
SAARLAND UNIVERSITY

Faculty of Mathematics and Computer Science
Department of Computer Science
BACHELOR THESIS



EyeLogin - Calibration-free Authentication Method For Public Displays Using Eye Gaze

submitted by
Omair Bhatti
Saarbrücken
February 2020

Advisor:

Michael Johannes Barz, M.Sc.
German Research Center for Artificial Intelligence
Saarland Informatics Campus
Saarbrücken, Germany

Dr. Daniel Sonntag
Prof. Dr. Antonio Krüger

Saarland University
Faculty MI – Mathematics and Computer Science
Department of Computer Science
Campus - Building E1.1
66123 Saarbrücken
Germany

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass die vorliegende Arbeit mit der elektronischen Version übereinstimmt.

Statement in Lieu of an Oath

I hereby confirm the congruence of the contents of the printed data and the electronic version of the thesis.

Saarbrücken,
(Datum / Date)

.....
(Unterschrift / Signature)

Acknowledgements

I want to thank my advisor Michael Barz for his tremendous support and assistance. His office door was always open when I had a question or faced any problem during the creation of the thesis. Moreover, I would like to express my deepest gratitude to Dr. Daniel Sonntag for providing me the opportunity to write this thesis at the German Research Center for Artificial Intelligence in Saarbrücken.

Furthermore, I want to express my appreciation to Prof. Dr. Antonio Krüger for his valuable feedback and for being the second reviewer of this thesis.

I especially want to thank my friends and family for supporting me during the creation of this thesis. And last but not least, special thanks to all the participants of the study.

Abstract

Recently, the usage of interactive public displays has increased, including the number of sensitive applications and, hence, the demand for user authentication methods. In this context, gaze-based authentication was shown to be effective and more secure but significantly slower than touch- or gesture-based methods. We implement a calibration-free and fast authentication method for situated displays based on saccadic eye movements. In a user study ($n = 10$), we compare our new method with *CueAuth* from Khamis et al. [20], an authentication method based on smooth pursuit eye movements. The results show a significant improvement in accuracy of 13.16% to 95.88%. At the same time, we found that the entry speed can be increased enormously with our method, on average, 18.28s down to 5.12s, which is comparable to touch-based input.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 2 | Related Work | 3 |
| 2.1 | Traditional Pin Input & Security on public displays | 3 |
| 2.1.1 | Attacks using residues | 3 |
| 2.1.2 | Attacks using observation | 5 |
| 2.2 | Eye-tracking based authentication | 7 |
| 2.2.1 | Accurate gaze-based authentication | 7 |
| 2.2.2 | Calibration-free authentication | 9 |
| 3 | Gaze-based Authentication | 16 |
| 3.1 | CueAuth | 17 |
| 3.2 | EyeLogin | 18 |
| 3.3 | Implementation | 19 |
| 3.4 | Visual Debugger | 20 |
| 4 | User Study | 22 |
| 4.1 | Participants | 22 |
| 4.2 | Conditions & Tasks | 22 |
| 4.3 | Design | 23 |
| 4.4 | Procedure | 23 |
| 4.5 | Apparatus | 24 |
| 4.6 | Dependent and Independent Variables | 24 |
| 4.7 | Hypotheses | 25 |
| 4.8 | Limitation | 25 |
| 4.9 | Results | 25 |
| 4.9.1 | Accuracy | 26 |
| 4.9.2 | Entry Time | 27 |
| 4.9.3 | Perceived Workload | 28 |
| 4.9.4 | SUS | 28 |
| 4.9.5 | Qualitative Feedback | 29 |
| 4.10 | Discussion | 30 |
| 4.11 | Limitations & Future Work | 31 |

| | | |
|----------|--------------------------------|-----------|
| 4.11.1 | Midas Touch Problem | 31 |
| 4.11.2 | Input Correction | 32 |
| 4.11.3 | Common Error Types | 32 |
| 4.11.4 | Camera-based Attacks | 34 |
| 4.11.5 | Interface design | 34 |
| 5 | Conclusion | 35 |
| | Literaturverzeichnis | 36 |

Chapter 1

Introduction

An increasing amount of use cases require an authentication of users prior to interaction with a public display. Typical situations include making purchases, retrieving sensitive information, or for identification purposes. A common class of authentication methods are knowledge-based approaches, e.g., PIN, password, or patterns are entered via touch-input on the public screen or via an external keyboard. However, these methods are prone to residues- or observation-based attacks. Smudges [3] or thermal residues [1] can be used to retrieve partial to full information of an entered PIN or password. Attackers might also observe the input by shoulder surfing attacks [10] or more sophisticated attacks involving a camera recording the PIN entry [39]. An alternative to knowledge-based approaches are biometric authentication methods such as fingerprint and iris scans. However, in this work we concentrate on knowledge-based and gaze-based authentication in the context of mobile gaze-based interaction [7].

Prior works propose authentication mechanisms based on different modalities to be more robust against attackers. For example, [21] use the pressure-signal of touch-based input to overcome shoulder surfing. Other works introduce gaze-based methods that track the user's gaze with RGB cameras or specialized eye-tracking sensors. These methods were shown to be more secure than, e.g., touch-based input [14, 20]. In addition, gaze-based input allows hands-free authentication and interaction, which is more hygienic. This is an important factor, due to the high contamination of public displays [17]. However, a major drawback of many gaze-based authentication methods is the need for prior calibration [22, 8]. These approaches are not suited for public displays, because calibration takes time and is perceived as cumbersome [27], while interaction methods for public displays shall be designed for immediate usability [19]. Calibration-free methods exist, but tend to be slow [20, 13, 11, 32] or suffer from low success rates for PIN entry [15, 20].

In this work, we describe and implement a novel gaze-based and calibration-free authentication method, *EyeLogin*, that addresses the limitations of prior approaches. Our system uses the direction of saccadic eye movements in a radial interaction design, similar to [8], that facilitates accurate and fast PIN entry. We use a low-cost remote eye-tracking sensor that allows broad integration into public displays and spontaneous user interaction. Other than the approach described in [8], our method is calibration-free. In addition, we implemented the state-of-the-art method *CueAuth*¹, that is described in [20], as a baseline system. In a user study ($n = 10$), we compare both authentication methods in terms of the accuracy in entering the correct PIN, the PIN entry time, the usability, and the perceived workload.

¹In *CueAuth* [20], three authentication methods are described based on touch, gesture and gaze input, respectively. We refer to the gaze-based version unless stated otherwise.

Chapter 2

Related Work

2.1 Traditional Pin Input & Security on public displays

Conventional authentication methods on public displays are knowledge-based, e.g., PIN, password, or patterns are entered via touch-input on the public screen or via an external keyboard. In the following chapter, we present multiple attacks against these authentication methods.

2.1.1 Attacks using residues

When interacting with a touchscreen or keypad residues are left of the screen. There are multiple ways of retrieving PIN or Pattern Lock on touch devices.

Smudges

In Aviv et al. [3], the authors presented an attack to retrieve the pattern using "smudges." These "smudges" are oily residues left from touches on the screen. They found that using these residues allow for partial and, in some cases, full PIN recovery.

Their experiment setup consisted of an attacker, who as a premise for the experiment had seized the smartphone of the victim. To obtain control of the phone secured with a Pattern Lock, the attacker has to determine the victim's pattern using lighting and fitting camera angles. Because of no prior studies about the properties of smudges on the screens, the authors had to figure out the best camera and lighting settings for a successful attack.



Figure 2.1: Smudges left on a touchscreen[3].

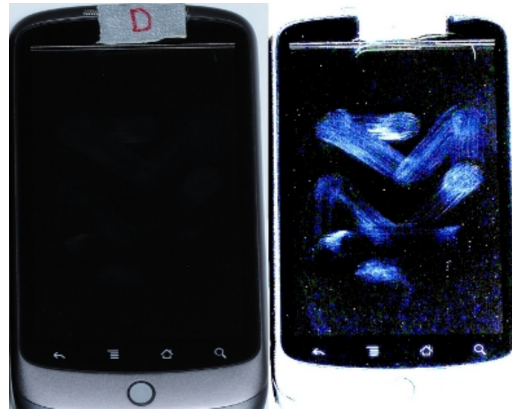


Figure 2.2: Original and post processed image side by side[3].

Furthermore, experiments investigating the negative effects of screen contact with clothes or normal usage were conducted. Remarkably, full retrieval was possible, even when the screen came in contact with clothes. However, the directionality of the pattern was lost. But the pattern possibilities are reduced to two - forward and reverse direction. Additionally, even after day-to-day usage only slightly affected the successful pattern retrieval rate.

Authentication on situated displays in public space with a Pattern Lock is even stronger affected by this attack. No seizing of any device is required as the attack can be directly executed after the user authenticates on the public display. The attacker only needs to wait for the user to authenticate and apply the attack, as mentioned in [3]. Similar investigations by Airowaily and Alrubaian [2] also confirmed security threat of smudges.

Thermal residues

Other residues based attack uses heat traces left on the screen[29, 1]. Abdelrahman et al. [1] introduce a thermal attack using heat traces on touch devices. After authentication, heat traces left on the screen can be captured and allow reconstruction of Patterns and PINs. Besides, this type of attack can even retrieve information about the order of PIN/Pattern Entry.

Thermal traces from the touchscreen fade away slowly. Attacking the user after authentication is possible and does not require the seizing of the mobile device as a picture with a thermal camera is enough. The attacker only has to wait for the possibility to take a photo with a thermal camera. This attack requires the victim to leave his smartphone unattended for a short period.

The authors conducted a study on how the different authentication methods are affected by different timing between the attack and password entry. And

additionally, how varying password properties affect the feasibility of thermal attacks.

In their study, 18 participants were asked to enter a set of PINs and patterns. Additionally, they were instructed to leave the smartphone on the table after the authentication. The attacks were conducted in 15 seconds intervals ranging from 0-60 seconds after the PIN/Pattern entry. The result showed that retrieving PINs within 30 seconds is very likely (78%), and PINs with duplicate digits are even more likely to be retrieved.

The authors propose to use authentication schemes untied to the touchscreen as an effective way to counteract thermal attacks. Smudges and thermal traces are left when a user authenticates on a public display. The attacker only needs a thermal picture after the authentication process, which is easily retrievable in a public space.

Residue-based attacks are highly effective on touch-based authentication on public displays. To prevent this attack, our approach is gaze-based. Thus, no residues are left on the screen when using a gaze-based authentication, making our approach resistant to this attack.

2.1.2 Attacks using observation

Besides using residues, an attacker can use observation during the authentication process. This "shoulder surfing" is a widely known attack (97%, 168 of 174 participants)[34, 16]. Ye et al. [39] introduced an advanced observational attack. This attack is video-based and uses computer vision algorithms tracking the victim's fingertip movement. In contrast to attacks using residues, it can be executed from a further distance ranging up to nine (and possibly more) meters. Their attack is focused on reconstructing a Pattern Lock on mobile phones. Unlike other observational attacks, no screen footage is required.

