# User-Generated Free-Form Gestures for Authentication: Security and Memorability

Michael Sherman[†], Gradeigh Clark[†], Yulong Yang[†], Shridatt Sugrim[†], Arttu Modig[*],
Janne Lindqvist[†], Antti Oulasvirta[‡⋆], Teemu Roos[*]
[†]Rutgers University, [‡]Max Planck Institute for Informatics
[⋆]Saarland University, [*]University of Helsinki

## ABSTRACT

This paper studies the security and memorability of free-form multitouch gestures for mobile authentication. Towards this end, we collected a dataset with a generate-test-retest paradigm where participants (N=63) generated free-form gestures, repeated them, and were later retested for memory. Half of the participants decided to generate one-finger gestures, and the other half generated multi-finger gestures. Although there has been recent work on template-based gestures, there are yet no metrics to analyze security of either template or free-form gestures. For example, entropy-based metrics used for text-based passwords are not suitable for capturing the security and memorability of free-form gestures. Hence, we modify a recently proposed metric for analyzing information capacity of continuous full-body movements for this purpose. Our metric computed estimated mutual information in repeated sets of gestures. Surprisingly, one-finger gestures had higher average mutual information. Gestures with many hard angles and turns had the highest mutual information. The best-remembered gestures included signatures and simple angular shapes. We also implemented a multitouch recognizer to evaluate the practicality of free-form gestures in a real authentication system and how they perform against shoulder surfing attacks. We discuss strategies for generating secure and memorable free-form gestures. We conclude that free-form gestures present a robust method for mobile authentication.

## Categories and Subject Descriptors

H.5.m. [**Information Interfaces and Presentation (e.g. HCI)**]: Miscellaneous

## Keywords

gestures; security; mutual information; memorability

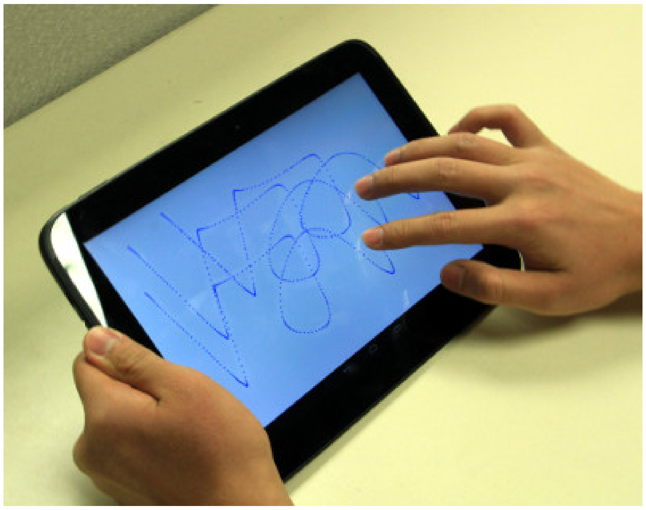


**Figure 1: This paper studies continuous free-form multitouch gestures as means of authentication on touchscreen devices. Authentication on touchscreens is normally done with a grid-based method. Free-form gesture passwords have a larger password space and are possibly less vulnerable to shoulder surfing. We note that there are no visual cues for the gestures, the gesture traces are shown only after creating the gesture. This is for the purposes of testing – a fully trained system does not show the traces.**

## 1. INTRODUCTION

Smartphones and tablets today are important for secure daily transactions. They are part of multi-factor authentication for enterprises [23], allow us to access our email, make one-click payments on Amazon, allow mobile payments [36] and even access to our houses [16]. Therefore, it is important to ensure the security of mobile devices.

Recently, mobile devices with touchscreens have made gesture-based authentication common. For example, the Android platform includes a 3x3 grid that is used as a standard authentication method, which allows users to unlock their devices by connecting dots in the grid. Compared with text-based passwords, gestures could be performed faster while requiring less accuracy. Although grid-based gestures better utilize the capabilities of touchscreens as input devices, they are limited as an authentication method. For example, a visual pattern drawn on a grid is prone to attacks such as shoulder surfing [43] and smudge attacks [1].

This paper studies *free-form multitouch gestures* without visual reference, that is, gestures that allow all fingers to draw a trajectory on a blank screen with no grid or other template. An example of the creation process is depicted in Figure 1, where the gesture traces are shown only after the gesture was created. This method bears potential, because it relaxes some of the assumptions that make the

*MobiSys'14,* June 16–19, op2014, Bretton Woods, New Hampshire, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2793-0/14/06 ...$15.00.
http://dx.doi.org/10.1145/2594368.2594375.

grid-based methods vulnerable. In particular, arbitrary shapes can be created. Moreover, as more fingers can be used, in principle more information can be expressed. Technically such gestures can be scale and position invariant, allowing the user to perform gestures on the surface without visually attending the display. Consider, for example, drawing a circle as your password. This may be beneficial for mobile users who need to attend their environment. Nevertheless, although no visual reference is provided, mnemonic cues referring to shapes and patterns can still be utilized for generating the gestures. Finally, when multiple fingers are allowed to move on the surface and no visual reference is provided, observational attacks may be more difficult.

Previous work on gestures as an authentication method has focused on a few directions: one was whether the same gesture can be correctly recognized in general [15, 31] or in a specific environment such as handwriting motion detected by Kinect-cameras [38], predefined whole-body gestures detected from wireless signals [28], and mobile device movement detected by built-in sensors [30]. Studies of the security of gestures look at either the protection of gestures from specific scenarios [43, 33, 38, 11], or an indirect measurement of security [17, 25]. Further, these works have focused on understanding performance of template gestures repeated by participants, not user-generated free-form gestures as the present work.

Our goal is to understand the security of this method by measuring mutual information and studying memorability in a dataset that allowed users to freely choose the kind of multitouch passwords they deemed best. We conducted a controlled experiment with 63 participants in a generate-test-retest design. At first, participants created and repeated a gesture (generate), then tried to recall it after a short break (test) and recalled it again after a period of time at least 10 days (retest). With this paradigm we were able to examine the effect of time on how participants memorize their gestures. To the best of our knowledge, we are the first to present a study on how people actually recall free-form multitouch gestures after a delay.

To analyze the security of the gestures, we use a novel information metric of mutual information in repeated multifinger trajectories. We base our metric on a recent one that was used for a very different purpose, specifically the estimation of throughput (bits/s) in continuous full-body motion [27], and it has not been used previously for authentication. Because multitouch gestures are continuous by nature the standard information metrics cannot be directly applied. What is unique to gesturing over discrete aimed movements (physical and virtual buttons) is that every repetition of a trajectory is inherently somewhat different [20]. However, when this variability grows too large, the password is useless, because it is both not repeatable by the user and not discriminable from other passwords. The information metric should capture this variability. In our metric, a secure gesture should contain a certain amount of "surprise", that is, some turns or changes, while still being able to be reproduced by the user itself. We also include a mutual information calculation to separate the complexity of controlled and intended features of the gesture and that of uncontrolled and unreproducible features.

Our results show that several participants were able to create secure and memorable gestures without guidance and prior practice. However, many participants used multiple fingers in a trivial way, by just repeating the same gesture. Our implementation of a practical multitouch recognizer shows that the free-form gestures can used as a secure authentication mechanism, and are resistant to shoulder-surfing attacks.

Our contributions are as follows:

1. Report on patterns in user-generated free-form multitouch gestures generated from 63 participants with a typical tablet;

2. Adaptation of a recent information theoretic metric for measuring the security and memorability of gestures;

3. A design and implementation of a practical multitouch gesture recognizer to evaluate free-form gestures applicability for authentication;

4. A preliminary study on a shoulder-surfing attack that indicates the potential of free-form gestures against such attacks.

## 2. RELATED WORK

In this section, we discuss related work on biometric-rich authentication schemes, graphical passwords, and password memorability.

**2D gesture authentication schemes.** Similar to free-form gestures, biometric-rich authentication schemes are based on the idea that when a user performs a gesture on a touchscreen they will do this in such a way that features can be extracted that will uniquely identify them later on [15, 31, 45, 3]. Similar ideas have been applied to recognizing motions with Kinect [38]. Specifically, Sae-Bae et al. [31] has shown that there is a uniqueness to the way users perform identical set of template 2D gestures based on biometric features (e.g. hand size and finger length). Frank et al. [15] demonstrated that the way a user interacts with a smartphone forms a unique identifier for that user, they showed that the way a user performs simple tasks (e.g. scrolling to read or swiping to the next page) is performed in a unique way such that the coordinates of a stroke, time, finger pressure, and the screen area covered by a finger are measurements that could be used to classify said user. Zheng et al. [45], operating on similar principles, have studied behavioral authentication using the way a user touches the phone – the features extracted included acceleration, pressure, size, and time. Bo et al. [3] performed recognition by mining coordinates, duration, pressure, vibration, and rotation. Cai et al. [8] examined six different features (e.g. sliding) and compared data such as the speed, sliding offset, and variance between finger pressures. De Luca et al. [11] developed a system for authentication by drawing a template 2D gesture on the back of a device using two phones connected back to back. The security of the gesture is analyzed through various methods by an attacker to replicate the original biometric or graphical password – there is no analysis performed as to the security content of the gesture, just its difficulty to be reproduced. Shazad et al. [35] worked on a template-based touchscreen recognition system on smartphones where they used distinguishing features of a gesture other than the shape to recognize users. To that end, they selected (besides the coordinates) features like finger velocity, device acceleration, and stroke time. The motivation here was to create a system where a gesture could not be stolen by sight alone. Their password space, however, is limited to ten gestures and is thus not user-generated or free-form.

**3D gesture authentication schemes.** 3D gesture recognition can be performed, most recently, using camera-based systems (e.g. Kinect) [38] or using wireless signals [28]. With the camera-based systems, a user would trace a gesture out in space and the image gets compressed into a two dimensional image and processed for recognition [38]. Pu et al. [28] have shown that three-dimensional gestures can be recognized by measuring the Doppler shifts between transmitted and received Wi-Fi signals.

