authentication and a value of 0 indicates authentication failure.

The MGPP framework architecture is designed so that information visible to an attacker during a login session is not sufficient to determine the underlying secret pattern. MGPP enables multiple secret pattern grids to be chosen by users, session dependent numeric values to be randomly added to generated secret patterns and secret pattern verification to be enabled while keeping the login process familiar to legitimate users.

### B. Registration Phase

The registration phase for MGPP (Multi-Grid Pattern-Project) framework comprises of steps to generate a secret user authentication pattern in the form of a multi-grid structure

having individual grid spaces. The user selects some of the cells from these individual grid spaces to form a secret authentication pattern or the primary login credential for a user. In traditional pattern-lock-based user authentication systems, a connection between all other cells for all connected cells within a user's pattern is stored. However, in the MGPP framework, the stored data for each user is only protected representations of the cells chosen by that user to be included in his/her pattern. The actual chosen pattern of a user cannot be figured out from the input given by that user during the registration process. Let the user identifier be represented as [see Eq. (13)]:

$$U = \{ID, Username\} \tag{13}$$

where, $ID$ denotes the unique user identifier and $Username$ denotes the user's login name. After successful account creation, the server generates and displays the MGPP interface consisting of four independent $3 \times 3$grids. The user is then required to select an ordered pattern consisting of eight distinct points distributed across the available grids. Using the notation introduced in Section 3, the registered pattern $P$consists of an ordered sequence of $L = 8$ selected positions. Each selected point is represented by its coordinate location [see Eq. (14)]:

$$p_i = (x_i, y_i) \tag{14}$$

where, $x_i$ and $y_i$ denote the column and row coordinates, respectively. The complete positional representation of the user's pattern is therefore expressed as [see Eq. (15)]:

$$P = \{(x_1, y_1), (x_2, y_2), \dots, (x_8, y_8)\} \tag{15}$$

Our proposed framework for generating patterns for authentication purposes constrains these in order to increase security even more and to enlarge the number of possible patterns even further. First, every pattern for authentication has to consist of a fixed number of 8 points. Second, all points of a pattern have to be distributed on several grids, in contrast to only one grid. Third, it is not allowed to choose for a given authentication sequence identical positional selections for a pattern in all grids. So the additional constraints for creating a pattern strongly dissuade users from choosing a very weak authentication sequence.

After the pattern is selected, the server converts the positional sequence into a canonical pattern vector defined as [see Eq. (16)]:

$$V_P = [x_1, y_1, x_2, y_2, \dots, x_8, y_8] \tag{16}$$

which contains the ordered coordinates of all selected points. Since the sequence of selection is critical for authentication, any change in the order of the coordinates produces a different pattern representation.

To protect the registered pattern from disclosure, the positional vector is transformed into a cryptographic representation. Let the protected pattern credential be denoted by [see Eq. (17)]:

$$H_P = h(V_P) \tag{17}$$

where, $h(\cdot)$ represents the secure one-way hashing function employed by the system. The hash value $H_P$ (stored credentials) is stored in the authentication database for the respective user ID. The storage of the hashed credentials using one-way hashing (as for the registration) does not reveal any information about the used positions, even if the database is attacked.

Following the registration process, all subsequent authentication sessions are carried out between the logged-in user and the authentication server. The user's logged-in session uses the user's registered pattern for verification against the stored hash value for that user, stored in the authentication database against the user's identifier $H_P$ is. The authentication server will only be aware of the user's identifier and the protected pattern that was created from the registration authentication information (that was used to create the stored hash value $H_P$ for that user's identifier) stored in the authentication database.

### C. Login Authentication Phase

The login authentication phase verifies whether the submitted authentication sequence corresponds to the pattern established during registration. The proposed MGPP framework differs fundamentally from conventional graphical authentication systems because users do not redraw the graphical pattern during login. Instead, users identify the previously selected pattern positions and enter the numerical values currently displayed in those locations.

At the beginning of each authentication session $t$, the server generates a new random numerical assignment for all cells in the authentication interface. The generated mapping changes dynamically for every authentication session such that [see Eq. (18)]:

$$M_t \neq M_{t+1} \tag{18}$$

thereby ensuring that identical patterns produce different visible numerical sequences across successive login attempts.

Using the stored knowledge of the secret pattern positions, the user locates the corresponding cells within the randomized grids and observes the displayed values. The entered authentication sequence $S_t$, defined previously in Section III, contains the numerical values observed at the selected pattern positions during session $t$.

The submitted sequence is transmitted to the authentication server together with the user identifier. Based on the current session mapping $M_t$, the server reconstructs the associated positional sequence [see Eq. (19)]:

$$P_t = \{p_1, p_2, \dots, p_8\} \tag{19}$$

by identifying the cells corresponding to the submitted digits and their ordering. The reconstructed pattern is then transformed into its positional vector representation [see Eq. (20)]:

$$V_{P_t} = [x_1, y_1, x_2, y_2, \dots, x_8, y_8] \tag{20}$$

which is subsequently converted into a protected credential using the same hashing function employed during registration [see Eq. (21)]:

$$H_{P_t} = h(V_{P_t}) \tag{21}$$

Authentication is performed through comparison between the reconstructed hash value and the stored hash value. The authentication decision function is defined as [see Eq. (22)]:

$$A(U, S_t) = \begin{cases} 1, & H_{P_t} = H_P \\ 0, & H_{P_t} \neq H_P \end{cases} \quad (22)$$

where, $A(U, S_t) = 1$ indicates successful authentication and $A(U, S_t) = 0$ indicates authentication failure.

The proposed login mechanism is also observation-resistant. The numerical values that the user can see during his/her login session are changing from time to time while the underlying secret pattern does not change. Therefore, any observation by an attacker of a user's login process would result in only a temporary numerical sequence of values without any information about the user's secret pattern. In addition to this property, the repeated occurrences of the same digit in different cells as well as in different grids increase the ambiguity of such an observation, and therefore, it would be very difficult for the user to recognize and to reproduce the correct secret pattern during the login process, as is the case with simple graphical authentication mechanisms that use only one grid of cells. For example, Fig. 3 shows a login area where the digit "5" occurs in different cells of a grid. As can be seen, there are several possible ways to interpret the shown numerical values during such an observation by an attacker, and only by following one of these possible ways the attacker is able to try to recognize the secret pattern that is used for login by the user. However, the recognition of the correct pattern is very difficult because of ambiguity that is introduced by all the possible ways for the interpretation of the numerical values shown to the user during his/her login session. This is in contrast to simple graphical authentication mechanisms that use only one grid and therefore, the observation-resistance of our proposed login mechanism is increased significantly.

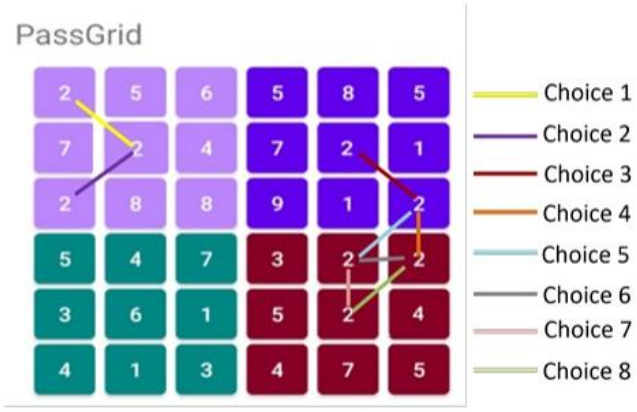


Fig. 3. An example of the several paths for the pattern that includes the number two twice, which can be very difficult for attackers to observe or guess.

The combination of dynamic numerical randomization, hidden positional verification, and cryptographic pattern validation enables the MGPP framework to authenticate legitimate users while simultaneously reducing susceptibility to shoulder surfing, brute-force guessing, and observation-based attacks.