The first step for a successful attack is recording the victim drawing his pattern(See Figure 2.1.2). Depending on the camera quality, higher distance attacks are possible. While smartphone cameras allow a distance of 2-3 meters, a digital reflex camera allows for an attack from 9 meters and more. Afterward, the start and endpoint of the authentication process have to be marked in the footage. Additionally, the fingertip and a fixpoint on the edge of the smartphone have to be manually selected. The author's TLD (Tracking-Learning-Detection) algorithm tracks the fingertip and the fixpoint. The trajectory generated from the fingertip could be affected by camera shake. To cancel the camera shake, the fingertip location is tracked with respect to the fixpoint of the device. This also counters slight movements of the device.

Furthermore, the filming angle has to be considered, so each frame of the generated trajectory needs to be transformed by the corresponding filming angle in the frame. The angle is calculated between a vertical line and the detected longer edge of the phone. An automatic process maps the results to a few possible

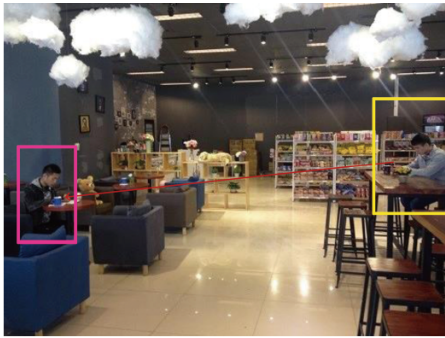


Figure 2.3: Video recording from a 2.5 meter distance[39].



Figure 2.4: The device screen as seen in the captured video[39].

patterns rejecting patterns not meeting the requirements (number of lines, length of lines, direction). Moreover, these patterns are sorted by heuristics, which takes multiple factors into account e.g., that a pattern used is most-likely starting from the left side of the phone, which is proven by multiple studies [26, 37].

To confirm the feasibility of their attack, an experiment was conducted with 10 students reproducing 120 unique pre-collected patterns from a survey with 215 participants. The students were recorded, drawing a pattern with different angles, shakiness, lighting, and different devices. The results show that over 90% of the patterns could be retrieved in five or fewer attempts. Moreover, as opposed to common belief, the more complex a pattern was, the easier it was to retrieve it successfully.

For situated public displays, observation-based attacks are more effective. In most cases, these displays are stationary. Additionally, their location is publicly accessible, which enables the attacker to prepare his attack. Hidden cameras could be placed, or high-quality cameras could be used to record footage of the authentication process from a distance. With the previously introduced attacks, the camera does not need to face the screen, making attacks even harder to detect due to more possible attack angles.

Our approach is not dependent on the finger movement but eye movement. It is much harder to track eye movement from a distance or a camera angle, not facing the eyes of a victim. Consequently making observation-based attacks much harder to achieve. Additionally, to tracking eye movements, the display has to be recorded to link the eye movement to the authentication process, making our approach robust against these attacks.

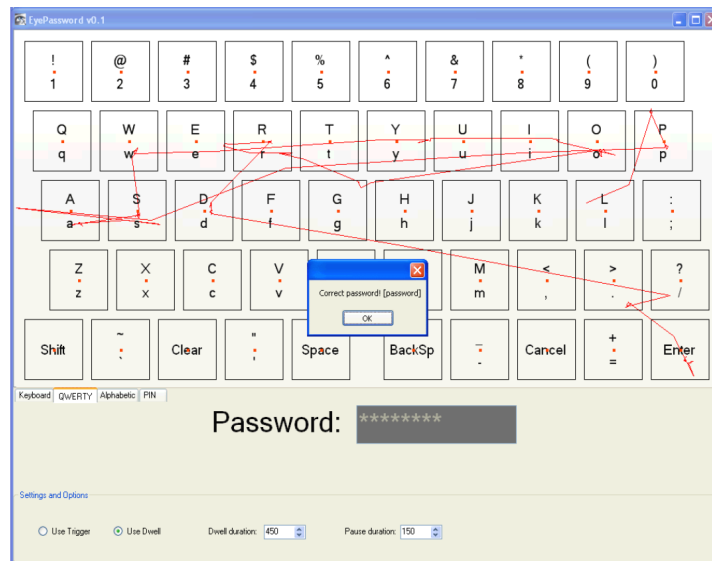


Figure 2.5: Gaze-pattern when the user enters "password" as the password[22].

2.2 Eye-tracking based authentication

In this section, we describe related works in the field of gaze-based authentication methods. Prior gaze-based authentication methods can be classified into "accurate gaze-based" and "gesture-based" methods. Accurate gaze-based authentication requires prior calibration and uses accurate gaze data. While gesture-based authentication systems make use of eye movement patterns that can be recognized automatically. These movements can be a combination of fixations, saccades, and smooth pursuit eye movements.

2.2.1 Accurate gaze-based authentication

The Midas-touch problem with eye-tracking is the problem of distinguishing between gaze gathering visual information and gaze used for selection. Research suggests using "dwell-time," where the users have to fixate on a element until a certain time is reached. The system then interprets the fixation as selection. If the dwell-time is not reached, the system interprets the gaze as just gathering visual information. In the following, we describe a "dwell-based" authentication method that uses accurate gaze data.

Dwell-based authentication

Kumar et al. [22] introduced "EyePassword," a dwell-based authentication method that displays a virtual keyboard or number pad for entering a password or a PIN, respectively. Instead of using fingers to press a button on a touchscreen, the user

aims his gaze at the characters.

There are some drawbacks when using gaze, which they had to consider when creating the layout. E.g. The size of the virtual buttons had to be big enough for the user to be accurate and small enough for the user not to move his head. Head movements could leak information to a potential attacker.

Furthermore, they had to decide between a dwell-based trigger mechanism [28], or a multi-modal solution (dedicated button), which potentially could leak timing information. Finally, to validate the user's input, audio or visual feedback has to be provided (e.g., asterisks in a password field, "beep" sound).

EyePassword was developed on Windows and used a Tobii 1750 eye tracker. They implemented a QWERTY and alphabetic layout. In a study with 18 subjects, the authors compared traditional keyboard password input with EyePassword using a hardware-based or dwell-based trigger. First, the subjects performed a calibration with the eye tracker. After a short practice session for each technique, the subjects had to enter a set of passwords.

The study compares the password entry speed and error rate. Finally, the participants were asked to fill out a survey about their subjective opinion of the techniques. The results showed that the least errors were recorded with dwell-based gaze input - even traditional keyboard input had a higher error rate. Also, the QWERTY layout was faster than the alphabetical. This could be explained with the participant's familiarity with the QWERTY layout. Although EyePassword was significantly slower (about 5-times) to keyboard input, the evaluation of the survey, showed that over 80% of the users would prefer to use "...a gaze-based approach over using a keyboard when entering their password in a public place" [22, p.17].

In their study, they confirm the feasibility of gaze-based authentication. Their paper gives clues about design decisions when creating a dwell-based authentication method. The size of the buttons has to be large enough to be accurately detected and should contain a fixed point in the middle. This point allows the user a more relaxed focus on the desired character.

Boundary-based authentication

Instead of using a dwell time for selection, Best and Duchowski [8] introduced a selection relying on the gaze data crossing boundaries of the digit elements on the screen (Figure 2.6). They implement a weighted voting scheme to determine which digit was chosen by the user. To minimize the entry time, they designed the interface in a rotary design, imitating a rotary dial phone. To enter a PIN, the user starts in the center and looks at the digit he wants to enter. By processing the gaze data, they recognize boundary entering and leaving and thus can identify a digit entry. They found out that they could minimize the entry time by deploying a rotary design and showed that an average entry time under five seconds is possible ($M = 4.62$). Drawbacks are that their system has low accuracy

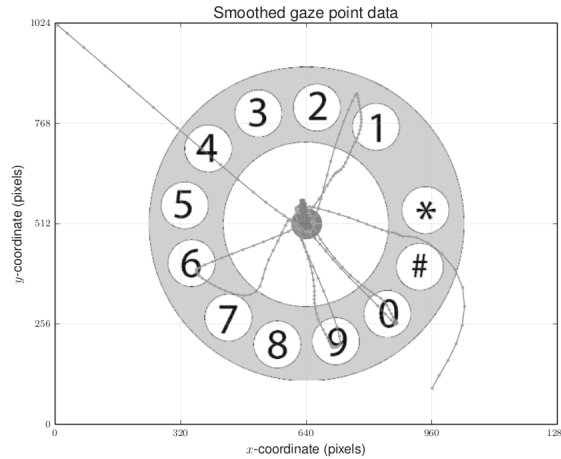


Figure 2.6: Rotarty Interface and visual representation of entering a the PIN: "9610"[8].

($M=71.16\%$), and their method is not implemented for real-time authentication.

While the rotary interface layout is similar to our approach, a significant distinction is that their gaze-based authentication systems require accurate gaze data. Thus a calibration is necessary before each use. Accurate gaze-based authentication methods are not suited for public displays, because calibration takes time and is perceived as cumbersome [27], while interaction methods for public displays should be designed for immediate usability [19]. Our approach does not need any calibration from the user. We only rely on the relative movement hence providing faster and more satisfying user experience.

2.2.2 Calibration-free authentication

Calibration-free eye-tracking methods address the problem of time-consuming and tedious calibration procedures, which is of particular interest for spontaneous interaction with public displays [19]. One approach to realize calibration-free gaze-based interaction is using gaze gestures.

Gaze-gesture based authentication

Instead of using the accurate gaze position, it is possible to use relative eye movements. These movements do not need any calibration nor expensive eye trackers. Each movement can be described as a gaze-gesture.

De Luca et al. [12] introduced an authentication method using only gaze-gestures. They designed a gesture alphabet representing digits from 0-9. The outlines of the respective numeric character represent each character. The user presses a button to trigger his input. While holding the button, he performs a gaze gesture

for each number in his PIN.

The prototype uses a commercial eye tracker ERICA, and similar to EyePassword, asterisks give the feedback for valid input. To confirm the utility, a study with 21 participants is conducted. Their prototype is compared with a numeric implementation of EyePassword, where the first implementation uses a dwell-time and the second using a button as a trigger.

They found similar results with the previously described authentication method [22] confirming the usability of eye-tracking for PIN entry, yet had a much higher error rate for dwell-based and trigger-based input in comparison. Surprisingly eye gesture input was least prone to false recognition. However, with an average time of 54 seconds, due to having to look up the "alphabet" for each numeric character and gaze gestures being something new for the users, it is not applicable for daily-usage. What the authors observed and confirmed was that users preferred and found it easier to remember a gesture instead of a PIN. So instead of deploying an "alphabet," as the authors did, it would be better to have a custom gesture for each user.