**Graphical and text-based passwords security and memorability.** Bonneau et al. [5] have studied alternatives to text-based passwords for web authentication and how to comparatively evaluate them. There has been considerable work on cued graphical passwords, a survey is offered by Biddle et al. [2] for the past twelve years. In particular, there has been analysis on how Draw a Secret (DAS) [19] type of graphical passwords measures up to text-based passwords in terms of dictionary attacks [26]. Oorschot et al. [26] go on to describe a set of complexity properties based on DAS passwords and conclude that symmetry and stroke-count are key in how complicated a DAS-password can be. They do not provide a direct measurement of this for DAS-password, the analysis is restricted to constructing a model to perform a dictionary attack and show that there are weak password subspaces based on DAS symmetry. For click-based graphical passwords (e.g. PassPoints [40]), Thorpe et al. [37] found they could seed attacks based on human choices and find hotspots for dictionary attacks. For text-based passwords Florencio et al. [14] studied people's web password habits, and found that people's passwords were generally of poor quality, they are re-used and forgotten a lot. Yan et al. [42] were among the first to study empirically how different password policies affect security and memorability of the text-based passwords. Chiasson et al. [9] conducted laboratory studies on how people recall multiple text-based passwords compared to multiple click-based graphical passwords (PassPoints [40]). They found that the recall rates after two weeks were not statistically significant from each other. Everitt et al. [12] analyzed the memorability of multiple graphical passwords (PassFaces [2]) through a longitudinal study and found that users who authenticate with multiple different graphical passwords per week were more likely to fail authentication than users who dealt with just one password.

**Security Analysis of Graphical Passwords and Gestures.** Most security analysis focus on preventing shoulder surfing attacks from hijacking a graphical password or gesture [43, 33, 11]. The methods depend on implementing techniques to make the input more difficult to attack (e.g. making the graphical password disappear as it is being drawn [43]). Another team designed an algorithm based on Rubine [29] that told users whether or not their gestures are too similar, although the metric for this is inherently based on the recognizer's scoring capabilities and not on a measure of the gesture by itself [22]. Schaub et al. [33] suggest that the size of the password space for a gesture is based on three spaces: design features (how the user interacts with the device), smartphone capabilities (screen size, etc.), and password characteristics (existing metrics of security, usability, etc). Security in this context refers to a measured resistance to shoulder surfing.

Continuing with security analysis, brute force attacks on gestures have been examined in some studies [38, 44, 2]. Zhao et al. [44] have examined the security of 2D gestures against brute force attacks (assisted or otherwise) when using an authentication system where a user will draw a gesture on a picture. A measure of the password space is developed and an algorithm under which a gesture in that space can be attacked. The attack is capable of guessing the password based on areas of the screen that a user would be drawn towards. This study does not concern itself with the security of the gesture drawn, instead it is focused on where a user would target in a picture-based authentication schema – it does not address free-form gesture authentication. Serwadda et al. [34] showed that authentication schema based on biometric analysis (including one by Frank et al. [15]) can be cracked using a robot to brute force the inputs using an algorithm that is supplied swipe input statistics from the general population.

Finally, on non-security related work, Oulasvirta et al. [27] studied the information capacity of continuous full-body movements. Our metric is motivated by their work. Specifically, they did not study 2D gestures or their security and memorability or use for an authentication system. When asked to create gestures for non-security purposes, previous work [17, 25] indicates that people tend to repeat gestures that are seen on a daily basis and are context-dependent (e.g. that the gestures people perform are dependent on whether they are directing someone to perform a task or receiving directions on a task).

# 3. SECURITY OF GESTURES

In this section, we present our novel information-theoretic metric for evaluating the security and memorability of gestures. We briefly discuss why existing entropy-based metrics used to evaluate discrete text-based passwords [4] are not suitable for gestures, and move to present our metric for security and memorability of continuous gestures. We have modified a recent metric on analyzing information capacity of full-body movements [27] to estimate the security of a multitouch gesture.

Multitouch gestures on a touchscreen surface produce trajectory data where the positions of one or more end-effectors (finger tips) are tracked over time. The continuous and multi-dimensional nature of multitouch gesture data poses some additional challenges for defining the information content compared to regular text-based passwords that only gauge information in *discrete movements* corresponding to key events (pressing the key down) caused by a single end-effector (e.g. finger, cursor) at a time. *Multitouch gestures* involve multiple end-effectors and continuous movement. To our knowledge, no information theory based security metric has been proposed for multitouch gestures as passwords.

The core idea is to demonstrate that there is an association between the security of a gesture password and the *information content* of the gesture. Intuitively, information content is a property of a message or a signal (such as a recorded gesture): it measures the amount of surprisingness, or unpredictability, of the signal with the important additional constraint that any surprisingness due to random (uncontrolled) component in the signal is excluded. Information-theoretically, the surprisingness of a message, or more precisely, of a source generating messages according to a certain probability distribution, can be measured by the entropy $H(x)$ associated with the random variable, $x$, whose values are the messages. For instance, the surprisingness of a key stroke chosen uniformly at random among 32 alternatives is $\log_2(32) = 5$ bits; five times that of an answer to a single yes–no question. A similar measure of surprisingness, differential entropy, can also be associated to continuous random variables, but it lacks the same meaning in terms of yes–no questions. For in-depth definitions of the used information-theoretic concepts and their properties, please see e.g. Cover and Thomas [10].

For text-based passwords, in the practically untypical case where a password is chosen uniformly at random, the relationship between entropy and security is straightforward. If the alphabet size is denoted by $|\mathcal{X}|$ and the password is of length $n$, then the entropy is given by $H(X) = n \log_2 |\mathcal{X}|$ in the uniformly random case. It then holds that the probability that an uninformed guess is successful is $2^{-H(X)} = |\mathcal{X}|^{-n}$ and the expected number of guesses required to guess correctly by an uninformed attacker is $(2^{H(X)} + 1)/2 = |\mathcal{X}|^n/2 + 1/2$, about half the number of possible passwords of length $n$. However, when the password is not chosen uniformly at random, the entropy has no direct relationship with security and the required number of guesses can vary significantly between different kinds of password generation strategies even if they have exactly the same entropy [4]. Despite this, the entropy $H(X)$ is used as a standard measure of password security even in the non-uniform case [7]. More accurate definitions that take into account the shape of the password distribution have been proposed, see Bonneau [4].

For continuous passwords, one needs to take into account the tolerance for repeatability and recognizability. From an information theoretic point of view, this aspect is captured by the concept of *mutual information*. The mutual information $I(x\,;\,y)$ of two random variables, $x$ and $y$, gives the reduction in the entropy of one random variable when another one becomes known:

$$I(x\,;\,y) = H(x) - H(x\,|\,y), \tag{1}$$

where $H(x\,|\,y)$ denotes the conditional entropy. This applies to both discrete as well as continuous variables. The mutual information characterizes the information capacity of noisy channels with the interpretation that $x$ denotes a transmitted signal and $y$ denotes the signal received at the other end of the channel. In the context of the present work, the transmitted signal is the intended gesture and the received signal is the recorded or repeated gesture. The more noisy the channel (less accurately repeated gesture), the lower the mutual information, $I(x\,;\,y)$, and the capacity of the channel, and *vice versa*. Note that a message can have high entropy (complexity) without the mutual information (information content) being high but not the other way around.[1]

The exact operational meaning of mutual information from a communication point of view is that given a channel for which the mutual information between its input and its output equals $I(x\,;\,y)$, it is possible to reliably transmit information at the rate of $I(x\,;\,y)$ bits per use of the channel. In other words, we can communicate a choice among $2^{I(x\,;\,y)}$ different messages so that the receiver can reliably recover the intended message. Getting back to the question of password security, if the mutual information between the intended and the recorded gesture is $I(x\,;\,y)$, and a gesture password is chosen uniformly at random among the $2^{I(x\,;\,y)}$ mutually distinguishable choices, we obtain a result analogous to the case of text-based passwords: the probability of success of an uninformed guess is given by $2^{-I(x\,;\,y)}$ and the attacker will need on the average $2^{I(x;y)-1} + 1/2$ attempts before the guessed gesture is identified as the correct one. Each bit of mutual information will double the effort by the attacker. Thus, the mutual information defines the *effective key-length* [6, 4] of gesture passwords, so that their security becomes directly comparable to the security of text-based passwords.

In practice, it can be questioned whether users will typically choose gesture passwords uniformly at random [37, 44], although the problem is hardly as pronounced as in the case of text-based passwords where a significant portion of passwords are vulnerable to simple dictionary attacks [4]. Another issue that interferes with the above counting argument is the design of the recognizer that is used to accept or reject a gesture. The information-theoretic limit provides an upper bound on the effective key-length. The better the recognizer, the closer to this limit the system will be able to operate.

Based on the above insight, we can now ask what makes a gesture password secure. Recalling Eq. 1, we see that the two factors affecting the mutual information are $i)$ the entropy (uncertainty or surprisingness) of the intended gesture, $H(x)$, and $ii)$ the conditional entropy (remaining uncertainty) of the intended gesture given the observed gesture, $H(x\,|\,y)$. These two factors correspond to the complexity and the accuracy of the gesture, respectively.

The metric we use for the information content in repeated gestures is defined as the mutual information between two realizations of the same gesture. The input to the metric consists of two multitouch movement sequences produced by asking a user to produce a gesture and repeat it.

The first realization will be used to represent the intended gesture, $x$, and the second realization will be denoted by $y$. The trajectories record the locations of each of the used fingers over duration of the gesture. In order to estimate the mutual information $I(x\,;\,y)$ between two multitouch trajetories, which describes the security of the gesture in the information-theoretic sense described above, a number of steps are required.

**Computation.** Computation of the mutual information involves a sequence of steps. First, we need to remove from the sequences their predictable aspects, as far as possible. To do so, we fit a second order autoregressive model for both of the sequences separately. For sequence $x$, the model is:

$$x_t = \beta_0 + \beta_1 x_{t-1} + \beta_2 x_{t-2} + \varepsilon_t^{(x)}, \tag{2}$$

where $\beta_0, \beta_1$ and $\beta_2$ are parameters that we estimate using the standard least-squares method. The benefit of a second-order model is its interpretability; it captures the physical principle that once the movement vector (direction and velocity) is determined, constant movement contains no information.

