Path Word: A Multimodal Password Entry Method for Ad-hoc Authentication Based on Digits’ Shape and Smooth Pursuit Eye Movements

Hassoumi Almoctar, Pourang Irani, Vsevolod Peysakhovich, Christophe Hurter

To cite this version:


HAL Id: hal-01936330
https://hal-enac.archives-ouvertes.fr/hal-01936330

Submitted on 27 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
**PathWord: A Multimodal Password Entry Method for Ad-hoc Authentication Based on Digit Shape and Smooth Pursuit Eye Movements**

Hassoumi Almoctar  
École Nationale de l’Aviation Civile  
Toulouse, France  
almoctar.haiz@gmail.com

Pourang Irani  
University of Manitoba  
Winnipeg, Manitoba, Canada  
irani@cs.umanitoba.ca

Vsevolod Peysakhovich  
ISAE-SUPAERO  
Toulouse, France  
vsevolod.peysakhovich@isae-sup Aero.fr

Chrisophe Hurter  
École Nationale de l’Aviation Civile  
Toulouse, France  
christophe.hurter@enac.fr

**ABSTRACT**

We present *PathWord* (PATH passWORD), a multimodal digit entry method for ad-hoc authentication based on known digits shape and user relative eye movements. *PathWord* is a touch-free, gaze-based input modality, which attempts to decrease shoulder surfing attacks when unlocking a system using PINs. The system uses a modified web camera to detect the user’s eye. This enables suppressing direct touch, making it difficult for passer-bys to be aware of the input digits, thus reducing shoulder surfing and smudge attacks. In addition to showing high accuracy rates (Study 1: 87.1% successful entries) and strong confidentiality through detailed evaluations with 42 participants (Study 2), we demonstrate how *PathWord* considerably diminishes the potential of stolen passwords (on average 2.38% stolen passwords with *PathWord* vs. over 90% with traditional PIN screen). We show use-cases of *PathWord* and discuss its advantages over traditional input modalities. We envision *PathWord* as a method to foster confidence while unlocking a system through gaze gestures.

**CCS CONCEPTS**

- Human-centered computing → Interaction design;

**KEYWORDS**

interaction; eye tracking; touch-free interaction; smooth pursuit;

![Figure 1. PathWord overview: (Left) A user's eye, as seen through the infrared eye camera, following the red stimulus moving on the digit “3”. The pupil center implicitly draws the digit shape. (Middle Left) PathWord’s interface. On each digit, a stimulus is drawn and travels its shape. As such, in addition to remaining visual clues as in the standard interfaces, the digits serves also as trajectories for the moving stimuli. (Middle Right) Aggregated pupil center positions of 12 participants trying to select the digit “3”. (Right) A user entering a PIN in a traditional interface. Intruders can easily steal the PIN.](image)

**1 INTRODUCTION**

Authenticating using Personal Identification Numbers (PINs) or passwords has become a crucial step for interacting with electronic devices. It grants full access to computing systems which allow us to communicate, buy, send or obtain sensitive information. While manual input using side buttons or touch screen continues to be the most used technique to enter PINs, it is prone to security issues commonly known as shoulder surfing.

Entering password using eye gaze could decrease the risk of privacy infringement. The idea of communicating using eye movements appeared as early as 1983 [17]. Gaze-based interactions are often dwell-based [40] where an item should be gazed at for a certain time period before being activated. Blink-based interactions also exist [29] but excessively penalize the visual system and therefore are seldom. An interesting alternative to these interactions is based on eye gestures [22, 48]. Gaze gestures offer a natural appeal for applications in Human-Computer Interaction. Smooth pursuit eye movements have demonstrated encouraging interaction modalities, they unveil a higher degree of privacy preservation, i.e., they allow users to interact with computing machines confidently [21, 36]. As such, their investigations in passwords and PINs entry mechanism for user identification are gaining considerable attention [8]. We present *PathWord*, a multimodal digit entry method for ad-hoc authentication based on the natural digits shapes and users’ relative eye movements. *PathWord* is a gaze-based input modality along
with a back-of-device touch switch, to decrease shoulder attacks when unlocking a system using PIN-based authenticating systems. The contribution of this paper is threefold. First, we outline the implementation of our system for PIN-based authentication without changing the classical representation of the standard user interface and keeping its inner affordance. Second, we address the various technical challenges and compare the robustness of different algorithms to recognize the intended digits through gaze gestures and through a back-of-device touch gesture to delimit gaze input, and finally, we show the feasibility and the effect of shoulder surfing attacks through user-studies, quantitative and qualitative analysis.

