SAFE: Secure Authentication with Face and Eyes

Arman Boehm[♭], Dongqu Chen[§], Mario Frank[♭], Ling Huang[†],
Cynthia Kuo[♯], Tihomir Lolic[♭], Ivan Martinovic[⋆], Dawn Song[♭]
[♭] University of California, Berkeley; [†] Intel Labs; [♯] Nokia Research; [⋆] Oxford University; [§] Yale University

***Abstract*—Face authentication is commonly offered as an alternative to passwords for device unlock. However, available face authentication systems are vulnerable to simple spoofing attacks. We demonstrate the impact of image quality on spoofing, using low resolution photo representative of those commonly posted online. We also show that videos and slideshows of images at different angles, and crude 3D avatars are effective. To defend against these vulnerabilities, we propose a face authentication system that includes a secrecy challenge. We present SAFE (Secure Authentication with Face and Eyes[1]), an improved face authentication method that uses a commodity gaze tracker to input a secret. During authentication, the user must not only show her face but also gaze at a secret icon that moves across the screen. Using a novel method for estimating the noise level in the gaze tracking data, SAFE adapts the system's parameters to enable secure, hands-free authentication.**

I. Introduction

Screenlock is an ubiquitous feature, available on almost all of today's laptops, tablets, and smartphones. A screenlock hides what the user was last viewing. It also prevents unauthorized use of the device and access to sensitive data.

Mobile device users employ screenlocks with distressing infrequency. Industry surveys estimate that between 38% and 70% of users do not even lock their mobile devices with passwords or PINs [19, 25]. As a result, biometric technologies are now offered as alternatives to passwords, including face authentication on devices with front-facing cameras [24]. However, face authentication is vulnerable to spoofing attacks, with attacks demonstrated as recently as Android's Jelly Bean release [13, 18, 20].

In the social media era, users' faces are often available online. Faces should be considered public knowledge, which means face authentication lacks a secrecy component. Secrecy is the strength behind passwords and PINs; to be secure, it may be something that face authentication must incorporate.

To address the issues with face authentication, we take the first step to design a Secure Authentication system with Face and Eyes (SAFE) for device unlock. SAFE is a hands-free, non-intrusive, and provably secure system that is appropriate when users and devices are physically co-located. In SAFE, we augment a face-based identity recognition module with a secrecy-based challenge-response protocol using a gaze tracker. During device unlock, a user tracks her secret icon with her eyes. The system recognizes the user's face and evaluates whether she has followed the correct icon using her gaze. SAFE integrates the passive and hands-free advantages of face recognition with additional security from the gaze-based challenge-response protocol. This leverages the advantages of both face recognition and secret-based authentication.

To account for the errors introduced by both human eyes and gaze trackers, we develop a mathematical framework to implement the challenge-response protocol using Principal Component Analysis (PCA). This framework allows us to quantify the robustness of the protocol and determine the system parametrization. The key contributions of our work include:

- The first analysis of the impact of image quality to successfully spoofing commercial face authentication systems. We show that attacks are successful even with low resolution images, videos, and animations.
- A unlock system that augments face recognition with a challenge-response protocol that uses gaze tracking.
- A secrecy challenge that could be deployed at any hands-free gaze tracking device such as head-mounted displays.
- A mathematical framework for estimating the gaze direction of a user. This framework is robust against the imprecision introduced by human eyes and gaze trackers and enables us to analyze the security of the challenge-response protocol and to determine the system parametrization.

The remainder of the paper is organized as follows. We evaluate the impact of image resolution on spoofing success in Section II. Next, we present the design and implementation of SAFE in Section III. Section IV outlines a mathematical model for analyzing system security and parametrization. We evaluate our system and model using an empirical user study in Section V. Finally, we review related work in Section VI and conclude in Section VII.

II. Spoofing Face Authentication

Photos taken in conditions similar to users' enrollment conditions most likely result in successful logins [6]. However, existing work fails to examine one important question: how "good" do these images need to be? How many pixels does an attacker need to conduct a successful attack? Is it possible to break a face authentication system with the low-quality photographs that are widely posted online?

We tested four commercially available face authentication systems: Dell FastAccess (version 2.4.95) [1]; HP Face

[1]Note that SAFE differs from SAfE, a method for ad-hoc pairing [7].

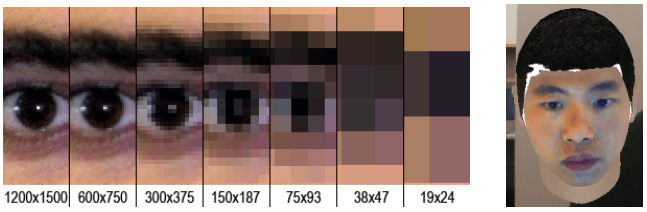


Figure 1: Left: Image quality declines with resolution. Image snippets are shown at slightly lower scale than the printed photographs. Right: Crude 3D avatar can spoof the Lenovo and Toshiba systems using liveness detection.