After parameter fitting, we obtain residuals $r_t^{(x)}$ for each frame $t$:

$$r_t^{(x)} = x_t - \hat{x}_t = x_t - (\hat{\beta}_0 + \hat{\beta}_1 x_{t-1} + \hat{\beta}_2 x_{t-2}) \tag{3}$$

The residuals $r_t^{(x)}$ correspond to deviations from constant movement and are hence the part of the sequence unexplained by the autoregressive model. They can be used to gauge the surprisingness of the trajectory. The same procedure is carried out for sequence $y$. We could now compute the differential entropy of the residual sequences, but as stated above, it alone has little meaning and we are in fact only interested in the mutual information between the two sequences.

Before we compute the mutual information, *dimension reduction* is performed whereby multitouch gestures are represented using only as many features per measurement as the data requires. The motivation for this step is that one cannot simply add information in the movement features in multitouch gestures together. Instead, any dependencies between the fingers should be removed. Intuitively, a multitouch gesture with all fingers in a fixed constellation contains essentially the same amount of information as the same gesture performed using a single finger. Dimension reduction is performed using principal component analysis (PCA), which removes any linear dependencies. Following common practice, the number of retained dimensions is set by finding the lowest number of dimensions that yields an acceptably low reprojection error (e.g. mean square error [27]).

Once the movement features have been processed by a dimension reduction technique (PCA), we treat them independently which amounts to simply adding up the information content in each feature in the end. Hence, the following discussion only considers the one-dimensional case where both $x$ and $y$ are univariate sequences.

Another issue in gesture data is that the two gestures $x$ and $y$ are often not of equal length due to different speed at which the gestures are performed. This can be corrected by *temporally aligning* the sequences using, for instance, Canonical Time Warping [46]. The result is a pairwise alignment of each of the frames in $x$ and

[1] In fact, to be precise, the inequality $I(x;y) \leq H(x)$ which follows directly from Eq. 1, holds only for discrete signals, such as text-based passwords, but not for continuous signals, such as gestures, because continuous signals can have negative (conditional) differential entropy [10]. However, even though there is no theoretical guarantee of it, the intuition that a trivial gesture such as a straight line cannot contain high information content holds true in our experiments; see below.

$y$ achieved by duplicating some of the frames in each sequence. These duplicate frames are skipped when computing mutual information to avoid inflating their effect.

Finally, we form pairs of residual values $(r_t^{(x)}, r_t^{(y)})$ corresponding to each of the frames in the aligned residual sequences and evaluate the mutual information. Since the mutual information is defined for a joint distribution of two random variables, we model the residual pairs $(r_t^{(x)}, r_t^{(y)})$ in each frame $1 \leq t \leq n$ using a bivariate Gaussian model, under which the mutual information is given by the simple formula

$$I(x\,;\,y) = -\frac{n}{2}\log_2(1-\rho_{x,y}^2), \tag{4}$$

where $\rho_{x,y}$ is the Pearson correlation coefficient between $x$ and $y$.

By substituting the sample correlation coefficient, $r$, estimated from the data in place of $\rho_{x,y}$ and subtracting a term due to the known statistical bias of the estimator (see [27]), we obtain the mutual information estimate

$$\hat{I}(x\,;\,y) = -\frac{n}{2}\log_2(1-r^2) - \log_2(e)/2, \tag{5}$$

where $\log_2(e) \approx 1.443$ is the base-2 logarithm of the Euler constant.

The total information content in the gesture, based on two repetitions, is estimated as the sum of the mutual information estimates in each of the movement features after dimension reduction.

**Summary.** The metric has some appealing properties to serve as an index of security. First, a distinctive feature of our framework that sets it apart from the work on text-based password security is that in dealing with continuous gestures, it is imperative to be able to separate the complexity due to intended aspects of the gesture from that due to its unintended, and hence non-reproducible, aspects. This is the main motivation to use mutual information as a basis for the metric. Second, as mutual information under the bivariate Gaussian model is determined by the correlation between the movement sequences (residuals), it is invariant under linear transformations such as change of scale, translation, or rotation. Hence, the user need not remember the exact scale, position, or orientation of the gesture on the screen. The metric is also independent of the size and the resolution of the used screen unless, of course, the resolution is so low that important details of the gesture are not recorded. Third, the time warping step ensures that variation in the timing within the gesture has only slight effect on the metric. Fourth, the metric enables comparison between gestures of unequal lengths and between single-finger and multi-finger gestures, as well as across different screen sizes and resolutions, on a unified scale (bits).

# 4. METHOD

Our study design builds on a generate-test-retest paradigm where participants were first asked to create a gesture, recall it, and recall again during the second session, a minimum of 10 days later. Participants were told that they should generate secure gestures as they would do in real situations and that their ability to recall them would be tested later. They were not given any hints about what a secure gesture might be. For understanding the generation and recall process, we used a mixed method approach: after generating a gesture, all participants filled a questionnaire on workload (NASA-TLX [18]) after each task and a short survey at the end of the second session.

We note that a somewhat similar generate-test-retest design has been used before by Chiasson et al. [9] to compare multiple password inference to recall between text-based and graphical passwords (PassPoints [40]). However, our work differs for the following reasons: we use TLX forms, focus on free-form multitouch gestures, include more repetitions and recalls, do not require a separate login phase, and postpone questions to the end of a trial.

Next, we describe our volunteer participants, our apparatus, data preprocessing, experiment design and procedure.

**Participants.** We recruited participants with flyers, email lists, and in-person in cafeterias. We required the participants to be 18 years old or over and familiar with touchscreen devices. We recruited 63 participants in all, from the ages of 18 to 65 (M = 27.2, SD = 9.9); 24 are male and 39 are female. Their educational background varies: 22 have high school diplomas, 23 have a Bachelor's degree, 16 have a graduate degree and two have other degrees.

All 63 participants completed session 1 of our study, and 57 of them returned and participated in session 2. As compensation, participants received $30 for completing the whole study. They also participated in a raffle of three $75 gift cards.

We recruited our participants in two batches: first in May 2013 (33) and second in June 2013 (30). Further, in order to analyze the effect of varying time on recall, the gap between the two sessions of the study varies. The mean time gap for the first participants is 14.53 (SD = 5.81) days and 29.52 (SD = 7.57) days for the second.

Our study was approved by our Institutional Review Board.

**Apparatus.** The gesture data was recorded on a Google Nexus 10 tablet with Android 4.2.2 as the operating system, at an average of 200 frames per second (FPS).

**Preprocessing.** The raw data files were preprocessed using MATLAB. In the preprocessing, each gesture file is resampled to 60 FPS, to reduce the effect of the uneven sample rate.

The resampling was done by cubic spline interpolation, via MATLAB's built in function interp1.m. For the $x$ and $y$ coordinates for each finger, it takes the recorded timestamps and resamples to a constant rate. The reduction from 200 to 60 FPS takes into account a large amount of duplicate data created by the touchscreen, and is necessary to prevent artifacts in the resampling.

FFT analysis showed that the frequency reduction combined with the cubic spline method results in low pass filtering of the data, removing high frequency jitter introduced by the touchscreen hardware, but preserving the low frequency content of the gesture data. To deal with the uneven sample rate in the non-interpolated data, the Lomb-Scargle [32] method was used.

At this stage, artifacts in the raw data were detected and corrected as well. The primary such artifact is when a participant fails to place their fingers on the touchscreen in the same order between consecutive trials. As fingers are numbered sequentially upon being detected by the touchscreen, this places them out of order, resulting in artificially low scores in the later analysis. This was corrected by comparing the starting coordinates of each finger, and correcting the order to be consistent.

**Experiment Design.** The experiment followed a 17 x 2 mixed factor design with a repeated measurement variable of gesture repetition (17 levels) and a between-subject variable of time gap between sessions (2 levels). In the 17 gesture repetitions, 10 were performed during the creation process, followed by another 2 repetitions after a short distraction, and 5 were performed in the second session.

**Procedure.** We conducted a two-session study. The second session was held after a minimum 10 days after the first session. The details of the procedure were as follows.

***First session:*** First, the participants were introduced to the study, which included reading and signing the consent form, discussion of their rights as participants and how they will be compensated.

**Gesture Creation (Generate):** Each participant was given the same tablet and was asked to create what they thought would be a secure gesture by drawing on it. The participants were asked to generate a gesture that they think others could not guess, but they could also remember later. The participants could retry until they felt satisfied with their gesture. Then the participant repeated the same gesture for an additional nine times on the same tablet. Participants were presented with a blank screen for drawing their gestures. The application did not limit the number of fingers participants use to create the gesture. However, the number of fingers used could not be changed during the drawing process, that is, the gesture has to be drawn continuously without lifting any fingers from the screen. Once it is completed, the gesture is displayed on the tablet's screen as a colored curve line, as shown in Figure 1. The display was checked visually, to verify that the gesture was recorded properly.

**Subjective Workload Assessment:** The participants were asked to fill out NASA-TLX form regarding the creation process.

**Distraction:** The participants were asked to perform a mental rotation task and count down from 20 to 0 silently.

**Gesture Recall 1:** The participants were asked to recall the gesture by repeating it twice on the same tablet using the same application.

**Demographic questions:** The participants were asked usual demographics questions that were aggregated and reported above.

***Second session:***

**Gesture Recall 2:** The participants were asked to recall their gestures by repeating it for five times on the same tablet using the same application.

**Subjective Workload Assessment:** The participants were asked to fill out a NASA-TLX form regarding the recall process.

**Short Survey:** The participants were asked some questions about the recall process and other thoughts about the study.