### D. Cryptographic Protection Mechanism

The security of the proposed MGPP framework can be considered from two aspects: the secret pattern selected by a user and the authentication credentials stored in the framework for verification. To protect a selected pattern from disclosure, a one-way hashing function is used to transform the positional representation of a user's pattern into a stored credential on the authentication server. To prevent disclosure of stored credentials in case of a database attack, the original pattern coordinates are not stored on the authentication server. Instead, the positional representation of a pattern is first transformed into a stored credential by using the hashing function and then the resulting credential is stored on the server. Let $P$ denote the registered pattern defined previously in Section III. The positional information is first transformed into a canonical pattern vector [see Eq. (23)]:

$$V_P = [x_1, y_1, x_2, y_2, \dots, x_L, y_L] \quad (23)$$

which preserves both the selected locations and their ordering. Since the order of selection is an essential component of the authentication credential, any modification to the sequence results in a different vector representation.

To generate the protected credential, the pattern vector is processed using a secure one-way hash function. The resulting authentication credential is defined as [see Eq. (24)]:

$$H_P = h(V_P) \quad (24)$$

where, $H_P$ = protected pattern credential, $h(\cdot)$ = cryptographic hash function, $V_P$= pattern vector representation. The generated hash value is stored in the authentication database and associated with the corresponding user identifier. Since the hashing operation is non-invertible, recovering the original pattern from the stored credential is computationally infeasible.

During the authentication phase, the submitted sequence is converted into its corresponding positional representation and transformed into a verification vector [see Eq. (25)]:

$$V_{P_t} = [x'_1, y'_1, x'_2, y'_2, \dots, x'_L, y'_L] \quad (25)$$

where, $t$ denotes the current authentication session. The verification hash is then computed as [see Eq. (26)]:

$$H_{P_t} = h(V_{P_t}) \quad (26)$$

Authentication succeeds only when the newly generated credential matches the credential stored during registration. Therefore, the verification condition is expressed as [see Eq. (27)]:

$$H_{P_t} = H_P \quad (27)$$

Equality holds for selected positions, i.e., the positions that were selected and their order, equal to the positions and order that were registered for the first time.

In addition to protecting registered positions, credential information such as a user's password is also protected within the system. The framework uses Bcrypt hashing for passwords. Bcrypt is an adaptive one-way hash function which is designed to be slow (i.e., to be computationally expensive) and thus, for the purposes of a brute-force attack, is far more expensive than

typical hash functions. The cost c of Bcrypt hashing is a parameter of the hash function. The hashing operation may therefore be represented as [see Eq. (28)]:

$$H_P = h(V_P, c) \tag{28}$$

where, increasing $c$ proportionally increases the computational resources required to evaluate candidate patterns. Large-scale offline password guessing attacks become much more expensive for the attacker.

The protection of stored and verified positional credentials in MGPP is completely independent from the numerical login randomization. This means that even if an attacker managed to get hold of the authentication sequence that has been transmitted or even gets access to the database where all authentication credentials are stored, the corresponding password pattern or passphrase cannot be discovered because all stored and verified positional credentials are encrypted. This multi-layered protection also covers attacks like credential theft, database attacks, replay attacks, and attacks that try to reconstruct a used password pattern or passphrase from already verified authentication credentials. Fig. 4 illustrates registration and authentication with credentials of MGPP. The process that transforms the given pattern information into the stored or verified protected positional credentials in the registration as well as in the login process is shown in more detail. Note that the real password pattern or passphrase information is not stored in plain text.

### E. Threat Model

In this study, we perform a formal security analysis of our MGPP framework. We first present a detailed threat model that describes the attacks of an opponent with a given set of capabilities. In the subsequent security analysis and the related entropy analysis, we restrict ourselves to this threat model.

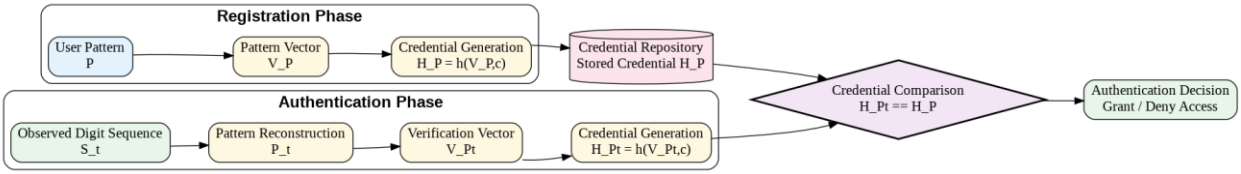


Fig. 4. Registration and authentication credential processing workflow in the MGPP framework.

This analysis will assume that the adversary has complete knowledge of the system and protocol under consideration, as well as the authentication function being used. Because of Kerckhoff's principle, the security of a cryptographic system resides in the authentication credentials for that system and not in the authentication algorithm(s) or protocol(s) for that system. Thus, the adversary has complete knowledge of the multi-grid (MGPP) framework under consideration as well as the pseudo-random location within each of the various grids used for pseudo-randomization of location, as well as complete knowledge of the login function in general.

Let the attacker be represented by the entity $\mathcal{A}$ whose objective is to recover the user's secret pattern or successfully impersonate a legitimate user. The attacker's capabilities are represented by the set [see Eq. (29)]:

$$\Gamma = \{\gamma_{obs}, \gamma_{rec}, \gamma_{net}, \gamma_{alg}\} \tag{29}$$

where, $\gamma_{obs}$ denotes the capability to visually observe the login interface, $\gamma_{rec}$ denotes the capability to record authentication sessions, $\gamma_{net}$ denotes the capability to intercept transmitted authentication messages, $\gamma_{alg}$ denotes knowledge of the authentication algorithm.

Under this threat model, the attacker may observe one or multiple authentication sessions and attempt to infer the underlying pattern from the visible numerical sequences entered by the user. However, the attacker is assumed to have no access to the user's memorized secret pattern and no direct access to the protected credentials stored on the server. Fig. 5 illustrates the threat model and attack surfaces considered in the security analysis of the proposed MGPP framework. The figure summarizes the adversarial capabilities, potential attack vectors, and the corresponding security objectives addressed by the framework. This model serves as the basis for the subsequent security evaluation and formal entropy analysis presented in Section VI-C.

The attacker's objective can therefore be expressed as maximizing the probability of successful authentication [see Eq. (30)]:

$$P_{succ} = Pr(\hat{P} = P) \tag{30}$$

where, $P$ denotes the legitimate user's pattern, and $\hat{P}$ denotes the attacker's estimated pattern.

The primary security goal of the MGPP framework is to minimize $P_{succ}$ by introducing uncertainty into the observation process.

The first attack scenario considered is shoulder surfing, in which the attacker directly observes the authentication process. Since the visible numerical values vary across authentication sessions according to the randomization function $M_t$, the observed authentication sequence $S_t$ does not directly reveal the underlying positional pattern $P$. Consequently, observation of a single session provides only temporary authentication information.

The second attack scenario involves multi-session observation. In this case, the attacker records multiple login attempts and attempts to correlate observed sequences. Let [see Eq. (31)]:

$$\mathcal{S} = \{S_1, S_2, \ldots, S_T\} \tag{31}$$

represent the set of observed authentication sessions, where $T$ denotes the number of recorded sessions. Because each session employs a distinct random mapping, the relationship between observed values and secret positions remains concealed, increasing the complexity of pattern reconstruction.

The third attack scenario considers brute-force guessing attacks. Let the total pattern space be represented by $| \Omega |$, where $\Omega$ denotes the set of all valid MGPP patterns. The probability of successfully guessing the correct pattern in a single attempt is therefore [see Eq. (32)]:

$$P_{guess} = \frac{1}{| \Omega |} \tag{32}$$

which decreases as the pattern space increases.