2 RELATED WORKS

Our system builds upon prior works on (1) gaze interaction, (2) password-entry mechanisms, and (3) multimodal approaches.

2.1 Gaze-based interactions

Inserting PINs via direct finger-touch in everyday devices has remained largely unmodified. However, the increasing growth of stolen PINs during authentication, or shoulder surfing attacks [28] motivated researchers to propose different authentication schemes based on eye tracking approaches [41]. Recent developments have increasingly focused on using smooth pursuit eye movements as a solution of the Midas touch problem [23], i.e. triggering activation even when the user has no such intention. They were adopted for eye tracking systems calibration [6, 20, 35], augmented reality [15, 30], semaphoric gaze gesture through the Hololens [11] and for evaluating users cognitive workload [42]. They have also been used for input via gaze interaction on smartwatches [14]. Cymek et al. [8], in a study that inspired the present work, employed moving numbers drawn on Microsoft PowerPoint and asked users to enter PINs by following the movement of each number. However, there method required calibration, induces crossing trajectories and may give an intruder a clue to which target the user is following due to the long trajectories. De Luca et al. [10] implemented a prototype of eye-based PIN entry that uses the well-known alphabet proposed by Wobbrock et al. [45]. However, entering a digit requires the user to look and to recall the new shape of the digit printed on a paper and placed next to the monitor, causing shifts between the monitor and the paper, and adding an extra effort. Subsequently, they proposed EyePassShapes [9] to enable entering a password by painting a shape using the eye in a predefined order. However, in addition to the password, it requires remembering the shape and the order that links them. Both approaches were evaluated on large screens.

2.2 Password entry

Progress in face detection, biometrics, and fingerprint minutiae extraction changed the way users access systems. These approaches, while promising, suffer from various threats. A direct example is the iris-based biometric spoofing using printed iris pictures [19]. Follow-up studies improved the algorithm and proposed an approach to check for aliveness of the pupil [34]. Face-based authentication systems necessitate the user to take a specific head position and the detection algorithms suffer from uncertainties caused by the lighting conditions, occlusions, cameras, and head pose. Therefore, PIN entry methods remain the prevailing mean for interacting with a computing machine for authentication. However, the manual PIN entry is prone to security breach by direct observation. For example, malicious persons or cameras, carefully placed in the immediate vicinity of an Automated Teller Machine (ATM), could record the digits entered by the user [12]. Abdelrahman et al. [1] showed that it is possible to retrieve PINs entered on a tactile screen by means of thermal cameras. As thermal cameras become ubiquitous and affordable, we face a new form of threat to user privacy on mobile devices. They were able to detect heat traces of the user’s finger-touches. A similar trend is the smudge attack which allows to reconstruct PINs and patterns [2].

2.3 Multimodal approaches

Seetharama et al. [38] proposed a look-and-shoot method where the user fixates on the digit and selects it by clicking on a button, however, this approach depends strongly on the accuracy of the tracking device. Moreover, no study on the target size was performed by the author for proper activation as in [16]. They also proposed two methods which relies on dwell time to select a digit and blink activation. The disadvantage of these methods is that users may often blink unconsciously leading to spurious activation. Another disadvantage of their methods is that a user calibration is needed whenever a user wants to interact with the system. The double input modalities based on both selecting the digits with finger touch and user gaze direction proposed in GazeTouchPin [27] affords an alternative to the long-established approaches, unfortunately, the user still uses on-screen touches, leaving traces of touch input on the device, thus being vulnerable to smudge attacks.