## 5. RESULTS

For each participant who completed both sessions, a total of 17 gesture repetitions were recorded. In all, 1038 recordings were generated, as six participants did not attend session 2, and three additional traces were not completed. The groups of repetitions are summarized in Table 1. Because the estimated Mutual Information ($\hat{I}$), is computed in pairs, $\hat{I}$ is reported as the mean for the relevant repetitions.

| Session | Trial # | Group |
|---|---|---|
| 1 | 1-10 | Generate |
| 1 | 11-12 | Recall1 |
| 2 | 13-17 | Recall2 |

**Table 1: Repetition Groups by Session # and Trial #. $\hat{I}$ was computed for all pairs of repetitions each group.**

### 5.1 Factors Affecting Security

Figure 2 shows the mean $\hat{I}$ of each repetition across all gestures versus the repetition number. During gesture creation in Generate, $\hat{I}$ trended upwards from repetitions 1-4, and then leveled off from repetitions 5-10. This shows that it takes at least three repetitions for a participant's gesture to become stable.

A second major feature is that by Recall1, repetition 11 and 12, the $\hat{I}$ has dropped suddenly, despite a delay of only a few minutes. Surprisingly, more than 10 days later, $\hat{I}$ did not drop much further in Recall2. The drop between Generate and Recall1 was more severe for single finger gestures, and the drop between Recalls 1 and 2 was more severe for multifinger gestures. In both cases, $\hat{I}$ stabilized, at around 27 bits for single finger and 15 bits for multifinger gestures, having dropped from an initial value of around 35 and 20 bits respectively.

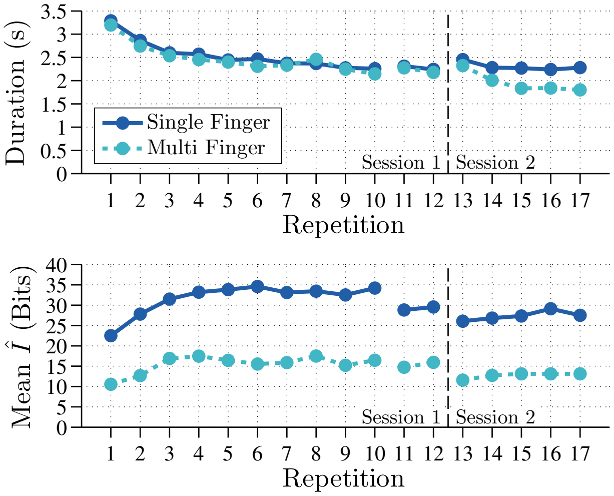


**Figure 2: Mean $\hat{I}$ and mean gesture duration vs repetition. Top: Within Generate, mean gesture duration trended downwards as gestures increased in speed. Bottom: Over the same repetitions, $\hat{I}$ trended upwards before leveling out. It then dropped quickly between Generate and the two Recalls.**

Figure 2 also shows the amount of time taken to record each repetition of each gesture. This duration also changed with repetition. As the number of repetitions increased, the mean duration of each repetition trended downwards from around 3 seconds to around 2. Unlike $\hat{I}$, it remained stable from there through the two recalls. Interestingly, during Recall2 the multifinger gestures sped up from 2.5 seconds to under 2 seconds, whereas the single finger gestures stabilized.

A plot of the mean $\hat{I}$ of each gesture versus mean duration appears in Figure 3. This shows that many gestures with a short duration also had a low $\hat{I}$. The highest $\hat{I}$ gestures had a duration of between 2 and 5 seconds. $\hat{I}$ increased with duration, but had a poor fit, explaining only 5% of the variation, as the highest duration gestures were either very high or very low $\hat{I}$. Long duration could thus indicate either a complex, careful gesture, or a relative lack of practice.

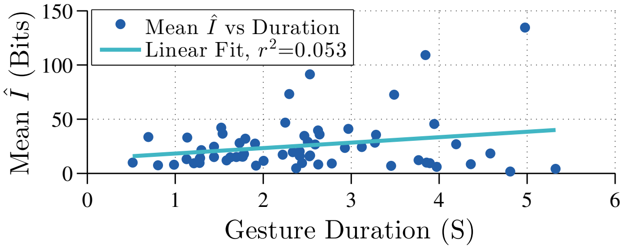


**Figure 3: Mean $\hat{I}$ vs. mean gesture duration. The $r^2$ value indicates a poor linear fit, showing little correlation. However, the highest $\hat{I}$ gestures were all longer than 2 seconds, suggesting that some degree of precision is required.**

Figure 4 shows that majority of the user-generated gestures had relatively low $\hat{I}$ with only a small tail of high $\hat{I}$. For Generate, the distribution had a mean of 27.72 bits, and a standard deviation of

26.30 bits. However, taking only the second, stable half of Generate, the mean rose to 33.42 bits, with a standard deviation of 31.13 bits. In both cases, the standard deviation was about the same size as the mean, indicating a large variability. The histogram shifted as well, becoming slightly more uniform. Although many user chosen gestures score poorly, some scored highly as well, suggesting that it is possible to create guidelines to emulate the characteristics of high scoring gestures.

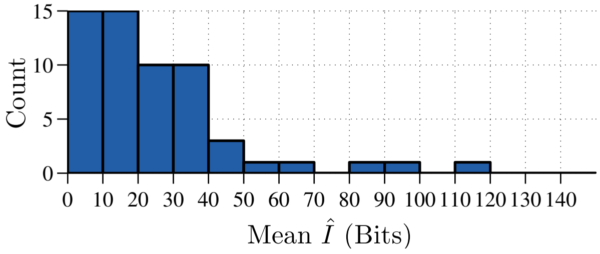


**Figure 4: Histogram of mean $\hat{I}$ of Generate, per gesture, showing low scoring, biased distribution.**

We compared mean $\hat{I}$ with participant age and gender. There was a mild negative relationship between $\hat{I}$ and age, with a $r^2$ of 0.08. Seven of the top eight gestures were created by participants under the age of 25, with the remaining one under the age of 30. Mean $\hat{I}$ for male participants (N=23) was 29.82 bits and mean $\hat{I}$ for female participants (N=40) was 25.00 bits.

Finally, we looked for defining visual characteristics of the highest and lowest scoring gestures. We ranked each gesture by its $\hat{I}$ in each of the five categories, and evaluated the top five in each category. There was a high correlation between the categories, and as such, the top five gestures for each category overlapped significantly, having only nine unique gestures. The best gestures fell into two groups, angular paths with many hard turns, and signatures. This matched our expectations, as the algorithm looks for both consistency between trials and for large deviations from a straight line. The defining visual feature of the lowest scoring gestures was having only a few, gentle curves. Many were multifinger, with the additional fingers merely copying the motion of the first. A gallery appears in Figure 5.

## 5.2 Security of Multitouch Gestures

As seen in Figure 2, the mean $\hat{I}$ of multifinger gestures is lower than that of single finger ones. We compared $\hat{I}$ of these gestures to estimate how much additional information is added by additional fingers. Figure 6 shows the higher mean $\hat{I}$ of single finger gestures, and the rarity of gestures using more than two fingers. Recall2 showed a greater difference in $\hat{I}$ than Generate, as a number of participants failed to use the same number of fingers when they returned in session 2. Of the 63 participants, 32 decided to create multifinger gestures, and 31 chose to create single finger gestures, with only three participants using more than two fingers. Participants were prompted that they could use as many fingers as they liked, but were not instructed on how many to use.

We performed regression analysis on the effect of the number of fingers on $\hat{I}$. The result shows that the effect is significant for the $\hat{I}$ of Generate, $b = 17.948, t(57) = 2.763, p = .0077$, while not for the $\hat{I}$ of Recall2, $b = 11.898, t(57) = 1.841, p = .07$. However, the regression model only explained 11.8% of variance in the $\hat{I}$ for the significant effect. In short, the number of fingers is not the most major factor affecting $\hat{I}$.

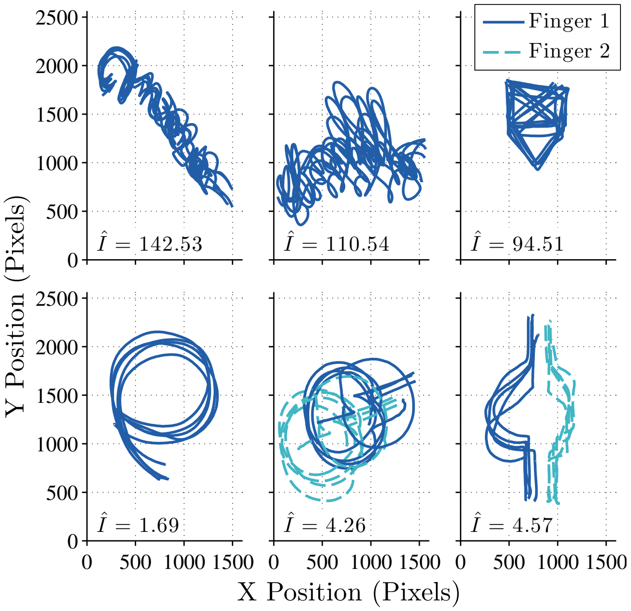


**Figure 5: Gestures ranked by $\hat{I}$ for Generate and Recall2. Top: Best three gestures, showing the many tight turns characteristic of high scoring gestures. Bottom: Worst three gestures, Low scoring gestures had few, gentle turns.**

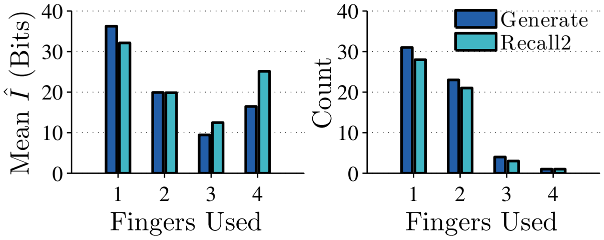


**Figure 6: $\hat{I}$ and number of fingers used to perform gesture, and change between sessions 1 and 2. This shows the both the higher performance of single finger gestures, as well as the rarity of using more than two fingers.**

## 5.3 Factors Affecting Memorability

Figure 7 shows the best remembered gestures. To evaluate memorability, we computed the cross-group $\hat{I}$ of Generate and Recall2, with pairs consisting of one from each group, instead of both from within a group. However, the large differences in $\hat{I}$ from gesture complexity obscured the differences from repetition accuracy, as the cross-group $\hat{I}$ has a linear fit with an $r^2$ of 0.65 with the $\hat{I}$ of Generate. We compensated by dividing the mean cross-group $\hat{I}$ for each gesture by its $\hat{I}$ from Generate.