Recognition for HP ProtectTools (version 2.0.1.651); Lenovo Veriface (version 3.0) [2]; and Toshiba Face Recognition (version 3.1.18) [4]. For each of the four systems, we enrolled eight individuals (four male, four female) across a variety of skin colors and races (Caucasian, East Asian, South Asian, Latino).

On the Dell, Lenovo and Toshiba systems, we increased the sensitivity levels to their highest level; there were no adjustable settings on the HP. After enrollment, individuals attempted to login to the system, without any change in position or lighting. If the individual was unable to login, we reduced the sensitivity by one level until the individual could successfully login. At that maximal sensitivity level, we tried to spoof the respective system. Also, Lenovo notebooks provide an option to enable liveness detection, which is disabled by default. To start with the most conservative setting, we enabled liveness detection at its highest level and reduced it until individuals could successfully login.

A. Photographs and Videos

We created a high-quality photograph of each individual using a Canon Powershot SD1200IS digital camera. The photos were taken in an indoor environment, with typical fluorescent office lighting. Individuals were photographed from the front against a plain, off-white wall while maintaining a neutral expression.

We cropped and scaled the images to 1200×1500 pixels, such that the face was about 750 pixels high, from the tip of the chin to the top of the head. Then we downsampled the photos to 600×750, 300×375, 150×188, 75×94, 38×47, and 19×24 pixels.

Each photo was scaled to page size and printed on letter-sized office paper using a Xerox Phaser 7400 color laser printer. Figure 1 (left) illustrates the decline in image quality with the decrease in resolution.

We also recorded VGA (640×480) videos of individuals rotating their heads slightly to the left and right. Faces were about 300±100 pixels large.

B. Spoofing Tests

We tried to login to each system with the photographs or video. For the Dell and HP systems, we held the printed photographs in front of the laptop's webcam. For the Lenovo and Toshiba systems, the video played on a 30" monitor facing the laptop's webcam. The distance of the images was adjusted to roughly match the size of a live user's head.

The Dell, Lenovo, and Toshiba systems all support five discrete sensitivity levels. Assuming level 5 is the highest sensitivity, individuals were able to login, on average, at level 4.4 on the Dell system (default: 3); levels 2.6 for liveness detection (default: off) and 5 for face recognition (default: 5) on the Lenovo system; and level 4.8 (default: 1) on the Toshiba system. The manufacturers most likely chose low sensitivity levels to lower the rate of false negatives.

Figure 2 summarizes our spoofing results. The results show that photographs with an extremely low resolution are capable of spoofing both the Dell and HP systems. The HP system was much easier to reliably spoof, perhaps because we could not increase the sensitivity level over the default. At higher sensitivity levels, the Dell system was sensitive to changes in lighting and position. We were able to reliably spoof the Dell system if we decreased the sensitivity level back to its default. A video was also able to spoof the Dell system. Turning off liveness detection on the Lenovo system also allowed us to authenticate using still images.

Commercial face recognition systems have been rumored to perform poorly with dark-skinned faces [8]. We observed that all four systems struggled with dark-skinned individuals. Our darkest-skinned participants were unable to enroll or login under standard office lighting. They even struggled to enroll next to a window with bright daylight. Similarly, login was also very unreliable, even in bright daylight. One individual was unable to login on the Toshiba system – at any level of sensitivity – after enrollment.

C. Alternatives to Videos

Since attackers may not have access to videos of users at specific angles, we tried two alternative strategies. First, we cycled through slide shows of photographs from several angles. The slide show attack was effective on both the Lenovo and Toshiba systems. Second, we generated a video of a 3D avatar rotating its head from side to side. Using one frontal photograph of the individual's face, we created a 3D model of the individual's head using FaceGen Modeler. Using 3ds Max, a 3D animation tool, we generated an animation of the model rotating and nodding its head. We show a screenshot of an example avatar in Figure 1 (right). The video of the avatar was also effective against the Lenovo and Toshiba systems.

D. Summary of Attacks

The Dell and HP face authentication systems are easily spoofed by low-quality, high-contrast photographs of users' faces, such as pictures on Facebook or profile photos on LinkedIn. The Lenovo and Toshiba face authentication systems support liveness detection, but we were able to

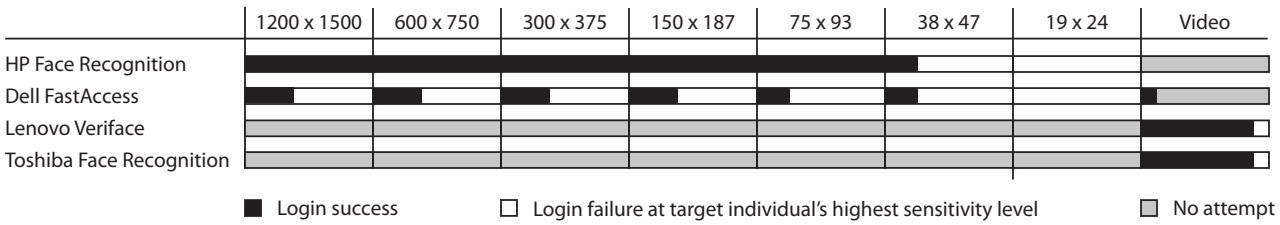


Figure 2: Summary of photo and video spoofing attempts at the systems' *highest* sensitivity levels. Note that we were able to reliably authenticate with images at the systems' *default* sensitivity levels.

spoof both these systems with videos and animations of the enrolled users.

