Master's Thesis

GestureMeter: Evaluating Gesture Password Selection on Smartphones with Strength Meter

Eunyong Cheon

Department of Human Factors Engineering

Graduate School of UNIST

2020

Master's Thesis

GestureMeter: Evaluating Gesture Password Selection on Smartphones with Strength Meter

Eunyong Cheon

Department of Human Factors Engineering

Graduate School of UNIST

2020

GestureMeter: Evaluating Gesture Password Selection on Smartphones with Strength Meter

Eunyong Cheon

Department of Human Factors Engineering

Graduate School of UNIST

GestureMeter: Evaluating Gesture Password Selection on Smartphones with Strength Meter

A thesis
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

Eunyong Cheon

December 10th, 2019
Approved by

Advisor
Ian Oakley

GestureMeter: Evaluating Gesture Password Selection on Smartphones with Strength Meter

Eunyong Cheon

This certifies that the thesis/dissertation of Eunyong Cheon is approved.

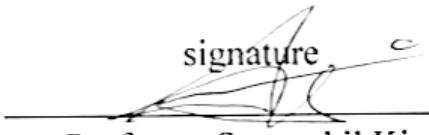
December 10th, 2019

signature 

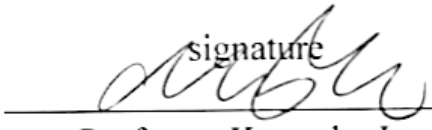
Advisor: Ian Oakley

signature 

Professor Ian Oakley

signature 

Professor Sung-phil Kim

signature 

Professor Kyungho Lee

Abstract

Gestures are potential authentication method for touchscreen devices and common tasks such as phone lock. While many studies have indicated gesture passwords can achieve high usability, evaluating their security remains a grey area. Key challenges stem from the small sample sizes in current gesture password studies and the requirement to use similarity-based recognition metrics which prevent the application of traditional entropy assessment methods. To overcome these problems, we perform a large-scale study online (N=2594). With the resulting data set, we develop a novel multi-stage discretization method and *n*-gram Markov models that enable us to assess the partial guessing entropy of gesture passwords and to create a novel clustering-based dictionary attack. We report then while partial guessing entropy appears to be greater than other common phone lock methods (e.g., Pin, pattern), gestures are highly susceptible to dictionary attack. To improve the security of gesture passwords, we develop a novel gesture password strength meter. Password strength meters has been previously proposed as an effective password policy that can improve the security of other authentication techniques such as passwords or pattern. Using the meter, we propose various mandated compliances in which users are restricted to meet certain level of strength: *default (none)*, *weak*, *fair*, and *strong*. We validate the effectiveness of gesture strength meter designs on security by performing a follow up online study and applying the security framework and attacks established in the first study. The *default policy* improves the gesture password security with small cost in usability. This thesis concludes that gesture password meters can be an effective technique for improving the security of gesture authentication systems that deserve further study.

Statement of Attribution

This thesis contains material from an article published in the following peer-reviewed conference in which the author of this thesis is listed as the first author.

E. Cheon, Y. Shin, J. H. Huh, H. Kim, and I. Oakley, “Gesture Authentication for Smartphones: Evaluation of Gesture Password Selection Policies”, in 2020 IEEE Symposium on Security and Privacy (SP). IEEE, May 2020

The contributions of the co-authors of this article are as follows:

- The aspects of this paper attributable to the author of this thesis
    - Multi-stage discretization method
    - *n*-gram Markov model simulations and optimal model selection
    - Assessment of partial guessing entropy

- The aspects of this paper attributable to other authors
    - Preprocessing methods and recognition metrics
    - Assessment of gesture dictionary based on clustering algorithm
    - Usability evaluation metrics
    - Design and implementation of online user study

Contents

List of Figures

List of Table

I. INTRODUCTION

1.1 Background

Users commonly choose simple and memorable passwords for unlocking their mobile devices whether it is PINs [1] or patterns [2]. Although fast and simple authentication can be highly usable, a malicious user may successfully crack such easy-to-guess passwords to assess private information. Gesture passwords are hand-drawn graphical passwords on touch screen devices and may serve as a potential alternative password scheme compensating security issues for several reasons: the theoretical space of possible gestures is extremely large compared to PINs or patterns [3] and authenticating with gesture passwords may require less visual attention [4] which is an attractive feature for mobile usage scenarios.

Despite gesture password's expected potentials, evaluation of the innate security and usability of gesture password is currently ambiguous. First of all, recent studies evaluating security of gesture password have collected small numbers of gesture samples in lab studies [5][6]. This contrasts to large scale data collection studies conducted with other password schemes such as PINs [7] and patterns [8]. Moreover, no common metrics have been established to assess the security of gesture passwords. This uncertainty makes comparison with other password schemes challenging. Currently, collecting large gesture data sets is necessary to establish appropriate evaluation metrics. Finally, current gesture studies lack password composition policies which can guide users toward secure password selections [9]. Nowadays, a number of websites adopt password strength meters to encourage users creating strong text passwords [10]. This is not the only case for text passwords, as prior study of pattern policies [8] also verify the effectiveness of policies such as mandating the start point during password creation.

To overcome uncertainties in evaluating security of gesture passwords, we conduct a large scale online study collecting 2594 gestures from online workers. We then establish two effective gesture password security evaluation methods: partial guessing entropy assessment and clustering based dictionary attack. To calculate partial guessing entropy of gesture data, we propose a novel multi-stage discretization method which enables raw gestures to be used to train *n*-gram Markov models. To minimize error, we explore out *n*-gram Markov model parameters such as smoothing methods and edge case handling to select the best performing model. We construct 270 models using a 5-fold process and select the optimal models based on three reasonable criteria: Crack rate, Similarity, and Completeness. We show how gesture password considerably outperforms other password schemes regarding partial guessing entropy analysis with our optimal *n*-gram Markov models. In contrast to the high entropy, we also show a large proportion of collected gesture passwords are cracked with our effective clustering-based dictionary

attack (54.18% to 58.37%). In light of weakness against the guessing attack, we develop novel password strength meters for gesture which can guide users toward more secure passwords [10][11]. To accurately measure a gesture password's strength, simple heuristics like password length and number of non-overlapping symbols are combined with advanced measures such as probability and dictionary match score. We diversify the designed meter by mandating compliance with a minimum strength of gesture password: *default(none)*, *weak*, *fair*, and *strong*. We show mandating compliance improves security against our dictionary guessing attack by 12% to 60%. We observe how mandating compliance negatively affects usability reducing recall performance by between 8% and 11%.

The contributions of this work are:

1) Effective gesture discretization method and diverse *n*-gram Markov models.

2) Security framework of partial guessing entropy assessment for gestures.

3) Novel gesture meter design improving the security of user-chosen gesture password.

1.2 Thesis statement

Gesture meters derived from large datasets of gesture password examples can be used as effective password composition policies. They can improve the security of gesture passwords for phone lock with acceptable costs to usability.

1.3 Thesis structure

This thesis is composed of five sections, the first being this introduction. In the second section, the background and related work are presented, covering gesture passwords and password strength meters. In the third section, we summarize the procedure of a large online study for collecting gesture password data on smartphones. We also introduce a security evaluation process via n-gram Markov model based partial guessing entropy assessment and a clustering based online guessing attack. In the fourth section, we explain how we develop a gesture password meter and how varying the compliance policy with that meter affects security and usability, as measured in a small-scale online study. In the final section, we present discussions and conclusions of this thesis.

II. LITERATURE REVIEW

2.1 Gesture passwords

2.1.1 Gesture

As advances in touch screen devices allows high resolution input over large areas, suggestions for new authentication techniques to protect users' information in their devices have been proposed [12]. Early work mainly explores authentication with gestures and inputs on the touch screen. This involves exploring robust and identifiable gestures [13] and successfully authenticating genuine users with single stroke approaches [14]. Prior gesture studies have also explored full-screen and multi stroke gesture inputs over various touch screen devices, while we note gesture passwords studies are yet to consider the smartphone usage scenario where input area is limited, and only single-stroke gesture passwords are allowed such as Android patterns. While the main focus of very early work was on exploring performance of gestures to be used as authentication system, recent gesture works highlight performance of recognizers, usability and security of gestures as a new authentication technique.