Since multiple people use public displays, therefore it makes sense to use calibration-free methods for authentication. This prototype introduces a calibration-free method, which lacks usability — input time, and the hassle to learn an alphabet before use are the most significant factors. The lack of the necessity of holding a button to indicate input is a different from our approach. Furthermore, we deliver a much faster entry time. Yet the authors could confirm that usage of gaze gestures can be robust against errors.

Thus using the conclusions from their previous work De Luca et al. [15] implemented a novel entry method called EyePassShapes. It combines an existing authentication method, "PassShapes," with Eye-Tracking. PassShapes allows the user to authenticate by painting combinations of strokes with a pen. Eight directions, each representing a different character, are used to create a password. The user selects a password and only has to remember the direction of the strokes. Using simple directions improves memorability. As this method does not use Eye-Tracking but drawing of shape, the same attacks as traditional PIN Input are effective against it.

EyePassShapes combined gaze-based authentication with PassShapes, allowing each stroke to be selected by the respective gaze-gesture. In their prototype, the user presses a button and then enters his shape with his eye movement, which is then analyzed by the system, and if recognized, access is granted.

The interface background design is either dotted(See Figure 2.7) or a grid since informal evaluations showed that only an advanced user could enter gestures on a blank screen. To acquire the most useful settings, a user study was conducted showing a dotted background was preferred by the user and had the best average input time.

After implementing these findings, the usability and security of their final pro-

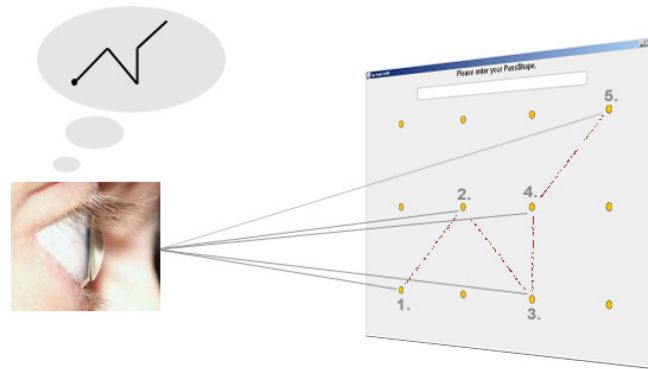


Figure 2.7: EyePassShapes: a user performing the gesture "93U9".[15].

prototype of EyePassShapes was evaluated and compared to traditional and the previously implemented PIN entry methods in two more studies. Traditional PIN entry with a keyboard was the fastest method, yet EyePassShapes could slightly outperform PassShapes, but only when the user performed their pattern in one stroke. The results show that with eye-tracking, the user could "draw" in a time comparable to drawing on a touchpad with a pencil.

Allowing the user to enter his shape with one stroke, yields a much faster password entry time (5.31s). The main difference to our approach is the usage of directions instead of a traditional numerical PIN. The authors that gaze-based authentication can be fast by using simple directions. Our approach also uses simple directional eye-movement, since they yield a faster PIN entry. The author did not publish exact accuracy values. Furthermore only critical errors (PIN entered wrong for more than 3 times) were counted as errors.

Smooth-pursuit based authentication

Whereas in gaze-based authentication, the user actively performs a gesture to interact with the system, another approach to facilitate calibration-free gaze input is based on smooth pursuit eye movements. These eye movements are correlated with trajectories of elements in a dynamic user interface for, e.g., selecting objects.

Liu et al. [25] introduced an android based authentication method using smooth pursuits. It is based on making the user track the corresponding moving object to his PIN. Four randomly sorted objects(See Figure 2.8), each assigned with a unique number between 1-4, are placed in the middle of the screen. For five rounds, those objects simultaneously move from the middle of the screen to the edge of the screen in a vertical or horizontal direction. They are then placed back on their original position. Each round, which represents a single digit of the user's PIN, the direction the user moves his eyes, is tracked with the front camera of the smartphone. If the movement matches the PIN, the authentication passes.

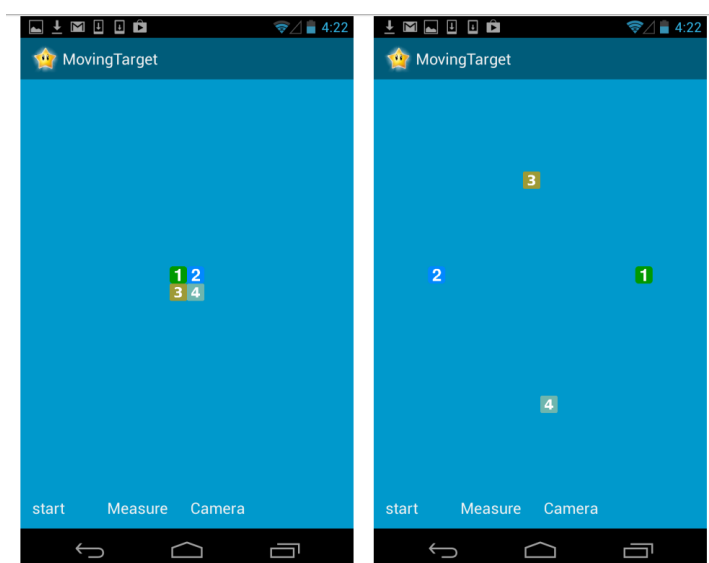


Figure 2.8: Four randomly sorted objects representing a number between 1-4 and each moving in random direction per round. [25].

They divided their system into a front-end and a back-end, whereas the front-end is responsible for presenting the objects and randomizing them for each authentication process. The back-end records the user's eye movement in each round with the front camera and then proceeds to extract the position of the eye frame-by-frame. Resulting in an eye movement trajectory, which is then matched to the targets objects movement trajectory in the "Decision Maker." The similarity of both trajectories is compared, and after five rounds, the final result is presented to the user. To increase security, not only are the objects randomly placed at the beginning of the authentication process, but also, no feedback to the user is given until five rounds are completed.

In an early prototype, the authors observed that a user's eye is not able to strictly follow a moving object due to either distraction of other moving objects or hand and head shakiness. Also, eye-tracking is not perfect. This led them to use a majority vote principle, where only 4 of the 5 rounds needed to be matching.

In an experimental setup, their android prototype was tested with 21 participants. The authors observed a 77.1% indoors to 79,3% outdoor accuracy rate but especially found out that in most cases, only one round was falsely detected. After using the Majority Vote, the detection rate rose between 91.6% to 97.3%.

This reduces the number of possibilities even more and makes it prone to brute-force attacks. Nonetheless, smooth-pursuits allow a calibration-free method for authentication. On situated public displays, the camera and the display are fixed, which drastically increases the accuracy since no shaking or handheld movement disturbs the eye-tracking. A possible solution would be to increase the possible directions and thus also the possible PIN combination, which could allow for a

relatively fast and accurate authentication method.

More recently, Khamis et al. [20] proposed a more secure authentication method utilizing smooth pursuit for situated displays. Their concept is to present a Numpad on display. Each digit on the Numpad is encapsulated and has a unique trajectory. The selection is made by following the digit's trajectory. This movement is then compared and matched. Multiple rounds are matched, and the corresponding PIN is identified. The trajectories used are either linear, circular, or zigzag-shaped.

This type of authentication is also classified as cue-based. The fundamental idea is to flood the screen with randomized cues. The attacker now has to track the cues but also the user's interaction with the cues. Making it hard for the attacker to follow and seize relevant information. Additionally, two extra cue-based authentication methods, using mid-air gestures and touch, are implemented. Both methods display a Numpad on the screen. Whereas the touch-based displays the clues as swipe direction and two different colors, each for the respective touch input area. Mid-air gesture instead uses the different colors corresponding to the left and right hand. Hence the swipe gesture has to be performed with one hand mid-air respective to the color.

They conducted a usability and security study with all three introduced authentication methods recruiting 20 participants. The participants were asked to perform the authentication methods depending on the Latin square order. Additionally, they were told to fill out a NASA TLX questionnaire to determine the workload.

All three authentication methods had a relatively high success rate ($>82\%$). Touch was less error-prone (93.38 %) in comparison to gaze (82.72%). But a significant difference was evident in the entry time. While touch took 3.73s on average, gaze was remarkably slower (26.35s). Accordingly, the NASA TLX Score gaze was experienced as less performant, more frustrating, and more physically and temporally demanding. Through a semi-structured interview, it was evident that gaze is described as secure and discrete but also as slow and causes straining eyes. Additionally, the participants proposed using different trajectories and allowing a refresh phase between each round. Also, on all three authentication methods, users asked for more visible feedback.

Furthermore, the security study showed that even with a repeated video attack, the success rate for gaze-based PIN was 0.05%, compared to 74% when using touch and 64% mid-air. This confirms the perceived security of the participants. The study proves the security of gaze-based authentication against an observation-based attack. Their approach for gaze-based authentication uses smooth Pursuits to achieve a calibration-free authentication method. But a major problem is their long entry time and perceived stressful input. This evaluation is one of our core sources during the study section, and their gaze-based scheme is implemented and then compared with our authentication method. With our approach, we aim to improve the entry time. Instead of using a predefined gesture for each digit, simply looking in the direction of the desired character

is enough. Thus reducing the workload and making the authentication process quicker and less stressful.

The presented gaze-based authentication methods are shown to be more secure than traditional PIN entry systems. However, all systems suffer from one or more major disadvantages including low accuracy in recognizing the entered PIN, high entry times or they need prior calibration (See Table 2.1).

In this work, we design and implement a novel calibration-free authentication system that addresses these flaws: Our goal is to enable gaze-based secure authentication on public displays with high accuracy and low entry times. We aim to reduce the input time and task load by introducing a more straightforward method of selecting the digit for the pin, where the user only has to direct his gaze to the desired digit for selection.

| Authors | Entry Time (avg.) | Accuracy(avg.) | Calibration | real-time | PIN |
|---------------------|-------------------|----------------|--------------|-----------|---------------|
| Best et al. 2018 | 4.62 s | 71,16 % | required | no | Numeric |
| Kumar et al. 2007 | 9.20 s | 96,00 % | required | yes | Alphanumeric |
| Liu et al. 2015 | 9.60 s | 79,30 % | not required | no | Numeric (0-4) |
| De Luca et al. 2009 | 12.52 s | n.a. | not required | yes | Gesture |
| Khamis et al. 2018 | 26.35 s | 82.72 % | not required | yes | Numeric |
| De Luca et al. 2007 | 54.00 s | 90.5 % | not required | yes | Numeric |

Table 2.1: Overview of related work. De Luca et al. 2009 did not state the accuracy of their authentication system.