Gestures that scored highly on the resulting ratio have the best consistency between Generate and Recall2, as compared to the consistency within Generate. The top gestures for memorability are shorter and simpler than the top gestures for security.

Once we had a way of comparing how well gestures are remembered, we investigated what might cause the large difference in $\hat{I}$ between sessions. We compared the memorability ratio to the time interval between Generate and Recall2, as seen in Figure 8. The

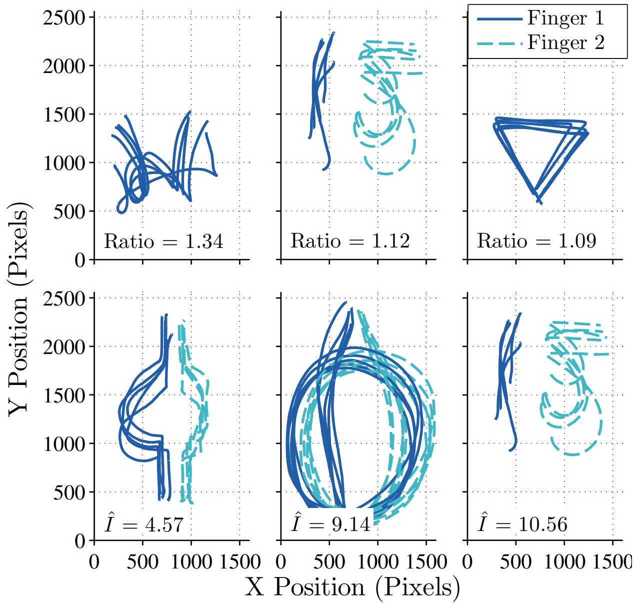


**Figure 7: Top: Best 3 Gestures by memorability. These have a shorter length and decreased complexity compared to high $\hat{I}$ gestures. Bottom: These 3 gestures had the greatest difference in path between fingers. Two of the three are a simple mirroring of the path (Left, Middle), while only (Right) adds a large amount of $\hat{I}$.**

linear fit however, had a $r^2$ of less than 0.03, showing minimal dependence on the delay between sessions.

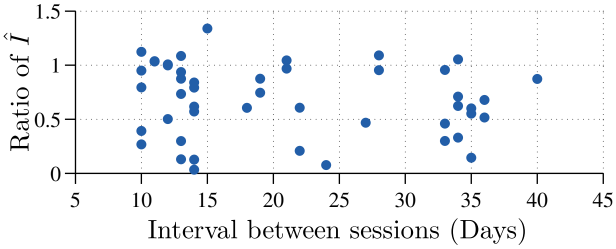


**Figure 8: Memorability vs interval between sessions. This plot compares the relative quality of gesture recall versus time between Generate and Recall2. There was no significant correlation with a ($r^2$ < 0.03).**

## 5.4 Individual Differences

Given the surprising deficiency of the multifinger gestures, we looked at specific examples. Only three participants used multitouch gestures with significantly different motions between fingers, and in two of the three cases they were simple mirrorings of the motion. These three gestures appear in Figure 7. All other cases were just translations of the same trace, as if the gesture were made with a rigid hand. Gestures with minimal additional information per finger were in part scored low because all gestures were run through a PCA algorithm to remove redundant information, prior to analysis of $\hat{I}$.

Participants also commonly performed several categories of error. Despite instructions, 19 of the 32 multifinger using participants placed their fingers down in an inconsistent order between sequential trials. This was fixed in preprocessing to ensure that computation of $\hat{I}$ used the same fingers. In addition, some participants rotated the tablet either during or between sessions. This stands out visually when comparing traces, but the metric is negligibly affected by rotations. This was verified by comparing the result with its gesture counterpart from session one or two which did not have a rotation. Some of the lowest scoring gestures featured a rotation between repetitions, and all gestures with rotation between sessions scored poorly on cross-session $\hat{I}$. However, the gestures with cross-session rotation also scored poorly within a session. Only one gesture with rotation between sessions scored significantly higher on its within session $\hat{I}$ than on its cross-session.

Given that error containing gestures scored poorly, even when excluding those repetition pairs containing the errors, we take these errors as an indicator of poor recall. Extra fingers or complex shapes are no guarantees of a high score, without consistent execution. Attention when creating the gesture is thus important, practicing accurate repetition rather than just going through the motions.

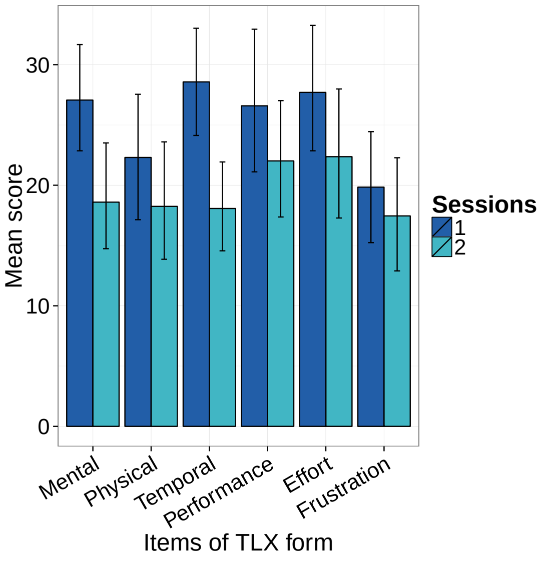


**Figure 9: Mean score and corresponding 95% confidence interval for each item on the TLX form, for the two sessions of our study. The mean scores of each item in session two were lower than those of session one.**

## 5.5 Subjective Task Load

This section contains the analysis of the study's TLX forms. For each session of our study we asked participants to fill out one TLX form, which is used to assess subjective workload of given task. Figure 9 shows the mean score and corresponding 95% confidence interval for each item in the TLX form, for both sessions.

Figure 9 suggests that the mean scores of session two are lower than those of session one. To prove the point we conducted a non-parametric repeated-measure Wilcoxon signed-rank test on the scores of each item from the two sessions, given the fact that the data does not follow a normal distribution. The result is shown in Table 2.

From Table 2 we can see that except for Frustration, all items show significant differences in scores between sessions. It is safe to say that given the data we have, it is very likely that participants felt the recall task was easier than the creation task in terms

| Item | Mental | Physical | Temporal |
|---|---|---|---|
| $p$ value | <0.001 | 0.025 | <0.001 |
| effect size | -.038 | -0.21 | -0.34 |
| Item | Performance | Effort | Frustration |
| $p$ value | 0.029 | 0.0013 | 0.072 |
| effect size | -0.20 | -0.30 | -0.17 |

**Table 2: Subjective task load components compared between the two sessions. Statistical significance calculated using Wilcoxon signed-rank test. The difference was significant at the 0.05 level for all components except Frustration.**

of workload. This could be because of increased practice or familiarity, and intuitively agrees with the increase in gesture speed seen in Figure 2.

## 6. PRACTICAL AUTHENTICATION SYSTEM IMPLEMENTATION

In this section, we describe our extension to a practical single touch gesture recognizer for multitouch gestures, and the results of how the participants' gestures would perform in a real authentication system based on our multitouch recognizer. We also present our trial on shoulder surfing attacks that indicates how free-form gestures are robust against shoulder surfing.

A recognizer works by taking a user's gesture, passing it through a recognition algorithm, and computing whether or not the gesture is a successful match for a stored template. The device will store a series of templates of which the gestures are compared to for authentication. The best score is used and compared to a threshold value. We have the following assumptions for the recognizer: 1) location invariance: No matter where the correct gesture is drawn on the screen, it should be authenticated correctly. 2) scale invariance: No matter what size the correct gesture is drawn to on the screen, it should be authenticated correctly. 3) rotation invariance: No matter what angle the correct gesture is drawn at on the screen, it should be authenticated correctly. Location and scale invariance are important when dealing with cross-platform authentication. The screen dimension inherently limits what size the gesture can be drawn to and the area over which a gesture can be performed would cause wild variations in where it would be drawn depending on the user. Rotation invariance is useful for reducing computational complexity when dealing with individualized free-form gestures as we have in our data set. We note that authentication system designers can opt to restrict or relax these assumptions. Scale invariance, for example, is not inherently necessary. The size of a gesture can be a feature of that gesture depending on how the recognizer is implemented.

We elected to implement and extend the Protractor [21] recognition algorithm, a popular nearest neighbor approach. Given the gesture templates obtained and the two recall sets, we would like to measure how well the gestures perform. Protractor is an improvement upon the $1 Recognizer [41], having both a lower error rate [21] and an effectively constant computational time per training sample as compared to $1's growing cost per training sample. Protractor presents itself further as an attractive algorithm for the data under consideration since it has low computational complexity compared to other techniques, for example, Dynamic Time Warping (DTW) [41] and Hidden Markov Models (HMM) [13, 24]. In general, Protractor's error rate falls with an increasing number of training samples and at 9-10, the error rate is less than 0.5% [21].

Below we describe first the single touch Protractor and we follow with our extension of it to multitouch gestures.

Upon the input of a gesture, the algorithm splits the work into four parts:

1. A gesture is resampled into N equally spaced coordinates.

2. The gesture is translated to the origin of the plane.

3. After translation, the angle between the first point of the gesture and the origin is measured. Then, the entire gesture is rotated until that angle becomes zero degrees.

4. To minimize the distance between the gesture and a template, they need to be aligned such that the angle between them is the smallest. This angle is easily calculated as being the dot product between the gesture and the template. The gesture is subsequently rotated by the dot product angle.

5. The cosine distance is then measured between the template and the gesture. The inverse of this distance defines the recognition score. The smaller the cosine distance, the better the match (and correspondingly, a larger score).

6. The score is compared to a threshold value. If it is greater than the threshold, the gesture and the template are said to match. If it is less than the threshold, they are said not to match.

