International Journal of Information Security
https://doi.org/10.1007/s10207-022-00642-2

REGULAR CONTRIBUTION

Bu-Dash: a universal and dynamic graphical password scheme (extended version)

Panagiotis Andriotis[1] · Myles Kirby[2] · Atsuhiro Takasu[3]

Abstract
Passwordless authentication is a trending theme in cyber security, while biometrics gradually replace knowledge-based schemes. However, Personal Identification Numbers, passcodes, and graphical passwords are still considered as the primary means for authentication. Passwords must be memorable to be usable; therefore, users tend to choose easy to guess secrets, compromising security. The Android Pattern Unlock is a popular graphical password scheme that can be easily attacked by exploiting human behavioristic traits. Despite its vulnerabilities, the popularity of the scheme has led researchers to propose adjustments and variations that enhance security but maintain its familiar user interface. Nevertheless, prior work demonstrated that improving security while preserving usability remains frequently a hard task. In this paper we propose a novel graphical password scheme built on the foundations of the well-accepted Android Pattern Unlock method, which is usable, inclusive, universal, and robust against shoulder surfing and (basically) smudge attacks. Our scheme, named `Bu-Dash`, features a dynamic user interface that mutates every time a user swipes the screen. Our pilot studies illustrate that `Bu-Dash` attracts positive user acceptance rates, it is secure, and maintains high usability levels. We define complexity metrics that can be used to further diversify user input, and we conduct complexity and security assessments.

Keywords Usability · Acceptance study · User authentication · Mobile device · Complexity · Android · Pattern unlock · Usable security

1 Introduction

User authentication is a ubiquitous task performed daily by millions of mobile device users. Personal Identification Numbers (PINs) have been widely used during mobile computing's adolescence. Since 2010, several proposals have been presented aiming to replace 4- or 6-digit PIN screen lock methodologies (alphanumeric, graphical, biometrics, implicit authentication). Android developers were among the first that attempted to introduce a graphical-based method for user authentication on mobile devices proposing the Android Pattern Unlock (APU) scheme during 2008 [2]. Earlier studies showed that the APU was used by at least 25% of Android users [2,3]. On the other hand, contemporary or implicit authentication methodologies, like the "Smart Lock" on Android, do not seem to be very popular among users [4].

The proliferation of biometrics nowadays is evident due to the increased usability they offer, urging consumers to replace traditional text or graphical passwords (knowledge-based) with fingerprint and face identification methods [5] (biometric-based). However, although biometrics seem to be the preferred user authentication methodology, there still exists the need to set up a knowledge-based password on the device in case the biometric sensor fails. Therefore, text or graphical-based passcodes are still necessary to ensure

This paper is an extended version of our work [1] that was presented at the HCI International 2022 conference in June 2022.
Myles Kirby was a student at UWE Bristol while this work was conducted.
Dr. Panagiotis Andriotis was an International Research Fellow of Japan Society for the Promotion of Science (Postdoctoral Fellowships for Research in Japan (Standard Pathway)) when this paper was submitted for publication.

✉ Panagiotis Andriotis
p.andriotis@bham.ac.uk

[1] University of Birmingham, Edgbaston, Birmingham B15 2TT, UK

[2] University of the West of England, Frenchay Campus, Coldharbour Ln, Bristol BS16 1QY, UK

[3] National Institute of Informatics, 2 Chome-1-2 Hitotsubashi, Chiyoda City, Tokyo 101-8430, Japan

Published online: 04 December 2022

smooth and untroubled authentication for mobile device users, advancing user's security [6].

Prior work on text-based authentication investigated the transition from 4- to 6-digit PIN passcodes and concluded that longer PINs attain only marginally improved security [7,8]. The transition from 4-digit to longer passcodes was the only notable change (or improvement) in this kind of knowledge-based user authentication for mobile devices. On the other hand, several graphical password schemes have been proposed aiming to provide more usable and secure solutions for mobile devices [9–12].

If we focus particularly on the APU scheme, we can list a handful of proposals that aimed to enhance its security. The implementation and addition of password meters as an improvement toward raising users' security awareness has been studied extensively in the past [13–15]. Although this line of work demonstrated that password meters can assist in the formation and adaptation of more secure passcodes, they haven't been widely adopted yet by vendors and developers. In addition, research has shown that similarly to the extension of the 4- digit to 6-digit format for passcodes [3], strategies like the expansion from the standard 3×3 to a 4×4 grid, do not offer significant security enhancements [16]. Other variations of the APU scheme include node re-arrangement [17], system-guided contact point selection [18], or dual superimposed input on the same 3×3 grid [2] aiming to prevent or minimize threats from shoulder surfing attacks.

The common characteristic of the aforementioned methods is their intention to propose (mainly minor) structural interventions to the original APU scheme that will not drastically harm users' familiarity with the interface. This is because a dramatic alteration might cause frustration and disapproval. Another example of this approach is the inclusion of tactile or pressure-based methodologies (in various platforms) that request from users to tap to certain areas or apply pressure periodically on the screen when forming their usual passcodes [19–21].

Despite the plethora of proposals to improve graphical passwords against smudge [22,23] and shoulder surfing attacks [24], to the best of our knowledge, there is no research work that attempts to blend the APU's popularity with the implementation of a dynamic grid. In this paper we introduce `Bu-Dash`, a proof of concept based on design principles found in the APU and in gaming platforms.[1]

We were initially inspired from the *Morse code* and its use of dots (or `Bullet` points •) and `Dashes` (–) to create an encoded vocabulary (or lexicon) to be used in telecommunications. Therefore, we envisioned a passcode scheme that comprises symbols instead of alphanumerical characters. However, because the use of only two symbols in a password would introduce security issues (i.e., limited password space), we propose to utilize additional shapes as the passcodes' potential building blocks: ◯, □, −, △, ×. These shapes should probably look familiar to gamers,[2] or other broader audiences.[3] Their selection was based on previous research [25] which demonstrates that these are the least complex shapes in a series of different candidates [26]. Moreover, these symbols are widely utilized daily in various settings, therefore people should feel comfortable in using them. Furthermore, as discussed later in Sect. 2, these symbols have been widely used in the literature for similar purposes in the past.

Additionally, to defend from shoulder surfing and smudge attacks we propose a novel approach in designing graphical password schemes. Instead of forming the password by swiping a finger on the nodes of a static grid, we propose the use of a dynamically changing grid. `Bu-Dash` is based on the popular APU 3×3 node interface which is well-known to mobile device users. But instead of having static nodes (i.e., •), `Bu-Dash`'s grid is dynamic, featuring randomly assigned shapes in its nodes (◯, □, −, △, ×). The shapes keep changing every time users move their fingers on the grid making the scheme more robust to shoulder surfing and smudge attacks, without drastically affecting its usability. Further details about the scheme's design are provided in Sect. 3.1.

In summary, this paper makes the following contributions:

- We propose a novel graphical password scheme based on Android's popular 3×3 node interface, but with a twist; instead of having a static grid as a base of the interface, we introduce the use of a dynamic one to evade shoulder surfing and smudge attacks.
- We develop a mobile application to showcase the `Bu-Dash` system and collect preliminary feedback from mobile device users. Thus, we conduct a series of pilot studies with users who volunteered to participate and comment on the feasibility of introducing such as scheme. Reported results can be perceived as an acceptance study of the scheme.
- We report early results that show usability is not drastically reduced due to the introduction of a shifting grid in the scheme's interface.
- We comment on the complexity characteristics of the proposed scheme and we discuss possible security implications.

The rest of this paper is structured as follows. Section 2 offers a comprehensive literature review of methods related to

[1] https://www.playstation.com/en-gb/legal/copyright-and-trademark-notice/.

[2] We used Google's "Material Icons" for this research work: https://fonts.google.com/icons.

[3] We refer to viewers of the popular series "Squid Game".

user authentication for mobile devices. In Sect. 3 we discuss our design strategy, we provide examples of how the scheme was communicated to users, and how a Bu-Dash password can be formed. Section 4 presents methods we used to collect data related to the use of the scheme. In Sect. 5 we present results from our online survey and from data collected by volunteers who downloaded and interacted with an application we developed as a proof of concept of the Bu-Dash scheme. Section 6 examines Bu-Dash's password space complexity and discusses possible security implications of biased input. Section 7 distils our findings and discusses limitations and further challenges. We conclude this paper in Sect. 8.

2 Related work

Graphical passwords for mobile devices have been introduced as a more usable solution for user authentication [27] because graphical information is more memorable by humans [28]. For example, von Zezschwitz et al. [29] demonstrated in the past that mobile device users would prefer to utilize a pattern rather than a PIN.

Criticism on the APU security Prior work on Android patterns, however, investigated users' biases and habits when interacting with the 3×3 node interface and found preferable starting and ending points, and N-grams [23], which are sometimes related to the influence of human factors such as users' handedness [30]. The APU security was quantified by Uellenbeck et al. [31] who found that, in theory, APU selection is as diverse as selecting a 3-PIN password [2]. The lack of APU's passcode diversity due to human aspects was also confirmed by Aviv et al. [16] in an online study, and more extensively by Loge et al. [32] whose work showcased users' poor security perceptions when forming passcodes in different contexts (e.g., authentication in banking or shopping apps).

The APU has been also studied as an attack surface with research focusing basically on side channel and guessing attacks. Aviv et al. [22] showed that smudges that unintentionally reside on mobile devices' screens can eventually aid guessing attacks against users' passwords. Andriotis et al. [23] combined insights retrieved from collected passcodes and performed an in situ lab study, experimenting with guessing attacks. At a later work, they commented on the feasibility of performing successful guessing attacks on the APU using common knowledge [30]. Another work by Cha et al. [33] also demonstrated that smudge attacks can boost attackers' guessing performance. Additionally, it was shown that built-in smartphone sensors may partially leak PIN and pattern information [34]. Android patterns are also susceptible to shoulder surfing [35,36] and video-based attacks [37,38]. However, it still remains among the most popular user authentication methods on Android [3] drawing researchers' attention.

Improvements of the APU robustness The lack of diversity in choosing Android patterns and the influence of human biases in the scheme's security have led researchers to propose a variety of solutions aiming to make the APU more robust to smudge attacks [39–41] and shoulder surfing [42,43]. Another strand of research proposes the use of password meters to diversify input and enhance awareness [13–15]. However, Golla et al. [44] demonstrated that meters based on visual estimators should be treated with caution. Nevertheless, they state that the inclusion of such measures (i.e., strength meters) can be eventually beneficial.

Other proposals: (a) incorporate dual input on the same 3×3 framework [2]; (b) feature extended 4x4 grid interfaces [16]; (c) utilize background images and animations to enhance passcode selection [45]; (d) employ assisted pattern formation [18,46]; or (e) integrate blocklists [47] to enable a more diverse pattern selection and incommode guessing attacks. Most of these solutions do not alter radically the well-known 3×3 interface, but they attempt to include small adjustments in the user authentication experience keeping the main grid in a static state. Tupsamudre et al. [17] propose an alternate circular layout (namely "Pass-O") simplifying the APU drawing rules. Their usability evaluation, however, shows that users tend to create shorter and less complex passwords under the Pass-O scheme [48].

Layouts, colors and dynamic grids Alternative layouts and dynamic grids have been proposed in the past for *PIN-based* authentication aiming to minimize the influence of shoulder surfing attacks. However, floating [10] or rotating [49] grids result in longer login times than conventional text-based systems. Gesture-based proposals, such as "SwiPIN" [12] might require long training periods for the user to become familiar with, and other methods, like the "Xside" [42] or the " Glass Unlock" [50] require specialized equipment (e.g., Google Glass equivalent gadgets) or they can only work on certain devices.