2.1.2 Gesture recognizer

Recognizing two relevant gesture passwords is usually more difficult than the case of text or PIN passwords where two passwords are matched exactly with their contents. Gesture passwords should be matched based on stroke similarity. Dynamic Time Warping is one candidate for recognizing gestures with excellent performance in assessing similarity between time series data [15] whose effectiveness in graphics has been explored [16]. Another well-established gesture recognizer is Protractor which uses cosine distance as similarity measure [17]. Although many recognizers have been proposed widely across the literature, variations in recognizers performance [5] often make gesture password analysis harder as no previous studies have such recognizer selection criteria.

2.1.3 Gesture usability and security

For the most part, researchers have analyzed passwords on usability and security to determine their effectiveness over other password schemes. Yang et al. [18] examine usability and memorability of user-chosen gesture passwords after one hour, one day, and one week compared with text passwords over multiple user accounts. According to their result, gesture password outperforms usability in terms of creation time (42%) and entry time (22%) and similar degree of memorability compared to text passwords. Sherman el al. [3] and Sahami et al. [19] perform shoulder-surfing attacks in which a malicious observer attacks a user's gesture password to assess the security of gesture passwords, while Liu et al. [5] perform automated brute-force attack generating random frequencies and applying low pass filter. Compared to rather simple guessing attacks such as brute-force attack and shoulder-surfing attack, a recent gesture study [6] successfully cracks between 47.71% and 55.9% of gesture password data set with novel dictionary based offline guessing attack with 10^9 guesses. The study generates an envelope known as Sakoe-Chiba band around dictionary gestures and guesses gesture by creating attack gestures within the band. We note here that guessing attacks with common and guessable gestures (i.e. star shape) are potentially effective.

2.2 Password Improvement Methods

2.2.1 Password composition policy

Designing a password composition policy is important, as it helps users to avoid guessable password [9] by mandating minimum requirements. However, it is important to notice when designing password composition policies that they may impact usability [20] by making it more difficult for users to recall or enter their passwords [9][21]. While numerous studies examine policies for text passwords such as minimum required password length and number of symbols [10][22], few studies regarding password policy are currently available for graphical passwords. Cho et al. [8] mandate composition points and explores the impact in terms of security and usability for Android pattern. Chen et al. [23] apply password meter to Android pattern to guide users creating secure passwords. They strengthen the security of pattern passwords by providing visual feedback, while highlighting issues in usability. Clark et al. [24] propose three policies that request users to create gestures that are fast, random or use multiple fingers. Evaluations indicate they have limited impact on security and may have negatively affected usability. Given the importance of password policies in ensuring the security of other forms of password

system, we identify the need for research on the development of policies that can help users select more secure gesture passwords.

2.2.2 Strength estimation

Password meter (PSM) is an effective policy that can measure the strength of password and provide informative feedback visually to users [10]. Egelman et al. [11] emphasize the effectiveness of password meter on security in a compulsory scenario. Most importantly, meters should correctly measure the strength of entered passwords [25]. Heuristics such as password length, number of non-overlapping symbols, use of upper and lower case and use of special characters [10][21][26] are commonly used to measure the password strength. Ur et al. [10] examine various meter designs and conclude users choose longer password when a meter is presented compared to a baseline condition. Recent studies suggest that simple heuristics are ambiguous to measure true password strength [7][26][27]. Recent work by Ur et al. [22] uses an artificial neural network to score passwords with multiple heuristics. Castellucia et al. base *n*-gram Markov model [28] and Houshmand et al. apply Probabilistic Context-Free Grammar (PCGF) to examine the password strength [29]. For graphical password Chen et al. apply password meter to pattern to estimate the strength of pattern [23]. Galbally et al. introduce multimodal approaches and fused measure to successfully score a password [30]. While many studies struggle measuring the accurate password strength with diverse approaches, we note no current studies exist for measuring the gesture password strength. Limited number of works on graphical password meter is done by Chen et al. applying password meter on Android pattern [23]. We note applying password meter to gesture passwords has not been explored and password meter for gestures can be a valuable security strategy as users are new to gesture password and do not know if their passwords are strong.

III. LARGE SCALE ONLINE STUDY

In this section, we establish security evaluation metrics for gesture passwords performing large scale online data collection study. We also describe a clustering based online guessing attack and a process for selection of a probabilistic model for calculation of partial guessing entropy. Together these techniques can assess the strength of gesture password set.

3.1. Security evaluation techniques

3.1.1 Preprocessing and recognition metric

It is always essential for gestures to be normalized correctly in advance of any security analysis. The variabilities of two gestures involving size, location, and orientation make comparison difficult otherwise. Previous works suggest preprocessing procedures by exploring the impact of applying scale, location, and rotation normalization that makes these properties invariant [5]. For instance, the two gestures depicting similar arrows in the same stroke order should be matched regardless of differences in the scale or location. However, the rotation normalization is typically not applied as we regard a gesture depicting a leftward arrow as different from one depicting a rightward arrow.

The next step involves selection of appropriate gesture recognition metric to assess gesture similarity. While there are many existing algorithms for gesture recognition, Dynamic Time Warping (DTW) [6][16] and Protractor [3][18] which have been widely used in recent gesture works and are applied as recognition metrics in this study. In this study, these recognizers were configured as described below.

DTW: Standard DTW implementation based on a Euclidean distance measure is used [16]. To optimize DTW, we set global constraint as Sakoe-Chiba band with 10% band size of sequence length. Global constraint speeds up the algorithm by setting boundary to search region. We follow 10% window size from previous gesture work [6].

Protractor: Reference $N Protractor implementation is used [31]. Gestures are compared by inverse cosine distance. Recognizer only allows single stroke gestures and allow gestures to be matched on shape. We applied default threshold value for rotation invariance.

The final step is re-sampling. As gesture recognizers require gestures to be exactly the same size, re-sampling is an essential part of preprocessing procedures. The optimal value for resampling size (n) can be determined by creating multiple sets of re-sampled gestures and comparing Equal Error Rates (EERs)

as in previous work [6]. For a given data set, False Rejection Rates (FRRs), which measure the proportions of users' genuine gestures that are rejected by a gesture algorithm, are calculated by matching different examples of each individual's gestures against each other. False Acceptance Rates (FARs), which measures the proportion of others' gestures being misclassified as users' own, are based on matching an individual's stored gesture template against those from all other individuals. FARs reflect the rate at which naive attackers might succeed in guessing users' gestures. By calculating FRRs and FARs for a range of distance threshold values, we can derive an EER at their intersection. Re-sampling size is then set to minimize EERs across the data sets being examined [3].

3.1.2 Entropy assessment

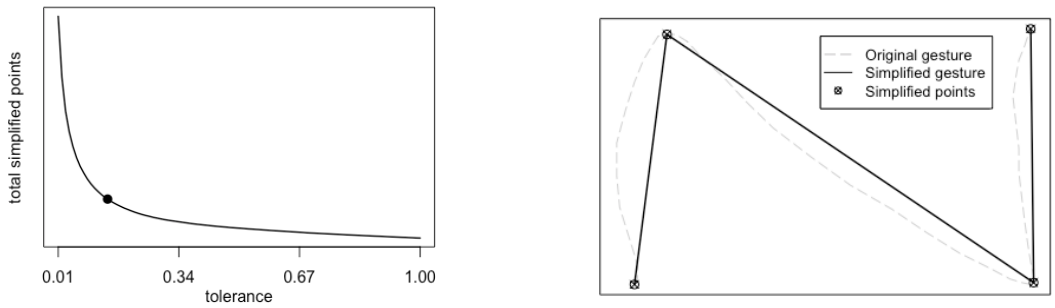
The degree of likelihood with which a given user would choose a given gesture password can be a significant metric for evaluating password strength - more commonly chosen passwords would be easier to guess or predict. This exact probability could be precisely determined from a dataset containing the real-world gesture password probability distribution for all passwords. However, in practice, since only a small portion of theoretically possible space of gesture passwords can be collected, it is necessary to develop probabilistic password models that use collected gesture samples to estimate the full probability distribution [32]. Developing *n*-gram Markov models is one way of achieving this that has been shown to be effective at estimating the probability distribution of other forms of graphical passwords [1][8]. In an *n*-gram Markov model, the probability of the next stroke in a graphical password is calculated based on a prefix of length *n.* The idea behind the model is that adjacent strokes in user-chosen passwords are not independent but follow certain high probability patterns. In *2*-gram Markov model (also known as 1-order Markov chain), probability of a password "$a_1a_2a_3 \dots a_k$" can be calculated as equation (1),