Again, Protractor is a single touch recognition algorithm. Other projects considering multitouch gestures have used more general techniques for gesture recognition instead of a nearest neighbor approach. For example, Sae-bae et al. [31] applied DTW to deal with their multitouch gesture set. For the authenticator to remain practical, it needs low computational complexity, high speed, and low error rate per template to be implemented on a mobile device – Protractor can meet this demand. As we are dealing with multitouch gestures, it is necessary to modify Protractor. The accounting procedure is as follows:

1. Each finger is split into its own set of points and passed through the algorithm and compared to templates of similar fingers and the score is computed for each individually.

2. They are then averaged together.

3. There are provisions built into place to ensure the authentication failure for the wrong number of fingers. In the case of $n$ fingers versus $m$, the number of fingers is compared. If $n$ is equal to $m$, then the recognizer continues to the next step. If $n$ is not equal to $m$, then the recognizer immediately stops the computation and registers the score as 0: a failure.

4. This score is then compared to the threshold value. If the score is greater than or equal to the threshold then it is considered a positive authentication, otherwise, it is negative.

It is important to note that the threshold should be set high enough such that authentication failure is all but guaranteed for gestures that are being matched to templates other than their own.

As a reminder, when the participants began the study, they were asked to repeat their gestures ten times. Each of these ten trials is used by Protractor as templates for that gesture. There are two authentication data sets under consideration here: the first is where participants were asked to replicate their gesture after a distraction and the second where they were asked to replicate their gesture after at least 10 days.

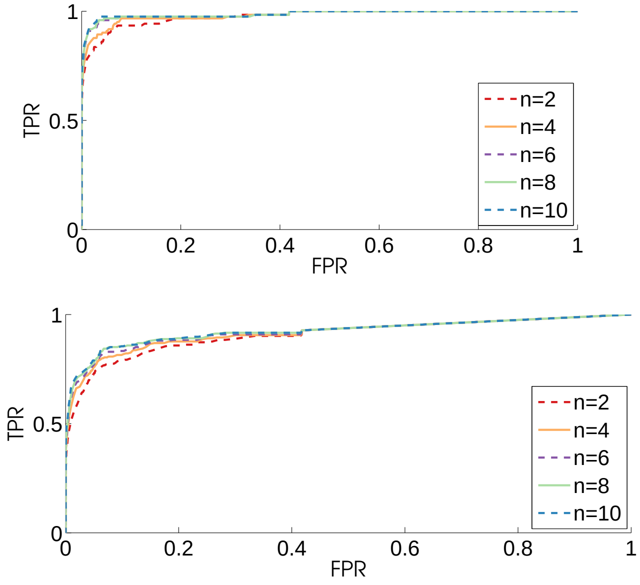


**Figure 10: This figure shows the ROC curve for the recognizer across the two data sets with variable template numbers – 'n' corresponds to the number of templates. The top plot is the first data set and the second plot is the second data set. The ROC curve is a measure of the performance of a binary classifier, the closer the top left corner of the plot moves towards the vertical axis, the better the performance. The first data set is closer to that corner than the second, showing that the second data set performed worse and must have a higher equivalent error rate. As the number of templates increases, the closer the curve shifts up and the better the recognizer performs. In general, the recognizer classified the first data set with a lower EER than the second set despite those being the same gesture types, indicating a weakness on the parts of participants to accurately replicate their gestures. The EER values can be read in Table 3.**

## 6.1 Recognizer Performance

We wanted to know how accurately the recognizer is performing across the different gestures in our data sets. To quantify this in terms of a numerical estimate and visualize it, we elected to obtain a Receiver Operating Characteristic (ROC) curve and derive from it an Equivalent Error Rate (EER). The ROC curve gives the visual representation of how our classifier is performing and the EER value gives us the numerical estimate for how it is performing: the lower the EER is, the more accurately the recognizer is performing.

To find the EER value we need to find the rate at which the True Positive Rate (TPR) is equivalent to the False Positive Rate (FPR). These are defined as:

$$TPR = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{6}$$

$$FPR = \frac{False\ Positives}{False\ Positives + True\ Negatives} \tag{7}$$

These values are dependent on the threshold parameter that the score computed by the recognizer is compared to. To generate an ROC curve, one must vary the authentication threshold parameter and measure a (TPR,FPR) value for each point and plot them. From there, we can determine the EER visually.

At low threshold values, the classification system would be accepting virtually any gesture as a validation against any template gesture. At high values, all input gestures would be rejected against any other template (even if it is being matched to the correct one). As the EER value climbs higher, the reliability of the gesture or the recognizer can be called into question since false positives are propagating through the system. In the case of our system, the templates are not matched to one another. We have two true cases for a given gesture in the first data set and five true cases for a given gesture in the second data set. All other gesture attempts in all other data sets (exclusively across these two sets, no intersections) are considered false cases.

As such, what this means is that the ROC curves and the EER values are across the entire data set and not per gesture. This is done to avoid generating ROC curves and EER values for every one of the different gestures, which would be an overabundance of data that is not altogether useful. The EER and ROC values given here can be thought of as a measure of performance of the recognizer across the entire set rather than the recognizer and a single gesture.

| # of Templates | Set 1 EER | Set 2 EER |
|---|---|---|
| 2 | 7.07% | 15.97% |
| 4 | 6.42% | 14.45% |
| 6 | 4.13% | 13.94% |
| 8 | 4.10% | 13.09% |
| 10 | 3.34% | 13.16% |

**Table 3: EER Values, Ranked by Template Number. Listed above are the EER values corresponding to Figure 10. As the number of templates increases, the lower the EER value drops and thus the lower the error in the system and the better the recognizer performs. The lower Set 2 values correlate to the shape of the curves represented in Figure 10: as EER decreases, the better the curve appears. The EER values for the recognizer reduce more slowly with 6 training templates, indicating this to be the ideal starting point when asking a user to train the system.**

As for how well the gestures and the recognizer performed in terms of accuracy across the data sets, that information can be gleaned from the ROC plots given in Figure 10 and the EER values shown in Table 3. As a reminder, the further away an ROC curve moves from being a 90-degree box (an EER of 0%), the worse it is performing. Figure 10 shows the ROC curves with a varying number of template sizes. As the number of templates increases, the ROC curves are pushed further towards the corner and the EER values are lowered, telling us that a larger number of training templates leads to improved accuracy. For the first set, the best result (with 10 templates) is 3.34% and the best result for the second set is 13.16%. Note that the higher EER on the second data set does not speak to the weakness of the recognizer but rather those of the participants – some participants who returned to attempt their gestures in the second set forgot the number of fingers they used, registering immediate authentication failure. As such, the increased error in the second set as compared from the first set can be attributed to recall problems rather than weaknesses in the recognizer's ability to classify gestures.

## 6.2 Threshold Selection

The final stage of the recognition process, the threshold comparision, determines whether or not a gesture is classified as a positive or a negative. So, naturally, the question becomes: How does one select the optimal threshold? Ideally, a perfect threshold would be set at a point where it is possible to accept only the correct gestures and reject all others.

The starting point for considering the optimal threshold would be the ROC curve and the EER. Recall that the EER is the rate at which the true positive rate equals the false positive rate. The EER value occurs at a specific tuning threshold. Meaning, at this specific threshold, the system rejects the same number of true positives as it accepts false positives. An illustrative example is this: if the EER is 2%, then that means 2 out of 100 true attempts are rejected and 2 out of 100 false attempts are accepted. Selecting a threshold above or below the EER threshold can tighten or relax the admission requirements.

At this point, the threshold selection becomes application dependent. For authenticating into a bank, for example, a 2% false acceptance rate is unspeakable. As such, the threshold would be set at a point far above the EER threshold. When this happens, the false acceptance rate drops drastically but the false rejection rate would increase just as drastically; now there is a threshold where is a 0.01% false acceptance rate but a 10% false rejection rate. This is a tradeoff that a bank could accept.

For most gestures in this set, false positives are not an issue. Individual EER ratings on a *per gesture basis* are quite small, as compared to the larger EER of the gesture set. Selection at this threshold for gestures in the data set would be acceptable for authentication.

## 6.3 Shoulder Surfing Attack Trial

We conducted a preliminary study to understand how free-form gestures would resist shoulder surfing attacks. Towards this end, we recruited seven participants from computer science and engineering schools who had considerable experience with touchscreens. We assume that these volunteers would likely to be more skilled with attacks than the general populace to limit confounding factors. One of the seven volunteers acted as the target who performs gestures and the other six would be attackers who try to replicate target's gestures.

We chose three qualitatively different gestures as shown in Figure 11 as examples. In the experiment, we first had the target of the attacks exercise and get familiar with all three gestures; then we collected gesture data from the target in a way matching the original dataset: for each chosen gesture, the target first repeated ten times; after a short distraction task, which included mental rotation and countdown, the target repeated another two times. Finally, we video recorded one additional repetition the target made for each gesture. Instead of having the target performing gestures in person for every attacker, we played video recordings of that process to the attackers, which ensured attackers would not be affected by any inconsistency or difference within the performance of the target.

During the shoulder surfing process, each attacker was presented with all of the three videos, each of which contained one of the chosen gestures. The attackers were always seated at the same spot, adjacent to the a chair at the table where the display was setup. This was done in order to emulate shoulder surfing. The order of videos played to each attacker was produced in a Latin square to prevent any carry over effect on the attackers' performance. Each video was played only once for each attacker. Then attackers were told to repeat the gesture they observed from the video for five times with the purpose of replicating it as well as possible.

We measured shoulder surfing effectiveness of a gesture using our multitouch recognizer discussed above. The templates we used are the 10 gestures from the Generate phase of the target. Table 4 shows the result. All scores displayed in the table are the maximum score of that category, that is, the best attempt of either the target or attackers to replicate the gesture. Scores that are rendered as a 0 are because of the fact that the recognizer immediately rejects template-gesture matching where the finger count is different.