Other proposals include the use of colors and shapes in the user authentication process. "ColorSnakes" [11] for example, uses fake paths on a grid of numbers to disguise user input, but lacks usability as it introduces higher authentication times and error rates. "Chameleon" is a hybrid scheme using a mixture of digits, colors and shapes but it is not clear whether it can fit in small screens like these used on smartphones [51]. Some of the shapes used in this work (○, □, △) are similar to those we utilize for the Bu-Dash scheme. Li et al. [52] use similar shapes as well in conjunction with behavioral biometrics. "FakePointer" [53] also uses shapes to obscure input and randomizes entries to defend against peeping attacks with video cameras. Similar symbols with the ones we use in this paper are also incorporated in Lee’s work [54].

“ColorPIN” uses colors to guide PIN entries in ATM settings [55], “PicassoPass” mixes colors, basic and theme-based shapes and characters to resemble passwords/ phrases [56], and “ShaPIN” combines multiple selectable items into a single input element [57] illustrating colored shapes, and characters (numbers, or letters) adding complexity that can be detrimental as far as usability is concerned. Additionally, these authentication methods are not inclusive because they exclude certain categories of users (e.g., those with color deficiencies). Finally, similarly to our scheme, “ SteganoPIN” [58] and “ SwitchPIN” [59] are using dynamic interfaces that randomly assign digits on 3×3 grids. The SteganoPIN creators, however, state that their system is more appropriate for ATM and PoS systems rather than mobile devices.

Threat model and our proposed scheme In this paper we propose `Bu-Dash`, the first graphical password scheme for mobile devices that adopts concepts from aforementioned work, but aims to present a more usable and simple authentication process. Our system uses the familiar and very popular 3×3 grid from APU, but instead of static nodes, it features dynamic and randomly mapped shapes as nodes of the 3×3 interface. Users envision and construct passcodes that consist of symbols only, which are formed on the randomized interface swiping their fingers, mimicking the APU experience. The randomized and dynamic node mapping ensures that `Bu-Dash` remains robust against shoulder surfing attacks, and the simple symbols used as constructing blocks contribute to the system’s usability.

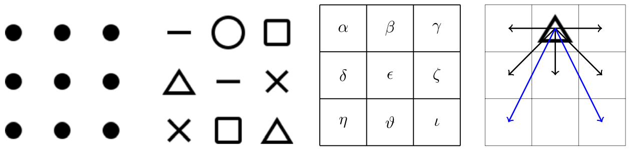
We consider that the scheme aims to protect against a non-targeted attacker that performs a physical observation (not video or camera-based) attack. Similarly to the APU concept, the attacker is only able to perform an “online” attack, meaning that they have limited attempts to guess the passcode before the device gets locked.

3 Proposed scheme

This section describes our design approach and how `Bu-Dash` is different from the APU (Sect. 3.1). Recognizing that the verbal description of a dynamically changing interface might not be sufficient for the reader, we also offer a concrete example and a detailed illustration of the interface to showcase how the scheme is used (Sect. 3.2).

3.1 Design considerations

The APU is a popular authentication method among Android users, but it has been flagged as susceptible to shoulder surfing attacks [35]. Many believe that this is because the 3×3 grid—which is used as a canvas to draw the pattern– is quite limited (it has only 9 nodes). However, research has shown that the expansion of the grid area, does not always affect security positively [16]. The same observation holds true for different graphical password schemes, such as LG’s “Knock Codes” [21].



(a) Static **(b)** Dynamic **(c)** Numbering **(d)** Neighbors

Fig. 1 The APU grid (**a**), and an instance of the ever-changing Bu-Dash grid (**b**). Part (**c**) shows the nomenclature for the nodes’ positions and Part (**d**) demonstrates eligible moves to neighbor nodes from position β. Blue color indicates “knight moves” [35]

Our proposed scheme adopts design concepts from the APU and uses symbols as the building blocks of the password (Fig. 1). We use the familiar 3×3 grid setting from Android (Fig. 1a). We also utilize the method of forming passwords by swiping the finger among different nodes on the grid. However, different from Android which uses a static grid, we propose the use of a dynamic grid that keeps changing when the users swipe their fingers (Fig. 1b).

We believe that this addition to the password scheme makes the authentication process more resistant to shoulder surfing attacks, and totally robust against smudge attacks. Previous research has already shown that the use of dynamic grids works well against shoulder surfing attempts [27]. The `Bu-Dash` grid is not static and it does not feature only bullet points as nodes (like the Android’s grid, see Fig. 1a). The `Bu-Dash` grid is dynamic and it includes 5 different symbols as nodes (Fig. 1b). The nodes (symbols) are randomly chosen and fetched by the system when the password scheme launches and they keep changing when users swipe their fingers to select the next node in the password chain. We implemented the following guidelines to assist users to get familiar with `Bu-Dash` passwords.

3.1.1 Design guidelines

The chosen password must adhere to the following guidelines:

- The password is formed as a sequence of symbols from the following set: $\{\bigcirc, \square, -, \triangle, \times\}$
- *Length*: The preferred password must be 4–9 symbols long (similar to the APU).
- *Diversity*: The preferred password must contain at least 2 different symbols.

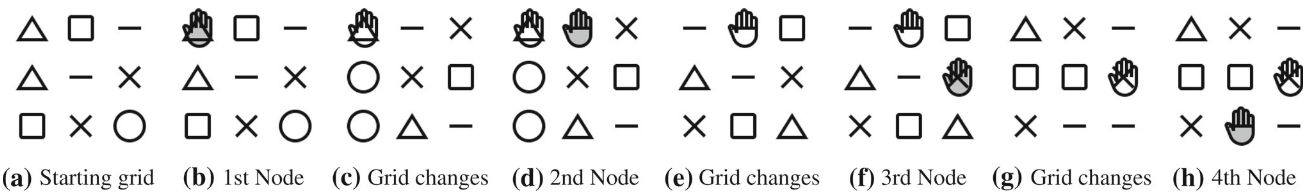

(a) Starting grid **(b)** 1st Node **(c)** Grid changes **(d)** 2nd Node **(e)** Grid changes **(f)** 3rd Node **(g)** Grid changes **(h)** 4th Node

Fig. 2 Example: Forming the password △ – × – : The starting grid (**a**) shows the 5 shapes in random order. The user starts at any node featuring the △ shape (**b**) and the node becomes visited. At that moment, the shapes on the grid change randomly (**c**) and the user tries to reach the next shape of the password, which is – (**d**). The – is reached and the grid changes again (**e**). The same process continues until the full password is formed (**f**–**h**)

- The password is formed when users swipe their fingers on a 3×3 grid that keeps changing when they visit a new position.
- *Allowed moves*: Users are allowed to swipe their fingers in the neighbor nodes only (similar to the APU), therefore “jumps” to a distant node are not feasible (e.g., from β to θ in Fig. 1c), unless the user choses a “knight move” [35], as seen in Fig. 1d).
- Users are allowed to revisit a node on the grid as many times they need. (This rule makes the scheme more usable compared to the APU that restricts access to already visited nodes.)

3.2 Example: forming a `Bu-Dash` password

To illustrate how the password scheme works with an example, let’s assume that a user chooses the following password to unlock a device: △ − ×− (see Fig. 2).

The starting grid places the 5 shapes at its nodes in random order (Fig. 2a). The user places a finger to any node that features the first building block of the preferred password (in this case, the △). In this particular example, the user chooses the top left node (position α) of the grid (Fig. 2b). This common bias can be seen in grid-based authentication schemes and it has been reported in numerous research works [21,30]. `Bu-Dash`’s design prevents this common bias to affect security because the starting grid is randomly initialized every time it launches. Additionally, as we can see in this particular example, the user has also the choice to start forming the password from position δ.

When the first node on the grid becomes visited, the rest of the nodes randomly change shapes immediately (Fig. 2c). Next, the user tries to reach the next shape of the password, which is the −, as it can be seen in Fig. 2d. The only available choice for reaching a − in this setting is to swipe the finger at position β. The − is reached and the grid changes again (Fig. 2e).

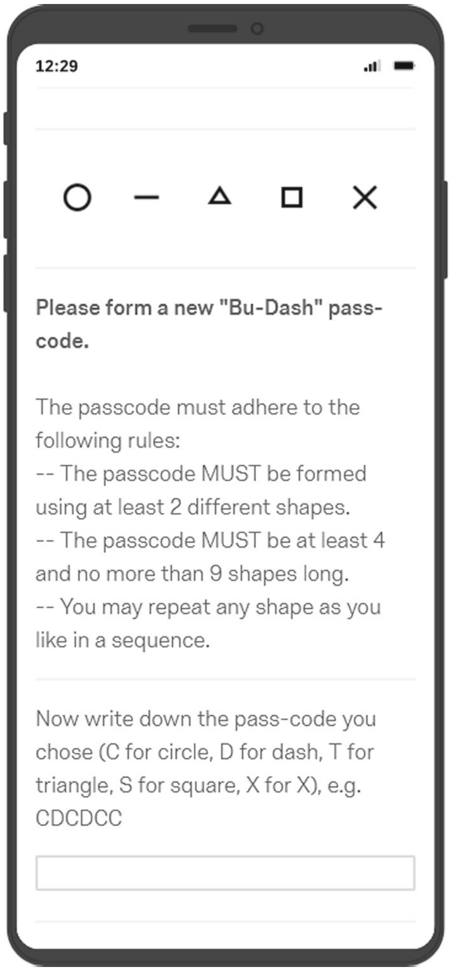
The next symbol in the password chain is the ×. As seen in Fig. 2e there are 2 available × in positions ζ and η (the latter move is also known as a “knight move” [35]). Therefore, `Bu-Dash` is more resistant to shoulder surfing attacks compared to the APU because users have more choices to form their passwords at any given time. In this example, the user selects position ζ (Fig. 2f) and the grid changes again (Fig. 2g). The last shape for this password is − and, as previously, there exist more than one nodes for the user to choose to form the last part of the password (positions γ, ϑ, ι). As seen in Fig. 2h, the user chose node ϑ and the password is now completed (△−×−). Regarding the scheme’s resistance to smudge attacks, it is apparent that oily residues left on the screen will not be of any use to an attacker, given that the grid is not static. Therefore, the possibility the shapes on the nodes of the grid to be identical in subsequent authentication attempts is minimal.

4 Methodology

In this section we discuss how we communicated the proposed scheme to volunteers that responded to our call to contribute to this research work. We also describe how data collection was performed. (Respondents were asked to consent and participate anonymously after reading our data and privacy policy.)

4.1 Exploratory study

First, we conducted an online survey requesting respondents (Android and iOS users) to provide a `Bu-Dash` password that they would possibly use on their devices (Fig. 3). The request was to provide an “easy-to-use and secure password”. In this digital “pen-and-paper” study, participants were not interacting with a device. They were asked to envision a *usable* and *secure* `Bu-Dash` password based on the constraints mentioned in Sect. 3.1.1. They also had the chance to view a short video that explained how the scheme works, showing an example about how they can swipe their fingers on the dynamic grid to form a password. We did not show any examples about how to form a certain password aiming to avoid introducing unwanted biases. Our primary intention was to gather information about how intelligible the pro-



12:29

○ — △ □ ✕

Please form a new "Bu-Dash" pass-code.

The passcode must adhere to the following rules:
-- The passcode MUST be formed using at least 2 different shapes.
-- The passcode MUST be at least 4 and no more than 9 shapes long.
-- You may repeat any shape as you like in a sequence.

Now write down the pass-code you chose (C for circle, D for dash, T for triangle, S for square, X for X), e.g. CDCDCC

Fig. 3 Explaining to survey participants how the Bu-Dash scheme works

posed scheme is. Additionally, we asked the participants *if they would prefer this scheme over the traditional APU*.

In this paper we refer to this group of participants as the "**Survey**" group ($n = 65$). The survey was communicated to a diverse mix of students and staff via emails and announcements in the learning platform of our University. We received responses from 85 individuals (who joined anonymously), but only 65 consented in participating and answered all given questions. Therefore, we ignored the input of 20 respondents. Additionally, as an incentive for their participation, individuals were given the option to be included in a raffle to win vouchers.