In the final attack scenario, an attacker performs eavesdropping and intercepts credentials. The attacker is able to capture all authentication messages that are transmitted between client and server. However, the protected cryptographic credentials used for verification in our approach, instead of the transmitted pattern information of the user, do not reveal any information.

These goals for the MGPP framework are used in Section VI-C for formal entropy analysis. They can also be used to evaluate the newly proposed framework for so-called observation-type attacks, as well as for typical brute-force attacks.

The MGPP framework proposed in this study is a unified system for object authentication. It selects a multi-grid pattern in a random numerical manner, and for a session, it protects the credentials by means of cryptography, and in an adversarial manner, it maps possible threats. In order to make authentication more robust against observation-based attacks and at the same time more practical for use, in the following section, the authors will describe the implementation of an MGPP-supported mobile system for object authentication (see Fig. 5).

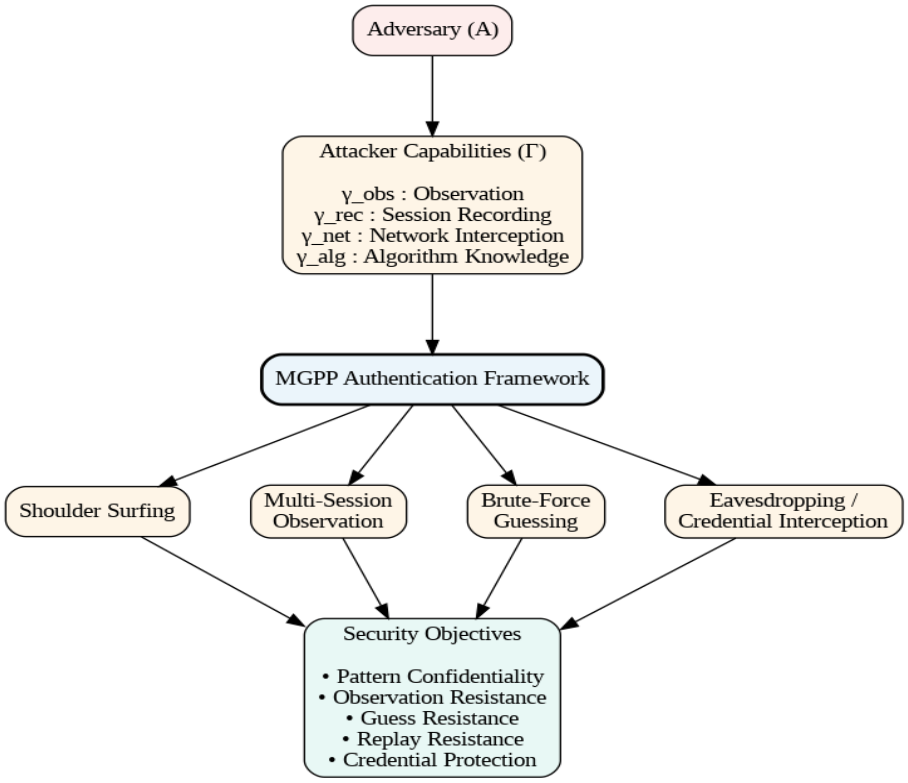


Fig. 5. Threat model and attack surfaces considered in the MGPP authentication framework.

All models were implemented in Python using TensorFlow and PyTorch libraries and executed on an NVIDIA GPU-based system. Reinforcement learning agents are DQN, PPO, and A2C.

## IV. System Development

### A. Mobile Application Architecture

In this work, a Multi-Grid Pattern-Based Password (MGPP) authentication framework is developed and implemented as a client-server mobile authentication system. The system consists of an Android client, an authentication server, and a database that holds the credentials in a protected form and other account information of users. The Android client is used for registration and login of users. The authentication server is used for managing the user's credentials, for generating the random grids for a session, for verifying an input authentication string against the stored credentials, and for database access.

MGPP architecture is multi-grid and portable. In the registration phase of the architecture, the secret pattern is spread on all of the grids given by the user. In the authentication phase, the mobile application of the user first gets a new random grid pattern for each session from the server and then shows it to the user. The user then enters the sequence of clicks as usual, and the user application sends the sequence to the server for verification. In this system, actual verification of credentials is done in a controlled environment of server. Thus, it is secure enough.

### B. User Interface

The graphical user interfaces for the MGPP framework are graphically illustrated in Fig. 6. The graphical user interfaces for the MGPP framework include interfaces for user registration, user login, and passgrid authentication. In order to keep the interfaces as simple as possible in order to support users with easy and comfortable authentication processes, we try to keep the high security standards of the MGPP framework.

During the registration process, users create their own secret authentication patterns. They select eight positions within the multiple grids provided, which are then changed to a protected format and saved on the server of the application. During login sessions, all grids for the session are provided to the user with all positions filled with dynamic numerical values. The user then selects the positions within the provided grids that correspond to the positions selected by the user during registration time. After correct position selection, the user enters the corresponding numerical values in the correct order.

The graphical user interface for the Multi-Grid Passgrid Pattern (MGPP) application is shown in Fig. 6. The interface does provide sufficient space to clearly show a Multi-Grid environment. However, to prevent attacks that rely on observation of the user interface as much as possible, the interface has been kept as simple as possible. The application makes use of multiple grids. The numerical values in the grids are changed on a dynamic basis. As a result of this, attacks that try to guess a user's pattern by observing the interface will be ambiguous and will most likely fail.

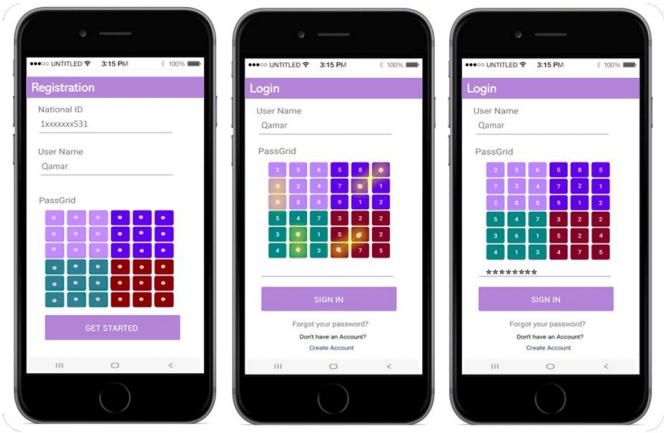


Fig. 6. Graphical user interface of the developed MGPP approach.

### C. Server Components

Server-side functionality of the authentication framework comprises authentication management, password protection, session generation as well as interaction with the database. The core of the server, which has been implemented in PHP, for every new login request, generates a random numerical representation for all grid cells, which then gets sent to the client program.

On the server side, when a login request is received, the server first reconstructs the positional pattern corresponding to the entered authentication sequence by referring to the active session mapping for the session received earlier. Then, the generated and protected credential representation of the reconstructed pattern is compared with the stored credential of the user's registration information. If both the credentials match, then the server grants authentication to that user.

As a second layer of protection, all servers have been configured to use the protection mechanism, as outlined in Section III-D. Thus, instead of storing the coordinates of the pattern entered by the user to form the login credentials, the protected form of the user's credential is stored. As previously described, should the worst happen and a database suffers a complete database compromise with subsequent unauthorized access by a third party, such an attack will not be able to retrieve and then use any previously submitted authentication login credentials. This system can also be used to help prevent the Credential Recovery Attack, where an attacker would attempt to retrieve previously submitted login credentials from a database in the hope that he can then use them to gain access to protected resources in subsequent login attempts. The integration of secure credential storage, dynamic session generation, and server-side authentication verification enables the MGPP framework to provide a secure and practical authentication solution for mobile environments.

## V. Evaluation of the MGPP Authentication System

The evaluation of the MGPP system has been carried out based on the assessment of two software quality aspects [45], [46]: performance and security. According to [47], it is a paramount requirement for cybersecurity testing to assess the technical implementation of the proposed authentication system and assess the human-system interaction. In addition to the experimental evaluation, the proposed framework is further analyzed through a formal threat model and entropy-based security analysis to provide theoretical evidence regarding its resistance to observation and guessing attacks.