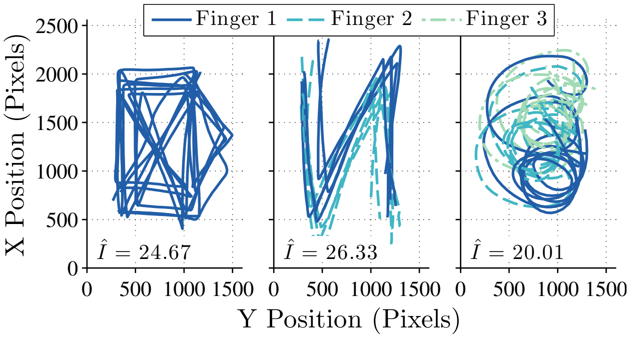


**Figure 11: Gallery of Gestures used for shoulder surfing. In order from left to right represents Gesture1, Gesture2, and Gesture3 generated by the Target.**

| Participant | Gesture1 | Gesture2 | Gesture3 |
|---|---|---|---|
| Target (Recall) | 4.36 | 4.31 | 6.75 |
| Target (Video) | 2.95 | 4.51 | 7.27 |
| Attacker 1 | 0.50 | 0.00 | 0.97 |
| Attacker 2 | 0.43 | 3.08 | 0.00 |
| Attacker 3 | 1.07 | 0.00 | 0.00 |
| Attacker 4 | 0.27 | 0.00 | 0.94 |
| Attacker 5 | 1.10 | 2.19 | 3.96 |
| Attacker 6 | 0.57 | 0.00 | 0.53 |

**Table 4: Table of best scores for each attacker for each gesture. The results show that none of the attacks were successful: the passing score for the recognizer can be set so that the Target can authenticate with ease and the attackers are not authenticated.**

From the Table 4 we see that there is not any overlapping for scores of target and attackers. This indicates our recognizer correctly differentiates attempts made by the target and that by attackers. In general, the target is always authenticating quite well across the the three gestures in comparison to the poorer scores of the attackers. The only opportunity where an attacker became close enough to steal the gesture (Attacker 2, Gesture 2) still has a one point cushion around it – high enough to prevent authentication by an attacker if the threshold is set appropriately.

# 7. DISCUSSION AND CONCLUSIONS

We have presented the first study of using free-form multitouch gestures for mobile authentication. Towards the end of analyzing the security and memorability of the gestures, we presented a novel metric based on computing the estimated mutual information of repeated gestures. We designed and implemented a practical multitouch recognizer as an authentication system, and studied the robustness of free-form gestures against shoulder surfing attacks.

Overall, the results are favorable to user-generated free-form gestures as a means of authentication on touchscreen devices.

Security, as estimated by $\hat{I}$, is high enough for most passwords the users generated. We learned that *multi*finger gestures do not show high security in this measure. It should be noted though, most of multifinger gestures in our dataset are gestures of multiple fingers repeating the same simple shape, for example, drawing a circle with three fingers. We believe that the participants may overestimate the increase in security by merely increasing number of fingers. When they decided to use multiple fingers, they tended to choose a simple shape because they might have believed multiple fingers gave them high security despite simple shape. Such inconsistency in participants' perception and the actual security could be advised against in the password generation user interface.

We also learned that, unlike with the length of a text-based password [39], the duration of a gesture does not play an important role in $\hat{I}$. Intuitively speaking, complex gestures with high $\hat{I}$ should take longer time to perform. However, we learned that even brief gestures can have high security. Gestures with duration less than two seconds have an average $\hat{I}$ less than 2% lower than the average $\hat{I}$ for all gestures.

By looking at our dataset, we found out that some simple shapes, even circles, would actually take more time to complete than complex ones like signatures. The possible reasons warrant further studies. At this point, we suspect that complex gestures are also more difficult to reproduce precisely. A good secure gesture should have both: inherent complexity and easiness to perform. It is interesting in this light that signatures are particularly good and resulted in very high $\hat{I}$. This means, although very complex to perform, participants still managed to repeat them quite well.

When it comes to memorability, the data shows that users need a few repetitions to achieve a stable password. Like with text-based passwords, the generation of passwords is experienced as more demanding by users than recall. After generation, $\hat{I}$ drops after an interval of more than 10 days by about 16%. However, they are still recognizable as unique passwords. In addition, for a participant, the value of $\hat{I}$ varies as they repeat the gesture. By continuously repeating, $\hat{I}$ tends to stabilize. Unlike the text-based passwords, which one has to input exactly, free-form gestures involve many sources of variance, which would be very difficult to keep constant across different attempts. Therefore, one alternative way of reporting security of gestures could be a range similar to confidence interval, instead of an exact value. Moreover, studying gesture variability is a good topic for future research, because a good balance must be found between memorability and security. Of course, it is important to note that the memorability could be in our favor as compared to truly random gestures. We asked the participants to generate gestures that would be repeated after at least ten days. As such, there is an incentive on the participants to create something they will be able to recall after some time.

Several participants were able to create highly secure and memorable gestures. Below, we sketch strategies for generating such gestures. We plan to develop and test the guidelines and their effect on creating gestures in further studies. These guidelines are illustrated with the best and worst gestures in Figure 5, especially with the worst gestures being simple multifinger circular motions.

1. General advice to promote consistency and retention: Practice different gestures first in order to get used to the touchscreen. Try out different gestures instead of picking the first one, to find one you prefer. Pick a gesture that will be used frequently to avoid forgetting it. Take more care and pay attention. Do not rush. Practice until faster and still accurate. Try to repeat each trace as closely as possible.

2. Characteristics of High $\hat{I}$ gestures to emulate: Use many sharp turns. Use a familiar gesture, for example, a signature. Use extra fingers to do different motions. Follow the above rules even when adding fingers.

3. Characteristics of Low $\hat{I}$ gestures to avoid: Do not use only few turns. Do not use gentle turns. Do not make turns only to go in the the same direction. For example: avoid doing a circle.

4. Specific errors to avoid: Place fingers down in the same order each time. Use the same number of fingers each time.

Our results from the recognizer show the capability of free-form multitouch gestures to work as passwords in a practical authentication system. The gestures were classified by the recognizer with relatively low error when being compared across all 63 participants, indicating the ability of the participants to generate passwords that a recognizer would not have trouble classifying when comparing against multiple templates. The recognizer generated much higher scores when evaluating a participant's gesture against their template (ranging from three to nine) as compared to when it compared to other templates (ranging from zero to one). Memorability of the free-form gestures are also displayed through the EER values in recognizer results: the lowest EER value for the first set is 3.34% and the lowest for the second set is 13.16%. The disparity in the data sets can be attributed to two factors: 1) the first data set had only two authentication trials compared to the second set's five trials and 2) the second data set was performed after a much longer time span than in the first set, thus, there were memorability effects between the two sessions. The multitouch recognizer we designed and implemented has room for consideration in the future, for example, the effects of rotation, scale, and position invariance as added degrees of freedom with free-form multitouch gestures.

There can be some cause for remark here about whether or not there is a trade-off between memorability and security of a gesture. Heightened EER values between the first and second recall sessions can give rise to the notion that some participants might have trouble remembering certain gestures, in spite of the $\hat{I}$ for those gestures. However, we note that the errors in recognition extend to the number of fingers – the general shape of the free-form gesture is always almost correctly reproduced. So we contend that the trade-off is in the multitouch aspect and not in the free-form aspect. It appears that remembering the number of fingers is difficult for some participants and we contend this is likely due to a lack of adoption of multitouch as an industry standard. Most people are used to using a single finger for interaction with touchscreens and that might have confused participants in the second session after more than ten days.

With our preliminary shoulder surfing attack trial, we also learned that free-form gestures are relatively robust against shoulder surfing. None of the attackers were able to repeat the gestures well enough to be accepted by a practical authentication system. We acknowledge that further more comprehensive studies with several different kinds of gestures and more opportunities for the attackers would be warranted. For example, we could separate attackers into different groups in which half of them are allowed to rewatch the video recordings as many times they want to.

Thinking towards the future, we wondered what would happen if gestural passwords like these become ubiquitous. How would people begin managing multiple passwords across different platforms? Would there be trouble in remembering the correct password for different applications? If that is the case – what is the best way to manage multiple gestural passwords such that someone can recall them? The obvious text-based analogue is a password manager. Although it is outside the scope of this paper, we can consider the concept of a gestural password manager to be important future work.

Going further, we can think about the effect of screen size on gestural password generation. It can be reasonably assumed that both a person's finger size and the screen size will both change what gestures are used for passwords. All of the participants in this study worked off of a Nexus tablet with a fairly large capture area. Future work could focus on seeing what passwords are generated for varying screen sizes or even to see if the gestures in the sets presented in this paper are comfortable to perform on smaller screens.

To conclude, our work shows that free-form gestures present a robust method of authentication for touchscreen devices.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Number 1228777. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## 8. REFERENCES

[1] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. In *Proc. of WOOT'10.*

[2] R. Biddle, S. Chiasson, and P. Van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Comput. Surv.*, Sept. 2012.

[3] C. Bo, L. Zhang, X.-Y. Li, Q. Huang, and Y. Wang. Silentsense: Silent user identification via touch and movement behavioral biometrics. In *Proc. of MobiCom '13.*

[4] J. Bonneau. The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In *Proc. of IEEE SS&P'12.*

[5] J. Bonneau, C. Herley, P. van Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Proc. of IEEE SS&P'12*, May 2012.

[6] S. Boztas. Entropies, guessing and cryptography. Technical report, RMIT University Research Report Series, 1999.

[7] W. E. Burr, D. F. Dodson, E. M. Newton, R. A. Perlner, W. T. Polk, S. Gupta, and E. A. Nabbus. NIST SP 800-63-1. Electronic Authentication Guideline, 2011.

[8] Z. Cai, C. Shen, M. Wang, Y. Song, and J. Wang. Mobile authentication through touch-behavior features. In *Proc. of Biometric Recognition*, 2013.