3 IMPLEMENTATION

Pathword leverages the potential of various techniques in order to propose a robust multimodal password entry mechanism. Firstly, because it does not need the exact user’s gaze position, no calibration is required [43]. Our approach builds upon, modifies and extends dynamic [15] and static matching techniques [11, 45]. Figure 1 (Middle Left) shows PathWord’s user interface. Ten digits from (6) to (9) are sketched as in a traditional interface. A stimulus (represented by the small red circle) is drawn on each digit. Each stimulus moves following the path of the corresponding digit. As such, the digits serve as trajectories of the moving stimuli. For each digit of the PIN code, a user has to follow the red dot corresponding to the digit with his eyes. Thus, for a 4-digit code, a user has to perform 4 smooth pursuit movements. This way the positions of the digits remain unchanged in the user interface. Our approach is a clear improvement on current methods because it does not need calibration, thus does not need to use saccades (the series of fixations and saccades obtained after calibration to detect planar gaze position), conserves the traditional disposition of the digits and uses smaller trajectories making the method more robust against shoulder surfing attacks and reducing the time-on-task. In the following section, we describe the data collection and preprocessing; then we detail the selection and representation of the targets (digits) on the user interface. Thereafter, we describe the methods for similarity matching between the digit the user is intending to enter and the set of available digits (6 - 9).
3.1 Graphical representation and requirements

The digits are drawn on the user interface using simple mathematical formulas (Circles and Lines equation only). For instance, the digit 0 consists of two half circles and the digit 1 is drawn using two line-segments as shown in Figure 3. We outline the design requirements of the password entry method as follows:

R1: The approach must increase confidentiality and reduce smudge attacks. This is addressed by proposing a touch-free input modality.

R2: The approach must reduce shoulder surfing; This is achieved by using gaze and back of the device gestures. An experiment was dedicated to evaluate this requirement.

R3: The interface and the digits’ representation must not change the traditional graphical PIN digits disposition as to keep the effect of inner consciousness and acquaintance of each digit position: This is accomplished by affixing a moving stimuli on the digits instead of moving or modifying the digits’ relative positions.

R4: The approach must be used without calibration: this is attained by directly using the raw pupil center positions in the camera imaging frame and processing them.

3.2 User Input

Gaze: When interacting with PathWord, the user follows the moving stimuli with his or her eyes. The user’s relative eye movements are captured with the eye camera of the Pupil Labs eye tracker which is a modified web camera with an IR bandpass filter and a surface mounted IR LED to enable illuminating the user’s eye and capturing images within a specific range of the IR spectrum. The pupil center locations in the imaging frame are collected (Figure 2). The positions of the moving stimuli of all digits are also stored. As a result, the system collects a set of point positions for all digits \(D_{16} \in [0,9]\) and the pupil center positions points \(P\). \(P\) is compared to all digit point positions and the digit which resembles the most is selected as the intended digit. Similarities and detection criteria are explained in the following section.

Preprocessing: The raw pupil data is cleaned before searching for similarities. Outliers were removed using DBSCAN [13] in order to group together data points that are closed in the 2D space and remove the points distant from the actual observations. Afterward, we used an adaptive aspect ratio normalization to reduce the shapes variations [33] and keep the data values between 0 and 1. Normalization of dimension is considered to be an efficient preprocessing technique [18]. An inappropriate normalization results in distortion and loss of information [33]. The aspect ratio used can be described as \(r = \min(W, H)/\max(W, H)\), such that \(r < 1\). Thereafter, the data was smoothed using \(\alpha\) Filter [5], a first-order low-pass filter with an adaptive cutoff frequency. The smoothing process allowed capturing the important shape variations while reducing points that are likely to appear in the data due to measurement errors.

Back-of-device interaction: While in the standard technique, the system is aware of the PIN entry by sensing the user’s touch at a specific position of the screen, we may consider an on-screen space (e.g., a rectangle located at the right bottom of the interface) where a user can tap and hold to enable the user input (or data collection). However, in such case, an intruder can be aware that the user is now entering a PIN, thus focusing on his eyes more attentively. This led us to implement a triggering mechanism for starting the data collection, that limits the intruder’s inspection from most common viewpoints. PathWord transforms the built-in rear camera into a touch-sensing tool for data collection. This input modality has been shown to provide an efficient interaction that circumvent fat-finger issues [3, 39, 47]. However, in our work, it is used for data collection solely. Conceptually, when interacting with the system, the user could subtly cover the camera with her finger and then follow the moving stimuli with the eyes. An analysis (Fisher’s exact test) of the primary outcome results of a pilot study indicates that using the back camera results in fewer intruder’s awareness of the PIN entry moment compared to using a button on the screen with a prevalence of 83.33\% (\(p = 0.015\)).