Chapter 3

Gaze-based Authentication

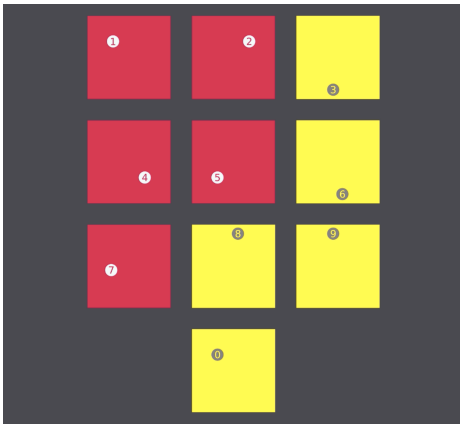


Figure 3.1: Interface of CueAuth

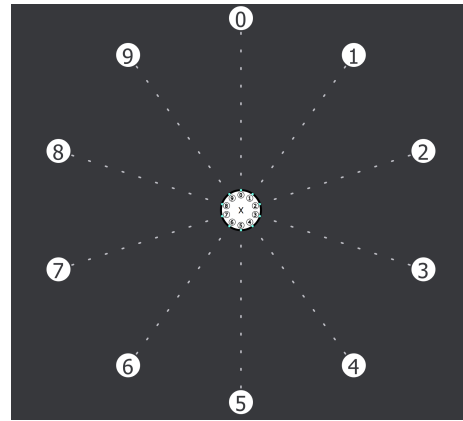


Figure 3.2: Interface of EyeLogin

In the following, we describe the design and implementation of the gaze-based *CueAuth* method [20] and our novel authentication method that is based on saccadic eye movements. We implement *CueAuth* as baseline system, because it is calibration-free, implements the same knowledge-based authentication method (four-digit PIN entry) and it is one of the most recent and comprehensive works that explores authentication on public displays.

Algorithm 1: CueAuth - PinDetection

Function *ObserveInput*

```

  while (PIN.size() < 4) do
    | OnAnimationStop += CalcDigit(gaze, trajectories);
  end
end

```

Function *CalcDigit* (*List gaze, List[] trajectories*)

```

  (corr_a, digit_a) = Correlate(gaze, trajectories, 2.0);
  (corr_b, digit_b) = Correlate(gaze, trajectories, 2.5);
  if (corr_a > 0.8 OR corr_b > 0.8) then
    | if (corr_a > corr_b) then
    | | PIN.Add(digit_a);
    | else
    | | PIN.Add(digit_b);
    | end
  end
end

```

3.1 CueAuth

We implement *CueAuth* as described in [20]. The central concept of *CueAuth* is matching smooth pursuit eye movements of a user with the trajectory of moving digits (0-9) in the interface (see Figure 3.1). The interface renders a virtual number pad. Each digit is presented in a small circle that moves with a pre-defined unique trajectory. The trajectories are either linear, circular, or zigzag-shaped, as proposed in [38].

To authenticate, the user needs to follow the movement of four digits in a row that form a PIN. For each iteration and match, the interface provides visual feedback in a separate text view by adding an asterisk symbol. Addressing the known limitations of *CueAuth*, we add a one-second break after the trajectory-based animation ends to allow the user to re-focus and to provide feedback when the matching process begins and ends. The actual matching begins after the animations of the digits stop (see Algorithm 1). We compare the trajectories of the interface controls with the relative eye movements of the same time-frame. We calculate the Pearson correlation for two axes (x and y) and average the correlation coefficients in the *Correlate* function. If the mean correlation $c \geq 0.8$, the digit is stored, and the user receives immediate feedback of the match (asterisk). If more than one trajectory reaches the detection threshold of 0.8, we choose the digit with a higher correlation. We call *Correlate* with two different time-windows: [2s-4s] and [1.5s-4s] that start 2s or 2.5s before the animation stops. The time-windows were manually optimized before the study. The digit with the highest correlation coefficient is appended to the stored PIN.

3.2 EyeLogin

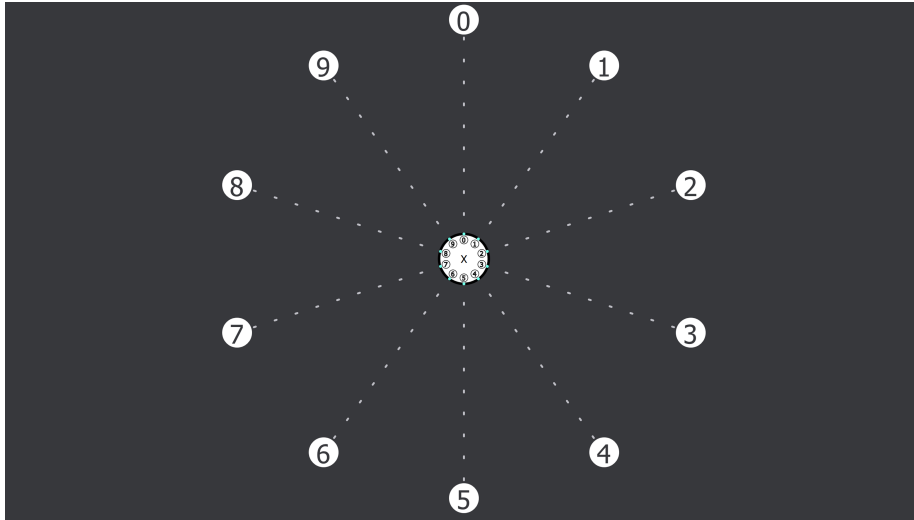


Figure 3.3: Interface of EyeLogin

We propose a novel algorithm for calibration-free authentication which is based on saccadic eye movements. *EyeLogin* shows the digits 0 to 9 in a radial design (see Figure 3.2), similar to [8]. At the center, we present feedback on the progress (one asterisk per entered digit appears) and show miniaturized digits as direction cues for the user to prevent errors. A dashed line connects the inner and outer digits to guide the user’s gaze. The user starts the authentication process by fixating at the center area and pressing the space bar. This trigger is required to overcome the Midas touch problem inherent in gaze-based interaction and could be replaced by any trigger in the future, e.g., presence detection in combination with a long fixation or speech-based hotwords known from digital assistants.

When the authentication process is started (trigger), the initial gaze position is stored as reference point $gaze_c$ and provided as input to *EyeLogin* (see Algorithm 2). The user can now enter a digit by fixating it and returning the focus to the center position afterwards, which then shows the recognition progress.

We leverage the quick nature of a saccadic eye movement between two fixations to determine the relative direction of the eye movement and to detect the digit of choice: First, we determine the farthest point max_p from the reference point $gaze_c$. Then, we calculate the direction vector dir_n with the $gaze_c$ as the origin and max_p as the destination point. The angle between the y-axis and the direction vector allows to infer the fixated digit in the `CalculateDigit` function: each digit is assigned to a certain angular sector. Upon detection, the system gives feedback by displaying an additional asterisk in the center area. Showing the feedback at the center region ensures that the user returns its focus to this point as expected by our algorithm. This process is repeated four times

to complete the PIN entry. One limitation is, that users might turn their gaze to the next digit before returning to the center area. This would cause an erroneous input. However, this error type occurred rarely in our study.

Algorithm 2: EyeLogin - PinDetection

```

Function ObserveInput (Point gazec)
  while (PIN.size() < 4) do
    if (Saccadic_Movement_Recognized()) then
      | CalcDigit(saccade, gazec);
    end
  end
end

Function CalcDigit (List saccade, Point gazec)
  | maxp = GetFarthestPoint(saccade, gazec);
  | dirn = CalculateDirection(gazec, maxp);
  | angle = CalculateAngle_To_Y_Axis(dirn);
  | digit = CalculateDigit(angle);
  | PIN.Add(digit);
end
  
```

3.3 Implementation

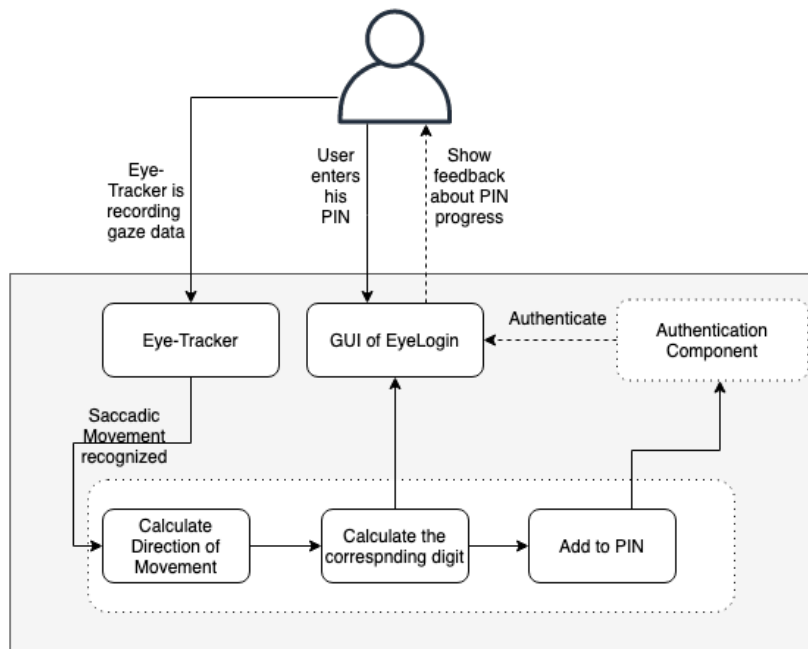


Figure 3.4: Architecture of EyeLogin

Both *EyeLogin* and *CueAuth* are implemented with C#. The Interface is implemented using WPF (Windows Presentation Foundation)². To retrieve the gaze data from the eye tracker, we use the Tobii Stream Engine(v2.2.2.363)³. Image processing is realised with EmguCV⁴.

3.4 Visual Debugger

The system stores the gaze data for *EyeLogin* into two log files. The first log file records all gaze-data during PIN entry with the timestamp provided by the Tobii Stream Engine. The second log file only contains the relevant gaze-data.

| gaze_x | gaze_y | digit_number | time |
|---------------|---------------|---------------------|-------------|
| 0.01 | 0.02 | 0 | 8575 |
| ... | ... | | ... |
| 0.03 | 0.005 | 3 | 9800 |

Table 3.1: EyeLogin .csv log file

The second log file only contains the four saccadic movements, each represented by *digit_number* (See Table 3.1). The offset *gaze_c* is taken into account before exporting to the log file. The gaze coordinates are normalized, and then the coordinate system is shifted. The *Center* coordinate is (0,0) (*Top* :(0,0.5) *Bottom* :(0,-0.5) *Left* :(-0.5,0) *Right* :(0.5,0)).

| gaze_x | gaze_y | time | digit_0_x | digit_0_y | ... | digit_9_x | digit_9_y |
|---------------|---------------|-------------|------------------|------------------|------------|------------------|------------------|
| 978 | 706 | 8575 | 185 | 719 | ... | 180 | 100 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 1538 | 1539 | 19800 | 185 | 719 | ... | 180 | 100 |

Table 3.2: CueAuth .csv log file. The gaze data is multiplied with the resolution.