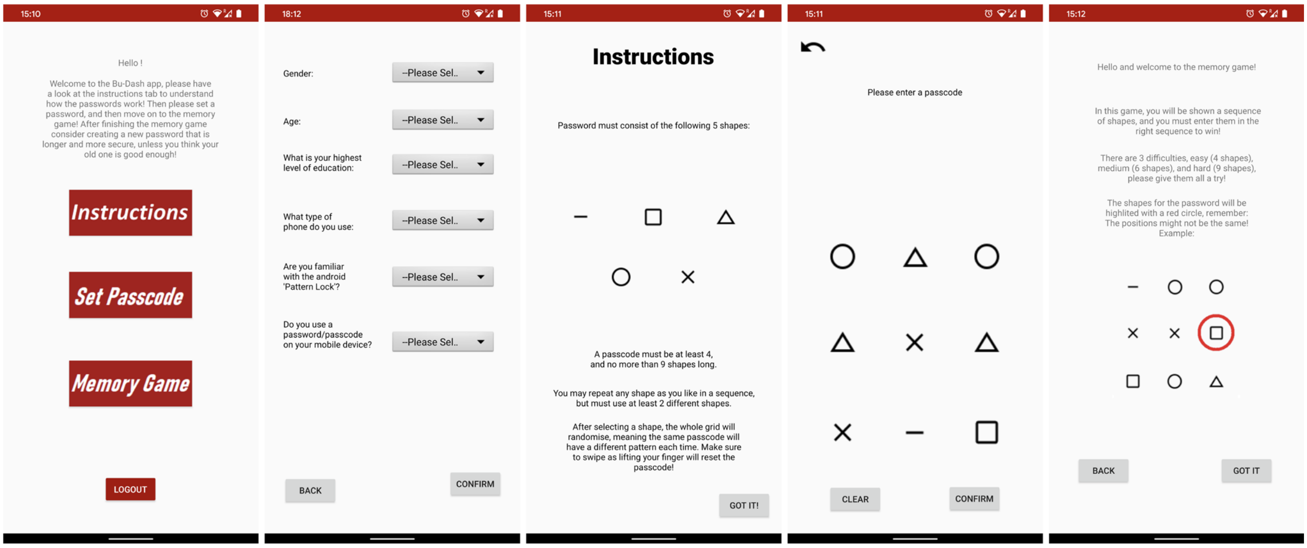
4.2 Interacting with the `Bu-Dash` scheme

At a second phase, we intended to gather and assess users' interactions with the proposed scheme. To this end, we developed an application, titled "Bu-Dash" (Fig. 4), which was distributed via the Google Play app store in the "Education" category. The application was featuring the `Bu-Dash` password grid and captured initially participants' input. The application was later updated to acquire very basic usability features at the latter stages of our experiments (Fig. 4a).

We released the first edition of our Bu-Dash application on Google Play and asked a small group ($n = 14$) of Android users (utilizing the same communication channels as previously) to interact with the `Bu-Dash` grid and provide passwords they would use on their devices. We refer to these respondents as the "**Pilot**" group in this work. Participants were asked to download our application on their Android devices and then launch it and follow the instructions provided by the application. The application asked them to provide basic demographics such as gender, age, education (Fig. 4b) and answer some generic, multiple-choice questions (the mobile OS they use; if they were familiar with Android's pattern lock screen; and which kind of authentication they use on their devices). Afterward, participants viewed a set of instructions about how to create a `Bu-Dash` password (Fig. 4c). It should be noted that the sequence of shapes was randomized every time the user was looking at the instructions. We followed this strategy to assess if provided passwords were affected by the shape the users were seeing first in the instructions. Finally, respondents were asked to form their preferred `Bu-Dash` password on their devices (Fig. 4d). Mimicking the same process while forming an APU passcode, the application was asking the participants (as a final step) to re-enter and confirm their `Bu-Dash` password. Additionally, users had the choice to be included in a raffle to win vouchers and then exit the application.

We then updated the application aiming to review usability characteristics of the proposed scheme. We added a "Memory Game" to the application and asked a different group of participants to play the 3 stages of the game (Fig. 4e). Users had the choice to play any of the "Easy", "Medium", or "Hard" games. The goal of the game was simple: After viewing the formation of (let's assume) an "Easy" password on the `Bu-Dash` grid (i.e., a sequence of 4 shapes: ×○×□), they were asked to re-enter this password. They were also given the chance to watch again the password formation on their screens as many times as they wanted. The "Medium" password consisted of 6 shapes and the "Hard" one consisted of 9 shapes. We did not use any complexity metrics [13] for this task, because our primary goal was to figure out if users would be able to recall at least a 4-node password. Therefore, we were aiming to assess the *short-term memorability* of the scheme. However, we offer a password complexity evaluation of the scheme in Sect. 6 in this paper. We refer to these participants as the auxiliary or "**Aux**" group ($n = 18$).

To conclude, results reported in the next section derive from answers we got from a set of 97 participants. Sixty-five of them belong to the *Survey* group which provided qualitative insights about the proposed password scheme. The rest of them (from the *"Pilot"* and *"Aux"* groups, i.e., $n =$



Hello !

Welcome to the Bu-Dash app, please have a look at the instructions tab to understand how the passwords work! Then please set a password, and then move on to the memory game! After finishing the memory game consider creating a new password that is longer and more secure, unless you think your old one is good enough!

Gender: --Please Sel..

Age: --Please Sel..

What is your highest level of education: --Please Sel..

What type of phone do you use: --Please Sel..

Are you familiar with the android 'Pattern Lock'? --Please Sel..

Do you use a password/passcode on your mobile device? --Please Sel..

Instructions

Password must consist of the following 5 shapes:

A passcode must be at least 4, and no more than 9 shapes long.

You may repeat any shape as you like in a sequence, but must use at least 2 different shapes.

After selecting a shape, the whole grid will randomise, meaning the same passcode will have a different pattern each time. Make sure to swipe as lifting your finger will reset the passcode!

Please enter a passcode

Hello and welcome to the memory game!

In this game, you will be shown a sequence of shapes, and you must enter them in the right sequence to win!

There are 3 difficulties, easy (4 shapes), medium (6 shapes), and hard (9 shapes), please give them all a try!

The shapes for the password will be highlited with a red circle, remember: The positions might not be the same! Example:

(a) Welcome screen **(b)** Demographics **(c)** Instructions **(d)** Bu-Dash grid **(e)** Memory Game

Fig. 4 Screenshots demonstrating the user interface of the Bu-Dash application in its various states

32) actually interacted with the `Bu-Dash` scheme on their mobile devices and provided responses quantified in the next section.

5 Results

First, we present results derived from the “Survey” group, comprising individuals that did not have access to the Bu-Dash application via their devices. Then, we discuss outcomes derived from two different user groups (namely “Pilot” and “Aux”) of our Bu-Dash application which utilized the proposed password scheme. Our aim is to identify common traits (if any) and attempt an initial assessment of `Bu-Dash`’s usability features.

5.1 Password space

Looking at the `Bu-Dash` password design constraints, we recall that the passcode must be at least 4 shapes long and its length can be up to 9 nodes. There are 5 different available shapes to choose from and there must be at least 2 different shapes in the password. Neighbor nodes can be visited as many times as necessary to form the password and the system ensures that there always exists at least one available shape (from the set of the 5) in the neighborhood of a visited node. This means that there exist $5 \cdot 5 \cdot 5 \cdot 5 - 5 = 620$ different 4-node `Bu-Dash` passcodes. Similarly, there are $\underbrace{5 \cdot 5 \cdot 5 \ldots 5 \cdot 5}_{9 \text{ nodes}} - 5 \approx 1.9M$ different combinations to form a 9-node passcode. We exclude those combinations that contain the same symbols in each passcode, e.g., those similar to ◯ ◯ ◯◯ for a 4-node passcode. Thus, we have more than 2.4M unique passcodes under this scheme (i.e., 2,441,220). This analysis shows that `Bu-Dash`’s password space is more than 6 times larger than the one defined by the APU scheme (which has 389,112 unique passcodes [15,30]). However, the APU has a wider space (if we consider that options are equiprobable) when we focus on passwords with 4–6 nodes, see Table 1. Nevertheless, as we discuss later in detail, the APU password space significantly shrinks in practice due to human factors and biases that often dictate user input [30].

5.2 “Survey” group

Most participants in the treatment group were undergraduate students. In the “Survey” group most respondents identified as males (66%), iOS users (78%), but the majority (94%) was familiar with the APU scheme (Table 2). In this survey we were basically targeting respondents that did not have access to the Bu-Dash application; this explains the prevalence of iOS users in the sample. Most participants (94%) said they use a passcode on their devices and the majority (72%) prefer biometric authentication methods (Fingerprint or Face ID).

After providing basic demographics, respondents were able to see a sequence of the available `Bu-Dash` shapes (◯ – △□×) with instructions about how to form a *valid* `Bu-Dash` password (as seen in Fig. 3). We then asked them the following two *open-ended* questions:

- **Q1**: “Write down the passcode you chose (C for circle, D for dash, T for triangle, S for square, X for X), e.g. CDCDCC”.

Table 1 Comparison of unique Bu-Dash and APU passwords

Nodes	Bu-Dash	APU
4	620	**1624**
5	3120	**7152**
6	15,620	**26,016**
7	**78,120**	72,912
8	**390,620**	140,704
9	**1,953,120**	140,704
Total	**2,441,220**	389,112

Bold shows prevelence of unique passwords

Table 2 "Survey" group demographics and answers

Question	Answer	Number	%
Age	18–21	45	**69%**
	22–25	10	15%
	26–29	5	8%
	30 and above	5	8%
Gender	Male	43	66%
	Female	21	32%
	Other/Prefer not to say	1	2%
Education	Undergraduate student	47	**72%**
	Other	9	14%
	University degree	7	11%
	No qualifications	2	3%
Mobile OS	iOS	51	**78%**
	Android	13	20%
	Other	1	2%
APU familiarity	Yes	61	**94%**
	No	4	6%
Use passcode	Yes	61	**94%**
	No	4	6%
Passcode used	Fingerprint	25	**38%**
	FaceID/unlock	22	**34%**
	PIN	11	17%
	Pattern/password/other	3	5%

Bold shows the most frequent answer

- **Q2**: " Would you use the "Bu-Dash" passcode scheme on your device? Which scheme you would use: (a) Android Pattern Lock, or (b) Bu-Dash? Please explain why.

Below we discuss insights resulted from their responses.

5.2.1 The `Bu-Dash` scheme is comprehensive

Although we did not offer any mechanisms to validate correct formation and input of the provided `Bu-Dash` passwords, invalid entries were not identified **(Q1)**. Thus, we can deduce that the scheme is intelligible and the provided instructions are sufficient and comprehensive.

5.2.2 Qualitative study: biometrics prevalence

Table 3 is a qualitative codebook derived from the answers to **Q2**. We can see that although the majority of respondents (i.e., 51 people, as seen in Table 2) are iOS users that utilize biometric authentication on their devices, 23 of them expressed positive views regarding the use of a `Bu-Dash` password on their devices. For example, **P51**: *"Yes, because you could easily remember the shapes"*. Additionally, 6 participants did not use a strong positive word (i.e., "Yes", "Definitely", etc.); hence, they were taxonomized as neutral. However, they eventually expressed a positive attitude toward the proposed scheme: e.g., **P42**: *"On a mobile device I would try it out. I like the idea that it moves about"*. Positively inclined respondents basically commented on the usability and the security that `Bu-Dash` provides: **P61**: *"Yes, it provides improved security for my device and is easy"*, and: **P41**: *"Cause it the same concept as using numbers its secure and easy to remember"*.

Negative answers for using `Bu-Dash` were basically focused around users' convenience with current methods and biometric authentication (13 users, marked with an asterisk* in Table 3). However, we should recall that knowledge-based methods are still important, because they are required as a complimentary method of authentication, in case the device remains idle for a long time (or after it restarts), or in case the biometric sensors fail (especially in the COVID era, when users used to wear face masks in closed spaces, thus methods such as "FaceID" were not usable).