$$P(a_1a_2a_3 \dots a_k) = P(a_1|a_s)P(a_2|a_1)P(a_3|a_2) \dots P(a_k|a_{k-1}) \quad (1)$$

$$P(a_i|a_{i-1}) = \frac{number\ of\ cases\ (a_{i-1}a_i)}{number\ of\ cases\ (a_{i-1})} \quad (2)$$

where a_s represents starting symbol attached to the beginning of every password. Transition probability in (1) can be calculated from dataset with equation (2), as $P(a_i|a_{i-1})$ can be obtained counting frequencies of "$a_{i-1}a_i$" divided by counting frequencies of "a_{i-1}". To accurately estimate the probability a gesture using this technique, it is essential to develop a reasonable probabilistic password model that approximates the probability distribution of real-world gesture passwords. To achieve this, 270 *2*-gram Markov models were built through 5-fold process. The different models varied stroke discretization and *2*-gram Markov parameters to identify the most appropriate model for further study. Higher order *n*-gram Markov model (e.g. *3*-gram) were assessed, as they resulted in a very high proportion of unseen *n*-gram sequences. The multi-stage process used to generate the n-gram Markov models and select a reasonable final model are described below.

Firstly, gestures should be in the form of a sequence of symbols in order to construct *n*-gram Markov model. However, gestures are composed of long a series of coordinates and two gestures with similar shapes might also contain quite different number of coordinates. Moreover, it is hard to regard the raw coordinates as discrete symbols. As the total number of possible raw coordinates is extremely large, effectively precluding use for training a reasonable *n*-gram Markov model. To solve this problem, we apply the Douglas-Peucker (DP) line simplification algorithm to all gestures [33]. DP line simplification is a well-defined technique and was selected for its reported accuracy over other candidate algorithms [34]. To set appropriate DP tolerance value, we check the relationship between the DP simplification tolerance value and the number of simplified strokes on large gesture data set and select the knee point as the optimal value [35]. See figure 1 (a) for an example.

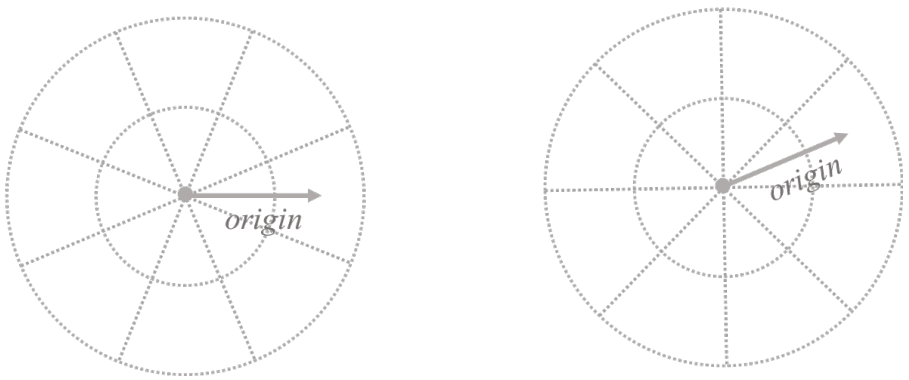


(a) DP tolerance and simplified points (b) Sample simplified gesture

Fig 1. (a) depicts a typical relationship between tolerance and total simplified points of DP simplification algorithm. The knee point is highlighted with a filled dot. (b) shows an example of how an "*N*" shaped user-chosen gesture password would be simplified to three basic strokes by applying the DP simplification algorithm.

Using the set of simplified strokes, we generate discrete symbols based on individual stroke length and stroke angle. We divide the angle and full range of all observed stroke lengths into equally sized segments and classify each simplified stroke into an angle/length category. Each angle/length category is associated with a unique symbol. Following this transformation, it is possible to represent each gesture as a series of symbols each representing as a single stroke.

To find a reasonable model, we consider multiple possible divisions of stroke length (dividing the full range of all observed strokes lengths into 2, 3, or 4 equally sized length regions) and angle (into 6, 8, 10, 12 and 14 equally sized angular regions). These variations of length and angle categories were chosen to result to sets of between 12 (2 length by 6 angle) and 56 (4 length by 14 angle) symbols, a range roughly equivalent to the number of symbols in a pattern (9 points) or PIN (10 symbols) and an alphanumeric password (about 95 symbols). To minimize error from the angle discretization process, we explore an additional variable: phase. Two possible phases for the angular regions are considered. The first *aligned* phase has its origin at 0◦ and the second *offset* phase has its origin at half the region angular width (e.g., 22.5◦ if there are 8 regions). These two phases are shown in Figure 2.



(a) *offset* phase (b) *aligned* phase

Fig.2. Example of *offset* and *aligned* phases in the angle discretization process. In this example, the length region is divided into 2 regions (short/long) and angular region is divided into 8 regions.

We further propose generating models using a grid search over two n-gram model variables: smoothing method and exclusion policy for short gestures. For smoothing, three well-known *n*-gram smoothing methods are applied to each model: 1) "add-1 Laplace smoothing", 2) "add-1/(number of symbols (N)) Laplace smoothing" and 3) "Good-Turing smoothing". The use of smoothing methods enables *n*-gram models to cover rare *n*-gram cases. These variations are described in the equations below.

Add-1 Laplace smoothing:

$$P(a_i|a_{i-1}) = \frac{number\ of\ cases\ (a_{i-1}a_i) + 1}{number\ of\ cases\ (a_{i-1}) + N} \quad (1)$$

Add-1/N Laplace smoothing:

$$P(a_i|a_{i-1}) = \frac{number\ of\ cases\ (a_{i-1}a_i) + \frac{1}{N}}{number\ of\ cases\ (a_{i-1}) + 1} \quad (2)$$

Good-Turing smoothing:

$$\text{Probability of unseen} = \frac{number\ of\ cases\ (c_1)}{number\ of\ cases\ (c_0)} \quad (3)$$

(c_k denotes the number of *n*-gram sequence where frequency is *k*.)

For the exclusion policy for short gestures, we propose three options: training on *all* gestures; training on all except for *single* stroke gestures and; training on all except for *dual and single* stroke gestures. These variations seek to avoid cases where extremely short gestures in the training set have their probabilities overweight, potentially biasing the resulting *n*-gram models. Those parameters produce a total of 270 models (30 (discretizations) by 3 (smoothing methods) by 3 (exclusion policies)). Each model is normalized with an end symbol to ensure probabilities of all possible gesture passwords sums up to 1 [32]. Specifically, an "*end symbol*" is appended to password "$a_1a_2a_3 \dots a_k$" where the last transition probability of maximal allowed length password $P(a_k|a_{Max\ len})$ is 1. Specifically, we can achieve this by setting transitional probability of $P(a_e|a_m)$ to 1, where a_e denotes the "*end symbol*" and *m* denotes the maximum observed password length in a data set.

Normalized probability of password "$a_1a_2a_3 \dots a_k$":

$$P(a_1a_2a_3 \dots a_k) = P(a_1|a_s)P(a_2|a_1)P(a_3|a_2) \dots P(a_k|a_{k-1})P(a_e|a_k)$$

Finally, three criteria are compared to select the most appropriate model; *crack rate* (CR), *similarity* (SR), and *completeness* (CP). C*rack rate* was intended to represent the accuracy of the probability distribution of collected gesture samples in the model. We calculate it by generating dictionaries containing the the top *k* gestures (i.e., those achieving the highest probability) from the train sets for each model. We then reconstitute a stroke representation of each dictionary gesture using an arbitrary starting coordinate and concatenating together sub-strokes (one per symbol) using the central values of the appropriate length/angle segment – see step 6 and 7 in figure 3. With these dictionary gestures, we determine how many gestures in the test set are matched a given threshold *t*. The number of cracked gestures is then divided by the total number of gestures to calculate the *crack rate*. The second criteria is *similarity* between each users' entered gesture and representation reconstituted from each n-gram model. The assumption behind *similarity* is that a close match to an original real-world gesture reflects a more accurate model. We use DTW distance to calculate *similarity*. The third criteria attempt to reflect how *complete* a model is, surmising that models in which we observe a larger proportion of possible *n*-gram cases will be more accurate. This can be calculated by dividing the number of seen *n*-gram cases by the total number of possible *n*-gram cases. After calculating these metrics for all models, the most appropriate three models over all three criteria are then selected through manual inspection.