CueAuth, on the other hand, saves the gaze data and additionally exports the movement of every moving digit (See Table 3.2). The gaze data and digit are represented with pixel coordinates.

²<https://docs.microsoft.com/dotnet/framework/wpf/>

³<https://www.nuget.org/packages/Tobii.StreamEngine/>

⁴<https://www.nuget.org/packages/EmguCV/>

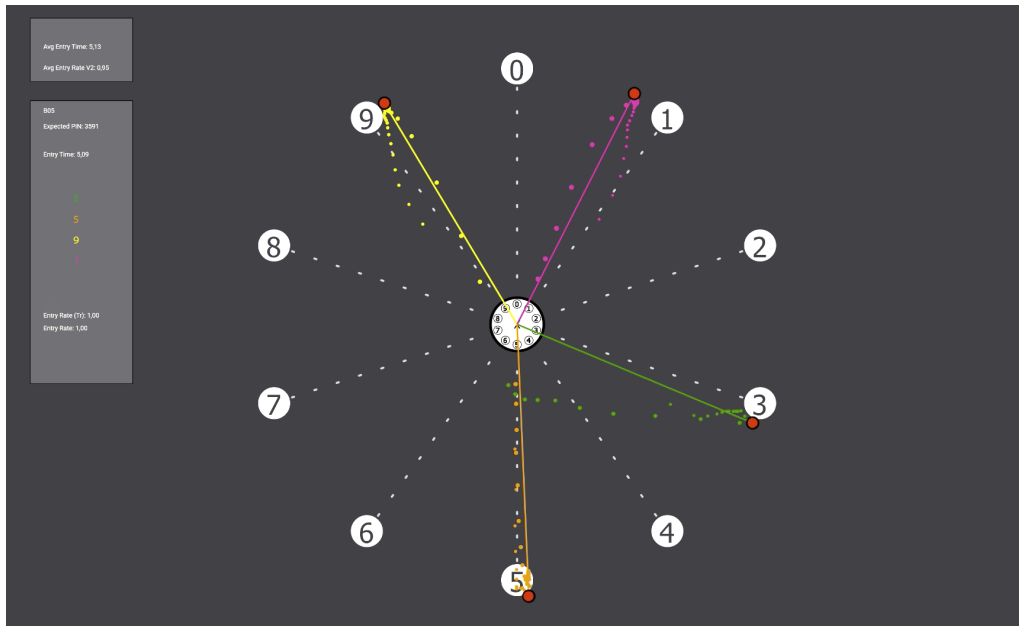


Figure 3.5: Interface of the Visual debugger for EyeLogin

By storing all the data into .csv files, we can "replay" every authentication attempt. Thus we wrote a Visual debugger, which visualizes the authentication attempt (See Figure 3.5).

The visualizer represents the gaze data for each digit with uniquely colored dots (green, orange, yellow, pink). The dot size relates to the order of gaze samples (increasing radius corresponds to increasing timestamps), and the red dot marks the point max_p from our algorithm.

On the left side of the screen, the average entry rate and time are calculated for each user. Additionally, on the top left side, the average entry time and rate for all users is calculated. To navigate between entries of the user, the button "n" for next and b for "back" are used. To change the user, "+" and "-" are used.

Chapter 4

User Study

We conduct a user study to compare our novel authentication method *EyeLogin* to the existing eye-tracking based authentication method *CueAuth*. We investigate the effectiveness, efficiency, usability, and perceived workload of both methods in a public display setting. We mostly revisit the experimental design and setting from [20] to ensure comparability with their results.

4.1 Participants

We recruited 10 students (two females and eight males) with normal or corrected to normal vision aged between 25 and 31. One participant with weak vision refrained from wearing eye correction but did not report any problems. Two of the participants had prior experience with eye-tracking.

4.2 Conditions & Tasks

We investigate the performance and usability of the previously introduced authentication methods *EyeLogin* and *CueAuth*. For each method, we ask participants to enter 11 PINs in a training phase and 17 PINs in the main phase, totaling 28 PINs per method and user. The instructor vocalizes the randomly selected four-digit PIN before the participant starts the authentication procedure by pressing the space-key. They receive automatic feedback about the progress as described above, but not whether a digit or the complete PIN was recognized correctly.

| Group | Sequence of performing the conditions |
|-------|---------------------------------------|
| A | EyeLogin CueAuth |
| B | CueAuth EyeLogin |

Table 4.1: Counterbalanced Order

4.3 Design

The study is designed as a repeated measures experiment and is conducted with the independent variable *authentication method* as the within-subject factor that includes *CueAuth* and *EyeLogin*. Since participants' performance can be influenced by conditions following other conditions, for example, to mental or physical demand from the first condition, it is crucial to counterbalance the order. Therefore we divide the participants into two groups (A and B). A will first perform the condition *EyeLogin* and then will perform the condition *CueAuth*. B will perform the condition in reverse - start with *CueAuth* and then perform *EyeLogin* (See Table 4.1).

4.4 Procedure

First, the participants are welcomed to the study and asked to sign a consent form. The instructor introduces and explains the study. The instructor presents one of the authentication methods (counterbalanced order) by demonstrating the interface, explaining how a digit is selected, and when a password entry is finished.

In the training phase (11 PINs), the participant can familiarise himself with the authentication method by entering three simple PINs, followed by eight random PINs. Each PIN will be randomly preselected. The user starts the authentication by pressing the "Space" button. He only receives feedback about each successful digit entry, but no feedback is given about the correctness of the entered PIN. If the instructor detects significant problems, the instructor corrects the user. The three first PINs are discarded for the analysis. In the main phase, the participant is asked to enter 17 PINs.

After finishing the tasks for one method, the participant fills in a NASA TLX [18] form on a computer to assess the perceived workload. This procedure is repeated with the remaining authentication method. After all tasks are completed, the participant fills in a questionnaire, including demographic questions and items of the System Usability Scale (SUS) [9] as well as open-ended questions for each authentication method.

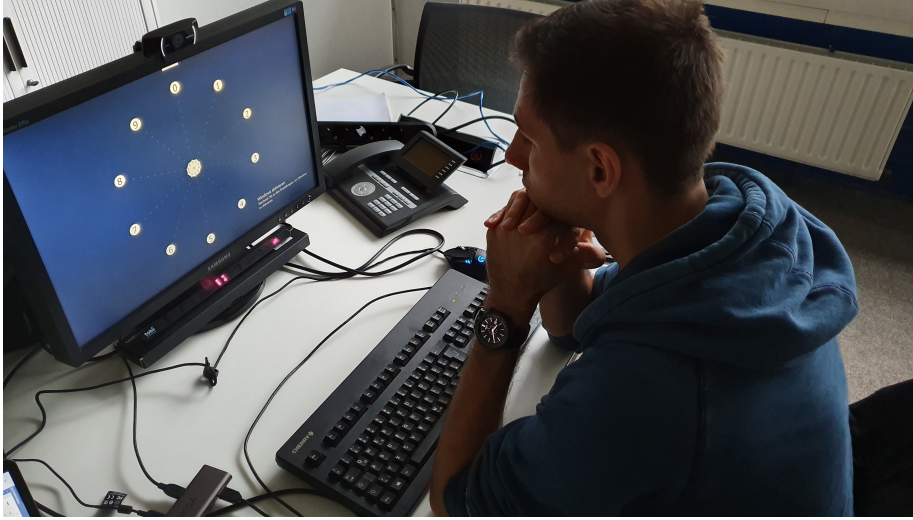


Figure 4.1: The setup of the study that is same for both methods

4.5 Apparatus

A 24-inch widescreen monitor with a resolution of 1920x1080 pixels is used to display the interfaces of the authentication methods. A webcam is placed at the top of the screen to record a video feed during PIN entry. Furthermore, We use the Tobii 4C remote eye tracker [35] with a 60Hz sampling rate, which is attached below the screen (see Figure 4.4). The eye tracker is calibrated once by the study instructor. We used the same calibration for all participants. For the study, participants are seated in front of the display with an approximate distance of 60cm. A keyboard is provided to start the authentication trials. After completing all 28 PINs, the participant fills in a NASA TLX form on the same screen. For analysis, we store the participant ID, the timestamped eye movements, and synchronized movements of all smooth pursuits stimuli in the interface for every PIN entry attempt. Furthermore, the results of the automatic PIN recognition and the correct PIN are stored.

4.6 Dependent and Independent Variables

The dependant variables are:

- PIN entry time: we measure the time from the moment the user presses the space bar until the system recognizes the fourth digit. We calculate this measure from the stored timestamps.
- PIN accuracy: We count a false entry if one or more digits are incorrect. We

calculate accuracy as the average number of correctly recognized PINs per participant.

- Usability score: System Usability Scale (SUS).
- Perceived workload: NASA TLX score.

The independent variable is the authentication method (*CueAuth* and *EyeLogin*). Both have a training phase and a main phase, which are analyzed separately.

4.7 Hypotheses

- **H1** Authentication with *EyeLogin* is more accurate than authenticating with *CueAuth*
- **H2** Authentication is faster with *EyeLogin* than authenticating with *CueAuth*
- **H3** The gains in effectiveness (accuracy) and efficiency (time) have no negative impact on the usability and the perceived workload

4.8 Limitation

As we replicate the setting of the study in [20], we face the same limitations. Repeated consecutive PIN entry is not a realistic use case and might have a negative impact on the usability and the perceived workload. However, a comparison between both methods is possible, because we test them under the same circumstances.

4.9 Results

For both methods, we observe the accuracy, the entry time, the NASA TLX and the SUS score for entering PINs. All metrics are measured for the training phase (8 PINs) and the main phase (17 PINs) per method. If not stated otherwise, we report the results of the main phase.

To test for statistical significance, we use the paired samples t-test⁵. The Shapiro-Wilk test is used to check whether the differences of the paired samples are from a normal distribution and, hence, no assumption of the dependent t-test is violated. We also checked whether the order of methods has an effect on our dependent variables, but found no significant differences using an independent t-test and the order as a between groups factor ($p > .05$).