Performance testing has been used to determine to what extent the system is powerful when applying different loads [48], [49]. More specifically, it has been implemented to assess how fast the aspects of the MGPP performance are under a specific workload, including load, response time, time complexity, stability, and scalability. A system-based evaluation method [50] was followed, where the experiment was achieved through a specific tool and collected the results immediately from the extracted log file from this tool. The performance evaluation was performed using the Junit 1 tool and under different conditions and working environments, such as using various network speeds and various mobile battery levels. In addition, the user-based testing method presented in [50]–[52] was used to ensure the compatibility of the applied method on different mobile devices with different specifications. During the evaluation, we applied some load on the app by increasing the number of users at a specific moment (1s) and over longer periods to check the performance in terms of: 1) desired load; 2) capability to scale up or scale out; 3) efficiency of the authentication process when a number of hits occur at a specific time; 4) the responsiveness of the authentication process in general. The average time complexity of number randomization was measured via a crowdtesting technique [52], where the average time complexity is measured during the experiment by different testers through 15 different Android mobile devices

with different specifications (device name, models, and Operating system versions) and different internet speeds.

Regarding security testing, different techniques have been performed to identify and minimize potential security risks in the proposed system and implement necessary countermeasures before releasing it to the public [47]. A static code analysis (SCA) [53] was executed using the "SonarLint"2 security testing tool by highlighting bugs and security vulnerabilities while developing the source code. This helped to identify potential issues and find remediation guidance on how to fix such issues to ensure the security posture and credibility of the proposed approach. In addition, vulnerability scanning/authentication scanning [53] methods were used to measure the capability of attackers to guess or observe the user's pattern and authenticate successfully. The security evaluation was conducted under the adversarial assumptions defined in the threat model presented in Section III-E, where attackers are assumed to possess knowledge of the authentication mechanism and may perform observation, recording, and guessing attacks. Furthermore, a document published by the Open Web Application Security Project (OWASP)3 has been adopted to ensure that the MGPP system is free of the top ten critical security risks. The experimental security assessment is complemented by a formal entropy analysis presented in Section VI-C, which quantitatively evaluates the theoretical password space and resistance of the proposed framework against partial-observation and brute-force attacks.

## VI. Results and Discussion

This section elaborates on the core findings derived from the applied methods (Section V) and explains the significance of those results compared to the existing state-of-the-art approaches to answer the proposed research questions. In addition to the experimental findings, the results are interpreted in light of the formal threat model presented in Section III-E and the entropy-based security analysis introduced in Section VI-C. This combined evaluation provides both empirical and theoretical evidence regarding the security and robustness of the proposed MGPP authentication framework.

### *A. Performance Analysis*

The results of performance testing on a variety of mobile device types and with different specifications show that the authentication process has a time complexity in the range from 1.5s to 2s for generating random numbers over the four grids. For the load testing, the results show that 40 users were dropped or disconnected when the load of the number of users reached 1000 at 1s, as shown in Fig. 7. This does not reveal a weakness in our approach; it is basically related to the network bandwidth capacity. This is consistent with the conclusion presented in the prior research [54], which states that as the load of user requests increases, the network becomes less powerful. This is the reason for dropping or disconnecting a few users in our results, where the MGPP approach does not reveal a weakness in up to 960 of 1000 users at 1s. The authors argue that the small number of dropped users and the slight increase in response speed prove the power and good reliability level of our approach in accepting multiple requests with low response time.


Fig. 7. Load testing results while increasing the number of users up to 1000 in 1s.

With respect to load and response speed, we tried to measure the response time with different values by increasing the number of users' authentication requests at 1s. The results from the test scenario clarify that the response speed of the completion of the authentication process and logging into the app requires less than 1s (97 ms) for each individual user. For the response speed of multiple authentication requests at 1s, it is clear from Fig. 8 that the average response speed of 750 multiple users' requests takes about 7.3 seconds, for 7500 users requests takes about 1.18 minutes, for 11250 users requests takes about 1.9 minutes, for 16500 users requests require about 2.5 minutes and for 32250 users requests take approximately up to 5 minutes. This indicates that the response time tends to increase very slowly and almost stabilizes with loads of 32250 and above. This small increase is considered a normal case for any software that uses network connections; this is because of the fact that when the load of requests is higher, more service requests are done [54], [55].

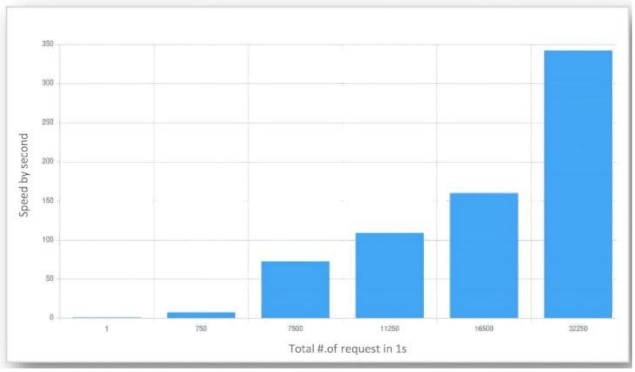


Fig. 8. The speed of the authentication process (response time) while the number of users' authentication requests is increased at 1s.

Fig. 9 presents the results of the stability and scalability evaluation of the proposed MGPP authentication process under increasing user workloads. The figure illustrates the behavior of the authentication system and database components as the number of concurrent users grows, providing insight into the ability of the framework to maintain stable performance under high-load conditions. For the scalability and stability of our approach, the findings showed a gradual increase in the overall period of time (1s to 53s) with a slight decrease in 7s with the load increase (reach 40.000 users). This decrease in the graph

from 15 seconds to 22 seconds does not indicate that a defect or error has occurred in our system regarding stability or scalability, but rather means that there were no requests, or some users were logged out of the system (in the scalability curve), as indicated by the results in the graph being synchronized with each other and no decrease was shown in one line rather than another. This is supported by the findings presented by Vahidnia et al. [56]. Consequently, this proves the capability of our authentication process and the database's ability to meet the growing needs through different conditions and maintains stability with the greatest number of users. From this, we can argue that our approach is highly stable and scalable regardless of the growth in user numbers. From this, we derived that the scalability of the system can impact its stability.

The performance testing of our approach demonstrated the high capability of the approach to complete the authentication process with a lower time complexity rate of random number generation (1.5s); a high-reliability level in handling up to 960 users at 1s, the ability to accept multiple requests with a low response time; the capability of maintaining stability and a high capacity to function more users, regardless of the growth in user numbers, we can answer RQ1 that the MGPP approach is efficient and can outperform a single-grid pattern password authentication approach. The performance results further indicate that the additional security mechanisms incorporated within the MGPP framework, including dynamic numerical randomization and protected credential verification, do not introduce significant computational overhead and therefore maintain the practical usability of the authentication process.

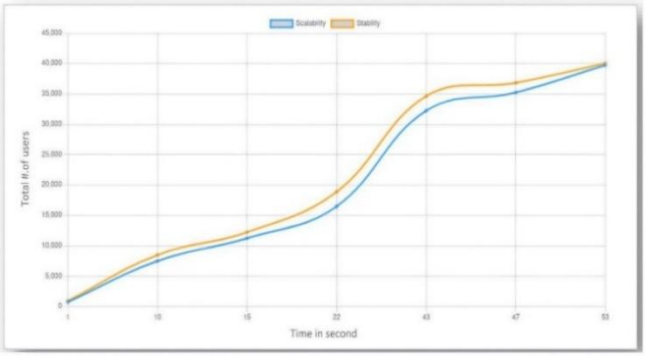


Fig. 9. The results of stability and scalability tests of the authentication process, given an increasing number of users.