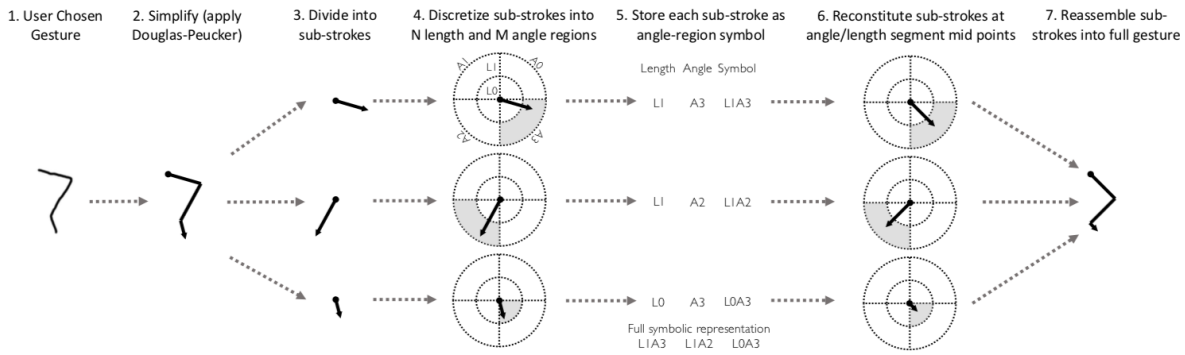


Fig 3. Overview of discretization process from a gesture (step1) to angle-region symbols for building *n*-gram Markov models (step 5) and the gesture reconstitution process for evaluating them (step 6 and 7). Illustration uses a simple model with 2 length and 4 angle discretization regions.

To further improve performance of the three *n*-gram Markov models selected from previous criteria, edge cases in discretization stage need be considered. Specifically, cases where strokes lie on or near the boundaries between regions may be miscategorized or the reliability of the categorization may be low. To deal with this, we propose an approach where all strokes that lie within b% of length or angle boundary edges result in increasing the *n*-gram sequence frequency of both the actual and adjacent length or angle region by *i* and *j* respectively, where the sum of these values is always 1. To instantiate this idea, we apply it to each of the models selected in the previous stage and perform a grid search over all values of *b* in the range 1% to 10% for angle and length with both *i* and *j* set to a constant 0.5. This generates an additional 100 models for each base model studied. One or more reasonable models is selected based on balanced improvements to metrics defined above.

With the selected n-gram Markov models, we calculated partial guessing entropy to evaluate the security of the gesture passwords [7]. Partial guessing entropy estimates are useful because real-world attackers might only be interested in cracking just a fraction of an entire password set - it is a popular technique for estimating the average number of trials needed to successfully crack a *fraction* (α) of an entire password set. We report these data in terms of "bits of information." To calculate this, first we calculate probabilities of all possible passwords generated by the selected optimized *n*-gram Markov models and order them in non-increasing way. We can derive rank index $\mu_\alpha(0 \leq \alpha \leq 1)$ which is defined as index j where sum of probabilities $\sum_{i=1}^{j} p_i$ achieves α. Noting actual covered fraction as $\lambda_\alpha = \sum_{i=1}^{\mu_\alpha} p_i$, we derive partial guessing entropy with following equation (1) and representation in terms of bits follows equation (2) [8].

$$\textit{Partial Guessing Entropy } \alpha = (1 - \lambda_{\mu_\alpha}) \times \mu_\alpha + \sum_{i=1}^{\mu_\alpha} (p_i \times i) \quad (1)$$

$$\textit{Bits of Information } \alpha = log(\frac{2 \times \textit{partial guessing entropy } \alpha}{\lambda_{\mu_\alpha}} - 1) + log(\frac{1}{2 - \lambda_{\mu_\alpha}}) \quad (2)$$

3.1.3 Dictionary assessment

We create an effective clustering-based gesture dictionary that can be used as another security assessment. We derive gesture dictionary with following steps. Firstly, to create gesture dictionary composed of common shapes, similarities between all gestures in the training set are calculated using a gesture distance metric. Affinity propagation clustering algorithm is then applied to the similarities [36]. It is an exemplar-based clustering algorithm that identifies typical examples within the data set. When clusters composed of similar gestures results from affinity propagation algorithm, we select the *geometric* centers of each cluster as dictionary which are optimal representative gesture from derived cluster. We are interested in online attack scenario, where an attacker can try a limited number of guesses to crack a genuine user's password. Specifically, with the dictionary of 20 center gestures, we match all test set gestures against the dictionary to calculate crack rate. The crack rate is the proportion of gestures in the test set that match at least one gesture in the dictionary. We report crack rates for a continuum of distance thresholds.

3.2 User study

We capture large gesture data set and analyze based on two security evaluations described in 3.1. The gestures were captured in a homogeneous study protocol and outside traditional lab or university environments. The ethical aspects of the study were approved by the host university IRB.

3.2.1 Gesture recognizer

To capture data for both studies, we should use a gesture recognizer to accept, reject or confirm gestures. We use Protractor which is often deployed in many related gesture studies [3][18].

3.2.2 Study design

We recruited experiment participants from Amazon Mechanical Turk (MTurk) and implemented the study online. The study requires online users to participate in experiment on their mobile phones by providing them with a link or QR code to the study site. Those who try to participate with other type of device (e.g., a PC) were screen out and re-provided with link and QR code for use with a mobile device.

When users start study clicking link or accessing QR code, they start with study instruction screen. It requests the users to create secure gesture password and incentive is provided to encourage him/her to create secure and memorable gesture password. In the next screen, they are requested to fill basic demographic information which is handedness, age, education level, occupation and ethnicity.

After answering demographic survey, they are asked to create and then confirm their gesture passwords in a square canvas in the smartphone screen. The input canvas is designed in a way that the majority of users can reach with their thumb. For confirmation process, they have to enter chosen gesture again to verify their decision. If they fail to match their gestures in this stage, they have to start again from initial stage of creating gesture password. At any stage, participants can cancel and start from the creation again.

Participants are requested for creating and confirming attack gesture to guess others' gestures in the next step. The attack entry is designed to distract users' working memory to recall gestures. For the final step, they finish the study by recalling creation gestures within 5 attempts.

(a) Create (b) Match (c) No Match

Fig 4. The figure illustrates example screens of the first study running on smartphone. Gestures are created in limited input region in (a), matched in confirmation stage as (b), and fail to match in (c).

3.2.3 Participants

In total, 2619 Amazon Mechanical Turk workers completed the study, each rewarded with 0.25 USD. 25 participants who were thought to have multiple Amazon accounts were removed from the set as they created essentially identical (and highly unique) gestures in close temporal proximity. The final gesture set contains data from 2594 MTurk workers.

3.2.4 Results

Usability

Demographics: The majority of participants identified as white (56.32%), Asian (18.47%), Hispanic (9.48%) or black / African American (7.94%) and fell in the 18-24 (31.8%), 25-34 (46.72%) or 35-44 (15.42%) age groups. Most were educated post-graduate (13.61%), college (49.46%) or high school level (33.73%) and they worked in a wide range of fields; the largest group were students (15.38%).

Measures	μ	σ	$\tilde{\mu}$
SC (#)	0.6	2.58	0
CF (#)	0.14	0.52	0
ST (s)	24.38	30.13	15.52
RT (s)	5.11	5.52	3.12

Table 1. The table summarized usability results of the first study in terms of mean (μ), standard deviation(σ) and median($\underline{\mu}$). (SC = *Setup Cancel* (#), CF = *Count Match Failures* (#), *ST* = *Setup Time* (s), RT = *Recall Time* (s))

Setup: We log the number of intentional cancellations of gesture set up (*Setup Cancel*), the time to create and confirm gestures (*Setup Time*), the number of failures to match creation gestures (*Count Match Failures*) by a user. These measures are summarized in terms of mean, standard deviation, and median as the positive skews in the data (Table1).

Recall: We log the time to recall creation gestures (*Recall Time,* see Table 1) and the number of participants who failed to recall their gesture within five attempts (*Recall Rate*). The recall rate in the study was 92.1%. Our recall rate is lower than previously reported recall rate of 98.9% [18]. Attack gesture negatively affects recall rates of users as 68% of errors occur along with confirmation of participants' attack gestures. These participants mistakenly opted to enter their attack gestures rather than their creation gestures.