⁵We use the 2-tailed paired samples t-test in SPSS with an alpha-level of 5%

4.9.1 Accuracy

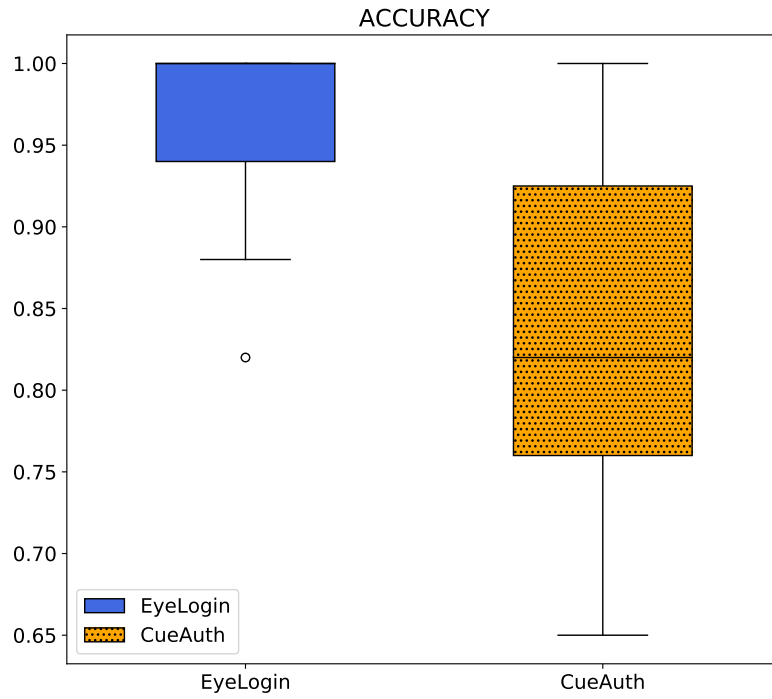


Figure 4.2: Box-plot for the PIN entry accuracy for the main phase of our study

Using our implementation of *CueAuth*, the users achieve a mean accuracy of 82.94% ($SD = 11.58$). This is close to the results from Khamis et al. [20] which reported 82.72% ($SD = 38.53$). On average, the accuracy is lower during the training phase ($M = 71.25\%$, $SD = 21.28$), but the difference is not significant ($t(9) = -1.491$, $p = .17$).

For our proposed method *EyeLogin*, we observe an accuracy of 95.88% ($SD = 6.23$), which is 12.94% better than the *CueAuth*-baseline (see Figure 4.9.1). This difference is statistically significant with $t(9) = 3.18$, $p = .012$. In addition, our gaze-based method performs better than the best method of Khamis et al. [20] that is based on touch interaction ($M = 93.38\%$, $SD = 26.05$).

Similar to *CueAuth*, the accuracy of *EyeLogin* during the training phase is 5.78% lower ($M = 90.00\%$, $SD = 18.45$). We did not test for statistical significance, because a Shapiro-Wilk test showed a significant departure from normality, $W(10) = .672$, $p < .00012$.

4.9.2 Entry Time

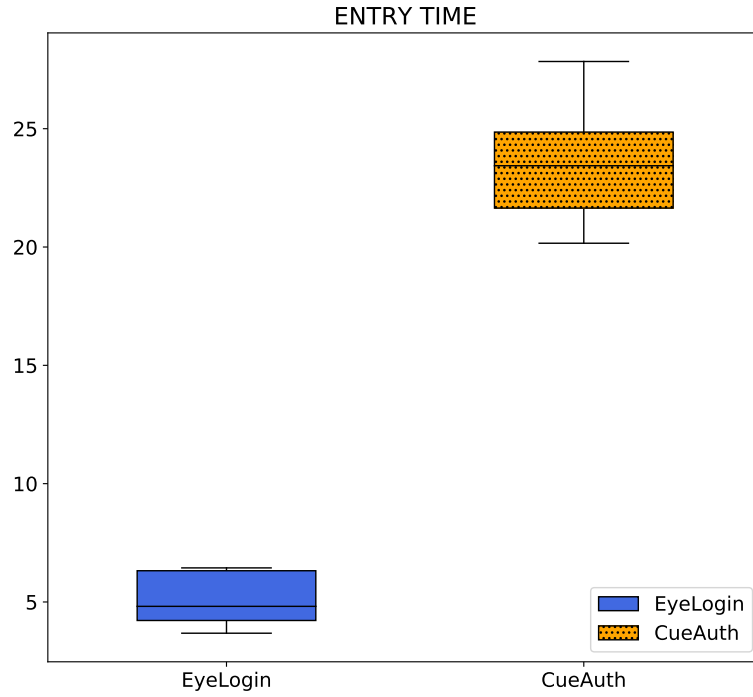


Figure 4.3: Box-plot for the PIN entry time (in seconds) for the main phase of our study

On average, we measure entry times of $23.41s$ ($SD = 2.28$) for *CueAuth*, which is similar to the result of $26.35s$ reported in the literature [20]. However, the reported standard deviation of 22.09 is much higher compared to our implementation. Using *EyeLogin*, we observe average pin entry times of $5.12s$ ($SD = 1.09$). The time saving of $18.28s$ compared to the baseline (see Figure 4.9.2) is statistically significant ($t(9) = 24.063, p < .001$). The touch-based method in [20] is reported to be the fastest and is, with an average of $3.73s$ ($SD = 1.07s$) only slightly faster than our proposed gaze-based approach. The entry times from the training phase and main phase do not differ significantly for both methods, *CueAuth* ($t(9) = -.045, p = .956$) and *EyeLogin* ($t(9) = -.301, p = .766$).

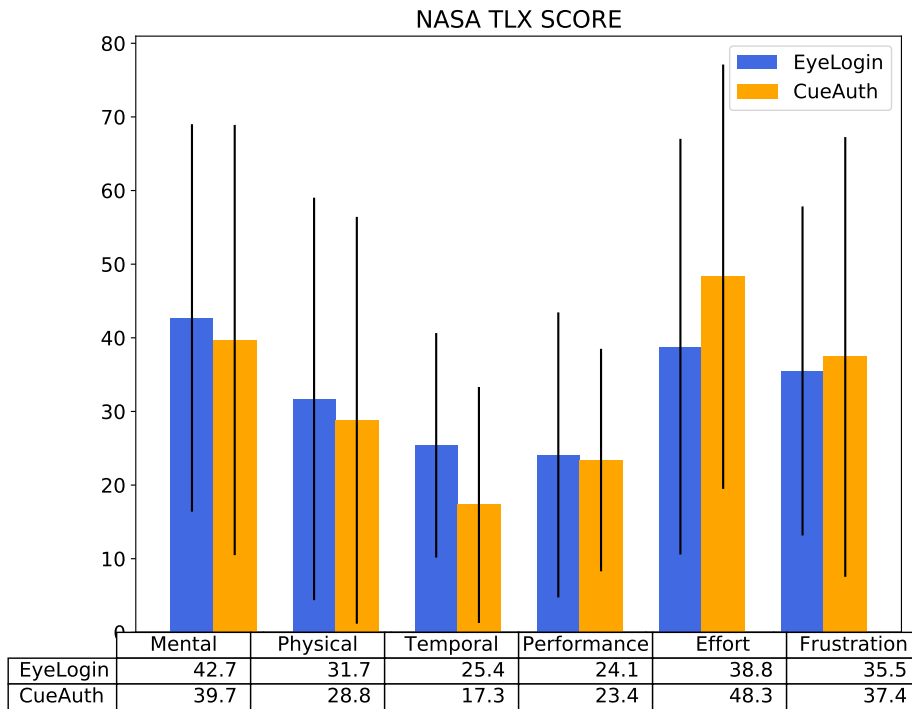


Figure 4.4: Bar chart diagram visualizing the NASA TLX results: mean \pm SD

4.9.3 Perceived Workload

We use the NASA TLX questionnaire to evaluate the perceived workload, as suggested in [20]. The mean scores for all dimensions of the test are reported in Figure 4.9.3. None of the differences are significant as determined by a paired samples t-test per dimension ($df = 9$; $p > .05$).

4.9.4 SUS

We ask the participants to fill in a SUS questionnaire, which gives us a subjective usability score, for both methods. We receive an average score of 66.5 ($SD = 18.72$) for *CueAuth* and 75.75 ($SD = 15.28$) for *EyeLogin* (higher is better). However, the difference of 9.25 points is not significant ($t(9) = -1.075$, $p = .31$).

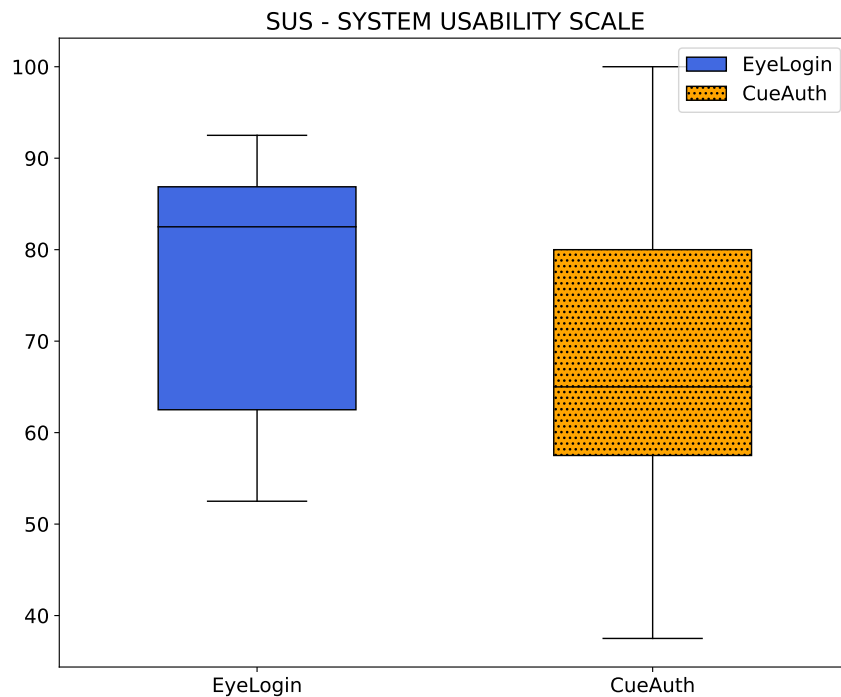


Figure 4.5: System Usability Scale

4.9.5 Qualitative Feedback

We collect qualitative feedback via open-ended questions. We ask participants to note the pros and cons of each method and to provide suggestions for improvements. Analyzing the answers, we find that *EyeLogin* is perceived as fast (7/10) and easy to use (7/10). Three participants criticize that blinks are likely to cause errors during pin entry.

For *CueAuth*, participants state as advantages that the layout is familiar (4/10) and easy to use (2/10). The participants perceive *CueAuth* as slow (7/10) and tiring (4/10). Two participants criticize that the system was not sensitive enough to recognize their input.

4.10 Discussion

The results of our evaluation show that our authentication method *EyeLogin* is significantly more accurate than the baseline system *CueAuth*. Users succeed in entering a PIN in 95.88% of all trials which is 12.94% better than the baseline and confirms H1. In addition, our gaze-based method achieved a similar accuracy than the touch-based version of *CueAuth* as reported in [20].

The PIN entry times for *EyeLogin* are tremendously lower than for *CueAuth* (significant). On average, users need 5.12s to enter a four-digit PIN which is 18.28s faster than measured for the baseline *CueAuth* (confirms H2). In addition, we measured lower PIN entry times for our implementation of *CueAuth* than Khamis et al. [20] for their implementation and our result is close to the PIN entry times of the touch-based version of [20] (3.73s). Additionally, we have a much lower deviation due to integrating a short stop after each animation, which gives the user enough time to re-focus and receive the feedback.