Although **Q2** requested from users to choose whether they would use a `Bu-Dash` or an APU password, several participants seem they would not give up the convenience provided by biometrics. This was made clear in their responses: e.g., **P49**: *"No, because Face ID is much faster"*. However, if we ignore these responses (given that they did not comment on their preference between the APU or `Bu-Dash`, but they just advocated for biometrics) we can see that the same amount of people in our sample are positively (29), or negatively (28) inclined to use `Bu-Dash`. We make this claim considering that 6 participants expressed a neutral view but they were eventually more keen to adopt the proposed scheme: e.g., **P52**: *"Maybe, it seems like an interesting and puzzling way to make your phone secure"* (Table 3).

5.2.3 Password characteristics

We gathered statistics from the acquired passwords and we will discuss them above. To make comparisons with results from users that interacted with the Bu-Dash application eas-

Table 3 Qualitative codebook from Q2 responses ("Survey" group)

	Freq.	Description	Example quote
Positive	**15**	Security and usability	"Yes, cause looks nice and safe"
	3	Passcode replacement	"I would rather use this than a number password as to avoid entering easily remembered information, e.g., DOB, etc."
	3	Biometrics alternative	"I would consider it as a backup method after fingerprint"
	2	Complexity	". . . the fact the shape assortment changes as you fill your password is a vital and nice touch! :)"
Neutral	2	Security	"If there was research to support that it was more secure, I would probably use it."
	4	Other	"On a mobile device I would try it out. I like the idea that it moves about", "Maybe, it seems like an interesting and puzzling way to make your phone secure"
Negative	**8**	Memorability	"No as it is not memorable, however, could become used to it eventually"
	6	Complexity	"No. Seems complicated"
	5	Design/Other	"No, I may confuse with the shape"
	4	Security	"No because it doesn't seem very secure"
	5	*Convenience**	"I'm more familiar with using my standard password"
	8	*Biometrics**	"No, because Face ID is much faster"

Bold italics demonstrate the answers related to the preference in biometrics and other standard password schemes (convenience)
Bold highlights most significant answers

Table 4 Bu-Dash symbols distributions as starting points and as password nodes

Symbols	Survey group		Pilot group		Aux group	
	Starts with	Appears	Starts with	Appears	Starts with	Appears
○	**22**	51	3	8	3	**14**
□	14	46	2	7	2	6
–	8	44	1	5	0	5
△	12	51	0	6	5	12
×	9	51	**8**	**13**	**8**	13

Bold shows dominant numbers

Table 5 Frequency analysis of password length and N^o of distinct symbols used

Groups	Password length (%)						N^o of distinct symbols used (%)			
	4	5	6	7	8	9	2	3	4	5
Survey	18.46	15.39	**27.69**	18.46	7.69	12.31	12.31	29.23	**32.31**	26.15
Pilot	**42.86**	21.43	21.43	0	14.28	0	**50.00**	21.43	28.57	0
Aux	**55.6**	27.78	5.55	0	11.11	0	33.33	**55.56**	11.11	0

Bold shows dominant numbers

ier, we showcase aggregated statistics from all groups in Tables 4, 5 and 6.

We mentioned previously that the "Survey" participants saw a sequence of the shapes they should use to create their Bu-Dash passwords; the circle was depicted first (○ – △□×). This might have created a bias toward starting their passwords with a ○. Table 4 shows that 22 (i.e., approximately 1/3) of the participants created a password starting with a ○. In the next sections we discuss how we managed to overcome this issue by randomizing the symbols that are fetched first in the tutorial part of the Bu-Dash application. Additionally, we noticed that the – and the × symbols were the least favorite to start a password in this sample of users. We also estimated how many times the distinct shapes appear in the set and we report that their frequency is almost uniform (with approximately 51 appearances each). However, □ and – appear to be used less frequently than the other symbols, with 46 and 44 appearances, respectively.

Table 5 aggregates frequency analysis results. We can see that most participants created a password with 6 nodes (27.69%) and approximately 32% of the participants in this sample used 4 shapes in their passwords. One can deduce that the "Survey" participants were mostly focused on the proposal of a secure password because they did not have the opportunity to actually interact with the Bu-Dash scheme on their devices.

Table 6 Statistics on password length and N^o of distinct shapes used

Groups	Password length			Symbols used		
	μ	σ	Median	μ	σ	Median
Survey	6.185	1.580	6	3.723	0.992	4
Pilot	5.214	1.424	5	2.786	0.893	2.5
Aux	4.833	1.295	4	2.778	0.647	3

Finally, we report the following attributes of the provided password set (as seen in Table 6). For the length of the passwords we have $\mu = 6.185$ and $\sigma = 1.580$ (μ: mean, σ: standard deviation). The average number of different symbols that were used in this treatment is as follows: $\mu = 3.723$ and $\sigma = 0.992$. The latter feature is particularly interesting for the quantification of the scheme's complexity, as we will discuss in Sect. 6. The median value of the password length is 6 and the median value of the number of shapes per password is 4.

5.3 "Pilot" group

Similar to the "Survey" group, most participants in the "Pilot" set of users were undergraduate students (79%), identified as males (71%), using biometric authentication (57%) and their main device was running Android (79%). This group of participants was the first to use the `Bu-Dash` scheme on their devices; therefore, insights from their provided passwords are very useful to understand the usability and security of the scheme. We gathered their responses to compare them with our initial results derived from the "Survey" group (see Tables 4, 5, 6).

5.3.1 Starting point

In Sect. 5.2.3 we discussed the possibility that our survey instructions might have introduced a bias regarding the starting point of the provided passwords. The Bu-Dash's application instructions, however, were illustrating the shapes in random order every time they were fetched, aiming to eliminate similar biases (Fig. 4c). Furthermore, we tracked the sequence of shapes shown in the instructions during our experiments and compared them with the provided passwords from the users. The results demonstrate that only 2 of the 14 users provided a password that started with the same shape as the one that was firstly depicted in the instructions. Therefore, we believe that our updated tutorial instructions did not subconsciously introduce biases. Additionally, Table 4 shows that the majority of the "Pilot" participants preferred to start their password with a ×. Furthermore, the × is the most common symbol that appeared in this password set.

5.3.2 Using the **`Bu-Dash`** grid

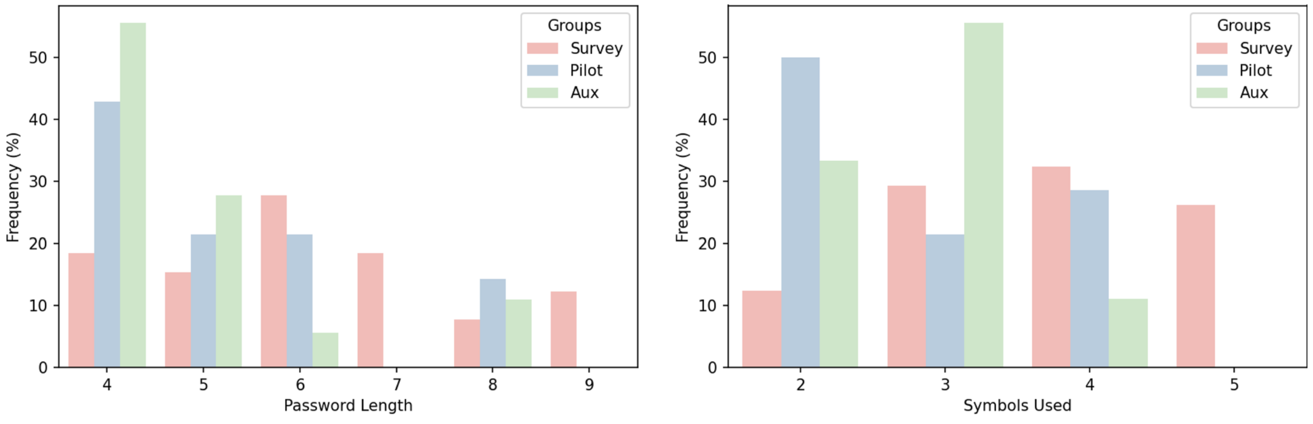
Table 4 demonstrates that although users envision and formulate on paper long and complex passwords (length: $\mu = 6.185$; shapes included: $\mu = 3.723$) aiming to advance security, they eventually end up with shorter and less complex passwords (length: $\mu = 5.214$; shapes included: $\mu = 2.786$) the first time they formulate a `Bu-Dash` "phrase" on their devices (median length: 5; median N^o of shapes 2.5). This is a common trend in grid-based password authentication [13]. Thus, in this treatment we can see that most respondents created a password with 4 nodes and half of the participants used 2 different shapes only. However, we advocate that the dynamic grid and the randomized order of the `Bu-Dash` starting grid are adequate to minimize shoulder surfing and smudge attacks. It would be interesting to see if the implementation of a password meter would urge users to create more secure passcodes.

Additionally, although the majority of participants in this group provided shorter passwords, we believe that the proposed scheme is more secure compared to the APU. Recent research illustrated [30] that due to common biases when users form APU passcodes (e.g., starting from top left), its available password space decreases dramatically (more than 90% for 4-node passcodes). Additionally, it is more feasible to extract parts of an APU password via observation (and then perform a guessing attack) because an attacker can easily see and recall edges that link nodes (since the grid is static), making the whole password less secure. On the contrary, `Bu-Dash` nodes are not visually linked with edges, thus an attacker cannot easily infer the next node in the password chain, if a node is already known.

To conclude, Table 4 showcases that the most favorite shape to begin a `Bu-Dash` passcode in this treatment was the ×. This shape also appears often in the password set along with ○. The least used symbol in the "Pilot" password set is the −. Additionally, as stated in the previous paragraphs, users in this treatment valued usability more than security and preferred less busy passcodes compared to the "Survey" participants.

5.4 "Aux" group

The "Aux" treatment comprised mainly participants identified as males (78%), Android users (83%), familiar with the APU (89%), using biometric authentication on their devices (67%). 56% were undergraduate students and the rest had at least one University degree. Results derived from this group's provided data (Tables 5, 6) confirm that when respondents use the `Bu-Dash` grid, they seem they choose shorter and less complex passcodes (length: $\mu = 4.833$; shapes included: $\mu = 2.778$). Median values for length is 4 and for the number of included shapes is 3. However, we can see the use of longer,



(a)

(b)

Fig. 5 Password length and symbols' diversity in collected passcodes. The figure at the left-hand side shows the number (%) of collected passwords with length of 4 up to 9 nodes in the three different participant groups. The figure at the right-hand side shows the number (%) of collected passwords that contained 2 up to 5 different symbols in the three participant groups

8-node passcodes, in some cases in both "Pilot" and "Aux" groups (Fig. 5a). Figure 5b also shows that participants that used `Bu-Dash` on their devices attempted to incorporate 3 and 4 symbols in some cases in their passcodes, aiming to enhance security.

5.4.1 Frequency analysis

Tables 4, 5 and 6 confirm trends we saw in the "Pilot" treatment. × is the most preferred starting shape in this treatment too (8 instances, i.e., 44.4%). Since this is not a large scale study (we report preliminary results here) we can only note that this finding might introduce security concerns related to the available password space, similarly with the APU scheme as we comment in Sect. 5.3.2. However, provided data from participants that interacted with the `Bu-Dash` grid (both from "Pilot" and "Aux" treatments) show that 68.75% of users that formed a short `Bu-Dash` code (4-nodes), preferred to include at least 3 shapes in their passcode. Therefore, we can see from these data that most users valued and considered security while forming easy-to-use passcodes aiming to add more shapes in the sequence. Additionally, similar to Sect. 5.3.2, "Aux" data show that the — is the least used shape in the password set.

Figure 5 illustrates results presented in Table 5. We can see that "Pilot", and "Aux" participants preferred to form shorter passcodes and used basically 2–4 symbols.