In general, face authentication systems are vulnerable because faces are publicly available information in the social media era. The optical appearance of a face can be reproduced with ease using consumer-grade equipment.

III. The SAFE System

While flawed, face authentication is still convenient, allowing users to unlock their devices in a hands-free, non-intrusive manner. Below we present the design and implementation of SAFE, a more secure hands-free system for device unlock. Our system requires a front-facing camera, as well as a gaze tracker consisting of two infrared light sources and a camera with an infrared band-pass filter.

The system has two phases: an enrollment phase and an authentication phase. In the enrollment phase, the user calibrates the gaze tracker and enrolls in the system. Enrollment consists of two parts: taking pictures of the user for the face recognition system, and selecting a set of secret icons for a multi-phase challenge-response protocol. To successfully login, a user needs to recognize her secret among a set of decoy icons and follow it using her gaze.

In the authentication phase, the user faces her device's front-facing camera. If the user's face is recognized as the face of an enrolled user, the SAFE user interface is displayed (there should be a password backup for adverse lighting conditions). A window opens with n icons populating the sides of the screen. One of the n icons is a member of the user's set of secret icons. Each icon sits on a line outlining its path of movement, as shown in Figure 3 (bottom). The system pauses for about one second so that the user can locate her secret icon. The icons start moving with uniform speed along their paths, and the user must follow her secret icon with her eyes. As the challenge is solved by a minimal gaze movement, it is very difficult for an observer to conduct shoulder surfing attacks.

After the icons move off the screen, another set of icons may appear. This second phase is similar to the first, except the set of icons on the screen is different. The user follows her next secret icon, and the process may repeat until she completes p total phases. The number of icons per phase n and the number of phases p depend on the system configuration, which in turn depends on the security requirements.

During device unlock, the video stream from the camera appears as a transparent overlay below the icons. White circles outline the target location of the user's eyes. The circles represent the location of the user's eyes during calibration to help the user maintain her head position during authentication. We deliberately implement a transparent overlay and circles of a certain size to make shoulder surfing attacks more difficult.

A. System Architecture

The SAFE architecture, illustrated in Figure 3 (top), is highly modular and comprises the following parts: the user interface, the gaze tracker, the face recognizer, the authentication model, and the user icon and profile databases. The user interface is shown in Figure 3 (bottom) and displays: 1) the challenge generated by the SAFE authentication module; 2) the location of the user's eyes during calibration, represented as white circles; and 3) the current video stream from the front-facing camera.

The gaze tracker is a software and hardware system that reflects infrared lights on users' eyes and calculates the eyes' gaze location through pupil movement. The face recognition system evaluates the similarity of the current user's facial features to the enrolled user(s). The face recognition system must identify the current user as an enrolled user to initiate device unlock. During unlock, the face recognition system constantly runs in the background and communicates its results to the authentication module.

The SAFE authentication module is responsible for two main functions: generating a challenge and evaluating the response; and matching a user's gaze data to one of the icon paths on the screen. The challenge-response generator retrieves one of the user's secret icons from the profile database and constructs a challenge for each phase based on the physical screen size, the screen resolution, the gaze tracking error, and the desired security level. A challenge includes a set of icons from the icon database, the location of the icons on the screen, and the trajectories that the icons will take. Using the measured gaze data, the gaze angle analyzer determines the angle of a user's gaze path and determines

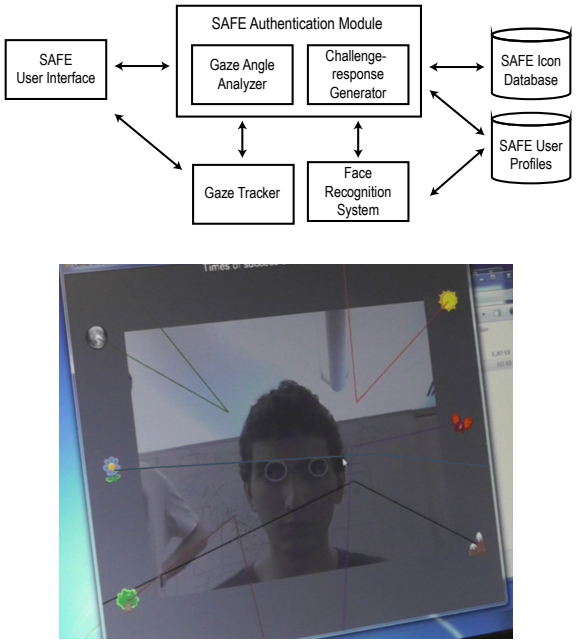


Figure 3: SAFE system architecture and the SAFE user interface during device unlock.

whether the gaze path matches the secret icon's path. We will describe this process in Section III-B.