Security

Partial guessing entropy: We follow the entropy analysis processes described in section 3.1.2. We create 270 different 2-gram models and applied selection criteria to choose a subset of models for optimization. We used DTW to calculate these metrics due to this algorithm's improved performance over Protractor in terms of EERs (See figure 7) and set the threshold to the value corresponding to DTW 10% FRR. Based on three decision parameters (CR, SR, and CP), we select the best three models to optimize for boundary. Table 2 presents the three best performing *n*-gram Markov after the optimization process. Although the first model "2X10" achieves the highest CR, it shows weakness on SR and "4X12" model shows weakness on CP, suggesting the 3X10 model may be the most reasonable choice. In addition to this result, we examine the distribution of frequencies for start, center and end strokes in these models (figure 5). General user behavior that can be assessed here is that their initial strokes tend to be rightward, central strokes tend to be short and curved, and final strokes are somewhat longer, but otherwise fairly evenly distributed. Thus, we believe that the model based on discretization into three length and ten angle regions (3x10) provides a well-balanced combination of high crack rate, close

accuracy to the original user-chosen gestures and high proportion of observed n-gram cases. We note that the model based on discretization into two length and ten angle regions (2x10) may perform better in terms of crack rate when gestures in a set are relatively simple.

Name	*Model Parameters*					*Model Performance*		
	Len	Ang	Phase	Smoothing	Excl.	CR	SM	CP
2×10	2	10	offset	Good-Turing	single	18.24%	62.72%	94.78%
3 ×10	3	10	offset	Add-1	dual	16.85%	86.16%	90.73%
4 ×12	4	12	offset	Add-1	dual	15.46%	84.70%	73.67%

Table 2. Three selected *n*-gram models after optimization showing *crack rate* (CR), *similarity* (SR), and *completeness* (CP) metrics.

We then calculated partial guessing entropy for the three optimized models, 4-digit PINs [2], and screen lock patterns [8]. This enables comparison across various password data sets. The results are presented in Table 3. Selecting optimal *n*-gram Markov model based on partial guessing entropy is not desirable as more complex model (4X12) tends to have higher partial guessing entropy values while showing poor model completeness. All three *n*-gram Markov models have higher partial guessing entropy values than PINs and patterns for every level of α. An interesting result is that partial guessing entropy values of the three *n*-gram Markov models show steep increase between α levels of 0.1 to 0.4. This suggests the presence of a "weak subspace" of gesture passwords [6] at low α levels that are simple and easy to guess.

Dataset	α					
	0.1	0.2	0.3	0.4	0.7	1.0
2×10 Pass-gestures	6.29	8.38	11.39	13.31	16.11	17.98
3 ×10 Pass-gestures	6.97	9.69	13.26	15.41	18.57	20.68
4 ×12 Pass-gestures	7.47	11.27	15.94	18.40	21.68	23.98
4-digit PINs	5.19	7.04	8.37	9.38	11.08	11.83
Patterns	5.04	5.82	6.54	7.19	9.20	12.71

Table 3. Table compares partial guessing entropy result across diverse password sets.

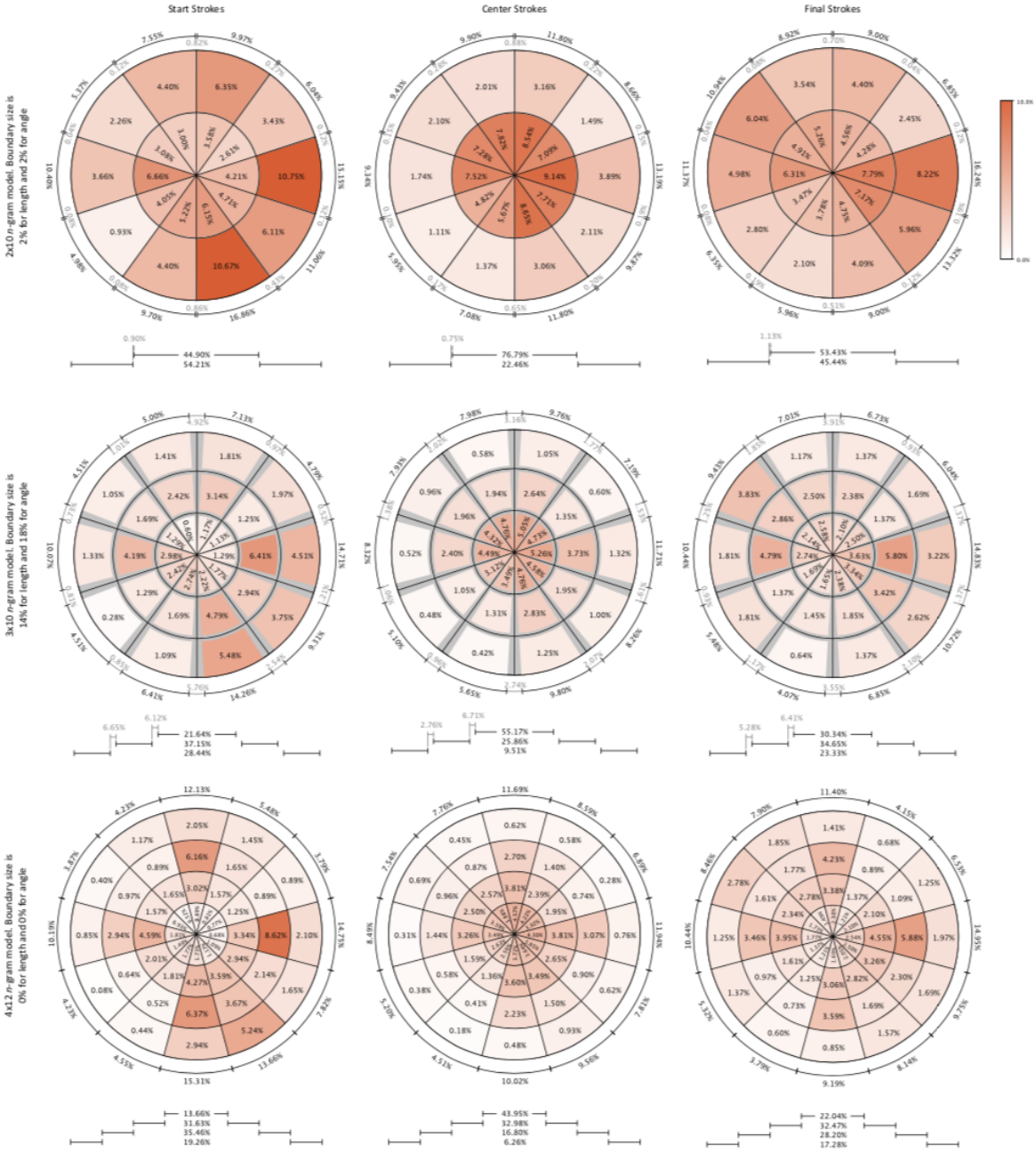
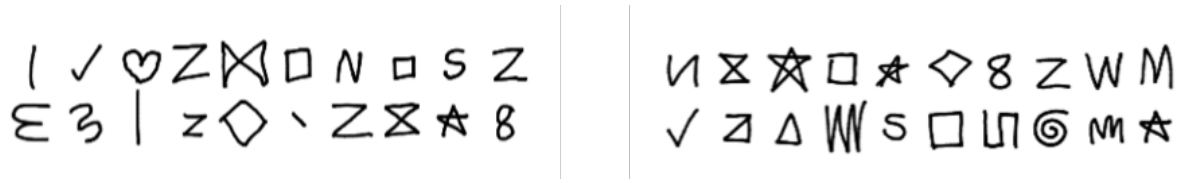


Fig 5. Distribution of strokes in optimized 2x10 (top row), 3x10 (center row) and 4x12 (bottom row) n-gram models in the first study. Left column shows data from start strokes, right column shows data from final strokes and all other strokes are shown in the center column. Each figure is divided into the discretization regions available in the given n-gram model, with the frequency of sub-strokes observed in each region marked in % and by color. Boundary regions are shown to scale as grey areas for both length and angle. The proportion of strokes used in each region (black text) and boundary (grey text) is shown at the boundary (for angles) and the bottom of each diagram (for lengths)

Clustering-based dictionary attack: For the next security assessment, we apply clustering-based dictionary attack to the gesture set to evaluate the set on crack rate metric. Following steps stated in Section 3.1.3, we generate dictionaries for both Protractor and DTW distances. We set the dictionary size *k* as 20, assuming an online attacker with 20 guesses. The top 20 dictionaries for Protractor and DTW are shown in Figure 6. On a continuum of EER threshold values, the clustering-based dictionary shows high effectiveness - see Figure 8. Dictionaries crack between 54.18% (DTW) and 58.37% (Protractor) of gestures at EER derived threshold of, respectively, 4.14% and 3.59% FRR. The DTW recognition metric outperforms Protractor in terms of both crack rate and EER value. We compare this result with dictionary attacks performed on other password data set. For patterns, they crack 13.33% of passwords in real-world settings [37], 32.55% in Mturk study [8].