We used the SUS and the NASA TLX questionnaires for measuring the usability and the perceived workload of both authentication methods. The results do not reveal any significant differences between the two considered methods, which suggests that we can confirm our hypothesis H3. Further, we observe a higher average SUS for *EyeLogin* (75.75) than for *CueAuth* (66.5). This might indicate that our authentication method has better usability than the baseline system. For comparison, other works using the SUS questionnaire achieve, on average, a score of 69.5 ($n = 273$) [4].

With the qualitative feedback, we can confirm that the majority of the users perceive *EyeLogin* as fast and easy to use. In comparison, only (2/10) described *CueAuth* as easy to use. Furthermore, the majority describe *CueAuth* as slow and tiring - most likely due to the high PIN entry time. The users' feedback strengthens the indication that our authentication method has better usability in comparison to *CueAuth*.

Other promising use cases could involve head-worn augmentation devices with integrated eye-tracking. Sensitive information displayed in Augmented Reality, like video recordings, for example of episodic memory support [36], or annotation [5] of confidential objects (e.g. "Keys for Safe") need to be encrypted. Our system can provide a fast and accurate authentication method.

Additionally, head-worn eye-tracking devices [24] can be used to extend our authentication method to be applied on public displays without having to connect an eye tracker. Furthermore, a seamless authentication on different devices could be possible [23] (e.g., smartphone to public display).

4.11 Limitations & Future Work

EyeLogin enables fast and reliable input for authentication on public displays on the same level of performance than more common touch-based methods. However, a few limitations remain, including a required start trigger to overcome the Midas touch problem, some error types that cause avoidable authentication fails, and potential vulnerabilities to camera-based attacks. We address each of these limitations and provide suggestions on how they could be solved.

4.11.1 Midas Touch Problem

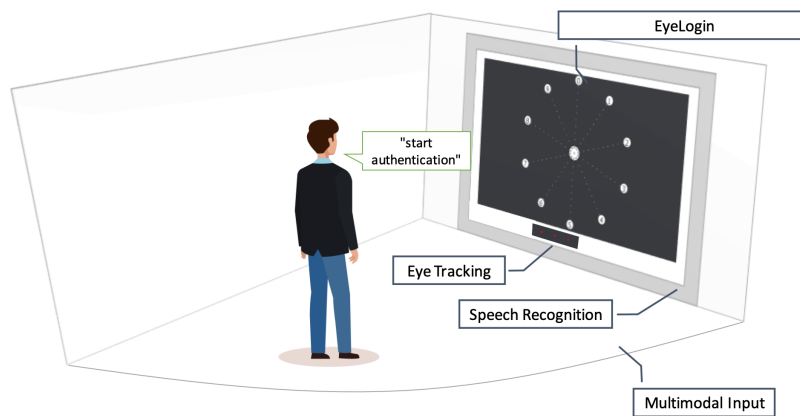


Figure 4.6: Multi-modal solution with speech-based trigger[33]

Our implementation requires the user to press the space bar to capture the reference gaze position $gaze_c$ and start the authentication process. Including an additional modality for disambiguating the gaze-based input [31, 30] is common practice.

However, on public displays, this trigger needs to be replaced by more suitable alternatives like touching a "start authentication" button on the screen. One solution can be a combination of eye-tracking with speech-based input [6]: an instruction can be shown at the central area of the user interface, asking the user to start the authentication by vocalizing a trigger word, also known as hotword (see, e.g., <https://snowboy.kitt.ai/>) (See Figure 4.6).

A pure gaze-based method can be realized as well: the instruction at the center can ask the user to fixate its area for a certain dwell-time to start the authentication. The presence of a user can be detected by the presence or absence of gaze data

from the eye-tracking sensor.

4.11.2 Input Correction

Currently, *EyeLogin* does not provide the functionality of correcting one's input. So the user has to finish entering the PIN, although he is aware that he already entered a wrong digit. To overcome this problem, our interface can be extended with two more elements in the radial design. One element contains a *revert* and one a *cancel* functionality, allowing the user to modify and correct his input actively or even cancel the whole PIN entry.

Other modalities overcoming the Midas touch problem can also be used to provide an input correction functionality - e.g., using "delete" or "cancel" hotwords via a speech-based input.

4.11.3 Common Error Types

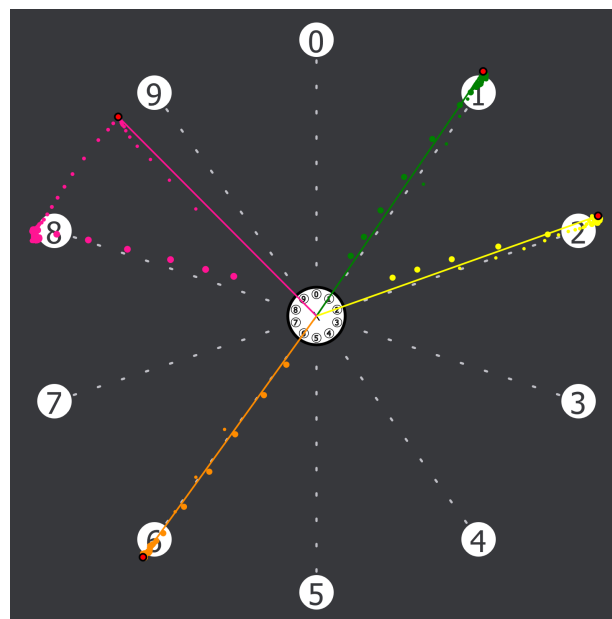


Figure 4.7: Visualization of falsely recognized PIN entry 1629

For *EyeLogin* the participant had 7 errors out of 170 PIN entries. Out of those 7 errors we observe two common error types that cause a wrong PIN entry. Figure 4.7) shows the raw gaze signal of a user that moved its gaze to the wrong digit (9), subsequently corrects the gaze position (8) and returns to the center. However, *EyeLogin* detects 9 as input resulting in a wrong digit sequence.

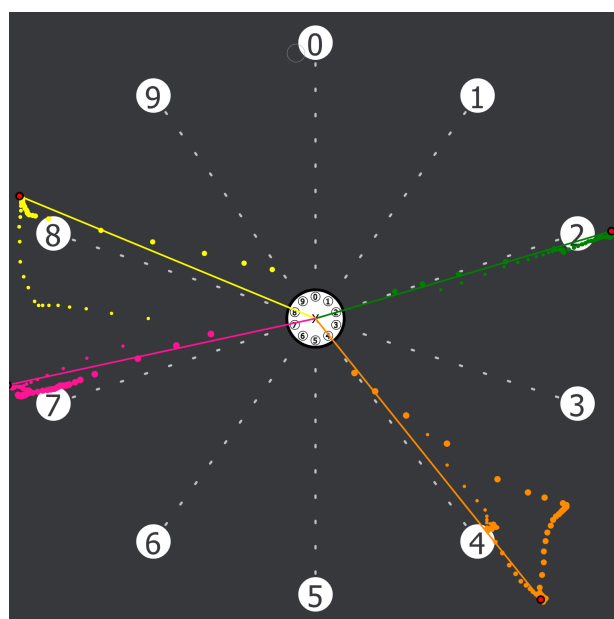


Figure 4.8: Visualization of correctly recognized PIN entry 2487

Figure 4.8 shows a similar case, but the correction (for 4 and 8) is done earlier and *EyeLogin* finally detects the correct sequence.

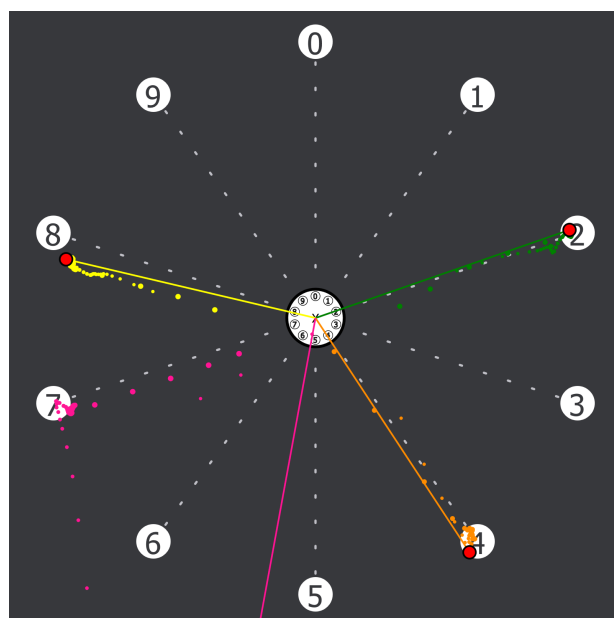


Figure 4.9: Visualization of blinking behaviour

Figure 4.9 shows a trial that failed due to a blink before fixating the last digit: 7. The blink resulted in a noisy gaze signal with distant samples, causing the algorithm to choose the wrong digit (5). A blink detection and a filtering could be applied to overcome the false recognition. Other error sources include misunderstandings with the instructor and users forgetting the PIN (3 out of 170).

4.11.4 Camera-based Attacks

EyeLogin is robust against traditional shoulder-surfing attacks, because an attacker would have to observe the display and the eyes of a user during PIN entry. However, more sophisticated attackers might attach a camera to the public display and infer the password from a video stream that captures the user's face and eye movements. One opportunity to overcome this vulnerability is to randomize the arrangement of the digits. This is perceived as more secure by all our participants. However, better security through randomly arranged digits probably needs to be traded off against usability. The video feed of the webcam can be used to do a security analysis in future work.

4.11.5 Interface design

To further reduce the PIN entry time with *EyeLogin*, the distance between the middle circle and the outer circle can be reduced. Previous work [8] indicates faster entry time with shorter saccadic eye movements. Thus future work could investigate if faster entry time is possible without loss of accuracy and usability.

Chapter 5

Conclusion

In this thesis, we presented a calibration-free and gaze-based authentication method for public displays. In a user study, we could show that our method *EyeLogin*, that leverages saccadic eye movements, performs significantly faster and significantly more accurate than *CueAuth*, a state-of-the-art gaze-based authentication system from the literature [20]. With this work, we presented the first calibration-free authentication method using gaze that is as effective and efficient than less secure input modalities such as touch- and gesture-based input.