3.3 Selecting trajectories length

Whole trajectories (Full length): The simple mathematical formula used to draw the digits allows obtaining a unique shape representation of every digit. The variabilities of the digits shape allow dissociating the raw digits formed by the unnoticeable small pupil movements. However, the moving stimuli must cover the whole trajectory to keep this dissociation. Consequently, the entry time is highly dependent on the trajectories length. The time required to enter a digit is \(\approx 2.3\) seconds on average for the whole length trajectories’ interface, (Figure 3 - Right top). This led us to try to reduce the trajectories length while keeping the variability. As such, we adapted a simulated annealing algorithm for this purpose.

Simulated Annealing trajectories (Reduced length): A non-negligible aspect of interacting with PathWord in real-time is that...
the rapidity of the method depends on the trajectories length. The longer they are, the more time it will take to recognize each digit. We optimized our method by searching the paths that best reduce the trajectories length while keeping the variabilities between them. To do this, we used simulated annealing to find the best solution [31] as shown in Figure 3. The method is a metaheuristic technique which consists on searching reduced parts of the digits that best differentiate them. The reduction of the trajectories length will allow minimizing the time required to enter a digit. We considered the whole trajectories length as the initial state, and the objective of the simulated annealing was to find a state where the variability \( V(s) \) among the trajectories is kept or maximized while the length of the trajectories \( L(s) \) are minimized. At each step of the process, the simulated annealing examines a neighboring state \( S_{i+1} \) of the current state \( S_i \), and a probabilistic approach decides whether the system should move to state \( S_{i+1} \) or examines another neighbor. The approach is iterated until a good solution is found and simulated annealing ensures avoiding local optimum. After obtaining the results of the method, the time required to enter a digit is \( \approx 1.5 \) seconds on average for the reduced length trajectories’ interface (Figure 3 - Right top).

![Figure 3](image)

**Figure 3.** (Left) PathWord user interface with stimuli moving on each digit. (Right Top) The trajectories followed by the stimuli for Whole Trajectories case. (Right Bottom) The trajectories followed by the stimuli for Reduced (Simulated Annealing solution) Trajectories case. Notice that the Simulated Annealing algorithm was able to obtain reduced trajectory lengths while keeping the variability between the digits for proper dissociation.

### 3.4 Detection algorithms

**RV coefficient for Correlation:** Our method was inspired by recent studies. We modified and extended the dynamic approach used in SmoothMoves [15] and Pursuits [43]. In their approaches, correlations between the eye and the target data for the x-axis and the y-axis are calculated using Pearson’s product-moment correlation coefficient. However, due to the assumptions of the data before using Pearson correlation, more precisely the normality of the data, we could not use this method with the solution given by the trajectories of the simulated annealing solution. For example, the path on the digit’s \( \overline{1} \) is a straight vertical line-segment (Figure 3 bottom right). While the correlation could be computed on the y-axis, the values on the x-axis are constant, making the correlation calculation impossible vertically. This statement is sustained by Pearson’s correlation formula shown below:

\[
 r = \frac{\mathbb{E}(\text{Eye}_x - \text{Eye}_x)\text{(Target}_x - \text{Target}_x)}{\sigma_{\text{Eye}_x} \sigma_{\text{Target}_x}}
\]

Where \( \text{Target}_x \) is the mean and \( \sigma_{\text{Target}_x} \) is the standard deviation which in the aforementioned case (shape represented by the digit \( \overline{1} \)) is equal to zero.

This brings us to use a variation of the correlation method. To the best of our knowledge, no previous studies have considered this issue. We used the RV coefficient [37], a multivariate generalization of the squared Pearson correlation coefficient. It measures the closeness of two sets of points that may each be represented in a matrix (2-dimensional matrix in our case). A unique correlation computation suffices and takes into account both x and y data in the calculation of the coefficient. The positions of the dot all along with the positions of the pupil centers feed the RV coefficient algorithm which returns the strength of the linear association between the sets of the recorded data.

**Shape Matching: 2D:** To evaluate the static matching between shapes, PathWord leverages techniques used in the 1S recognizer algorithm [46], EdgeWrite [45] and Moment Invariants [4]. More concretely, shapes are considered to be similar is the sum of the distance between the points representing them is the lowest. We included the preprocessing steps and modified the rotation invariance angle selection to fit our needs. Dynamic Time Warping was used to compute the distances between digits and eye points.

**Character recognition:** When the pupil data is in the matching reference (Figure 2-1e), it can be considered to be a unique character, that is, character recognition can be used to find similarities. We considered the Levenshtein distance which has been proposed as a recognizer based on shape similarity. Different features are extracted from each shapes, and each feature is replaced by a unique character, thus a shape can be represented by a sequence of character. Thereafter, the similarity is evaluated using the Levenshtein distance metric [7].