5.4.2 Commonly used passwords

Another noteworthy finding is that we did not encounter any particular passcode to be prevalent in the whole password set (Survey-Pilot-Aux, namely *S.P.A.*). We recognize that reported results come from a limited sample of participants ($n = 97$) and that diversity in the provided passcodes should be expected. However, only 5 different passcodes were seen to exist –twice– in the provided data. These are as follows: △◯××, ×◯×□, ◯□△×◯, ×××□□□, ×△×△×△×△. On the contrary, APU users frequently form passcodes that resemble letters of the latin alphabet, aiming to make them memorable [30], introducing subsequently additional security implications.

5.4.3 Preliminary usability assessment

We asked the "Aux" Group's respondents to participate in a Memory Game that was added in the final iteration of our experiments. As explained in Sect. 4.2, respondents were asked to play a Memory Game which featured 3 complexity levels ("Easy", "Medium", "Hard"). We did not explicitly tell them how many levels they should attempt to play. As we did not use any complexity metrics to assess how difficult it would be for an individual to memorize these passcodes, we randomly formulated one 4-node, one 6-node, and one 9-node passcode as an "Easy", "Medium", and "Hard" `Bu-Dash` password, respectively. Participants should choose the level of complexity they would like to play, and then they would see the password while it was formed on their screens (Fig. 4e). There was no limit on how many times they could watch the tutorial. Afterward, they had to recall and form that password on the `Bu-Dash` grid. The Bu-Dash application logged how many times they tried to play a game and if they successfully recalled the passcode. Results are as follows (Table 7).

Most participants in the "Aux" Group attempted to play the *Easy* game, but only 8 and 5 tried to solve the *Medium* and

Table 7 Memory game completions

Level	Attempted	Completed	Average attempts to completion	Failed or no completion	Failure rate (%)
Easy	18	17	1.13	1	5.6
Medium	12	8	1.88	4	33.3
Hard	5	3	2.67	2	40.0

Table 8 Comparison of password creation/confirmation times (preliminary)

Password schemes	μ (s)	σ (s)
`Bu-Dash`	19.84	8.81
DPatts [2]	25.41	14.57
Knock codes 2×2 [21]	16.2	7.7
Knock codes 2×3 [21]	18.4	11.0

Hard levels, respectively. Seventeen users successfully completed the *Easy* challenge; the average number of attempts to completion was approximately 1.13 attempts. Two "Aux" initial entries were considered as outliers and were excluded from the former estimation because users seemed they did not manage to complete the challenge after a reasonable number of attempts (more than 10 attempts each). The *Medium* challenge was undertaken by 12 individuals, and 8 of them successfully completed it with an average of 1.88 attempts. The *Hard* challenge was attempted by only 5 respondents; 3 of them successfully formed the –admittedly– challenging to recall password with an average of 2.67 attempts. These numbers confirm the expectation that when a password becomes longer, it eventually gets less usable and difficult to recall. However, there exist passcodes like the following one that are long, but very memorable: × × ×□□□ ○ ○○. Thus, password length is not the only feature that contributes to complexity, as we will see in the next section. Further experiments are needed to properly assess long-term memorability and the effects of password length in the password's complexity.

Furthermore, we randomly chose 7 candidates from the "Aux" Group and logged `Bu-Dash` password creation - confirmation times. We deduce that users needed 19,838.7 ms on average to perform the task: $\mu = 19.84$ s, $\sigma = 8.81$ s (see Table 8). In similar settings, Forman and Aviv [2] report that the Control treatment during their "Double Patterns" (i.e., *DPatts*) experiments needed 25 s to setup a *DPatts* password: $\mu = 25.41$ s, $\sigma = 14.57$ s. Samuel et al. [21] report the following statistics for LG's "Knock Codes": (a) For 2×2 grids: $\mu = 16.2$ s, $\sigma = 7.7$ s, and, (b) for their larger (experimental) 2×3 grids: $\mu = 18.4$ s, $\sigma = 11.0$ s. Although these insights are preliminary, they allow us to infer that `Bu-Dash`'s usability is not dramatically affected by the moving grid and it is favorably comparable with APU alternative methods.

6 Complexity analysis

In this section we report additional attributes found in passwords provided by the Bu-Dash application users ("Pilot" and "Aux" Groups). First, we attempt a theoretical complexity (and security) analysis of the proposed method. Then, we discuss basic complexity characteristics of the collected passcodes and we provide information for commonly used N-grams.

6.1 Measuring complexity

It is apparent that a passcode is less memorable if it is longer and if it comprises more than two symbols. To formalize this intuition, we define a complexity metric (namely *BDc*) aiming to quantify collected passcodes from all groups of our experiments. Thus, we use Lothaire's Combinatorics theory [60] and the definition of a finite *word*, followed by the simple metric *complexity of a word*. Additionally, we borrow the term *k-mers* from Bioinformatics [61] and combine them to define the `Bu-Dash` complexity metric (*BDc*) as follows.

6.1.1 Formal definition

We assume that the `Bu-Dash` symbols $\{\bigcirc, \square, -, \triangle, \times\}$ can be mapped to the following letters, respectively: C, S, D, T, X. Therefore, following Lothaire's theory [60], we define the finite, nonempty set $A = \{C, S, D, T, X\}$, as an *alphabet* that contains the *letters* (or `Bu-Dash` symbols) C, S, D, T, X.

A `Bu-Dash` passcode can be represented by any string (or *word*) $s = b_1 b_2 \ldots b_n$, where $n \in \mathbb{N}$ and $4 \leq n \leq 9$, given that at least one of the *letters* $b_1, b_2, \ldots, b_n \in A$, is not the same. Additionally, we assume that any word (passcode) can be formed by smaller *subwords* or *factors* with a binary operation called *concatenation*, as defined in [60]. For example, the passcode $CCDT$ can be formed by the concatenation of the factors CC and DT. In other words, we assume that a `Bu-Dash` passcode can be divided in smaller distinct pieces (subwords, or factors).

In Bioinformatics, in a similar fashion, *k-mers* are defined as substrings of length $k \in \mathbb{N}$ which are contained within a biological sequence (i.e., a sequence created from the letters: C, G, A, T). Therefore, we inherit the idea of using k-mers

Table 9 Complexity estimation of the passcode: XXXCCT

k	Distinct k-mers: frequency	N^o of distinct k-mers
2	"XX": 2, "XC": 1, "CC": 1, "CT": 1	4
3	"XXX": 1, "XXC": 1, "XCC": 1, "CCT": 1	4
4	"XXXC": 1, "XXCC": 1, "XCCT": 1	3
5	"XXXCC": 1, "XXCCT": 1	2
	BDc(XXXCCT)	**13**

Bold shows dominant numbers

for the estimation of the Bu-Dash passcodes' complexity. k-mers are actually the factors (substrings), as defined previously for the Bu-Dash passcode complexity estimation, and we can deduce that $k \in \{2, 3, \ldots, 9\}$. Thus, we can estimate the number of k-mers with $k \in \{2, 3, \ldots, 9\}$ in any Bu-Dash passcode.

Hence, we can define as complexity of a Bu-dash passcode (i.e., *BDc*) the number of distinct k-mers that exist in the passcode. This is the same definition as Lothaire's [60] *subword complexity* or *complexity of a word*. However, a long passcode is not always complex, e.g., × × ×□□□ ◯ ◯◯, or × × × × × × □ × ×. Nevertheless, there exist 6-mers, 7-mers, 8-mers, 9-mers that will cumulatively contribute to the complexity score of these passcodes, skewing the fact that these are actually very memorable passcodes. For this reason, we heuristically decided to consider only k-mers up to $k = 5$ for the complexity estimation of the Bu-Dash passcodes.

Finally, *complexity of a* Bu-dash *passcode* (i.e., ***BDc***) is defined as the number of distinct k-mers that exist in the passcode, where $k = \{2, 3, 4, 5\}$.

6.1.2 Complexity estimation example

As an example we estimate the complexity of the following passcode: ×××◯◯△, resembled by the word "XXXCCT". Table 9 lists the different k-mers of the word "XXXCCT" featuring their frequency in the passcode. Thus, there exist 4 distinct 2-mers, 4 distinct 3-mers, 3 distinct 4-mers, and 2 distinct 5-mers. As discussed previously in Sect. 6.1, we ignore 6-mers–9-mers for the *BDc* estimation.

Hence, *BDc*(XXXCCT) = 4 + 4 + 3 + 2 = 13.

6.2 Bu-Dash's password space complexity and security assessment

In this section we dissect the password space of the Bu-Dash scheme focusing on the passcodes' complexity (*BDc*). Table 10 aggregates basic statistics and Fig. 6 visualizes the complexity of the password space for all Bu-Dash passcodes with 4–9 nodes.

Figure 6 shows that $BDc \in \mathbb{N}$ and $5 \leq BDc \leq 26$ in the Bu-Dash password space. As expected, most 7- to 9-node passcodes are very complex, and in general, 4-node passwords cannot be considered as complex as the rest ones in the space. However, if we compare Bu-Dash short passwords with the analogous in the APU, we can argue that the APU ones are more vulnerable to shoulder surfing attacks, given that a 4-node APU password visually defines 2 or 3 edges (flat lines) that are easy to recall when a shoulder surfing or smudge attack is performed.

Table 10 Complexity of Bu-Dash's password space

Nodes	Count	*BDc* Mean	*BDc* Std
4	620	5.903	0.296
5	3120	9.763	0.544
6	15,620	13.585	0.772
7	78,120	17.367	0.982
8	390,620	21.108	1.178
9	1,953,120	24.809	1.364

In addition, as discussed in various papers [2,30], APU users provide biased input when forming their passcodes. For example, half of them start their patterns from the top left node. If one considers only this characteristic, then the APU password space shrinks dramatically. It was demonstrated [30] that if we consider this bias, the unique 4-node APU passcode number drops from 1624 to 154, which is a 90.5% decrease. Also, the same study shows that if we also consider the fact that most people tend to use the central node of the grid, then the unique 4-node APU passcode number drops from 1624 to 82, which is a 95% decrease. On the other hand, our study so far has only showed that our participants seem to prefer the × as a starting point. If we accept this preliminary finding as a bias, we infer that the unique 4-node APU passcode number drops from 620 to 496, which is a 20% decrease. Thus, even if such a bias exists in the Bu-Dash scheme, its overall 4-node password space is 3 times larger than the one that the most common biased APU input creates.

To take this further, we extend this analysis to assess how biased input affects the Bu-Dash and APU password spaces, based on the aforementioned assumptions. We use estimations made in [30], which describes the effect of the top left node bias in APU input. We compare these findings with

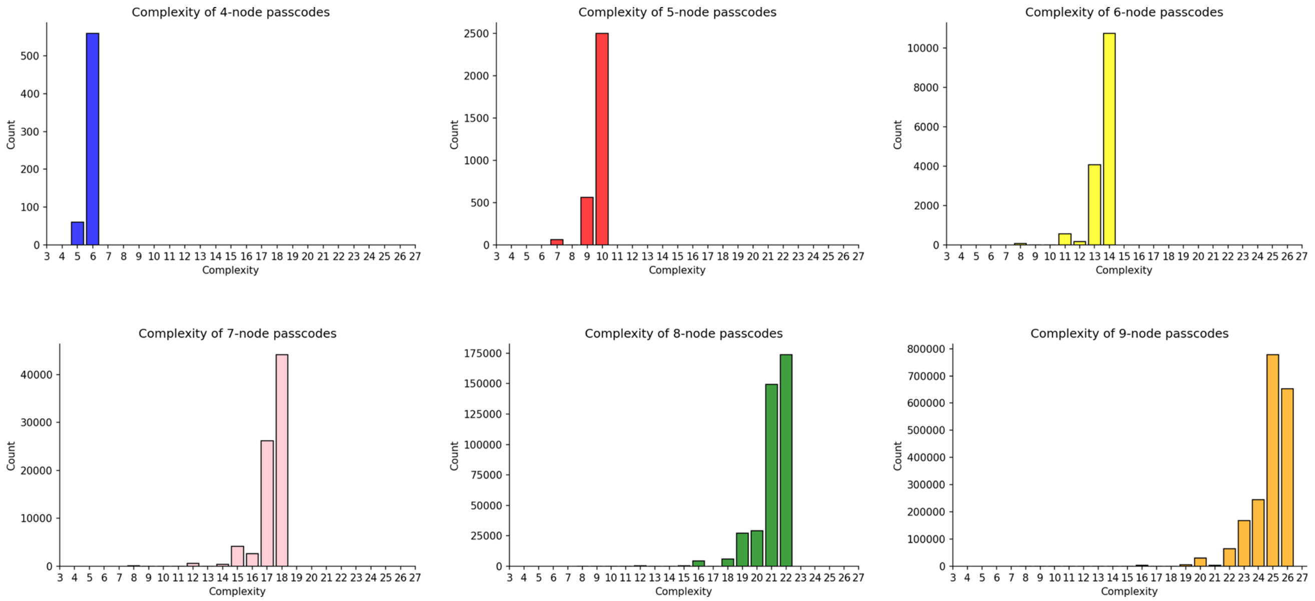


Fig. 6 Screenshots demonstrating the user interface of the Bu-Dash application in its various states