The SAFE user profiles database stores profiles of enrolled users. Profiles include facial feature data, as well as the user's set of secret icons and user-specific sets of decoy icons. The SAFE icon database stores all of the icons used in the authentication challenges.

B. Implementation

We implemented the SAFE user interface and authentication module on Windows 7 using the Microsoft Foundation Classes (MFC). The initial development was completed on a Samsung Series 7 Slate tablet with 64GB memory, but the software works on any Windows 7 machine. We used off-the-shelf technology for face recognition and gaze tracking. SAFE's modular architecture allows us to accommodate other face recognition techniques or gaze tracking packages as needed.

We use the ITU Gaze Tracker (version 2.1), an open-source video-based gaze tracker developed by the IT University of Copenhagen [22]. The gaze tracking software is paired with two Sony HVL-IRM infrared lights and a Thorlabs DCC1545M camera. The camera is fitted with an Arecont Vision MPL8-16 and an Opteka HD^2 37mm R72 720nm infrared filter. The infrared lights and camera are mounted to a custom-made aluminum rack using Giottos MH 1004 mini ball heads and wing screws.

1) Calculating Gaze Trajectory Angles: All icons move across the screen along different trajectories composed of multiple line segments. A user gazes at her secret icon as

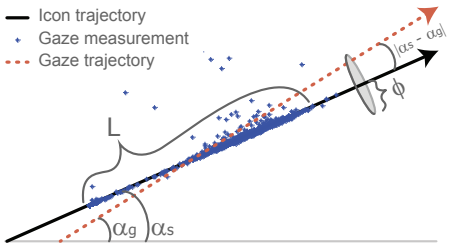


Figure 4: Diagram illustrating how SAFE compares gaze trajectory α_g and secret icon trajectory α_s. For a successful authentication with one-line trajectories, $|\alpha_s - \alpha_g|$ must be less than system tolerance ϕ.

it moves across the screen. All gaze measurements while the icon moves along one line segment constitute a set of gaze points in the 2-D plane. For each set of gaze points, we compute the angle of the user's gaze direction using Principal Component Analysis (PCA). PCA can discriminate the main direction of the gaze from noise in the gaze measurements in a way that best explains the variance in the set of points. We set the gaze direction as the first principal component (PC) of the gaze point set, and refer to its angle, relative to the horizontal line, as α_g. The dotted line in Figure 4 exaggerates how noisy data may cause α_g to deviate slightly from α_s.

2) Matching Icon and Gaze Trajectories: Let α_s be the angle of trajectory of the user's secret icon, and let ϕ be the tolerance given by the system. The tolerance ϕ defines the maximum angle that the user's gaze can deviate from the trajectory of the user's secret icon where the two angles are still considered a match. In other words, the system considers α_s and α_g to be the same if $|\alpha_s - \alpha_g| < \phi$. Figure 4 illustrates how SAFE matches a user's gaze trajectory to an icon trajectory.

When icons change direction partway across the screen, we split the trajectory into two line segments. There are then two icon angles, α_{s1} and α_{s2}, and two measured gaze angles, α_{g1} and α_{g2}, each of which is associated with one line segment. These two polylines are considered to be the same if $(|\alpha_{s1} - \alpha_{g1}| < \phi) \wedge (|\alpha_{s2} - \alpha_{g2}| < \phi)$.

3) Face Recognition: We use an online face recognition module from [27] implemented in C/C++ using OpenCV [3]. To build a user's identity model, we use two categories of features extracted from user faces: the eigenface and the geometry (triangle) between two eyes and the mouth. For every new image, a face detection procedure uses pre-trained Haar Cascade classifiers to detect faces [28], from which we locate the eyes and mouth. We then proceed to extract features from the face image and geometry. During enrollment, we add the facial features to the user's identity model based on 8 pictures of the user. During authentication, we test the facial features against the user's identity model: the face recognition system captures images from the video

stream continuously and runs each image through the face recognition.

4) Device Unlock Criteria: Access is granted when the user manages to satisfy following three conditions:

1) The user exceeds a certain threshold of positive identifications from the face recognition system.
2) The user follows her secret icons in such a way that sufficiently many (*e.g.*, more than 10) gaze points overlap with a substantial fraction of the icon line.
3) The directions derived from gaze points, α_{g1} and α_{g2}, satisfy $(|\alpha_{s1} - \alpha_{g1}| < \phi) \wedge (|\alpha_{s2} - \alpha_{g2}| < \phi)$.

Imposing a lockout penalty between unsuccessful unlock attempts can render brute force attacks impractical, particularly when the penalty increases between successive unlock attempts. SAFE pairs a weak password with an exponential lockout penalty to resist brute force attacks. After the third unlock attempt, a one-second penalty is added that doubles after every subsequent unsuccessful attempt.

IV. System Security and Parametrization

Several interrelated system parameters influence the security and usability of SAFE. In this section, we present a mathematical framework that guides the system parametrization. The parameters that affect the security and usability of our system are:

- L: the length of the line that the user follows,
- n: the number of icons on the screen during one phase,
- σ^2: the error (noise) in the gaze data,
- m: the number of gaze points measured per line,
- ϕ: the angle matching tolerance.