**Combined coefficient:** While the RV coefficient, the modified 1S recognizer, and the Levenshtein distance metrics produce different measures separately, they can be prone to error and thus give misleading matches. Thus, we explore the use of an aggregated coefficient based on the three precedent measures. In other words, a digit is selected only if results of two of the three algorithms give the same digit.

### 4 EXPERIMENT 1: EVALUATION OF THE RECOGNITION ALGORITHMS

Our evaluation proceeded similar to Delamare et al. [11] and Esteves et al. [14]. The first experiment aimed at assessing the consistency and accuracy of our approach based on the number of corrected digit intended to be entered by the users and the effective entered digits. The experiment had four goals. First, we explored the accuracy and robustness of the aforementioned algorithms for PIN entry context.
Secondly, we investigated the effect of the digits’ trajectory lengths based on the number of corrected digits detected by the algorithms. This serves to better understand the impact of using the whole digits trajectories or using reduced trajectories. Third, we inspected three different speeds based on the algorithms recognition rate, in order to understand how the speed levels affect the accuracy of the algorithms. Finally, we verified the effect of speed and trajectory length on the number of undetected digits and the number of false activations.

4.1 Participants and Apparatus

12 participants (5 females) took part in the experiment. Their age varied between 17 and 32 (M=28.5, SD=4.7) and they were active mobile phone users. Two participants were wearing glasses. Five participants had already used an eye-tracking system.

A C# desktop software was built for the experiment, running on a XPS 15 9530 Dell Laptop 64 bits1, 16 GB of Random Access Memory. We used a 24 inches Dell 2408WFP screen2 which has a resolution of 1920x1200 pixels and 24 milliseconds response time. The interface was shown on a desktop application (as to simulate an ATM). Participants were sitting at ~75 cm from the screen.

4.2 Design

The experiment was conducted in a controlled lab setting at our institution. We ensured that the same setup was applied to all participants. The experimental room has no window, the light was controlled. Only one eye camera 3 obtained from the head-worn Pupil Labs Eye tracker [24] was used. The method doesn’t necessitate a world camera because only raw pupil center positions were recorded. No calibration was done for the experiment. The data collection was processed at 114 frames per second on average, the remaining 6 frames were lost due to the processing of the pupil detection algorithm. The gaze position accuracy is 0.4° according to the manufacturer. We arranged a scenario in which a random digit is provided by the system and we asked the participant to enter the selected digit by following the red stimulus moving on its shape.

The study was structured as a 4 × 2 × 3 repeated measures within-subjects design wherein each participant completed all the conditions. We investigated the following independent variables:

- **Algorithm**: We explored the accuracy of the RV coefficient, IS recognizer, Levenshtein distance algorithm and the accuracy of the combination of the three precedent algorithms.
- **Path**: Two different path lengths were tested. The whole path scenarios in which each moving target travels the whole trajectory, and the scenario wherein the moving target follows the reduced trajectory obtained from the simulated annealing solution.
- **Moving target Speed**: We considered three levels of speed. The targets were moving at slow speed (6°, 7.8 cm, 294.80 pix)/s, medium (10°/s, 13.22 cm, 499.65 pixel/s) and fast (14°/s, 18.69 cm, 706.39 pixel/s) speeds.

The order of the condition was counterbalanced using a partial Latin-square design. The dependent variables used in this experiment were the number of corrected digits guessed by the algorithms, the number of incorrect digits (false activations) and the rate of undetected digits. We asked each participant to enter ten digits. Consequently, each participant performed 2 Path Lengths × 3 Moving target speeds × 10 digits = 60 trials. The same data were used to compute the coefficient of the four different algorithms corresponding to 4 × 60 trials = 240 records for each participant, for a total of 240 × 12 participants = 2880 records.