## B. Security Analysis

The results of the security evaluation and analysis show that the MGPP authentication system can protect users against security issues that are relevant to different types of attacks such as information leaks, shoulder surfing, pattern passgrid theft, and man-in-the-middle attacks. The security assessment was conducted under the adversarial assumptions defined in the threat model presented in Section III-E, where attackers are assumed to possess observation, recording, interception, and algorithm knowledge capabilities. The types of security issues produced from the static code security testing through the SonarLint tool revealed that the proposed code is capable of securing the proposed authentication process from expo-sure to cyberattacks, as there were no critical security issues discovered (see Fig. 10). Most of these issues were software bugs and were successfully resolved during the implementation phase to enhance the performance and ensure that the MGGP system was even more secure. This is supported by the conclusions drawn by Jayalath et al. [57] and Dasanayake et al. [58], in which SonarLint is effective in detecting security vulnerabilities and source code quality issues.

*1) Shoulder Surfing (SS) attacks:* The security analysis demonstrated that the proposed approach prevents a shoulder surfing attack, where it would be very difficult for the attacker to observe the entry or attack the passgrid pattern [59]. This prevention is due to the effectiveness of the proposed MGPP system, which implements a number repetition algorithm that would distract the user. It is also because of the implementation of the hidden pattern-entering technique where the numbers within the cells that represent the pattern are entered securely without clicking on the cells. This certainly helped reduce the risk of exposure to the pattern during the authentication process. This result is consistent with the findings explained in [59]. From the perspective of the proposed threat model, the attacker's probability of successfully reconstructing the secret pattern from a single observed authentication session remains low because the observed numerical sequence does not directly reveal the underlying positional pattern.

*2) Eavesdropping attack:* The results of security testing and implementation of the cryptographic algorithm described in Section III-B show the strength of the MGPP system in protecting the authentication process from eavesdropping attacks such as a Man-in-The-Middle (MiTM) attack, which could lead to unauthorised access to confidential data. This is due to the method of securing the position of selected pattern cells, where the cells' positions are hashed into a 60-character hash value. Predicting or analyzing the hash function to obtain the hash value by attackers will be very difficult, if not nearly impossible. This observation is consistent with the threat model assumptions, where intercepted authentication messages do not provide direct access to the protected pattern representation stored within the system.

*3) Brute Force (BF) attacks:* With regard to brute force attacks (guessability), the findings stemming from the security testing show the power of the MGPP approach in reducing the patterns' guessability opportunities for attackers. This is because of the use of 8 dots for the selected pattern and duplication of the numbers on the four grids' cells, in addition to the random change of these numbers in each new process of communication. This interpretation is supported by the finding described in [60]. The authors compare the time complexity for brute-forcing different pattern-based passwords based on their length in a 3x3 grid. The greater the number of connected dots, the more time it takes to be guessed. For example, a 4-dot pattern would take 1,079 days, a 5-dot pattern would take 4,949 days, a 6-dot pattern would take 18,049 days, and so on, as shown in Fig. 11. Therefore, this shows that the patterns with 8 and 9 connected dots provide more security against password attacks than other patterns with lower numbers. This is for the case of using the connected dots with a 3x3 grid.

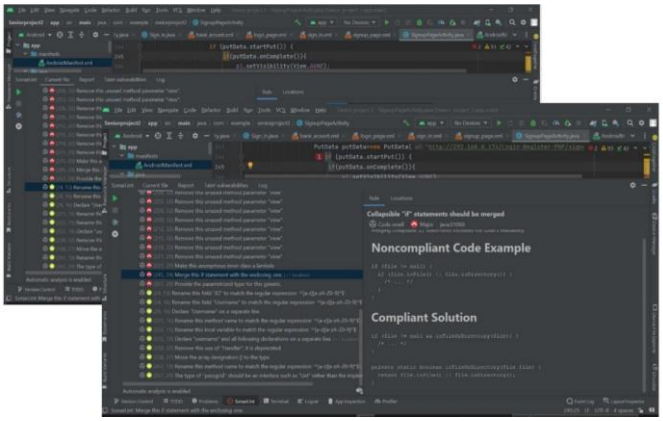


Fig. 10. A sample of the results obtained from static security testing via the SonarLint tool. No critical security vulnerabilities in the code have been found. All the issues discovered - ranging from major (red) to minor (green) levels - are related to coding mistakes, maintainability issues, and network configurations.

In the proposed approach, the authors considered using a discrete lines pattern with a length of 8-dots from four 3x3 grids, resulting in an overall 6x6 grid, which offers massive combinations of the possible generated patterns. This would result in an enormous increase in the number of possible combinations of the pattern generated and would most certainly decrease the chances of attackers guessing the user's pattern due to the increase in pattern combinations and the long time required to attempt to guess the passgrid pattern correctly. The same conclusion was reached by Potocký and Štulrajter [60], the more dots, the more time it takes for the attacker to guess the right pattern. While the experimental analysis demonstrates the practical resistance of the proposed approach against brute-force attacks, a more rigorous quantitative evaluation of the MGPP password space is presented in Section VI-C through formal entropy analysis. This analysis provides a theoretical estimation of the attacker's probability of successfully guessing the correct pattern under different observation scenarios.

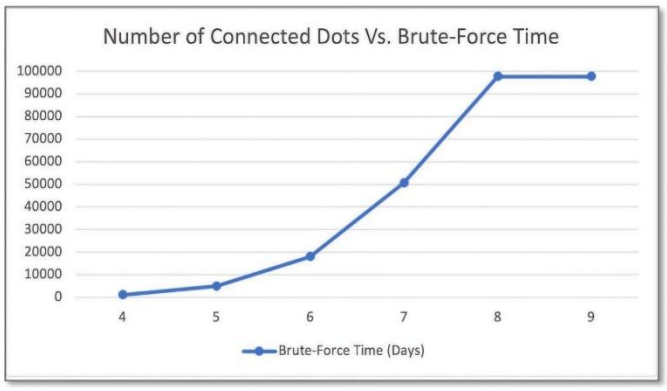


Fig. 11. Number of dots and the minimum time that might be needed by the attackers to guess the correct pattern.

### C. Formal Entropy Analysis

While the experimental security evaluation demonstrates the practical resistance of the proposed MGPP framework against observation-based and brute-force attacks, a theoretical assessment is necessary to quantify the complexity of recovering the secret authentication pattern [61]. The entropy of an authentication credential gives an indication of the uncertainty of this credential. In graphical authentication, the entropy of a credential is used to determine the strength of an authentication system against guess attacks. The higher the entropy of a credential, the larger the pattern space of this credential and hence the lower the probability that an adversary can reconstruct the used pattern successfully.

Let Ω denote the set of all valid authentication patterns that can be generated within the MGPP framework. The total number of possible patterns depends on the number of available cells and the length of the selected pattern. As introduced in Section III, the proposed framework utilizes four $3 \times 3$grids, resulting in a total authentication space of $N = 36$ available cell positions. Let $L$ denote the pattern length, where $L = 8$.

Assuming that each pattern consists of eight distinct selections and that the order of selection is significant, the total number of possible authentication patterns can be expressed as Eq. (33):

$$| \Omega | = P(N, L) = \frac{N!}{(N-L)!} \tag{33}$$

Substituting the values of $N$ and $L$ yields:

$$| \Omega | = \frac{36!}{28!} = 1{,}220{,}096{,}448{,}000 \tag{34}$$

possible authentication patterns.

The entropy of the authentication space is therefore defined as:

$$H(P) = \log_2 | \Omega | \tag{35}$$

which represents the minimum number of binary decisions required to uniquely identify a valid pattern. Substituting Eq. (34) into Eq. (35) gives [see Eq. (36)]:

$$H(P) = \log_2(1{,}220{,}096{,}448{,}000) \approx 40.15 \text{ bits} \tag{36}$$

This result indicates that the proposed MGPP framework provides substantially higher uncertainty than conventional graphical password systems operating on a single $3 \times 3$grid.