Compared to this result with offline attack, recent gesture password work by Liu *et al* [6] cracks 55.9% of gestures with DTW recognizer with guesses. This can be explained in diverse aspects. 1) The previous study collects gestures in more controlled and strict settings leading participants to create more complex and diverse gestures than MTurk workers [38]. 2) Our work also constrains users to create single stroke gestures on a small-sized input canvas which possibly leads users toward simpler gestures than gestures in multi-stroke and large input size settings [5][6]. Nonetheless, the result is meaningful to highlight the vulnerability of gesture passwords due to the high proportion of “weak subspace” gestures composed of simple, similar or otherwise weak strokes [6].



(a) Protractor Dictionary (b) DTW Dictionary

Fig 6. The figure illustrates top 20 dictionary from full set using affinity clustering algorithm for (a) Protractor and (b) DTW.

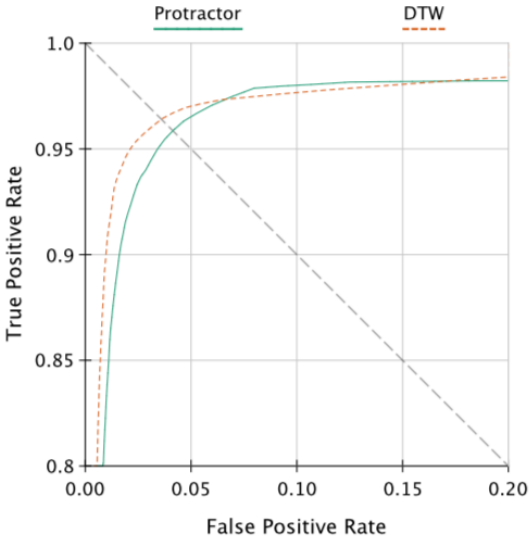


Fig 7. Receiver Operating Characteristic (ROC) curves contrasting relative FRR and FAR performance for Protractor and DTW recognizers. The Protractor EER is 4.14% (AUROC: 0.974) at a threshold value of 1.25 and the DTW EER is 3.59% (AUROC: 0.984) at a threshold value of 18.4.

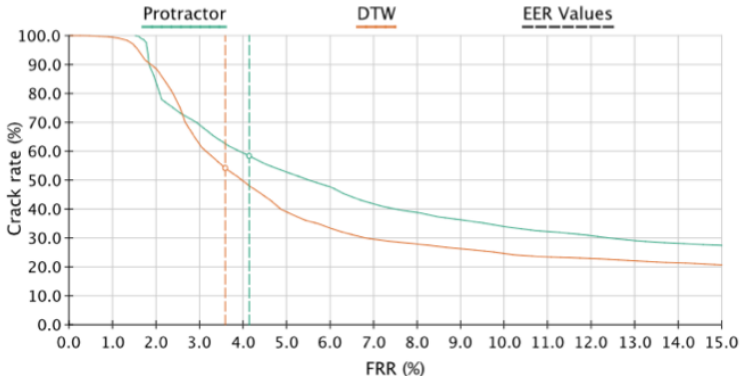


Fig 8. Proportion of cracked gestures with dictionaries at FRRs from 0%- 15% with Protractor and DTW recognizers in the first study. Vertical lines show EER values for each recognizer.

IV. METER ONLINE USER STUDY

From the first study, we observe that a large proportion of gestures can be cracked with a dictionary attack despite high entropy. To guide online users creating more secure gesture passwords, we design password meter policy for gestures in following section. The most important feature to design a password meter is a system that can provide a valid assessment of a given password's strength [25]. To meet this condition, gesture password scoring techniques including four metrics are explored. We also test four compliance policies encouraging users to create more secure gesture passwords. Besides the strength evaluation, it is essential to provide appropriate feedback about their password security to encourage users toward more secure passwords. Hence, we also cover visual design elements (e.g. colors, texts, and bars) for the gesture meter. Finally, we perform another MTurk study (albeit small scale) to evaluate security and usability of the gestures created using the meter and compliance policies.

4.1. Scoring mechanisms

4.1.1 Password scoring

Traditionally, the simplest metrics used for assessing password security is length and number of symbols in a password [10]. These are attractive security measures because they are simple to calculate and evaluate. However, it is hard to evaluate security solely on length and symbol numbers of a password [26][27]. For example, a zig-zag shaped gesture tends to have a long password length, although it is a very common shape that many users propose. A round shape has a lot of symbols (curvature produces lot of symbols in our discretization process) while it may also be, in practice, a very common choice. Thus, simple heuristics (password length and symbol number) need be combined with advanced metrics, *n*-gram Markov gesture probability and dictionary match score, to assess the genuine strength of a gesture [25]. *n*-gram Markov model has been known for its effectiveness of assessing password guessability in many security studies [9][28], and dictionary match score can work as quantitative dictionary metric which prevents users from selecting easy-to-guess passwords. To create these scoring metrics, the large gesture password dataset from the study outlined in section 3 was used to create automatic scoring approaches that can assess new gestures. We use this study data set as the basis for the meter for several reasons. Specifically, we argue using this set is appropriate because it has a 1) large sample number (N = 2594) and 2) the result of qualitative categorization reasonable follows other recent gesture studies [6] and 3) there are no public datasets containing gesture passwords.

Our basic scoring algorithm generates a rating for each gesture submitted to it. This is composed of several sub-metrics. We assign between 0 and 50 points for each of password length and number of symbols. For the Markov *n*-gram gesture probability and dictionary match score, we map percentile rank to the range between 0 to 100 points for each. Thus, the total score of a new gesture is between 0 and 300 points. Each scoring metric is pre-identified and ordered from the large gesture data set to create percentile rank reference data. In this way, the system can easily refer the percentile rank for each metric when a new gesture enters the meter. We describe the sub-metrics in more detail below.

4.1.2 Password length and symbol number

Common and simple heuristics to estimate password strength are the password length and the number of symbols present in a password [10]. The notion behind those heuristics is that long and multi-symbol passwords are likely to be secure. A gesture needs to be composed set of symbols by following DP line simplification, sub-stroke division and discretization for length and angle as described in section 3.1.2. When a gesture is discretized with optimal discretization parameters (3×10), we can assess a password length and number of symbols by counting the number of total and non-overlapping symbols in a set of symbols. For instance, the password length for a password "a-b-b-c-c-d" is 6 and the number of symbols is 4. We calculate the scores for the two metrics by finding the percentile rank of relevant gesture from pre-collected large training set. We need to consider that the number of length and symbol of a discretized gesture actually relates to security measures. We perform simple linear regressions independently to check this relation. Specifically, we first calculate two metrics and apply unsupervised equal frequency binning to parse gestures into range of figures. We then perform a dictionary attack to binned gestures of each score metric with top 20 dictionaries derived from the study to get proportion of gestures cracked. Significant effects on crack rate are figured with password length ($F_{1,7}$= 734.4, $p < .0005$, R^2= 99.06) and symbol count ($F_{1,7}$ = 588.9, $p < .0001$,R^2= 98.38).

4.1.3 Markov n-gram probability

Evaluating probability of password is an effective strength estimation strategy as it reflects how it commonly occurs in the real world. Regression result verifies that probability is a valid measure of strength as it shows a strong linear relationship. ($F_{1,7}$= 618.6, $p < .0005$, R^2= 98.72) Although it is theoretically possible to access estimated password probability distribution with *n*-gram Markov models, it is often too costly regarding memory and time to consider the full possible gesture space as user anticipates immediate feedback from the system. To solve this, the probabilities for all gestures in pre-collected large training set are calculated in advance by applying best performing *n*-gram Markov model derived from the first study, and the probabilities are ordered in non-decreasing way. When a user gesture is entered, it is initially discretized into symbols and *n*-gram probability is calculated with optimal *n*-gram Markov model. We then find the percentile rank of relevant gesture to generate the score for probability.