4.3 Procedure

Upon arrival, the participant was handed a demographic questionnaire and asked to sign a consent form. The global procedure was explained and written instructions were given to the participant. When he finished reading the instructions, the task began. The experimenter launched the desktop application on which the moving stimuli followed their respective path as explained in section 3. Before the first trial, the participant was instructed to take a moment to get familiar with the moving stimuli paths and the interaction. When ready, the data collection started. The experiment facilitator pressed the keyboard space bar to collect the eye movements data. After the data is collected, an asterisk symbol is drawn as a visual feedback. We asked the participant to take a pause after every three consecutive trials to reduce fatigue effects. The participant performed 60 trials (2 Path Lengths × 3 Moving target speeds × 10 digits to select). The experiment lasted around 35 minutes for each participant. The algorithms check if the data is correctly collected and provide the guessed digits for further analysis. In addition to the guessed digit, the recognition coefficients for each algorithm along with the x and y eye and digit’s coordinates are stored. The order of each trial is chosen randomly in order to minimize learning effects. After the trials were completed, the participant filled a NASA Task Load Index questionnaire for qualitative evaluation.

4.4 Results and Discussion

We performed a factorial ANOVA test with Greenhouse Geisser and Bonferroni correction. We computed the detection rates of all algorithms and we tested for effect on these metrics. Data from 6 trials were removed because of the bad pupil center tracking quality from a participant who wore eye mascara. The removed records concerned the entries of the digit 0 and 5 for the third-speed scenario. A main significant effect of Moving dot speed on detected digits was found (F(2,9)=12.26, p < .001). Post-hoc analysis showed that the comparison between level of speeds were significantly different (p < .0001). There was no statistically significant difference in mean detected digits between the whole and reduced Path (F(1,9)=.32, p = .56) (Figure 4). No speed × Path interaction was found (p = .38). Descriptive statistics (Figure 5a) showed that on average, the four algorithms together have detected 87.1% correct digit entries using the 6°/s velocity (V3), 74% correct digit entries using the 10°/s velocity (V2) and 62% correct digit entries using the 14°/s velocity (V1) for the global trajectories scenario. For the reduced trajectories (Figure 5b), the algorithms have detected 88%, 76% and 70% correct digit entries using V3, V2 and V1 respectively.

---

1Intel(R) Core(TM) i7-4712HQ CPU @ 2.30GHz,2301 MHz, 4 core(s), 8 process
2L x W x H Dimensions: 22 x 8.17 x 15.62 inches
3Sampling rate: @120Hz, 640X480 pixels
We recruited 42 participants for this experiment (36 M) aging from 17 to 36. The participants were students and administrative assistants. 1 participant wore glasses, 2 wore contact lenses and 2 had already participated in an experiment with an eye tracking system.

5 EXPERIMENT 2: EVALUATION OF SHOULDER SURFING ATTACKS

Experiments 1 investigated the effect of moving targets’ speed and path lengths on the detection rate of the similarity algorithms. In this experiment, we evaluate PathWord against shoulder surfing attacks on a mobile device and explore to what extent attackers can steal PINs while users are authenticating. We run a pilot study, combined with methods of previous studies [25, 26] to chose numbers and relative position of attackers. We test the effect of the number of attackers on the number of digits recognized by attacker(s). We consider the adequate parameters obtained from the first experiment, namely the simulated annealing result’s path length coupled with low speed (6°/s).

5.1 Method and apparatus

The apparatus consisted of an Android mobile phone running the 7.0 (Nougat) operating system. The pupil positions were captured using a single eye camera as described in study 1. Experimental software was developed in Java with the Android SDK.

5.2 Participants

We recruited 42 participants for this experiment (36 M) aging from 17 to 36. The participants were students and administrative assistants. 1 participant wore glasses, 2 wore contact lenses and 2 had already participated in an experiment with an eye tracking system.

5.3 Design

Participants were instructed to enter a PIN by following the moving stimuli as accurately as possible. For each digit selection, the participant had to cover the back camera with her finger then follow the stimuli with her eyes. To complete a PIN entry, the participant repeated the same scenario for the four digits. Each PIN entry session was recorded with 1, 2 or 3 cameras. We explored the number of stolen PINs by elaborating three different attack configurations:

1 attacker configuration: In this scenario, one participant is asked to authenticate while a camera (simulating attacker A1) is recording the digits. The Camera A1 was placed at 1.5 m from P1 as shown in Figure 6a.

2 attackers configuration: The participant (P1) is authenticating while two Cameras (C1 and C2) are recording (Figure 6b). C1 was placed as in the scenario 1 and C2 was placed behind P1. In this scenario, we investigated cases where two persons agreed to proceed to a shoulder surfing attack to steal a PIN.

3 attackers configuration: P1 is authenticating while C1, C2, and C3 were recording (Figure 6c). C1 and C2 were placed as in the scenario 2 and C3 was placed at a position diametrically opposed to C2. This scenario is rare, but we evaluated the robustness of the method against more sophisticated shoulder surfing attacks.