[9] S. Chiasson, A. Forget, E. Stobert, P. C. van Oorschot, and R. Biddle. Multiple password interference in text passwords and click-based graphical passwords. In *Proc. of CCS'09.*

[10] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

[11] A. De Luca, E. von Zezschwitz, N. D. H. Nguyen, M.-E. Maurer, E. Rubegni, M. P. Scipioni, and M. Langheinrich. Back-of-device authentication on smartphones. In *Proc. of CHI '13.*

[12] K. M. Everitt, T. Bragin, J. Fogarty, and T. Kohno. A comprehensive study of frequency, interference, and training of multiple graphical passwords. In *Proc. of CHI'09.*

[13] J. Fierrez, J. Ortega-Garcia, D. Ramos, and J. Gonzalez-Rodriguez. HMM-based on-line signature verification: Feature extraction and signature modeling. *Pattern Recogn. Lett.*, Dec. 2007.

[14] D. Florencio and C. Herley. A large-scale study of web password habits. In *Proc. of WWW'07.*

[15] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Trans. on Information Forensics and Security*, Jan 2013.

[16] Gogogate. www.gogogate.com. Ref. Dec 3, 2013.

[17] S. A. Grandhi, G. Joue, and I. Mittelberg. Understanding naturalness and intuitiveness in gesture production: insights for touchless gestural interfaces. In *Proc. of CHI '11.*

[18] S. G. Hart and L. E. Staveland. *Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research.* 1988. P. Hancock & N. Meshkati (Eds.), Human mental workload.

[19] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin. The design and analysis of graphical passwords. In *Proc. of USENIX Security'99.*

[20] L. A. Jones and S. J. Lederman. *Human hand function*. Oxford University Press, 2006.

[21] Y. Li. Protractor: a fast and accurate gesture recognizer. In *Proc. of CHI '10.*

[22] A. C. Long, J. A. Landay, and L. A. Rowe. "Those look similar!" issues in automating gesture design advice. In *Proc. of PUI'01.*

[23] Microsoft. Windows azure multi-factor authentication. www.windowsazure.com/en-us/documentation/services/multi-factor-authentication. Ref. Dec 3, 2013.

[24] D. Muramatsu and T. Matsumoto. An HMM on-line signature verifier incorporating signature trajectories. In *Proc. of ICDAR '03.*

[25] U. Oh and L. Findlater. The challenges and potential of end-user gesture customization. In *Proc. of CHI '13.*

[26] P. C. v. Oorschot and J. Thorpe. On predictive models and user-drawn graphical passwords. *ACM Trans. Inf. Syst. Secur.*, jan 2008.

[27] A. Oulasvirta, T. Roos, A. Modig, and L. Leppanen. Information capacity of full-body movements. In *Proc. of CHI'13.*

[28] Q. Pu, S. Gupta, S. Gollakota, and S. Patel. Whole-home gesture recognition using wireless signals. In *Proc. of MobiCom '13.*

[29] D. Rubine. Specifying gestures by example. In *Proc. of SIGGRAPH '91.*

[30] J. Ruiz, Y. Li, and E. Lank. User-defined motion gestures for mobile interaction. In *Proc. of CHI '11.*

[31] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon. Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In *Proc. of CHI '12.*

[32] D. Savransky. Lomb (lomb-scargle) periodogram, 2008. http://www.mathworks.com/matlabcentral/fileexchange/20004-lomb-lomb-scargle-periodogram. Ref Dec 9, 2013.

[33] F. Schaub, M. Walch, B. Könings, and M. Weber. Exploring the design space of graphical passwords on smartphones. In *Proc. of SOUPS '13.*

[34] A. Serwadda and V. V. Phoha. When kids' toys breach mobile phone security. In *Proc. of CCS '13.*

[35] M. Shahzad, A. X. Liu, and A. Samuel. Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it. In *Proc. of MobiCom '13.*

[36] Square. www.squareup.com. Ref. Dec 3, 2013.

[37] J. Thorpe and P. C. van Oorschot. Human-seeded attacks and exploiting hot-spots in graphical passwords. In *Proc. of USENIX Security'07*.

[38] J. Tian, C. Qu, W. Xu, and S. Wang. Kinwrite: Handwriting-based authentication using kinect. In *Proc. of NDSS '13*.

[39] B. Ur, P. Kelley, S. Komanduri, J. Lee, M. Maass, M. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, L. Cranor, S. Egelman, and J. Lopez. Helping users create better passwords. *;login*, Dec. 2012.

[40] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon. Passpoints: design and longitudinal evaluation of a graphical password system. *Int. J. Hum.-Comput. Stud.*, 63(1-2):102–127, July 2005.

[41] J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proc. of UIST '07*.

[42] J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password memorability and security: Empirical results. *IEEE Security and Privacy*, Sept. 2004.

[43] N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan. Shoulder surfing defence for recall-based graphical passwords. In *Proc. of SOUPS'11*.

[44] Z. Zhao and G.-J. Ahn. On the security of picture gesture authentication. In *Proc. of USENIX Security'13*.

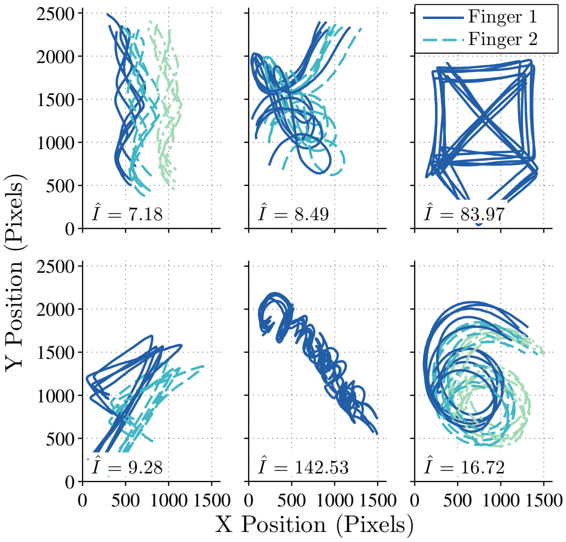
[45] N. Zheng, K. Bai, H. Huang, and H. Wang. You are how you touch: User verication on smartphones via tapping behaviors. Technical report, Dec. 2006.

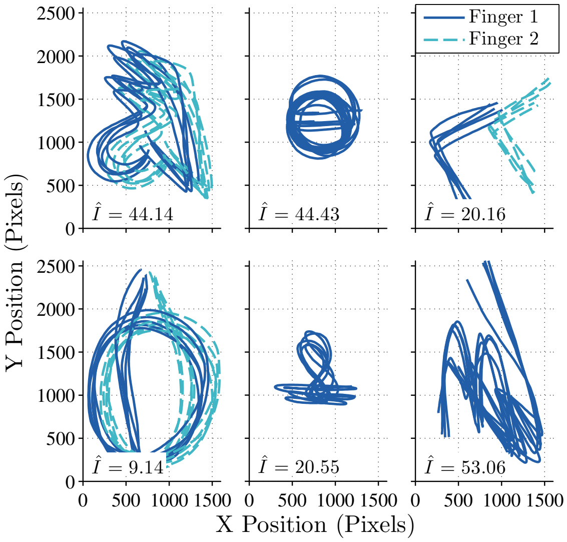
[46] F. Zhou and F. De la Torre. Canonical time warping for alignment of human behavior. In *Proc. of NIPS'09*.

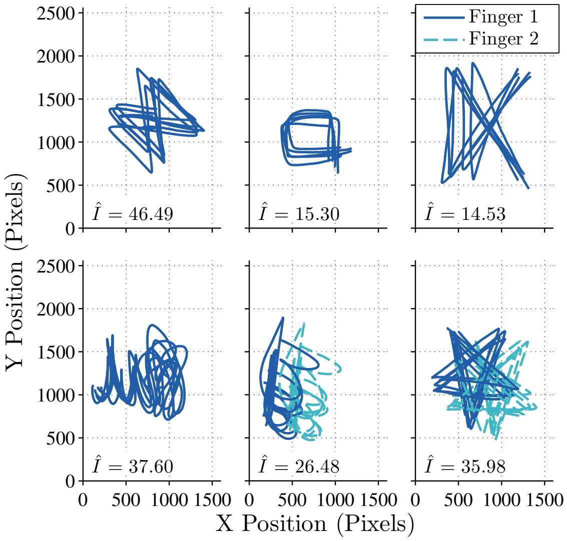
# APPENDIX

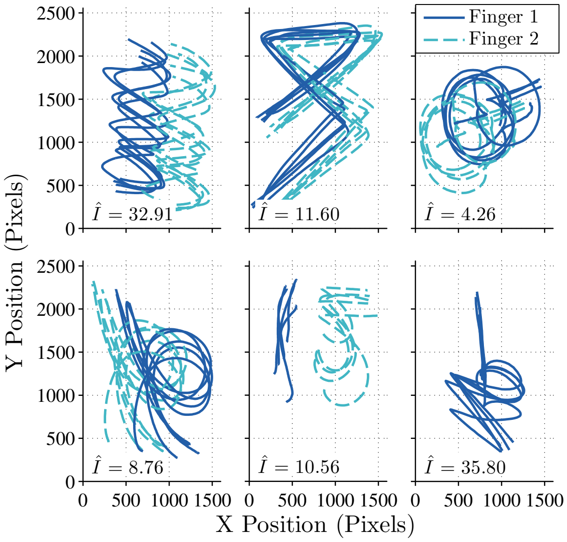
## A. EXAMPLES OF USER-GENERATED GESTURES

The following presents a selection of gesture passwords recorded during this study.


Finger 1
Finger 2
Y Position (Pixels)
X Position (Pixels)
Î = 7.18
Î = 8.49
Î = 83.97
Î = 9.28
Î = 142.53
Î = 16.72


Finger 1
Finger 2
Y Position (Pixels)
X Position (Pixels)
Î = 44.14
Î = 44.43
Î = 20.16
Î = 9.14
Î = 20.55
Î = 53.06


Finger 1
Finger 2
Y Position (Pixels)
X Position (Pixels)
Î = 46.49
Î = 15.30
Î = 14.53
Î = 37.60
Î = 26.48
Î = 35.98


Finger 1
Finger 2
Y Position (Pixels)
X Position (Pixels)
Î = 32.91
Î = 11.60
Î = 4.26
Î = 8.76
Î = 10.56
Î = 35.80