The probability of successfully guessing a valid pattern in a single attempt is given by Eq. (37):

$$P_{guess} = \frac{1}{| \Omega |} \tag{37}$$

which yields Eq. (38):

$$P_{guess} = 8.20 \times 10^{-13}. \tag{38}$$

Therefore, under random guessing conditions, the probability of successfully authenticating through a single brute-force attempt is negligible.

The threat model introduced in Section 3.5 further considers adversaries capable of observing a subset of the authentication process. Let $m$denote the number of pattern positions correctly inferred by an attacker through observation. The remaining search space is then given by Eq. (39):

$$| \Omega_{rem} | = P(N-m, L-m) = \frac{(N-m)!}{(N-L)!} \quad (39)$$

where, $0 \leq m < L$. Consequently, the residual entropy after partial observation becomes Eq. (40):

$$H_{rem} = \log_2 | \Omega_{rem} | \quad (40)$$

The search space is still very large even after some of the positions in the pattern have been disclosed to the observer. This is also a feature of the MGPP framework where the dynamic numerical randomization in the positions is separate from the positional pattern. Thus, by viewing the numerical pattern that has been disclosed to the observer, no information is disclosed as to the corresponding positions of the cells. Therefore, even when part of the search space has been disclosed by revealing some of the positions in the pattern, the size of the search space that can still be explored by testing all possible combinations of numbers for the revealed positions is very large. Table I shows the password space sizes for MGPP compared with other systems.

TABLE I. COMPARISON OF THE AUTHENTICATION SEARCH SPACE AND ENTROPY ACROSS COMMON AUTHENTICATION SCHEMES.

| Authentication Scheme | Password Space | Approximate Entropy (bits) |
|---|---|---|
| 4-Digit PIN | $10^4$ | 13.29 |
| 6-Digit PIN | $10^6$ | 19.93 |
| Android Pattern Lock (3×3) | 389,112 | 18.57 |
| Single 3×3 Grid (8 positions) | 362,880 | 18.47 |
| **Proposed MGPP (4×3×3 Grids, 8 positions)** | **1,220,096,448,000** | **40.15** |

The MGPP framework significantly increases the search space of needed authentication patterns when compared to simple pattern-based authentication schemes. Due to the existence of more than one grid, the total number of positions in all cells far exceeds that of a single simple grid. Although all cells are sequentially filled in a certain order to generate a specific pattern, an attacker cannot try all the possible patterns one by one to discover the secret pattern. In fact, the total search space of needed authentication patterns that are generated by the MGPP framework is so large that a brute force approach is entirely impractical. Hence, the MGPP framework provides an efficient and secure user authentication scheme.

Theorem 1. The MGPP framework provides a larger authentication search space than a conventional single-grid pattern authentication system of equal pattern length.

Proof. A conventional $3 \times 3$ pattern authentication system contains only nine selectable positions. For a pattern length of eight, the number of possible ordered patterns is Eq. (41):

$$P(9,8) = \frac{9!}{1!} = 362{,}880. \quad (41)$$

In contrast, the MGPP framework contains thirty-six selectable positions, yielding Eq. (42):

$$P(36,8) = 1{,}220{,}096{,}448{,}000. \quad (42)$$

Since [see Eq. (43)]:

$$P(36,8) \gg P(9,8) \quad (43)$$

The authentication search space of the MGPP framework is significantly larger than that of the conventional single-grid approach. Therefore, the proposed framework offers substantially stronger resistance against brute-force guessing attacks.

The theoretical findings are consistent with the experimental security results presented in Section VI-B. Both analyses indicate that increasing the number of selectable positions and introducing dynamic numerical randomization significantly increases the difficulty of pattern reconstruction. Consequently, the proposed MGPP framework achieves enhanced resistance against brute-force attacks, observation-based attacks, and pattern-guessing attempts while maintaining practical usability for mobile authentication applications.

In answer to the RQ2, the assessment of the approach proved the strength of the MGPP approach against brute-force attacks, eavesdropping attacks, and shoulder surfing attacks. Furthermore, the authors can confidently say that the proposed MGPP authentication approach is successful in minimizing financial fraud and reducing the risk of exposure to known software threats, as well as reducing exploitable vulnerabilities. Thus, this would increase security and protect credentials in online banking transactions. Based on experimental findings, the formal threat model, and the entropy-based security analysis, it can be concluded that pattern length, pattern structure, and pattern-space complexity are the primary factors influencing resistance against brute-force and observation-based attacks.

## VII. CONCLUSION

In conclusion, this research proposed a new approach for reducing financial fraud issues and cyberattacks and researched the reasons behind their existence to date. On the basis of this, a novel pattern-based password authentication approach has been developed that focuses on increasing the number of randomizations using four 3x3 grids. The main motivation behind this is to propose a more secure authentication algorithm to resolve the previously mentioned issues, as they remain a major concern to online banking users. The proposed approach (MGPP) has been developed as an Android mobile app, and it has been evaluated through performance testing and security testing. In addition, a formal threat model was developed to characterize the capabilities of potential adversaries and to evaluate the security objectives of the proposed authentication framework under observation, recording, interception, and guessing attack scenarios. The results from performance testing show the high level of performance achieved when adopting the MGPP approach, as it offers a lower time complexity rate through random number generation (1.5s to 2s). Results also show that the MGPP approach is highly capable of keeping users safe whilst the functionality is not negatively affected with the growth in user numbers. The assessment also demonstrated the capability of the MGPP approach to protect its users from threats, financial fraud, and cyber-attacks. Furthermore, a formal entropy-based security analysis demonstrated that the proposed MGPP framework significantly increases the authentication search space compared with conventional pattern-based

authentication systems, thereby reducing the probability of successful brute-force and pattern-guessing attacks. This experimental evaluation, formal threat modeling, and entropy analysis show that MGPP-based designs for secure and scalable authentication for mobile applications are feasible. For future work, this study will carry out large-scale usability studies and implement MGPP-based designs for online banking environments, as well as more thoroughly and in greater detail test security-related aspects of this design by means of, for example, penetration testing, ethical hacking, and a detailed risk analysis. Furthermore, this study investigates the use of more adaptive grid structures, dynamic grid morphing, and the incorporation of multi-factor authentication into the MGPP framework in order to increase the security and robustness of our design.

## ACKNOWLEDGMENT

The authors gratefully acknowledge the contributions of Lara S. Alghamdi, Sarah N. Alansari, Hams M. Alzahrani, Afaf M. Alogaili, and Reema A. Alebbie who participated, under the supervision of the authors, in the design, development, and testing of selected project components. Their dedication, technical efforts, and commitment provided valuable support to the successful completion of this work.

## REFERENCES

[1] J. T. Quah and M. Sriganesh, ‘‘Real-time credit card fraud detection using computational intelligence,’’ Expert systems with applications, vol. 35, no. 4, pp. 1721–1732, 2008. [Online]. Available: https://doi.org/10.1016/j.eswa.2007.08.093

[2] A. E. Omolara, A. Alabdulatif, O. I. Abiodun, M. Alawida, A. Alabdulatif, H. Arshad et al., ‘‘The internet of things security: A survey encompassing unexplored areas and new insights,’’ Computers & Security, vol. 112, p. 102494, 2022. [Online]. Available: https://doi.org/10.1016/j.cose.2021.102494

[3] S. Ruoti, J. Andersen, and K. E. Seamons, ‘‘Strengthening password-based authentication,’’ in Way@ Soups, 2016.