Before starting, the participants were asked to sign a consent form. The experiment facilitator explained how to enter a PIN and took one minute to give examples of the back of the device and eye gesture interaction. As such, we gave participants a few minutes to gain familiarity with the system and get accustomed to the novel interactions. Afterward, the shoulder surfing attack purpose was explained to the attackers. The eye movements were recorded with a head-worn Pupil Labs Eye camera [24]. No calibration was done, only raw pupil center positions data were recorded. The study was configured as a 3x2 within-subjects factorial design. The independent variables were the attack configuration: 1, 2 or 3 attackers and the input technique: PathWord and baseline (standard PIN entry method). Participants filled NASA-TLX worksheets after the experiment.

Hypothesis and dependent variables The hypothesis was H1: Successful attacks would be lower for PathWord than for standard PIN entry method. Detection success is the main dependent variable and is defined as the ability for the attacker(s) to successfully detect...
Analysis

The experiment was a 3 within-subjects design, with categorical factors for method (baseline, PathWord) and attack configuration (A1, A2, A3). We gave the recorded video to the next participants to try to guess the PIN P1 was intending to enter. The response measure was the successful attacks (stolen PIN). Since the response was dichotomous, i.e. took one of only two possible values representing success or failure (coded 1 if a PIN is correctly detected and 0 otherwise), a binary logistic regression was well-suited to these data and was therefore used. The basic aim of our analysis was to describe the way in which stolen PINs (detected) varied by method and attack configuration employed.

5.5 Results

Effect of Input Technique

The overall successful attacks was 92.85% (SD+.25) for Baseline and only 4.7% (SD+.21) for PathWord. A binary logistic regression was performed to ascertain the effects of technique on the likelihood that participants have their PIN stolen. The logistic regression model revealed a significant main effect of technique on detection success ($\chi^2(1) = 57.96, p < .0001$). As shown in Figure 8, it was far harder for attackers to steal PINs entered with PathWord. Both Baseline and PathWord detection rate improved as the number of attackers increased, from 83.33% (SD+2.35) and 2.38% (2.72) to 100% (1.5) and 7.14% (4.8) respectively. However, in all the three attack configurations, PathWord resulted in fewer stolen PINs. These results suggested an important decrease of stolen PIN using PathWord compared to the standard input technique (Baseline). As a result H1 is confirmed. No technique × attack interaction was found on detection success ($p > .5$).

Effect of Attack Configuration

The more the number of attackers, the more stolen PINs we obtained. Aggregating Baseline and PathWord results for each configuration, we obtained 50% (SD+.53) and 53.57% (+.51) successful attacks for 2 and 3 attackers configuration respectively. The 1 attacker-configuration was less successful, with 42.85% (+.49) stolen PINs. Analysis showed no significant main effect of attack configuration on detection success ($\chi^2(1) = 3.6, p > .1$). Post-hoc analysis showed a significant differences between 2 and 3 attackers-configuration ($\chi^2(1) = 3.48, p < .001$), no differences were found between 1 and 3 attackers-configuration ($p > .25$), and between 1 and 2 attackers-configurations($p > .05$). However, descriptive analysis (Figure 8) showed that the scenario involving 3 attackers resulted in more detected PINs.

5.6 Usability and Mental Workload

In the following, we were interested in the usability and mental workload using PathWord compared to Baseline, replicating the evaluation by Langlotz et al. [32]. The independent variable was the technique with 2 levels (PathWord and Baseline). We used the results of the NASA TLX and a customized SUS questionnaire, filled by the same cohort of participants in study 2 as the dependent variables. We hypothesized that PathWord and Baseline techniques have similar mental workload (H2) and efficiency (H3).

Results: Statistical analysis was conducted using the nonparametric Aligned Rank Transform (ART) [44] which enables the use of ANOVA for non-parametric factorial data. Overall, we found a significant main effect of input technique on the results of NASA
The moving stimuli speed significantly affected the recognition rate, i.e., the intruder(s) recognized fewer PINs compared to Baseline.