Bibliography

- [1] Yomna Abdelrahman, Mohamed Khamis, Stefan Schneegass, and Florian Alt. Stay cool! understanding thermal attacks on mobile-based user authentication. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3751–3763, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4655-9. doi: 10.1145/3025453.3025461. URL <http://doi.acm.org/10.1145/3025453.3025461>.
- [2] Khalid Airowaily and Majed Alrubaian. Oily residuals security threat on smart phones. In *Proceedings of the 2011 First International Conference on Robot, Vision and Signal Processing*, RVSP '11, page 300–302, USA, 2011. IEEE Computer Society. ISBN 9780769545813. doi: 10.1109/RVSP.2011.92. URL <https://doi.org/10.1109/RVSP.2011.92>.
- [3] Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies*, WOOT'10, pages 1–7, Berkeley, CA, USA, 2010. USENIX Association. URL <http://dl.acm.org/citation.cfm?id=1925004.1925009>.
- [4] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *J. Usability Studies*, 4(3):114–123, May 2009. ISSN 1931-3357. URL <http://dl.acm.org/citation.cfm?id=2835587.2835589>.
- [5] Michael Barz and Daniel Sonntag. Gaze-guided object classification using deep neural networks for attention-based computing. In Paul Lukowicz, Antonio Krüger, Andreas Bulling, Youn-Kyung Lim, and Shwetak N. Patel, editors, *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp Adjunct 2016, Heidelberg, Germany September 12-16, 2016*, pages 253–256. ACM, 2016. doi: 10.1145/2968219.2971389. URL <https://doi.org/10.1145/2968219.2971389>.
- [6] Michael Barz, Peter Poller, and Daniel Sonntag. Evaluating remote and head-worn eye trackers in multi-modal speech-based hri. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '17, page 79–80, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348850. doi: 10.1145/3029798.3038367. URL <https://doi.org/10.1145/3029798.3038367>.

- [7] Michael Barz, Florian Daiber, Daniel Sonntag, and Andreas Bulling. Error-Aware Gaze-Based Interfaces for Robust Mobile Gaze Interaction. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, pages 24:1–24:10, New York, NY, USA, 2018. ACM. ISBN 9781450357067. doi: 10.1145/3204493.3204536. URL <http://doi.acm.org/10.1145/3204493.3204536>.
- [8] Darrell S. Best and Andrew T. Duchowski. A rotary dial for gaze-based pin entry. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ETRA '16, pages 69–76, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4125-7. doi: 10.1145/2857491.2857527. URL <http://doi.acm.org/10.1145/2857491.2857527>.
- [9] John Brooke. Sus-a quick and dirty usability scale. 1996.
- [10] Frederik Brudy, David Ledo, Saul Greenberg, and Andreas Butz. Is anyone looking? mitigating shoulder surfing on public displays through awareness and protection. In *Proceedings of The International Symposium on Pervasive Displays*, PerDis '14, pages 1:1–1:6, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2952-1. doi: 10.1145/2611009.2611028. URL <http://doi.acm.org/10.1145/2611009.2611028>.
- [11] Dietlind Cymek, Antje Venjakob, Stefan Ruff, Otto Lutz, Simon Hofmann, and Matthias Roetting. Entering pin codes by smooth pursuit eye movements. *Journal of Eye Movement Research*, 7:1–11, 08 2014.
- [12] Alexander De Luca, Roman Weiss, and Heiko Drewes. Evaluation of eye-gaze interaction methods for security enhanced pin-entry. In *Proceedings of the 19th Australasian Conference on Computer-Human Interaction: Entertaining User Interfaces*, OZCHI '07, page 199–202, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595938725. doi: 10.1145/1324892.1324932. URL <https://doi.org/10.1145/1324892.1324932>.
- [13] Alexander De Luca, Roman Weiss, and Heiko Drewes. Evaluation of eye-gaze interaction methods for security enhanced pin-entry. In *Proceedings of the 19th Australasian Conference on Computer-Human Interaction: Entertaining User Interfaces*, OZCHI '07, pages 199–202, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-872-5. doi: 10.1145/1324892.1324932. URL <http://doi.acm.org/10.1145/1324892.1324932>.
- [14] Alexander De Luca, Roman Weiss, Heinrich Hussmann, and Xueli An. Eyepass - eye-stroke authentication for public terminals. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 3003–3008, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-8. doi: 10.1145/1358628.1358798. URL <http://doi.acm.org/10.1145/1358628.1358798>.

- [15] Alexander De Luca, Martin Denzel, and Heinrich Hussmann. Look into my eyes!: Can you guess my password? In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pages 7:1–7:12, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-736-3. doi: 10.1145/1572532.1572542. URL <http://doi.acm.org/10.1145/1572532.1572542>.
- [16] Malin Eiband, Mohamed Khamis, Emanuel von Zezschwitz, Heinrich Hussmann, and Florian Alt. Understanding shoulder surfing in the wild: Stories from users and observers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI'17)*. ACM, New York, NY, USA, volume 11, 2017.
- [17] Charles Gerba, Adam Wuollet, Peter Raisanen, and Gerardo Lopez. Bacterial contamination of computer touch screens. *American Journal of Infection Control*, 44:358–360, 03 2016. doi: 10.1016/j.ajic.2015.10.013.
- [18] Sandra G. Hart and Lowell E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In Peter A. Hancock and Najmedin Meshkati, editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139 – 183. North-Holland, 1988. doi: [https://doi.org/10.1016/S0166-4115\(08\)62386-9](https://doi.org/10.1016/S0166-4115(08)62386-9). URL <http://www.sciencedirect.com/science/article/pii/S0166411508623869>.
- [19] Mohamed Khamis, Andreas Bulling, and Florian Alt. Tackling challenges of interactive public displays using gaze. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, UbiComp/ISWC'15 Adjunct, pages 763–766, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3575-1. doi: 10.1145/2800835.2807951. URL <http://doi.acm.org/10.1145/2800835.2807951>.
- [20] Mohamed Khamis, Ludwig Trotter, Ville Mäkelä, Emanuel von Zezschwitz, Jens Le, Andreas Bulling, and Florian Alt. Cueauth: Comparing touch, mid-air gestures, and gaze for cue-based authentication on situated displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(4), December 2018. doi: 10.1145/3287052. URL <https://doi.org/10.1145/3287052>.
- [21] David Kim, Paul Dunphy, Pam Briggs, Jonathan Hook, John W. Nicholson, James Nicholson, and Patrick Olivier. Multi-touch authentication on tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1093–1102, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753489. URL <http://doi.acm.org/10.1145/1753326.1753489>.

- [22] Manu Kumar, Tal Garfinkel, Dan Boneh, and Terry Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd Symposium on Usable Privacy and Security, SOUPS '07*, pages 13–19, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-801-5. doi: 10.1145/1280680.1280683. URL <http://doi.acm.org/10.1145/1280680.1280683>.
- [23] Christian Lander, Sven Gehring, Antonio Krüger, Sebastian Boring, and Andreas Bulling. Gazeprojector: Accurate gaze estimation and seamless gaze interaction across multiple displays. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST '15*, page 395–404, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450337793. doi: 10.1145/2807442.2807479. URL <https://doi.org/10.1145/2807442.2807479>.
- [24] Christian Lander, Markus löchtefeld, and Antonio Krüger. Heyebriid: A hybrid approach for mobile calibration-free gaze estimation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(4), January 2018. doi: 10.1145/3161166. URL <https://doi.org/10.1145/3161166>.
- [25] Dachuan Liu, Bo Dong, Xing Gao, and Haining Wang. Exploiting eye tracking for smartphone authentication. In *International Conference on Applied Cryptography and Network Security*, pages 457–477. Springer, 2015.
- [26] Marte Dybevik Løge. Tell me who you are and i will tell you your unlock pattern. Master’s thesis, NTNU, 2015.
- [27] Päivi Majaranta and Andreas Bulling. *Eye Tracking and Eye-Based Human-Computer Interaction*, pages 39–65. Springer London, London, 2014. ISBN 978-1-4471-6392-3. doi: 10.1007/978-1-4471-6392-3_3. URL https://doi.org/10.1007/978-1-4471-6392-3_3.
- [28] Päivi Majaranta, Anne Aula, and Kari-Jouko Räihä. Effects of feedback on eye typing with a short dwell time. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications, ETRA '04*, pages 139–146, New York, NY, USA, 2004. ACM. ISBN 1-58113-825-3. doi: 10.1145/968363.968390. URL <http://doi.acm.org/10.1145/968363.968390>.
- [29] Keaton Mowery, Sarah Meiklejohn, and Stefan Savage. Heat of the moment: Characterizing the efficacy of thermal camera-based attacks. In *Proceedings of the 5th USENIX Conference on Offensive Technologies, WOOT'11*, page 6, USA, 2011. USENIX Association.
- [30] Sharon. Oviatt, Björn Schuller, Philip R. Cohen, Daniel Sonntag, Gerasimos Potamianos, and Antonio Krüger, editors. *The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations*. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA, volume 1 edition, 2017. ISBN 9781970001679. doi: 10.1145/3015783. URL <https://dl.acm.org/citation.cfm?id=3015783>.

- [31] Pernilla Qvarfordt and Pernilla. Gaze-informed multimodal interaction. In *The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations*, pages 365–402. ACM, volume 1 edition, apr 2017. ISBN 978-1-97000-167-9. doi: 10.1145/3015783.3015794. URL <http://dl.acm.org/citation.cfm?id=3015794>.
- [32] Vijay Rajanna, Seth Polsley, Paul Taele, and Tracy Hammond. A gaze gesture-based user authentication system to counter shoulder-surfing attacks. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA '17*, pages 1978–1986, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4656-6. doi: 10.1145/3027063.3053070. URL <http://doi.acm.org/10.1145/3027063.3053070>.
- [33] studiogstock / Freepik. empresarios-grupo-personaje-avatar-posterior, 2020. URL <http://www.freepik.com>.
- [34] Furkan Tari, A. Ant Ozok, and Stephen H. Holden. A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In *Proceedings of the Second Symposium on Usable Privacy and Security, SOUPS '06*, pages 56–66, New York, NY, USA, 2006. ACM. ISBN 1-59593-448-0. doi: 10.1145/1143120.1143128. URL <http://doi.acm.org/10.1145/1143120.1143128>.
- [35] Tobii Gaming. Tobii eye tracker 4C, 2019. URL <https://gaming.tobii.com/tobii-eye-tracker-4c/>.
- [36] Takumi Toyama and Daniel Sonntag. Towards episodic memory support for dementia patients by recognizing objects, faces and text in eye gaze. In Steffen Hölldobler, Rafael Peñaloza, and Sebastian Rudolph, editors, *KI 2015: Advances in Artificial Intelligence*, pages 316–323, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24489-1.
- [37] Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. Quantifying the security of graphical passwords: the case of android unlock patterns. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 161–172. ACM, 2013.
- [38] Mélodie Vidal, Andreas Bulling, and Hans Gellersen. Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '13*, pages 439–448, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1770-2. doi: 10.1145/2493432.2493477. URL <http://doi.acm.org/10.1145/2493432.2493477>.
- [39] Guixin Ye, Zhanyong Tang, Dingyi Fang, Xiaojiang Chen, Kwang In Kim, Ben Taylor, and Zheng Wang. *Cracking Android pattern lock in five attempts*. 10 2016.