Noise σ^2 originates from two sources: (1) the imprecision with which the user gazes at points on a line; and (2) the statistical errors of the measurement process. Error (2) can be measured and is reported in our findings in Section V. The other parameters must be set when configuring the system.

The primary parameter that impacts security and usability within a single challenge-response phase is the angle tolerance ϕ. It defines the maximum angle that a user's gaze path can deviate from that of the secret icon where the two are still considered a match. In this way ϕ determines the number of icons allowed to display and move over the screen, which determines the security level of the protocol. Thereby, ϕ itself depends on σ^2, m, and L.

As follows, we derive a theoretical dependency between all these parameters. We first show how ϕ determines the maximal number of icons n in Section IV-A, and then derive how ϕ depends on the given parameters in Section IV-B.

A. Number of Icons n as a Function of ϕ

As described in Section III-B2, the user's gaze is matched with the true icon if the difference of the gaze angle to the true line is within angle tolerance ϕ. We derive the maximal number of icons that we can put on the screen as a function of ϕ. For simplicity, we start our analysis assuming that icons move across the screen in one straight line. Then we extend this analysis to a two-line trajectory where icons change direction mid-screen. Since screens across different phases are independent and similarly configured, we only need to analyze the system with a single phase (one screen).

Theorem. Given the tolerance ϕ, the system can at most securely distinguish $n = \lfloor \frac{360^\circ}{2\phi} \rfloor$ number of icons that move along one-line trajectories. Please see [5] for a complete proof. Intuitively, with more than $\lfloor \frac{360^\circ}{2\phi} \rfloor$ icons on the screen, there must be at least two icon lines with an angle difference below 2ϕ, where the factor of 2 comes from using the absolute value of the angle difference. Then, by the setup for matching gaze to flying icons in Section III, an attacker may match her gaze to multiple lines by following only one line. As a consequence the extra icons (exceeding $\lfloor \frac{360^\circ}{2\phi} \rfloor$ icons) do not improve security.

Two-Line Trajectories. When icon movements have one change of direction, the trajectory of the icon consists of two line segments with different angles. Given ϕ, the system can at most securely distinguish $n = (\lfloor \frac{360^\circ}{2\phi} \rfloor)^2$ number of icons that move along two-line trajectories. This can be seen as follows. Consider $(\lfloor \frac{360^\circ}{2\phi} \rfloor)^2$ icons on the screen. We construct two sets of single distinguishable lines and then combine them to distinguishable pairs. Let the sets of single lines be $S_1 = \{l_i = 2i\phi, i \in \{1, \ldots, \lfloor \frac{360^\circ}{2\phi} \rfloor\}\}$ and $S_2 = \{l_j = 2j\phi, j \in \{1, \ldots, \lfloor \frac{360^\circ}{2\phi} \rfloor\}\}$. The set of distinguishable polylines is then the Cartesian product $P = S_1 \times S_2$. The cardinality of this set is equal to $(\lfloor \frac{360^\circ}{2\phi} \rfloor)^2$.

B. Derivation of Angle Tolerance ϕ

We use a statistical method based on PCA to estimate the direction of the gaze line. As described in Section III-B1, the gaze direction is the first principal component (PC) of a set of user's gaze points associated with one line segment.

When a user tracks a moving icon on the screen, her gaze often deviates from the exact icon position with some randomness as illustrated in Figure 4. In addition, the gaze tracker produces gaze measurements with a statistical error around the true gaze points. These two effects are the primary sources of error when measuring users' gaze locations. We want to set the tolerance ϕ as small as possible (to maximize security) while the system is still robust to the noise in gaze data.

We view the user's derived gaze trajectory as a noisy version of the icon's true trajectory. We determine the tolerance ϕ such that, with high probability, the deviation of the first PC from the icon (line) trajectory is smaller than ϕ. To this end, we make the following assumptions: (1) All components of the differences between the measured gaze points and the moving icon points are independent and identically distributed random variables with mean 0, variance σ^2, and fourth moment μ^4. As in [15], our theoretical analysis can also accommodate anisotropic distributions. (2)

The user follows a moving icon over a distance L that is significantly larger than σ.

With this setup, we are equipped with two versions of data of equal sizes: one version is the data points defined by an icon moving at uniform speed along the icon trajectory line, while another version is the measured gaze points, which are distributed around the corresponding icon points according to the first assumption. For 2D data points uniformly distributed on a line, the first PC is the line itself, and the two eigenvalues of its covariance matrix are $\lambda_1 = \frac{L^2}{12}, \lambda_2 = 0$, respectively.