Table 11 Impact of biased input in the APU and `Bu-Dash` password spaces (numbers show available distinct passcodes)

Nodes	`Bu-Dash`	APU	Biased `Bu-Dash`	Biased APU [30]
4	620	**1624**	**496**	154
5	3120	**7152**	**2496**	684
6	15,620	**26,016**	**12,496**	2516
7	**78,120**	72,912	**62,496**	7104
8	**390,620**	140,704	**312,496**	13,792
9	**1,953,120**	140,704	**1,562,496**	13,792
Total	**2,441,220**	389,112	**1,952,976**	38,042

Bold shows best cases in the comparisons

the impact that the favorite starting point might has on the `Bu-Dash` scheme. Results are gathered in Table 11, and as we can see, `Bu-Dash`'s password space decreases 20% from biased input, while the APU shrinks 90.22%. The `Bu-Dash` scheme is more concrete even if we focus on the passwords comprising 4 up to 6 nodes, where the APU is in theory superior.

6.3 Complexity features of the collected passcodes

Since the collected password set from the Bu-Dash application users was not extensive enough, we report in this paragraph (Table 12) the frequency of node and shape numbers per provided passcode as a naive complexity metric. Further work to assess security and complexity of passcodes provided by humans is needed and other suitable metrics can be also used [13,15,21,31,62]. Table 12 shows that the Bu-Dash application users preferred to form short codes, but in most occasions they also opted-in to form codes containing more than 2 shapes, having probably in mind that this choice will make a shorter code more secure.

Table 12 Complexity in password set ("Pilot" and "Aux" groups)

Node/Shape N^o per passcode	Frequency	Percentage
4–2	5	15.7%
4–3	10	**31.4%**
4–4	1	3.1%
5–2	2	6.2%
5–3	2	6.2%
5–4	4	12.5%
6–2	4	12.5%
8–2	2	6.2%
8–3	1	3.1%
8–4	1	3.1%

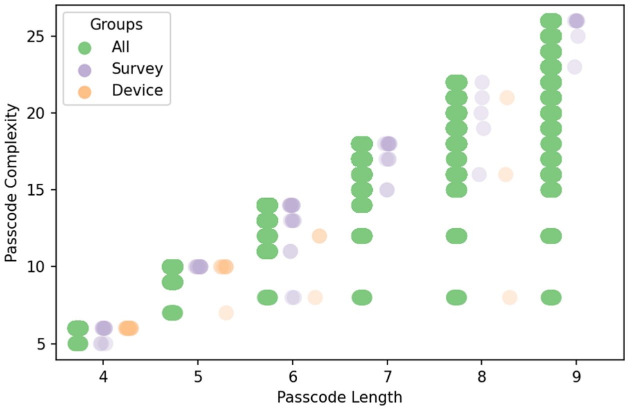


Fig. 7 `Bu-Dash` password space complexity

Table 13 Complexity statistics for the S.P.A password set

Group	Count	*BDc* Mean	*BDc* Std	*BDc* Min	*BDc* Max
Survey	65	13.95	6.20	5	26
Pilot	14	8.71	3.10	6	16
Aux	18	8.22	3.77	6	21

6.3.1 Using complexity

In this section we use the *BDc* metric to demonstrate in a scatter plot the complexity of the collected passcodes (from the S.P.A. treatments) in relation to their length. We also plot the whole password space in Fig. 7 aiming to show that although the majority of the Bu-Dash application users preferred to provide a shorter passcode, there also exist participants that formed complex passcodes. Note that the dots (resembling passcodes) in Fig. 7 are transparent, therefore density is represented by more solid colors. We can see that the "Survey" respondents in general provided more complex passcodes in a more diverse space. Participants that used the Bu-Dash application (marked as "Device" in the plot, i.e., "Pilot" and "Aux" groups) mostly provided shorter passcodes. Nevertheless, we can also see that there exist instances of more complex ones. As a matter of fact, those formed 4- and 5-node passcodes, chose to add complexity in their sequences. Table 13 shows the mean complexity values in the three study groups and some additional statistics. We can see that the Bu-Dash application users provided passcodes with average complexity of $BDc \approx 8.5$ and they avoided to use the least complex passcodes ($BDc = 5$).

6.3.2 Common shape combinations and N-grams

Finally, we extracted combinations of shapes and N-grams from the passcodes provided by the Bu-Dash application users. We did not consider which shape appears first in the sequence to estimate shape combinations. For example, ◯,□ is considered the same as □,◯. Table 14 aggregates frequencies of shapes appearing together in passcodes in the "Pilot" and "Aux" treatments. We can see that shapes ◯, △, × were mostly used by the participants either in pairs or by combining them together.

Table 14 Shape combination frequency analysis

Combinations	"Pilot"	"Aux"	Sum
◯,□	4	5	9
◯,−	2	3	5
◯,△	4	8	**12**
◯,×	7	10	**17**
□,−	1	1	2
□,△	4	3	7
□,×	7	4	11
−,△	1	3	4
−,×	4	3	7
△,×	6	8	**14**
◯,□,×	1	2	3
◯,−,△	0	1	1
◯,−,×	1	2	3
◯,△,×	1	3	**4**
□,−,△	0	1	1
−,△,×	0	1	1
◯,□,△,×	3	2	**5**
□,−,△,×	1	0	1

Bold shows the most frequently seen complexity in the dataset.

Table 15 shows the most common N-grams (N = 2, 3, and 4 respectively) in passcodes provided by the "Pilot" and "Aux" Groups. In this Table, order matters, i.e., ◯□ is different from □◯. As expected, we can see that respondents used frequently in their passcodes pairs of the same symbol, i.e., ◯◯, □□, −−, △△, ××, probably to make them more memorable. Additionally, confirming observations presented in the previous sections, we can see that most participants mainly utilized ◯, △, × for their `Bu-Dash` password formation. This information can be particularly valuable in case we need to implement blocklists that would make the scheme more secure against those biases.

7 Discussion

We envisioned an authentication system that would be easy to comprehend and adopt, and at the same time, it would be secure and usable. The aim of this work was to propose a robust graphical password scheme against shoulder surfing and mainly smudge attacks.

Table 15 N-gram frequency analysis (“Pilot” and “Aux” set)

N-grams	Frequency	Percentage
××	16	10.06%
◯×	16	10.06%
△×	14	8.81%
×◯	13	8.18%
×△	12	7.55%
◯◯	9	5.66%
□△	9	5.66%
△◯	8	5.03%
−−	6	3.77%
□□	6	3.77%
×□	6	3.77%
△△	6	3.77%
×△×	9	5.70%
× × ×	6	3.80%
× ◯ ×	6	3.80%
△ × △	6	3.80%
◯ × ◯	5	3.16%
×△ × △	6	3.82%
△ × △×	5	3.18%

Bold shows the most frequent combinations in passwords containing 2, 3, 4 symbols respectively

Novelty In Sect. 2 we discussed relevant schemes and their characteristics. Bu-Dash is the first method that combines the popularity of the APU (utilizing swiping gestures on the familiar 9-node interface) and the use of a dynamic grid. Most of the APU deficiencies relate to its static nature, therefore Bu-Dash overcomes vulnerabilities that derive from the APU, but at the same time maintains its familiar design. Dynamic approaches have been utilized for PIN-based authentication in the past; however, Bu-Dash is the first attempt to incorporate dynamic features on a graphical password scheme that extends the APU concept. Our acceptance study demonstrated: (a) positive user attitude toward the scheme, and, (b) acceptable usability standards. As discussed in Sect. 2, other graphical password schemes that use shapes and colors tend to be more complex, introducing usability and inclusivity problems. Bu-Dash uses symbols (shapes) and a dynamic grid, but it looks familiar to mobile device users given that its design concept derives from the popular APU.

Inclusivity and universality We believe that Bu-Dash is a universal scheme because it can be employed for user authentication in various settings. It can be utilized on smartphones and tablets, or it can be adjusted to work on even smaller screens (e.g., smartwatches). Our proposed method can be used on portable computers (utilizing the trackpad, or the mouse) and desktops. It is also *universal* because its building blocks are common shapes that can be recognized and used easily by any human. Therefore, there are no language, or other cultural (or even educational) burdens that could discourage people from using it.

Its dynamic 3×3 interface ensures that current mobile device users will not feel unfamiliar with the authentication process. Bu-Dash works similarly with the APU, requiring users to swipe their fingers on the mobile device screen in order to form the password. Compared to the APU, it has less restrictions (for example, a node can be visited as many times as needed) and its password space is more than 6 times larger. Our online survey indicated that the scheme is comprehensive and easy to perceive because, most of the respondents (“Survey” group), despite being iOS users employing primarily biometrics for user authentication, did not provide any invalid passwords when asked to create one after reading our basic instructions. In order to avoid having those outliers mentioned in Sect. 5.4.3, we might need to implement animated tutorials to further assist users perceive the scheme’s constraints.

By looking at the passwords provided by participants from groups “Pilot” and “Aux” we can infer that the scheme provides the opportunity to diversify users’ input compared to the APU. We did not find several repeating passcodes, but we recognize that our sample is not extended enough. However, we only saw a few trends in the sample that might be linked with human habits; e.g., the preference in using × as a starting point, or the fact that − seems to be the least favorite shape to use in general. Additionally, our analysis demonstrated that when participants were asked to form a Bu-Dash passcode on their devices, they chose *shorter* passcodes aiming probably to make them more memorable and usable. However, early indications show that when they did that they aimed to *add complexity* to the passcode using at least three shapes.

Mnemonic strategies We believe that the design of the proposed scheme allows the deployment of different mnemonic strategies to create a usable and secure Bu-Dash passcode. First, one can use the symbols ◯,− to create passcodes resembling Morse code sequences (if there still exist people that are actually aware of this lexicon). As an alternative, we can use Bu-Dash shapes to create conceptual paradigms to visually resemble common memorable phrases. For example, “home” or *a house* is usually symbolized with graphics like the following: ⌂. One can recognize that a square and a triangle might suffice to resemble conceptually the word “home”. Therefore, the password *“My home is 3 miles away from the lake”* might be translated as the following Bu-Dash passcode: □△−−−◯. According to Sect. 6.1, the complexity of that code is $BDc(STDDDC) = 13$. Although this is a comparatively high complexity, the use of a mnemonic strategy makes this passcode a very memorable secret. Hence, we can argue that a Bu-Dash passcode can be memorable and diverse when the user adopts similar mnemonic rules. And given that each person can have theoretically their own mnemonic habits and strategies, we could infer that their implementation can assist in diversifying users' input.

So far, our results showed that participants used pairs of shapes as a generic strategy to enhance memorability. As with most graphical passcodes, our intuition dictates that frequent use of the Bu-dash scheme will eventually assist in formulating memory muscle and will result in added usability. As a matter of fact, to further support this claim, we can cite a participant's comment (from the "Survey Group", as seen in Table 3) who stated the following: *"No as it is not memorable, however, could become used to it eventually"*.