4.1.4 Dictionary match score

In the previous study, we successfully guess a substantial amount of gestures using a clustering-based guessing attack. We believe the gesture dictionary derived from affinity clustering algorithm is effective from this security result. Thus, a gesture is assumed to be secure and unique if it is not similar to items in the dictionary. We take multi-stage approach to calculate dictionary match score. Firstly, we generate distance matrix with DTW matches from full-cluster and perform affinity propagation clustering method to separate groups composed of similar gestures, leading to 290 clusters with center. Secondly, we calculate a maximum distance score (the most similar) of entered gesture by matching the 290 dictionary centers to obtain dictionary match score value. To derive percentile reference of dictionary match score, all distance scores are processed by matching all training set gestures to all dictionary gestures and the greatest distance is selected as the final match score. High dictionary match score of a gesture represents low maximum distances, implicating uniqueness of the relevant gesture. On the contrary, low dictionary match score of a gesture represents high maximum distances which imply similarity with the derived dictionary. When the maximum distance is higher than DTW 10% FRR level, we give dictionary match score of 0 to the relevant gesture indicating vulnerability.

4.2 Meter compliance policies

One concern is whether the designed gesture meter motivates online workers to achieve high scores for their gesture [39]. From the previous study, we observe significant number of participants entered simple gestures as their password. This is an important issue for large scale online study that can pollute data quality. To avoid this kind of issue, one option to consider is setting minimum compliance, encouraging online users to create safer passwords [10]. Thus, four compliance policies are: 1) *default* meter and 2) *weak*, 3) *fair*, 4) *strong* compliances meter. While users can choose their gestures without any constraint for *default* policy, users are mandated to satisfy minimum scores of four ranges, these are 75, 150, and 225 points respectively for *weak*, *fair* and *strong* compliance policies (out of 300 points). Compliance policies force users to choose stronger gesture passwords (as measured by our meter). Nevertheless, this may irritate online users and cause an unwanted burden. Thus, the main goal under these compliances is to explore the tradeoff between usability and security of these policies so that the optimal compliance where two metrics balance can be established.

4.3 Meter visuals

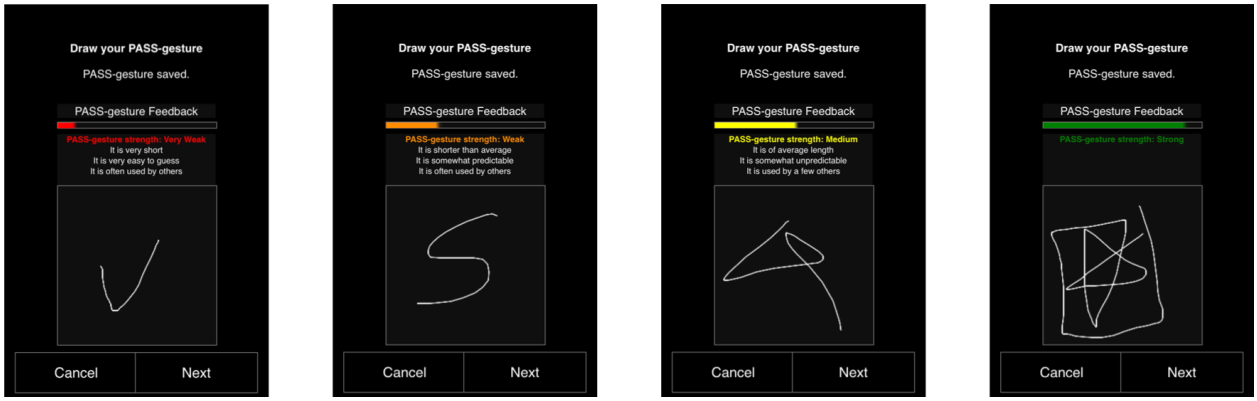
4.3.1 Colored bar

As the security of user-chosen passwords increases, the bar is filled accordingly. We follow the recommendations from recent string password meter study and use four colors [10]. The bar displays as red if it is filled between 0% to 25% implying user-chosen password is very weak regarding security, as orange from 25% to 50% which means user-chosen password is generally *weak*, as yellow from 50% to 75% as *fair* strength, and green meaning high strength of user-chosen passwords as the score reaches above 75%.

4.3.2 Text

General text feedback along with colored bars are designed to improve entered gesture password [10][22]. Overall rating text with color of a gesture password is displayed below the bar. In response to the total security score, a gesture password is categorized "*Very Weak*", "*Weak*", "*Medium*", and "*Strong*" respectively if users achieve more than 0, 75, 150, and 225 out of 300 points. A consideration for text feedback is that users will fail to improve the password security unless they can understand how to improve their gestures. We solve this problem by displaying data-driven text feedback in addition to

the overall rating for each of three security metrics to assist users' awareness. When a gesture is scored as "*Strong*", we do not present text feedback for individual score metrics. We opt to choose general sentence for probability and match score metrics to provide understandable feedback for users. Three text feedbacks are designed for each metric regarding the score (maximum 100 points for each). 1) For password length and symbol, meter displays "It is very short" (0 to 25), "It is shorter than average" (25 to 50), and "It is of average length" (50 to 75). 2) For password probability score, meter displays "It is very easy to guess" (0 to 25), "It is somewhat predictable" (25 to 50), and "It is somewhat unpredictable" (50 to 75). 3) For password match score, meter displays "It is often used by others" (0 to 25), "It is sometimes used by others" (25 to 50), "It is used by a few others" (50 to 75). See Figure 9 for example screens.



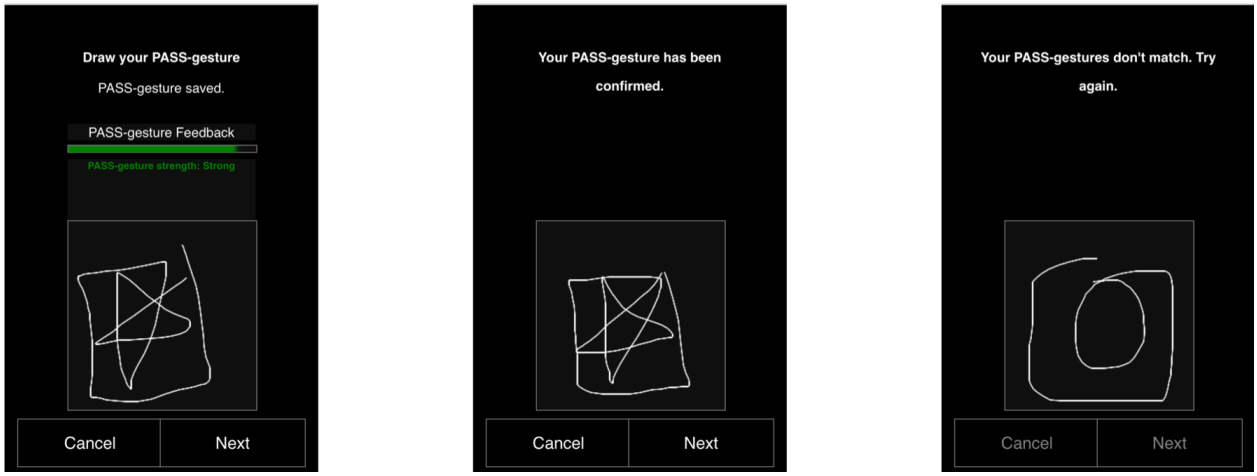
(a) Very Weak (b) Weak (c) Fair (d) Strong

Fig 9. Example gestures are displayed on a smartphone screen. The gestures are scored very weak, weak, fair, and strong in *default* compliance policy meter. (from left) Each gesture's score is represented visually with colors (red, orange, yellow, and green) and lengths (short, middle short, middle long, and long) of score bar and text feedbacks for three score categories (length/symbol, probability, and match score).