According to matrix perturbation theory [26], ϕ can be bounded by the following inequality:

$$|\sin\phi| \leq \frac{\|\Delta\|_F}{\lambda_1 - \lambda_2} = \frac{12\|\Delta\|_F}{L^2}, \quad (1)$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, and Δ is the difference between the covariance matrix of measured gaze points and that of moving icon points. According to a statistical perturbation result on PCA [15], $\|\Delta\|_F$ can be bounded by expectation using:

$$\mathbb{E}(\|\Delta\|_F) \quad \leq \quad 2\sqrt{\frac{\sigma^2 L^2}{6m}} + \sqrt{2\sigma^4 + \frac{2(\mu^4 + \sigma^4)}{m}}. \quad (2)$$

Using this expectation bound and assuming $m \gg 1$, we have

$$|\sin\phi| \leq \frac{12\|\Delta\|_F}{L^2} \lesssim \frac{12\sigma}{L}\left(\sqrt{\frac{2}{3m}} + \frac{\sqrt{2}\sigma}{L}\right). \quad (3)$$

With more data points (larger m) it is more likely that the noise cancels out. This leads to a smaller bound on ϕ. L/σ is a measure of signal-to-noise ratio (SNR). More noise means that the SNR is smaller and the bound on ϕ is larger. Recall that σ is tied to individual users and specific gaze trackers. By calibrating individuals with a given gaze tracker, we may obtain a personalized value of σ and adapt the system to a user and her device.

V. System Validation and User Study

Based on the model in the previous section, we parametrize SAFE with calibration data from 10 users. Then we conduct a user study to understand how many phases users can reasonably handle in practice. Some work has explored the use of gaze tracking for security applications [10, 12, 14, 17, 29], but they were not based on following moving objects.

A. Setting System Parameters

In this section, we set SAFE's parameters to achieve an acceptable level of security and a practical level of usability. The free parameters of our system, m and L, are primarily determined by gaze error σ and the bound in (3).

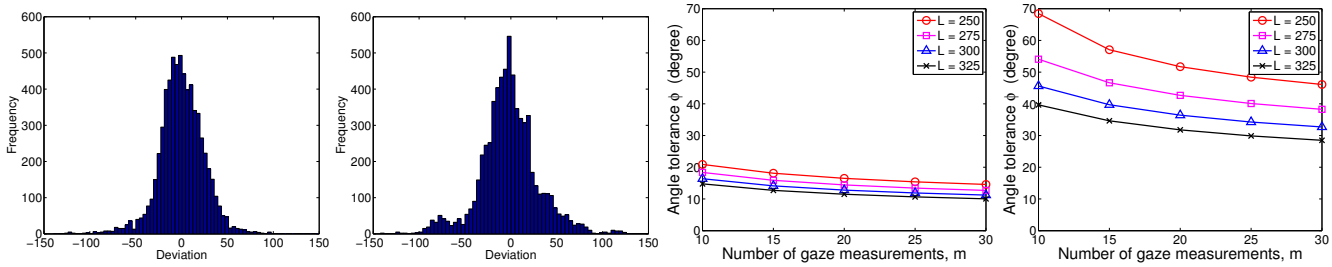
Calibration and estimating σ: To estimate the precision of our gaze tracker's measurements, we asked 10 users to complete the calibration exercise. Calibration allows us to estimate σ using the differences between the measured gaze points and the corresponding true (icon) points on the screen. We performed nine rounds of calibration for each of the 10 participants. In each round, a participant gazes at nine locations on the screen. The locations are presented in a random order, and the gaze tracker collects around 35 points for each location. Thus, we collected more than 3,000 gaze points for each user. We compute the mean and the standard deviation σ for each participant's calibration data. The histograms of gaze errors for two users are shown in Figure 5 . The means are very close to 0, and σ ranges from 20 pixels to 40 pixels.

Number of gaze points per line m: Our theoretical analysis in Section IV shows that m influences ϕ and the number of icons that we can put on the screen in any given phase. For several realistic values of σ, we explore a variety of values for m and L to see their effects on the angle tolerance ϕ. Figures 5c and 5d show three important results. First, σ has a significant impact on ϕ. For almost every pair of m and L values, ϕ becomes more than three times larger when σ changes from 20 to 40. Second, m has a relatively small effect on ϕ. The marginal benefit of increasing m is small, particularly when $m \geq 20$. Finally, at the greatest error level that we saw in our participants ($\sigma = 40$), the bound on ϕ is 54.1° even when only 10 samples are collected as the user tracks the icon along a line segment of length $L = 275$. This allows us to display up to $(\lfloor\frac{360^\circ}{108.2^\circ}\rfloor)^2 = 9$ different icons per phase. If the eye tracker sampled at a higher rate, we could collect more samples and support a larger number of icons. However, during the user study, participants exhibited some difficulty with nine icons on the screen. They preferred fewer (six) icons on the screen.

B. User Study

We conducted a user study to answer two questions. First, how many phases can users handle during login? Second, does the theoretically derived angle tolerance ϕ provide sufficient security while avoiding false rejections?

Thirteen participants (five female, age 25–45 years) completed the study. Participants were required to complete the study without eyeglasses. Participants first selected their set of secret icons from a predetermined pool of icons. Next, they calibrated the gaze tracker using 9 calibration points on the screen. Participants were asked to locate and follow their secret icons. Each user logged in with two to five consecutive phases with randomized order. During each phase, six icons moved at uniform speed across the screen. The trajectories were randomly selected such that the distance between all pairs of first line segments and all pairs of second line segments were at least 2ϕ, respectively. The length of each line segment was 275 pixels or more. To successfully follow her secret icon, a participant's gaze must be measured along each line segment of her secret icon for 10 or more points.