[4] S. K. Sood, A. K. Sarje, and K. Singh, ‘‘Cryptanalysis of password authentication schemes: Current status and key issues,’’ in 2009 Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS). IEEE, 2009, pp. 1–7. [Online]. Available: https: //doi.org/10.1109/ICM2CS.2009.5397977

[5] N. Sharma and B. Bohra, ‘‘Enhancing online banking authentication using hybrid cryptographic method,’’ in 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT). IEEE, 2017, pp. 1–8. [Online]. Available: https://doi.org/10.1109/CIACT. 2017.7977275

[6] V. K. Sharma, P. Mathur, and D. K. Srivastava, ‘‘Secure electronic fund transfer model based on two level authentication,’’ in 2018 second international conference on electronics, communication and aerospace technology (ICECA). IEEE, 2018, pp. 1338–1342. [Online]. Available: https://doi.org/10.1109/ICECA.2018.8474883

[7] S. Zhao and W. Hu, ‘‘Improvement on otp authentication and a possession-based authentication framework,’’ International Journal of Multimedia Intelligence and Security, vol. 3, no. 2, pp. 187–203, 2018. [Online]. Available: https://doi.org/10.1504/IJMIS.2018.096406

[8] D. E. Kurniawan, M. Iqbal, J. Friadi, F. Hidayat, and R. D. Permatasari, ‘‘Login security using one time password (otp) application with encryption algorithm performance,’’ Journal of Physics: Conference Series, vol. 1783, no. 1, 2021. [Online]. Available: https://doi/org/10.1088/1742-6596/1783/ 1/012041

[9] Y. S. Mezaal, D. A. Hammood, and M. H. Ali, ‘‘Otp encryption enhancement based on logical operations,’’ in 2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC). IEEE, 2016, pp. 109–112. [Online]. Available: https: //doi.org/10.1109/ICDIPC.2016.7470801

[10] J. Murkute, H. Nagpure, H. Kute, N. Mohadikar, and C. Devade, ‘‘Online banking authentication system using qr-code and mobile OTP,’’ International Journal of Engineering Research and Applications, vol. 3, no. 2, pp. 1810–1815, 2013.

[11] A. R. L. Reyes, E. D. Festijo, and R. P. Medina, ‘‘Securing one time password (OTP) for multi-factor out-of-band authentication through a 128-bit blowfish algorithm,’’ International Journal of Communication Networks and Information Security, vol. 10, no. 1, pp. 242–247, 2018.

[12] Y. S. Lee, N. H. Kim, H. Lim, H. Jo, and H. J. Lee, ‘‘Online banking authentication system using mobile-otp with qr-code,’’ in 5th International Conference on Computer Sciences and Convergence Information Technology. IEEE, 2010, pp. 644–648. [Online]. Available: https://doi.org/10.1109/ICCIT.2010.5711134

[13] F. A. A. A. Maqbali, ‘‘Strengthening password-based authentication,’’ Ph.D. dissertation, University of London Royal Holloway, 2019.

[14] S. W. Jirjees, A. M. Mahmood, and A. R. Nasser, ‘‘Passnumbers: An approach of graphical password authentication based on grid selection,’’ Journal homepage: http://iieta. org/journals/ijsse, vol. 12, no. 1, pp. 21–29, 2022. [Online]. Available: https://doi.org/10.18280/ijsse.120103

[15] R. Vadivukkarasi and K. Kuppusamy, ‘‘Enhanced authentication method using keyboard pattern based password generation,’’ International Journal for Modern Trends in Science and Technology, vol. 3, no. 8, 2017.

[16] S. Dedeepya, P. Swetha, and Y. RAJU, ‘‘Securing mobile banking in intelligent mobile devices-pattern based authentication approach,’’ International Journal of Scientific Research in Computer Science (IJSRCS), vol. 1, no. 3, 2013.

[17] S. Kiljan, H. Vranken, and M. van Eekelen, ‘‘Evaluation of transaction authentication methods for online banking,’’ Future Generation Computer Systems, vol. 80, pp. 430–447, 2018. [Online]. Available: https://doi.org/ 10.1016/j.future.2016.05.024

[18] L. Sharma and M. Mathuria, ‘‘Mobile banking transaction using fingerprint authentication,’’ in 2018 2nd International Conference on Inventive Systems and Control (ICISC). IEEE, 2018, pp. 1300–1305. [Online]. Available: https://doi.org/10.1109/ICISC.2018.8399016

[19] C. Lupu, V.-G. Găitan, and V. Lupu, ‘‘Fingerprints used for security enhancement of online banking authentication process,’’ in 2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). IEEE, 2015, pp. 217–220. [Online]. Available: https://doi.org/10.1109/ECAI.2015.7301177

[20] A. Muley and V. Kute, ‘‘Prospective solution to bank card system using fingerprint,’’ in 2018 2nd International Conference on Inventive Systems and Control (ICISC). IEEE, 2018, pp. 898–902. [Online]. Available: https://doi.org/10.1109/ICISC.2018.8398930

[21] B. A. Sahar, A. F. Rahardian, E. Muchtar et al., ‘‘Fingershield atm–atm security system using fingerprint authentication,’’ in 2018 International Symposium on Electronics and Smart Devices (ISESD). IEEE, 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ISESD.2018.8605473

[22] M. H. Ali and Q. M. Al Azze, ‘‘Design and implementation a security system for bank using voice recognition,’’ International Journal of Power Electronics and Drive Systems, vol. 10, no. 4, p. 2126, 2019. [Online]. Available: https://doi.org/10.11591/ijpeds.v10.i4.pp2126-2129

[23] E. D. Dimaunahan, A. H. Ballado, F. R. G. Cruz, and J. C. D. Cruz, ‘‘Mfcc and vq voice recognition based atm security for the visually disabled,’’ in 2017IEEE 9th international conference on humanoid, nanotechnology, information technology, communication and control, environment and management (HNICEM). IEEE, 2017, pp. 1–5. [Online]. Available: https://doi.org/10.1109/HNICEM.2017.8269516

[24] A. E. Omolara, A. Jantan, O. I. Abiodun, H. Arshad, and N. A. Mohamed, ‘‘Fingereye: improvising security and optimizing atm transaction time based on iris-scan authentication,’’ International Journal of Electrical & Computer Engineering (2088-8708), vol. 9, no. 3, 2019. [Online]. Available: https://doi.org/10.11591/ijece.v9i3.pp1879-1886

[25] R. C. Agidi, ‘‘Biometrics: the future of banking and financial service industry in nigeria,’’ International Journal of Electronics and Information Engineering, vol. 9, no. 2, pp. 91–105, 2018. [Online]. Available: https://doi.org/10.6636/IJEIE.2018129(2).04)

[26] Z. Erlich and M. Zviran, ''Authentication practices from passwords to biometrics,'' *Encyclopedia of Information Science and Technology, Third Edition*, pp. 4248–4257, 2015. [Online]. Available: https://doi.org/10.4018/978-1-4666-5888-2.ch417

[27] N. A. Karim and Z. Shukur, ''Using preferences as user identification in the online examination,'' dynamics, vol. 19, no. 20, p. 21, 2016. [Online]. Available: https://doi.org/10.18517/ijaseit.6.6.1412

[28] M. Awang, M. Mohamed, R. Mohamed, A. Ahmad, and N. Rawi, ''A pattern-based password authentication scheme for minimizing shoulder surfing attack,'' International Journal on Advanced Science, Engineering and Information Technology, vol. 7, no. 3, pp. 1049–1055, 2017. [Online]. Available: https://doi.org/10.18517/ijaseit.7.3.1517

[29] C. Paar and J. Pelzl, The Advanced Encryption Standard (AES). Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 87–121. [Online]. Available: https://doi.org/10.1007/978-3-642-04101-3_4

[30] T. Nie and T. Zhang, ''A study of des and blowfish encryption algorithm,'' in Tencon 2009-2009 IEEE Region 10 Conference. IEEE, 2009, pp. 1–4. [Online]. Available: https://doi.org/10.1109/TENCON.2009.5396115

[31] D. Soundari, R. Aravindh, S. Abishek et al., ''Enhanced security feature of atm's through facial recognition,'' in 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2021, pp. 1252–1256. [Online]. Available: https://doi.org/10.1109/ICICCS51141.2021.9432327

[32] R. A. Rashid, N. H. Mahalin, M. A. Sarijari, and A. A. A. Aziz, ''Security system using biometric technology: Design and implementation of voice recognition system (vrs),'' in *2008 international conference on computer and communication engineering*. IEEE, 2008, pp. 898–902. [Online]. Available: https://doi.org/10.1109/ICCCE.2008.4580735

[33] P. M. Chauhan and N. P. Desai, ''Mel frequency cepstral coefficients (mfcc) based speaker identification in noisy environment using wiener filter,'' in *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*. IEEE, 2014, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICGCCEE.2014.6921394

[34] G. Tzanetakis, G. Essl, and P. Cook, ''Audio analysis using the discrete wavelet transform,'' in Proc. conf. in acoustics and music theory applications, vol. 66. Citeseer, 2001.