The analysis of the NASA-TLX revealed significant effects for mental (p < .05) and physical demand (p < .05). Post-hoc analysis showed that PathWord was considered as the most secure system compared to Baseline (p < .05). However, 82% participants stated that they would rather use Baseline for everyday use because of the input rapidity. The majority of the participants (96.87%) found useful to use PathWord to unlock their system, especially if the system manages potential private information (e.g., ATM). Of the 42 participants, 9 preferred to have the ability to select the authenticating type. The reasons for this preference is that there are situations where the user is not observed. In this case, the traditional password entry method is more interesting in terms of rapidity and flexibility. It may be more appropriate to activate a PathWord entry method mode when people are in the vicinity of the user. Participant 2, an expert in security, suggested using our approach to enable entering card credentials on sales websites to reduce key-logger threats. Overall, some users deplored the time it takes (1.5 × 4 digits = 6 seconds) compared to the standard input modalities. Other users find it fun and were convinced of the security benefits of this approach.

6 DISCUSSION

Study 1 tested PathWord on a desktop application and study 2 compared PathWord and the standard input on a mobile device based on the number of stolen PINs from different threat models. A paired T-test showed that there is no significant difference in accuracy for using PathWord on a mobile or desktop application (t(5) = 1.10, p = 0.3). The small difference is likely to result from various factors including relative eye-device distance, different devices refresh rates and different participants (smooth pursuit dynamics). From our experiment, it is clear that PathWord resulted in fewer stolen PINs (97.62% undetected PINs for 1 attacker-configuration), outperforming the standard entry method (16.67%). The results are accentuated by the fact that PathWord relies on small eye drift to enter the PIN, making it difficult and tedious for an attacker to be aware of the digits the user was intending to enter.

The moving stimuli speed significantly affected the recognition rate. We found that V3 resulted in the best accuracy. This may be because at V1 and V2, the moving dots were moving so fast that users make saccadic eye movements to catch up the moving dot, circumventing smooth pursuit which is the base of the input method. This was confirmed by a participant who clearly stated that he could not follow the stimulus at V1 and V2. At V3 however, the speed was neither too fast nor too slow, thus resulting in an adequate parameter for higher recognition. These parameters may need to be personalized in an actual deployment.

7 LIMITATIONS

Our method focused on digits entry capability because current PIN authentication interfaces still heavily incorporate only digits from 0 to 9 (e.g., mobile phone, credit card cryptogram, ATM). However, detection rate can differ for alphabet-based passwords as there are more individuals in such cases, thus more alphabets that can have similar shape before (e.g., Ø ≈ Ø) and after applying the simulated annealing to reduce the individuals’ trajectories length (e.g., a part of Ø ≈ Ø, a part of Ø ≈ Ø). These conditions outlines challenging issue, that is, finding proper variabilities among characters (letters, digits and even special characters) and we are investigating this interesting perspective for future works. Results so far have been encouraging but there is still a need to address limitations in both the pupil and its center detection. First, pupil center detection methods still face problems caused by lighting conditions, contact lenses, glasses or head pose. This makes the method prone to false positives, yielding unintended digits’ entry.

In addition, PathWord provides an efficient way to enter a PIN code at the expense of sacrificing the time required using traditional inputs. Moreover, the digits must have an acceptable size otherwise the center positions of the pupil will appear to be static or unchanged. This problem may also occur when the user interface is located far away from the user. However, as most users hold their mobile device in their hand and interact with ATMs using finger touch, such distances are applicable for PathWord as it was demonstrated in the previous experiments. Also, it is much easier to install an eye tracking system at withdrawal-payment points than using mobile phones. A substitute of the back-of-the-device modality for ATMs may be a mobile icon that the user would follow before starting data collection for the PIN. However, as we have not focused on attacks in Experiment 1 (ATM), we did not implement it and we plan to propose a new modality in future work.

8 CONCLUSION

We presented PathWord, a novel multimodal PIN input approach that exploits back-of-device activation and user’s gaze for digit selection. This paper contributes to authentication methods resistant to shoulder surfing attacks. We first showed that our input recognition algorithm is sufficiently accurate to be used as a viable approach. Second, we provide evidence that our technique allows users to enter code digits with a sufficient security level. Since PathWord removes the use of the on-screen touch to interact with the device, it also allows reducing keylogger and smudge threats. Our technique is applicable to both mobile and desktop application and opens new possibilities for text-entry. For future work, we plan to investigate additional computation techniques to speed-up entry recognitions. As such, we would like to target a system where such input modalities can be applied to different application domains. We believe that our technique can provide a possible alternative for disabled people and text input at large.