(a) User A, $\sigma = 24.5$ (b) User B, $\sigma = 32.0$ (c) $\sigma = 20$ (d) $\sigma = 40$

Figure 5: 5a and 5b: Histogram of gaze errors, the differences (in unit of pixels) between the measured gaze points and their corresponding icon points on the screen. For each user, more than 3000 gaze points are collected during the calibration stage, which are used to estimate σ. 5c and 5d: The effects of m, σ, and L on the bound of ϕ.

We recorded the number of attempts required to successfully login, the time required to successfully login, and participants' gaze data. Analysis of the gaze data enables us to validate ϕ below. Also, participants provided qualitative feedback by answering a 7-point Likert scale [23] for the statement, "Overall, the task was:".

Results: On average, participants were able to successfully login with 2 to 4 phases in 1.1 attempts. This jumped to an average of 1.5 attempts with 5 phases. Participants expressed that 5 phases required too much concentration. This is reflected in the questionnaire in Figure 6a. Participants spent an average of $8s$ in each phase. Figure 6b shows the time required to successfully login, including failed attempts. Our results suggest that 4 or fewer phases work best for the SAFE system.

Validating the bound of ϕ: From the 13 participants, we accumulated around 450 sets of gaze data. Each set of gaze data contains 10 or more gaze points associated with an icon line segment. For each set of gaze data, we compute the first PC of the gaze points. Next, we calculate the angle difference between the PC and the corresponding line segment of the secret icon. With the angle differences for all participants, we compute the mean difference, which is $0.4°$, and the standard deviation, which is $25.5°$. Figure 6c shows a histogram of the angle differences. We find that our theoretical bound $|\phi| = 54.1°$, indicated by the red dash lines in Figure 6c, is indeed a high probability bound and covers more than 95% of the angle differences in our user study. Thus, it is unlikely that the system will fail to detect when the user follows the correct line. At the same time, the bound is not overly loose, providing maximal security for the given error rate. The key space of our system is limited by the imprecision of the device, not by the method itself. As more precise eye trackers become available, SAFE will support a larger key space and will have increased usability.

VI. Related Work

Prior work in gaze tracking and security focuses on methods for input selection and resistance to shoulder-surfing attacks. Kumar et al. implemented a system that uses gaze tracking to input the characters in a password or PIN, in lieu of a standard keyboard [17]. Similarly, De Luca et al. used gaze tracking to enter PINs into an ATM interface [11]. Both show that gaze-based input takes more time than keyboard entry. More recently, researchers have tested gaze tracking with non-alphanumeric passwords. Dunphy et al. tested ATM interfaces using faces instead of numbers [12]. Forget et al. proposed Cued Gaze-Points, in which users select secret points on a sequence of images [14]. De Luca et al. evaluated an authentication system, EyePassShapes, based on gaze gestures [10].

Biometrics-based authentication schemes are gaining popularity on mobile devices [16]. Extensive research has been done and substantial progress has been achieved in developing authentication schemes based on fingerprints [21], iris recognition [9] and facial patterns [27]. Our work shows the benefit of combining multiple factors to achieve secure authentication.

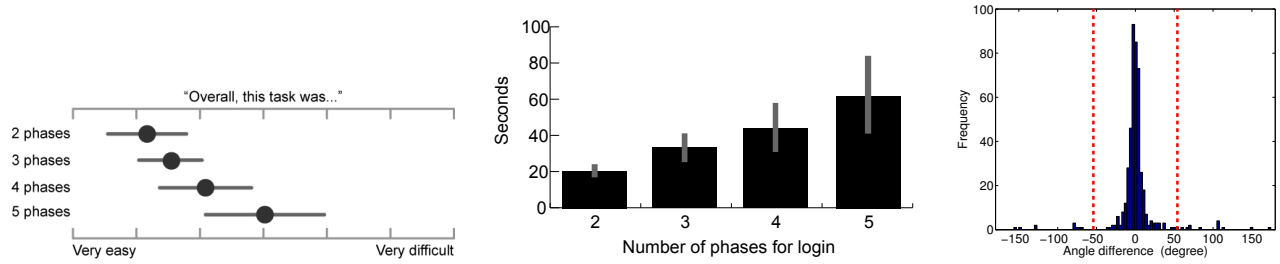
VII. Conclusion

We present SAFE, a system for secure device login that augments face recognition with gaze tracking. SAFE combines face recognition for identification with gaze tracking for the input of a secret. During login, while a face recognizer continuously checks the identity of the user, a number of icons are displayed on the screen; as the icons move in incongruent line patterns, the user follows her secret icon with her eyes. By adapting the number of icons and repetitions, one can control the security of the system.