Additionally, we propose the use of this graphical password scheme mainly as a solution for mobile devices. Therefore, similarly to all authentication schemes for mobile devices nowadays, if users forget their Bu-Dash password, they can always use their registration accounts to reset the passwords. This common solution is also implemented by the major mobile operating systems in case the device gets locked after a number of unsuccessful login attempts.

Usability and scalability In this paper we report preliminary usability results. Although our data are valuable because they come from users that interacted with our scheme on their actual devices, we cannot confirm if they generalize well. This is a limitation of this work. The collected Bu-Dash passcodes (derived from 97 participants in different settings) might provide a good first impression of how users would utilize the scheme, but a longitudinal and large-scale study would confirm the insights of the discussion provided in this section. Our initial results in Sect. 5.4.3 (Tables 7, 8) show that this is a usable scheme that can be employed in various devices. Future work should also focus on long-term memorability assessment and also on other usability metrics such as the average unlock times on participants' devices. These statistics can be inferred by longitudinal studies. The scope of our current work was to perform a pilot (or "acceptance") study of our proof of concept.

Security The collection of a larger dataset in the future will enable us to perform a more robust security analysis using metrics, such as *α-guesswork* ($\tilde{G_\alpha}$) or *β-success rate*, as proposed by Bonneau [62]. In this work we talked about the password space defined by the Bu-Dash scheme and we mentioned that it is larger from the one defined by the APU. However, the set of unique Bu-Dash passcodes with a shorter length is smaller than the one in the APU scheme. As discussed in Sect. 6.3, Android pattern formation is usually driven by human habits and biases, significantly shortening the APU password space. We advocate that Bu-Dash is a robust and secure authentication method, because it uses a dynamic grid which is randomly initialized every time it is launched. Bu-Dash can be used as an additional (or alternative) option for user authentication. We believe that it has certain strengths against the APU, i.e., it is more robust against shoulder surfing, smudge attacks, and human biases.

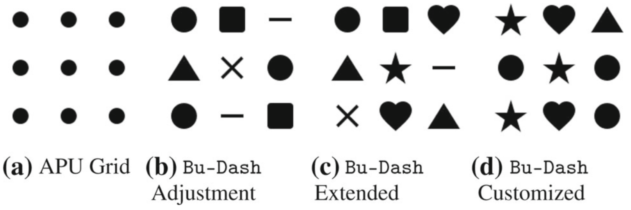


Fig. 8 Potential adjustments in the use of the Bu-Dash scheme

Smudge attacks against Bu-Dash are unlikely to be successful because the dynamic grid design ensures that different symbols are fetched to the nodes anytime the password system is initiated. Therefore, any smudges left on the screen from previous attempts (resembling edges connecting nodes of the password) cannot be used for inference, since the password's elements appear in different positions every time the scheme is launched. The proposed scheme also restricts the feasibility of successful shoulder surfing attacks because Bu-Dash does not use visual indicators such as an edge that connects nodes when forming the password. Previous research [35] showed that security is improved when we remove feedback lines on Android patterns, or if we incorporate "special moves" in our graphical passcodes [43]. Our scheme requires users to choose different paths to form their passcodes every single time they need to authenticate themselves. Moreover, they have plenty of choices about how to form their passcode, as we saw in Sect. 3.2 and they can use "knight moves" as well. Similar schemes that use photo gallery rearrangements in recognition-based authentication methods [63] claim that successful shoulder surfing attacks are reduced by the implementation of dynamic features. Additionally, prior research [27] showed that the use of dynamic grids advances the robustness of graphical passwords against shoulder surfing attacks.

Extending the scheme Furthermore, in Fig. 8 we propose a couple of adjustments for the Bu-Dash scheme that adhere to the APU design concepts. Fig. 8a shows the current static APU grid embedded in any Android version. Our analysis and results are based on data derived from volunteers that engaged with the Bu-Dash grid as shown in Fig. 4d. However, Fig. 8b features a more precise adaptation of the Bu-Dash scheme to the APU design concept. It would be useful as future work to see if there exists any significant implication if the latter design prevails as a preferred visual improvement.

Additionally, another path for future work would be to assess how the addition of complimentary shapes affects users' choices, usability, and security. In theory, adding more shapes as potential choices during the password formation process will strengthen security, because the password space increases dramatically, even when we are interested in only 4-node passcodes. For example, if we keep the basic Bu-Dash principles and symbols but at the same time we add 2 more shapes in the edges (as seen in Fig. 8c), we will end up with $7 \cdot 7 \cdot 7 \cdot 7 - 7 = 2,394$ unique 4-node Bu-Dash passcodes; this set is almost 4 times larger than the original Bu-dash set of 4-node passcodes. As a secondary question, it would make sense to study if it is reasonable (and therefore beneficial) to ask the user to first pick 5 shapes from a set of 7 (or more) available shapes (Fig. 8c) and then form a Bu-Dash passcode with the preferred shapes, as seen in Fig. 8d, customizing further the Bu-Dash passcode.

However, these interventions would require readjustment of the design constraints. For example, let's assume that a user visits position α (or γ, η, ι, due to symmetry) on the grid. Under the current design concept and constraints, there exist only 5 possible moves. This is why there are only 5 available symbols at the original Bu-Dash scheme, which need to be fetched randomly any time users swipe their fingers on the grid. Customizations as those seen in Fig. 8 might require alterations in the original design (therefore in the constraints) that might eventually affect usability. Similar questions, however, remain open for further research.

Ethical considerations Volunteers provided informed consent before participating in the study anonymously according to the Institution's policy for low-risk projects. No identifiable data were permanently stored and we cannot foresee any ethical issues deriving from our research, as it relates and presents a proof of concept which is not employed yet as a real authentication system on the participants' mobile devices. All volunteers were encouraged to uninstall our Bu-Dash application when they concluded their participation and the application was removed from the online store as soon as we concluded the research.

8 Conclusion

We presented a novel graphical password scheme, named Bu-Dash. This is a universal and inclusive user authentication method because it is not restricted by any language or other constraints such as educational background or technical knowledge. Users create passcodes comprising sequences of simple shapes in an intuitive manner. We conducted a series of studies asking volunteers to interact with Bu-Dash and gathered data that allow us to report a positive attitude toward adopting the scheme as a primary authentication method for mobile devices. Preliminary results demonstrate the scheme's diversity and its extended password space. The dynamic grid features randomly mapped edges that constitute the basis of the Bu-Dash scheme and ensures that the authentication process is secure and robust against shoulder surfing and smudge attacks. However, we noticed some human biases against using specific shapes (e.g., $-$) and we concluded that the users in our sample mostly preferred to start their passcodes with a certain symbol ($\times$). We showed that the latter bias does not affect significantly the password availability compared to the starting point bias in the APU scheme. Our frequency analysis showcased popular N-grams that could be used in the future to create partial blocklists, if needed. Finally, we assessed usability features and reported that the scheme is comprehensive, and usable. Furthermore, we illustrated mnemonic rules and strategies that can be employed to make them even more memorable and diverse.

To conclude, this paper demonstrated the feasibility of adopting the proposed scheme as a user authentication method that can be employed in multiple settings, ranging from smartphones to desktops, and other portable devices. Moreover, we presented open research questions stemming from the introduction of our proof of concept aiming to showcase how our scheme can be extended in the future.

Declarations

Conflict of interest The authors declare they have no conflict of interest.

Ethical approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. This UWE Bristol project has been classified as "low risk".

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Andriotis, P., Kirby, M., Takasu, A.: Bu-dash: a universal and dynamic graphical password scheme. In: Moallem, A. (ed.) HCI

for Cybersecurity, Privacy and Trust, pp. 209–227. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-05563-8_14

2. Forman, T., Aviv, A.: Double Patterns: A Usable Solution to Increase the Security of Android Unlock Patterns, ACSAC '20, pp. 219–233. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3427228.3427252
3. Markert, P., Bailey, D.V., Golla, M., Dürmuth, M., Aviv, A.J.: This pin can be easily guessed: Analyzing the security of smartphone unlock pins. In: IEEE Symposium on Security and Privacy (SP), pp. 286–303 (2020). https://doi.org/10.1109/SP40000.2020.00100
4. Mehrabi Koushki, M., Obada-Obieh, B., Huh, J.H., Beznosov, K.: Is implicit authentication on smartphones really popular? On android users' perception of "smart lock for android". In: 22nd International Conference on Human–Computer Interaction with Mobile Devices and Services, MobileHCI '20. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3379503.3403544
5. Zimmermann, V., Gerber, N.: The password is dead, long live the password: a laboratory study on user perceptions of authentication schemes. Int. J. Hum. Comput. Stud. **133**, 26–44 (2020). https://doi.org/10.1016/j.ijhcs.2019.08.006
6. Andriotis, P., Li, S., Spyridopoulos, T., Stringhini, G.: A comparative study of android users' privacy preferences under the runtime permission model. In: Tryfonas, T. (ed.), pp. 604–622. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58460-7_42
7. Wang, D., Gu, Q., Huang, X., Wang, P.: Understanding human-chosen pins: Characteristics, distribution and security. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ASIA CCS '17, pp. 372–385. Association for Computing Machinery, New York, NY, USA (2017). DOI: https://doi.org/10.1145/3052973.3053031
8. Markert, P., Bailey, D.V., Golla, M., Dürmuth, M., Aviv, A.J.: On the security of smartphone unlock pins. ACM Trans. Priv. Secur. **24**(4), 1–36 (2021). https://doi.org/10.1145/3473040
9. Seyed, T., Yang, X.-D., Tang, A., Greenberg, S., Gu, J., Zhu, B., Cao, X.: Ciphercard: a token-based approach against camera-based shoulder surfing attacks on common touchscreen devices. In: Abascal, J., Barbosa, S., Fetter, M., Gross, T., Palanque, P., Winckler, M. (eds.) Human–Computer Interaction—INTERACT 2015, pp. 436–454. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22668-2_34
10. Kim, S.-H., Kim, J.-W., Kim, S.-Y., Cho, H.-G.: A new shoulder-surfing resistant password for mobile environments. In: Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication, ICUIMC '11. Association for Computing Machinery, New York (2011). https://doi.org/10.1145/1968613.1968647
11. Gugenheimer, J., De Luca, A., Hess, H., Karg, S., Wolf, D., Rukzio, E.: Colorsnakes: using colored decoys to secure authentication in sensitive contexts. In: Proceedings of the 17th International Conference on Human–Computer Interaction with Mobile Devices and Services, MobileHCI '15, pp. 274–283. Association for Computing Machinery, New York (2015). https://doi.org/10.1145/2785830.2785834
12. von Zezschwitz, E., De Luca, A., Brunkow, B., Hussmann, H.: SwiPIN: Fast and Secure PIN-Entry on Smartphones, pp. 1403–1406. Association for Computing Machinery, New York (2015). https://doi.org/10.1145/2702123.2702212
13. Andriotis, P., Tryfonas, T., Oikonomou, G.: Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method. In: Tryfonas, T., Askoxylakis, I. (eds.) Proceedings of the 2nd International Conference on Human Aspects of Information Security, Privacy, and Trust (HAS 2014), pp. 115–126. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07620-1_11
14. Song, Y., Cho, G., Oh, S., Kim, H., Huh, J.H.: On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks, pp. 2343–2352. Association for Computing Machinery, New York (2015). https://doi.org/10.1145/2702123.2702365
15. Sun, C., Wang, Y., Zheng, J.: Dissecting pattern unlock: the effect of pattern strength meter on pattern selection. J. Inf. Secur. Appl. **19**(4), 308–320 (2014). https://doi.org/10.1016/j.jisa.2014.10.009
16. Aviv, A.J., Budzitowski, D., Kuber, R.: Is bigger better? Comparing user-generated passwords on 3×3 vs. 4×4 grid sizes for android's pattern unlock. In: Proceedings of the 31st Annual Computer Security Applications Conference. ACSAC 2015, pp. 301–310. Association for Computing Machinery, New York (2015). https://doi.org/10.1145/2818000.2818014
17. Tupsamudre, H., Banahatti, V., Lodha, S., Vyas, K.: Pass-o: A proposal to improve the security of pattern unlock scheme. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '17, pp. 400–407. Association for Computing Machinery, New York (2017). https://doi.org/10.1145/3052973.3053041
18. Cho, G., Huh, J.H., Cho, J., Oh, S., Song, Y., Kim, H.: Syspal: system-guided pattern locks for android. In: IEEE Symposium on Security and Privacy (SP), pp. 338–356 (2017). https://doi.org/10.1109/SP.2017.61
19. Krombholz, K., Hupperich, T., Holz, T.: Use the force: evaluating force-sensitive authentication for mobile devices. In: Twelfth Symposium on Usable Privacy and Security (SOUPS 2016), pp. 207–219. USENIX Association, Denver (2016)
20. Meng, Z., Kong, J., Li, J.: Utilizing binary code to improve usability of pressure-based authentication. Comput. Secur. **103**, 102187 (2021). https://doi.org/10.1016/j.cose.2021.102187
21. Samuel, R., Markert, P., Aviv, A.J., Neamtiu, I.: Knock, knock. Who's there? On the security of LG's knock codes. In: Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020), pp. 37–59. USENIX Association, Virtual (2020)
22. Aviv, A.J., Gibson, K., Mossop, E., Blaze, M., Smith, J.M.: Smudge attacks on smartphone touch screens. In: Proceedings of the 4th USENIX Conference on Offensive Technologies, WOOT'10, pp. 1–7. USENIX Association (2010)
23. Andriotis, P., Tryfonas, T., Oikonomou, G., Yildiz, C.: A pilot study on the security of pattern screen-lock methods and soft side channel attacks. In: Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '13, pp. 1–6. ACM, New York (2013). https://doi.org/10.1145/2462096.2462098
24. Khan, H., Hengartner, U., Vogel, D.: Evaluating Attack and Defense Strategies for Smartphone PIN Shoulder Surfing, pp. 1–10. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3173574.3173738
25. Chen, Y., Sundaram, H.: Estimating complexity of 2d shapes. In: IEEE 7th Workshop on Multimedia Signal Processing, pp. 1–4 (2005). https://doi.org/10.1109/MMSP.2005.248668
26. Dai, L., Zhang, K., Zheng, X.S., Martin, R.R., Li, Y., Yu, J.: Visual complexity of shapes: a hierarchical perceptual learning model. Vis. Comput. **38**, 419–432 (2021). https://doi.org/10.1007/s00371-020-02023-z
27. Lin, D., Dunphy, P., Olivier, P., Yan, J.: Graphical passwords and qualitative spatial relations. In: Proceedings of the 3rd Symposium on Usable Privacy and Security, SOUPS '07. Association for Computing Machinery, New York (2007). https://doi.org/10.1145/1280680.1280708
28. De Angeli, A., Coventry, L., Johnson, G., Renaud, K.: Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems. Int. J. Hum. Comput. Stud. **63**(1), 128–152 (2005). https://doi.org/10.1016/j.ijhcs.2005.04.020