[35] Z.-b. Wu and J.-q. Yu, ''Vector quantization: a review,'' Frontiers of Information Technology & Electronic Engineering, vol. 20, no. 4, pp. 507–524, 2019. [Online]. Available: https://doi.org/10.1631/FITEE.1700833

[36] S. Venkatraman and I. Delpachitra, ''Biometrics in banking security: a case study,'' Information Management & Computer Security, vol. 16, no. 4, pp. 415–430, 2008. [Online]. Available: https://doi.org/10.1108/09685220810908813

[37] B. A. Scaria and R. K. Megalingam, ''Enhanced e-commerce application security using three-factor authentication,'' in 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2018, pp. 1588–1591. [Online]. Available: https://doi.org/10.1109/ICCONS.2018.8662831

[38] A. T. Kiyani, A. Lasebae, K. Ali, and M. Ur-Rehman, ''Secure online banking with biometrics,'' in 2019 International Conference on Advances in the Emerging Computing Technologies (AECT). IEEE, 2020, pp. 1–6. [Online]. Available: https://doi.org/10.1109/AECT47998.2020.9194214

[39] I. A. Abdullah and J. J. Stephan, ''A survey of face recognition systems,'' Ibn AL-Haitham Journal For Pure and Applied Sciences, vol. 34, no. 2, pp. 144–160, 2021. [Online]. Available: https://doi.org/10.30526/34.2.2620

[40] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, ''Face recognition systems: A survey,'' Sensors, vol. 20, no. 2, p. 342, 2020. [Online]. Available: https://doi.org/10.3390/s20020342

[41] A. Nema and A. Ranjan, ''Pattern point based graphical authentication,'' International Journal of Engineering and Techniques, vol. 4, no. 2, 2018. [Online]. Available: https://doi.org/10.29126/23951303

[42] P. Sriramya and R. A. Karthika, ''Providing password security by salted password hashing using Bcrypt algorithm,'' ARPN Journal of Engineering and Applied Sciences, vol. 10, no. 13, pp. 5551–5556, 2015.

[43] ''Php: password_hash - manual.'' [Online]. Available: https://www.php.net/manual/en/function.password-hash.php

[44] N. Katrandzhiev, D. Hristozov, and B. Milenkov, ''A comparison of password protection methods for web-based platforms implemented with php and mysql,'' International Journal on Information Technologies & Security, №, vol. 11, no. 2, pp. 97–106, 2019.

[45] P. A. Laplante and M. Kassab, What every engineer should know about software engineering. CRC Press, 2022.

[46] D. Tobin, A. Bogomolov, and M. Golosovskiy, ''Model of organization of software testing for cyber-physical systems,'' in Cyber-Physical Systems: Modelling and Industrial Application. Springer, 2022, pp. 51–60. [Online]. Available: https://doi.org/10.1007/978-3-030-95120-7_5

[47] L. Maghrabi, E. Pfluegel, L. Al-Fagih, R. Graf, G. Settanni, and F. Skopik, ''Improved software vulnerability patching techniques using cvss and game theory,'' in 2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security), 2017, pp. 1–6. [Online]. Available: https://doi.org/10.1109/CyberSecPODS.2017.8074856

[48] Z. A. Hamza and M. Hammad, ''Testing approaches for web and mobile applications: An overview,'' International Journal of Computing and Digital Systems, vol. 9, no. 4, pp. 657–665, 2020. [Online]. Available: http://dx.doi.org/10.12785/ijcds/090413

[49] A.-R. Mawlood-Yunis, ''Debugging and testing using junit, espresso, and mockito frameworks,'' in Android for Java Programmers. Springer, 2022, pp. 135–181. [Online]. Available: https://doi.org/10.1007/978-3-030-87459-9_4

[50] Q. Naith, ''Crowdsourced testing approach for mobile compatibility testing,'' Ph.D. dissertation, University of Sheffield, 2021.

[51] Q. Naith and F. Ciravegna, ''The key considerations in building a crowd-testing platform for software developers,'' in Proceedings of the 4th International Conference on Crowd Science and Engineering, 2019, pp. 50–57. [Online]. Available: https://doi.org/10.1145/3371238.3371247

[52] Q. Naith and F. Ciravegna, ''Definitive guidelines toward effective mobile devices crowdtesting methodology,'' International Journal of Crowd Science, vol. 4, no. 2, pp. 209–228, 2020. [Online]. Available: https://doi.org/10.1108/IJCS-01-2020-0002

[53] M. Felderer, M. Büchler, M. Johns, A. D. Brucker, R. Breu, and A. Pretschner, ''Security testing: A survey,'' in Advances in Computers. Elsevier, 2016, vol. 101, pp. 1–51. [Online]. Available: https://doi.org/10.1016/bs.adcom.2015.11.003

[54] J. Lima, R. Dantas, C. Kamienski, and J. Kelner, ''Performance analysis of network composition in ambient networks,'' in 2007 Brazilian Networking Conference, 2007.

[55] F. Alqahtani, M. Amoon, and A. A. Nasr, ''Reliable scheduling and load balancing for requests in cloud-fog computing,'' Peer-to-Peer Networking and Applications, vol. 14, pp. 1905–1916, 2021. [Online]. Available: https://doi.org/10.1007/s12083-021-01125-2

[56] M. H. Vahidnia, A. A. Alesheikh, and S. K. Alavipanah, ''A multi- agent architecture for geosimulation of moving agents,'' Journal of Geographical Systems, vol. 17, pp. 353–390, 2015. [Online]. Available: https://doi.org/10.1007/s10109-015-0218-2

[57] L. Jayalath, K. Dharshana, and R. Rathnayake, ''Towards secure software engineering,'' South asian Res J Eng Technol, vol. 2, no. 6, pp. 45–53, 2020. [Online]. Available: https://doi.org/10.36346/sarjet.2020.v02i06.001

[58] S. L. Dasanayake, A. Mudalige, and M. L. T. Perera, ''Framework for secure coding: An algorithmic approach for real-time detection of secure coding guideline violations,'' 2021.

[59] J. Lai and E. Arko, ''A shoulder-surfing resistant scheme embedded in traditional passwords,'' in Proceedings of the 54th Hawaii International Conference on System Sciences, 2021.

[60] S. Potocky` and J. Štulrajter, ''The human interface device (hid) attack on android lock screen non-biometric protections and its computational complexity,'' Science & Military Journal, vol. 17, no. 1, pp. 29–36, 2022. [Online]. Available: https://doi.org/10.52651/sam.a.2022.1.29-36

[61] L. Bošnjak, J. Sreš, and B. Brumen, ''Brute-force and dictionary attack on hashed real-world passwords,'' in 2018 41st international convention on information and communication technology, electronics and microelectronics (mipro). IEEE, 2018, pp. 1161–1166. [Online]. Available: https://doi.org/10.23919/MIPRO.2018.8400211