Our proof-of-concept is supported by a theoretical analysis that quantifies the impact of real-world measurement errors on the user experience. SAFE is a novel system that is ideal for hands-free login on mobile devices.

References

[1] Dell FastAccess. http://www.sensiblevision.com/en-us/support/dellsupport.aspx.
[2] Lenovo Veriface. http://support.lenovo.com/en_US/detail.page?LegacyDocID=MIGR-72561.
[3] OpenCV. http://opencv.willowgarage.com.



(a) (b) (c)

Figure 6: User study results. (a) Difficulty rating for the number of phases required to login. Bars represent 95% confidence intervals. (b) Time required to successfully input a set of secret icons, including failed attempts. (c) Histogram of angles between the first principal components of the gaze points and the icons' true lines. The histogram is computed from around 450 trials of 13 users. The two red dash lines mark $|\phi| = 54.1^\circ$ computed from Eqn. (3) covering more than 95% of angles.

[4] Toshiba Face Recognition. http://us.toshiba.com/computers/research-center/technology-guides/face-recognition.

[5] SAFE: Secure Authentication with Face and Eyes. *Technical Report*, available upon request. 2012.

[6] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino. 2d and 3d face recognition: A survey. *Pattern Recognition Letters*, 28(14):1885–1906, 2007.

[7] I. Buhan, J. Doumen, P. Hartel, and R. Veldhuis. Secure ad-hoc pairing with biometrics: Safe, 2007.

[8] D. Cryer. HP computers are racist. http://www.youtube.com/watch?v=t4DT3tQqgRM.

[9] J. Daugman. How iris recognition works. *IEEE Trans. Circuits Syst. Video Techn*, 14(1):21–30, 2004.

[10] A. De Luca, M. Denzel, and H. 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.

[11] A. De Luca, R. Weiss, and H. 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.

[12] P. Dunphy, A. Fitch, and P. Olivier. Gaze-contingent passwords at the ATM. In *Proceedings of COGAIN 2008: Communication , Environment and Mobility Control by Gaze*, COGAIN 2008, pages 59–62, 2008.

[13] J. Fierrez, J. Galbally, A. Anjos, C. McCool, F. Alegre, N. Evans, A. Thiebot, A. Hadid, S. Li, G. L. Marcialis, J. Carter, J. Bustard, and J. Acedo. D2.3: Specifications of spoofing attacks. TABULA RASA: Trusted Biometrics under Spoofing Attacks, 2011.

[14] A. Forget, S. Chiasson, and R. Biddle. Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, CHI '10, pages 1107–1110, New York, NY, USA, 2010. ACM.

[15] L. Huang, X. Nguyen, M. Garofalakis, A. Joseph, M. Jordan, and N. Taft. In-network PCA and anomaly detection. In *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, B.C, 2006.

[16] A. K. Jain, A. Ross, and S. Pankanti. Biometrics: a tool for information security. *IEEE Transactions on Information Forensics and Security*, 22(11), 2006.

[17] M. Kumar, T. Garfinkel, D. Boneh, and T. 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.

[18] M. D. Nguyen and Q. M. Bui. Your face is NOT your password. In *Black Hat DC 2009*, 2009.

[19] Norton. Norton survey reveals one in three experience cell phone loss, theft. http://www.symantec.com/about/news/release/article.jsp?prid=20110208_01.

[20] J. Rice. Android Jelly Bean's Face Unlock "Liveness Check" Circumvented With Simple Photo Editing. http://www.androidpolice.com/2012/08/03/android-jelly-beans-face-unlock-liveness-check-circumvented-with-simple-photo-editing/.

[21] A. Ross, J. Shah, and A. K. Jain. From template to image: Reconstructing fingerprints from minutiae points. *IEEE Trans. Pattern Anal. Mach. Intell*, 9(4):544–560, 2007.

[22] J. San Agustin, H. Skovsgaard, E. Mollenbach, M. Barret, M. Tall, D. W. Hansen, and J. P. Hansen. Evaluation of a low-cost open-source gaze tracker. In *Proceedings of the 2010 Symposium on Eye-Tracking Research; Applications*, ETRA '10, pages 77–80, New York, NY, USA, 2010. ACM.

[23] J. Sauro and J. S. Dumas. Comparison of three one-question, post-task usability questionnaires. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, CHI '09, pages 1599–1608, New York, NY, USA, 2009. ACM.

[24] S.Furnell, N.Clarke, and S.Karatzouni. Beyond the PIN: Enhancing user authentication for mobile devices. *Computer Fraud and Security*, August 2008.

[25] Sophos Naked Security blog. Survey says 70% don't password-protect mobiles: download free Mobile Toolkit. http://nakedsecurity.sophos.com/2011/08/09/free-sophos-mobile-security-toolkit/.

[26] G. W. Stewart and J. G. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[27] M. Valko, B. Kveton, D. Ting, and L. Huang. Online semi-supervised learning on quantized graphs. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

[28] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of CVPR*, 2001.

[29] J. Weaver, K. Mock, and B. Hoanca. Gaze-based password authentication through automatic clustering of gaze points. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2011)*, 2011.