29. von Zezschwitz, E., Dunphy, P., De Luca, A.: Patterns in the wild: a field study of the usability of pattern and pin-based authentication on mobile devices. In: Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '13, pp. 261–270. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2493190.2493231
30. Andriotis, P., Oikonomou, G., Mylonas, A., Tryfonas, T.: A study on usability and security features of the android pattern lock screen. Inf. Comput. Secur. **24**(1), 53–72 (2016). https://doi.org/10.1108/ICS-01-2015-0001
31. Uellenbeck, S., Dürmuth, M., Wolf, C., Holz, T.: Quantifying the security of graphical passwords: the case of android unlock patterns. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS '13, pp. 161–172. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2508859.2516700
32. Loge, M., Duermuth, M., Rostad, L.: On user choice for android unlock patterns. In: European Workshop on Usable Security, EuroUSEC, vol. 16 (2016). https://doi.org/10.14722/eurousec.2016.23001
33. Cha, S., Kwag, S., Kim, H., Huh, J.H.: Boosting the guessing attack performance on android lock patterns with smudge attacks. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '17, pp. 313–326. Association for Computing Machinery, New York (2017). https://doi.org/10.1145/3052973.3052989
34. Aviv, A.J., Sapp, B., Blaze, M., Smith, J.M.: Practicality of accelerometer side channels on smartphones. In: Proceedings of the 28th Annual Computer Security Applications Conference, ACSAC '12, pp. 41–50. Association for Computing Machinery, New York (2012). https://doi.org/10.1145/2420950.2420957
35. Aviv, A.J., Davin, J.T., Wolf, F., Kuber, R.: Towards baselines for shoulder surfing on mobile authentication. In: Proceedings of the 33rd Annual Computer Security Applications Conference, ACSAC 2017, pp. 486–498. Association for Computing Machinery, New York (2017). https://doi.org/10.1145/3134600.3134609
36. Schaub, F., Deyhle, R., Weber, M.: Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In: Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia, MUM '12. Association for Computing Machinery, New York (2012). https://doi.org/10.1145/2406367.2406384
37. Ye, G., Tang, Z., Fang, D., Chen, X., Kim, K.I., Taylor, B., Wang, Z.: Cracking android pattern lock in five attempts. In: Proceedings of the 2017 Network and Distributed System Security Symposium 2017 (NDSS 17) (2017). https://doi.org/10.14722/ndss.2017.23130
38. Ye, G., Tang, Z., Fang, D., Chen, X., Wolff, W., Aviv, A.J., Wang, Z.: A video-based attack for android pattern lock. ACM Trans. Priv. Secur. **21**(4), 1–31 (2018). https://doi.org/10.1145/3230740
39. Kwon, T., Na, S.: TinyLock: affordable defense against smudge attacks on smartphone pattern lock systems. Comput. Secur. **42**, 137–150 (2014). https://doi.org/10.1016/j.cose.2013.12.001
40. Schneegass, S., Steimle, F., Bulling, A., Alt, F., Schmidt, A.: Smudgesafe: geometric image transformations for smudge-resistant user authentication. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '14, pp. 775–786. Association for Computing Machinery, New York (2014). https://doi.org/10.1145/2632048.2636090
41. Kabir, M.M., Hasan, N., Tahmid, M.K.H., Ovi, T.A., Rozario, V.S.: Enhancing smartphone lock security using vibration enabled randomly positioned numbers. In: Proceedings of the International Conference on Computing Advancements, ICCA 2020. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3377049.3377099
42. De Luca, A., Harbach, M., von Zezschwitz, E., Maurer, M.-E., Slawik, B.E., Hussmann, H., Smith, M.: Now you see me, now you don't: protecting smartphone authentication from shoulder surfers. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '14, pp. 2937–2946. Association for Computing Machinery, New York (2014). https://doi.org/10.1145/2556288.2557097
43. von Zezschwitz, E., De Luca, A., Janssen, P., Hussmann, H.: Easy to Draw, but Hard to Trace? On the Observability of Grid-Based (Un)Lock Patterns, pp. 2339–2342. Association for Computing Machinery, New York (2015). https://doi.org/10.1145/2702123.2702202
44. Golla, M., Rimkus, J., Aviv, A.J., Dürmuth, M.: On the in-accuracy and influence of android pattern strength meters. In: Workshop on Usable Security, USEC, vol. 19 (2019). https://doi.org/10.14722/usec.2019.23025
45. von Zezschwitz, E., Eiband, M., Buschek, D., Oberhuber, S., De Luca, A., Alt, F., Hussmann, H.: On quantifying the effective password space of grid-based unlock gestures. In: Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia, MUM '16, pp. 201–212. Association for Computing Machinery, New York (2016). https://doi.org/10.1145/3012709.3012729
46. Tupsamudre, H., Vaddepalli, S., Banahatti, V., Lodha, S.: TinPal: an enhanced interface for pattern locks. In: Workshop on Usable Security, USEC, vol. 18 (2018). https://doi.org/10.14722/usec.2018.23021
47. Munyendo, C.W., Grant, M., Philipp Markert, P., Forman, T.J., Aviv, A.J.: Using a blocklist to improve the security of user selection of android patterns. In: Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021), pp. 37–56. USENIX Association, Virtual (2021)
48. Vaddepalli, S., Nivas, S., Chettoor Jayakrishnan, G., Sirigireddy, G., Banahatti, V., Lodha, S.: Passo—new circular patter lock scheme evaluation. In: 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '20. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3406324.3417167
49. Chen, Y.-L., Ku, W.-C., Yeh, Y.-C., Liao, D.-M.: A simple text-based shoulder surfing resistant graphical password scheme. In: International Symposium on Next-Generation Electronics, pp. 161–164 (2013). https://doi.org/10.1109/ISNE.2013.6512317
50. Winkler, C., Gugenheimer, J., De Luca, A., Haas, G., Speidel, P., Dobbelstein, D., Rukzio, E.: Glass Unlock: Enhancing Security of Smartphone Unlocking Through Leveraging a Private Near-Eye Display, pp. 1407–1410. Association for Computing Machinery, New York (2015). https://doi.org/10.1145/2702123.2702316
51. Ku, W.-C., Liao, D.-M., Chang, C.-J., Qiu, P.-J.: An enhanced capture attacks resistant text-based graphical password scheme. In: IEEE/CIC International Conference on Communications in China (ICCC), pp. 204–208 (2014). https://doi.org/10.1109/ICCChina.2014.7008272
52. Li, W., Wang, Y., Li, J., Xiang, Y.: Toward supervised shape-based behavioral authentication on smartphones. J. Inf. Secur. Appl. **55**, 102591 (2020). https://doi.org/10.1016/j.jisa.2020.102591
53. Takada, T.: fakePointer: an authentication scheme for improving security against peeping attacks using video cameras. In: The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, pp. 395–400 (2008). https://doi.org/10.1109/UBICOMM.2008.76
54. Lee, M.-K.: Security notions and advanced method for human shoulder-surfing resistant pin-entry. IEEE Trans. Inf. Forensics Secur. **9**(4), 695–708 (2014). https://doi.org/10.1109/TIFS.2014.2307671

55. De Luca, A., Hertzschuch, K., Hussmann, H.: ColorPIN: Securing PIN Entry through Indirect Input, pp. 1103–1106. Association for Computing Machinery, New York (2010). https://doi.org/10.1145/1753326.1753490
56. van Eekelen, W.A.J., van den Elst, J., Khan, V.-J.: Picassopass: a password scheme using a dynamically layered combination of graphical elements. In: Extended Abstracts on Human Factors in Computing Systems, CHI EA ’13, pp. 1857–1862. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2468356.2468689
57. Bianchi, A., Oakley, I.: Multiplexed input to protect against casual observers. In: Proceedings of HCI Korea, HCIK ’15, pp. 7–11. Hanbit Media, Inc., Seoul (2014)
58. Kwon, T., Na, S.: SteganoPIN: two-faced human-machine interface for practical enforcement of pin entry security. IEEE Trans. Hum. Mach. Syst. **46**(1), 143–150 (2016). https://doi.org/10.1109/THMS.2015.2454498
59. Kwon, T., Na, S.: Switchpin: Securing smartphone pin entry with switchable keypads. In: IEEE International Conference on Consumer Electronics (ICCE), pp. 23–24 (2014). https://doi.org/10.1109/ICCE.2014.6775892
60. Lothaire, M.: Combinatorics on Words, vol. 17. Cambridge University Press, Cambridge (1997)
61. Marçais, G., Kingsford, C.: A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics **27**(6), 764–770 (2011). https://doi.org/10.1093/bioinformatics/btr011
62. Bonneau, J.: The science of guessing: analyzing an anonymized corpus of 70 million passwords. In: IEEE Symposium on Security and Privacy, pp. 538–552 (2012). https://doi.org/10.1109/SP.2012.49
63. Lapin, K., Šiurkus, M.: Balancing usability and security of graphical passwords. In: Biele, C., Kacprzyk, J., Kopeć, W., Owsiński, J.W., Romanowski, A., Sikorski, M. (eds.) Digital Interaction and Machine Intelligence, pp. 153–160. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-11432-8_15