4.4. User study

4.4.1 Study design

We follow a similar process in the gesture meter study to the first study. We use MTurk to gather online participants and run the study online. Again, we only accept mobile device users by sending QRs or links. We use the same gesture recognizer, Protractor in the gesture meter study [3][18]. After presenting study instruction to create secure and memorable gesture password, we request demographic surveys to the participants. Then users move on to the gesture creation stage. In this stage, gesture score feedback section where users will be evaluated based on three predefined score metrics with colored bar and texts is newly added from the previous study. Users are forced to create their gesture again if the scores are barred from compliance policy when it's mandated. Being satisfied with the choice and safe from gesture compliance policies, users click on next button to proceed confirmation. Users are asked to start creation again if they fail to match chosen gesture as in the first study, while they can cancel to start again at any stage of creation process. Due to negative effect on recall, we replace attack gesture creation session to memory game session used in previous work [8]. They also practiced their gesture 10 times and finally recall their creation gesture. Participants were invited (via email) to participate in a day-2 recall session 24 to 72 hours after completing the initial session. In that session, they were simply asked to recall their creation gesture one more time. Following the initial recall test, they had five attempts to achieve this task.



(a) Create (b) Match (c) No Match

Fig 10. The figure illustrates example screens of the second study running on smartphone. Gestures are created with meter feedbacks in limited input region in (a), matched in confirmation stage as (b), and fail to match in (c).

4.4.2 Participants

In total 182, Amazon Mechanical Turk workers participate in the second study. 1 USD is rewarded for completing meter study which is increased from the first study as the new study involves new practice session and memory game.

4.4.3 Results

Usability

Demographics: Most of the participants are Asian (66.5%), followed by white (28%) and Hispanic ethnic group (2.7%). Dominant age group is 25-34 (76.4%), 18-24 (12.6%) and 35-44 (8.8%). Educational level is mainly college (61%), post-graduate (33.5%) and their major was diverse.

Setup Cancel, Setup time, Recall Time: Summary of usability statistics is summarized in table 4. As all usability measures show skewness in their data, we perform Kruskal-Wallis test over the measures. Significances are found in Setup Cancel ($\chi^2 = 17.56$, $p < 0.001$), Setup Time ($\chi^2 = 12.95$, $p < 0.005$) and Recall Time ($\chi^2 = 8.58$, $p < 0.05$). We apply *post-hoc* Wilcoxon tests to these measures: 1) Significant differences are found in Setup Cancel between *strong* & *default* ($p < 0.0005$), between *strong* & *weak* ($p < 0.005$) and between *fair* & *default* ($p < 0.05$). Participants are less likely to change their passwords when strong compliance is given. 2) Significant differences are found in Setup Time

between *strong* & *default* ($p < 0.005$), between *default* & *weak* ($p < 0.005$) and between *strong* & *weak* ($p < 0.005$). Gesture setup time increases with stronger compliance policy. 3) Significant differences are found in Recall Time between *fair* & *weak* ($p < 0.05$)and between *strong* & *weak* ($p < 0.01$). Recall Time increases with stronger compliance policies.

	Default			*Weak*			*Fair*			*Strong*		
	μ	σ	$\tilde{\mu}$	μ	σ	$\tilde{\mu}$	μ	σ	$\tilde{\mu}$	μ	σ	$\tilde{\mu}$
SC	1.25	2.58	0	0.56	1.43	0	0.27	0.92	0	0.02	0.15	0
CF	1.64	5	0	1.9	8.11	0	0.57	1.5	0	0.86	1.99	0
ST	66.39	132.15	26.06	55.38	55.35	35.14	86.05	111.91	48.73	139.81	152.57	73.59
PM	8.57	2.82	10	9	1.89	10	8.11	3.16	10	7.95	3.21	10
RT	8.79	16.47	3.36	4.21	4.84	2.94	7.35	9.06	3.93	7.49	9.58	3.7
RA	0.46	1.34	0	0.25	0.99	0	0.7	1.62	0	0.67	1.63	0

Table 4. The table summarized usability results of the second study in terms of mean (μ), standard deviation(σ) and median($\underline{\mu}$). (SC = *Setup Cancel* (#), CF = *Count Match Failures* (#), *ST = Setup Time* (s), PM = *Practice Match* (#), RT = *Recall Time* (s), RA = *Recall Attempts*(#))

Recall: Day-1 Recall Rates are 93.18% for *default* meter, 96% for *weak* meter, 90.91% for *fair* meter, and 88.1% for *strong* meter. We compare these figures with recall rate of 98.9% from the prior study [18]. Fisher's exact test results reveal significant differences in recall rates between 1) prior study and *fair* compliance meter and between 2) prior study and *strong* compliance meter ($p < 0.05$). We conclude that when we mandate compliance with stronger gesture ratings, usability is negatively influenced; Setup Time and Recall Time increase and Setup Cancels and Recall Rates decrease.

Very small numbers of participants returned in day-2 recall session for all compliance policies: 6.82% returned in *default,* 19.2% in *weak*, 13.64% in *fair*, and 19.04% in *strong*. The low participation rate precludes reliable assessment of the usability in day-2 recall task. As a positive sign, we note that, overall, just two participants failed to recall their gestures - both with the *weak* compliance policy. Further work is needed to capture more data and provide a more meaningful characterization of gesture recall performance over multiple and temporally separated gesture password entry sessions.

Security

Partial guessing entropy: In the first study, we choose our best *n*-gram Markov model (length and angle divided into 3 and 10 regions respectively, offset phase alignment, add-1 smoothing, dual stroke exclusion, and boundary optimized 14% for length 18% for angle) on multiple criteria. We use this model to assess partial guessing entropy of gesture set collected from different compliance policies. Partial guessing entropy increase greatly from the first study. Over various portions (α) of password *default* meter generally outperforms others. While *strong* password meter set reveals to have high partial guessing entropy at low alpha value (10%) which roughly equates online attack scenario, its strength decreases as fraction of password set increases. While the partial guessing entropy of *weak* password meter is poor at lower (0.1, 0.2 and 0.3), it outperforms *fair* and *strong* meter at higher (0.7 and 1.0). This result can be interpreted as the minority of users in *weak* meter policy create unique and hard to guess passwords that attackers are impossible to guess full set of passwords. In contrast, participants in *fair* and *strong* password meters are enforced to create complex gestures, reducing the size of weak subspace, increasing entropy at lower alpha.

Dataset	α					
	0.1	0.2	0.3	0.4	0.7	1.0
First Study	6.97	9.69	13.26	15.41	18.57	20.68
Default Meter	12.82	18.80	20.45	21.28	22.59	23.19
Weak Meter	11.87	17.88	19.81	20.80	22.36	23.07
Fair Meter	12.91	18.51	20.00	20.83	22.25	22.98
Strong Meter	13.91	18.56	19.89	20.68	22.09	22.87

Table 5. The table summarizes partial guessing entropy results of the second study.

Dictionary attack: We use full cluster top 20 dictionaries derived from the first study for online attack scenario. As we show the effectiveness of dictionary cracking large portion of gesture set, its effectiveness is tested across meter compliance policy designs. We use DTW as our recognition metric as it shows effectiveness over Protractor on EER and Crack rate in the first study. Security against online dictionary attack is increased from 11.92% to 60.45% (47.72% for *default*, 36.36% for *fair*, and 21.43% for *strong*) compared previous result (54.18% of gesture cracked) except for *weak* compliance meter (59.62%) at system threshold FRR of 3.59%. We apply Fisher's exact test to examine significant differences between the first study and compliance policies. Crack rates of *weak* and *default* compliance meters are not significantly different from the first study at system threshold ($p > 0.1$). We then examine crack rates at specific FRR rates 2.5%, 5% and 10% to see significant difference between compliance

policies. All comparisons show significant difference ($p < 0.05$) except for the following: 1) At 2.5% FRR rate, *default* compliance policy does not show any significant difference with *weak* compliance policy ($p = 0.38$) and between *fair* versus *strong* compliance policies ($p = 0.89$). 2) *fair* compliance policy also does not show significant difference at the 5% FRR with *strong* compliance policy ($p = 0.051$). 3) At 10% FRR, *default* compliance policy does not show significant difference with *fair* compliance policy ($p = 0.17$). We observe that *weak* compliance meter does not improve security of gestures, implying "weak subspace" [6] of this data set. This result complies with partial guessing analysis result that large proportion of gestures are actually simple and homogeneous. Security of *default* compliance meter improves in higher percentages of FRR which is not significantly different from *fair* compliance meter. *fair* compliance meter generally performs well on lower percentages FRR without any significant difference compared to the performance of *strong* compliance meter.

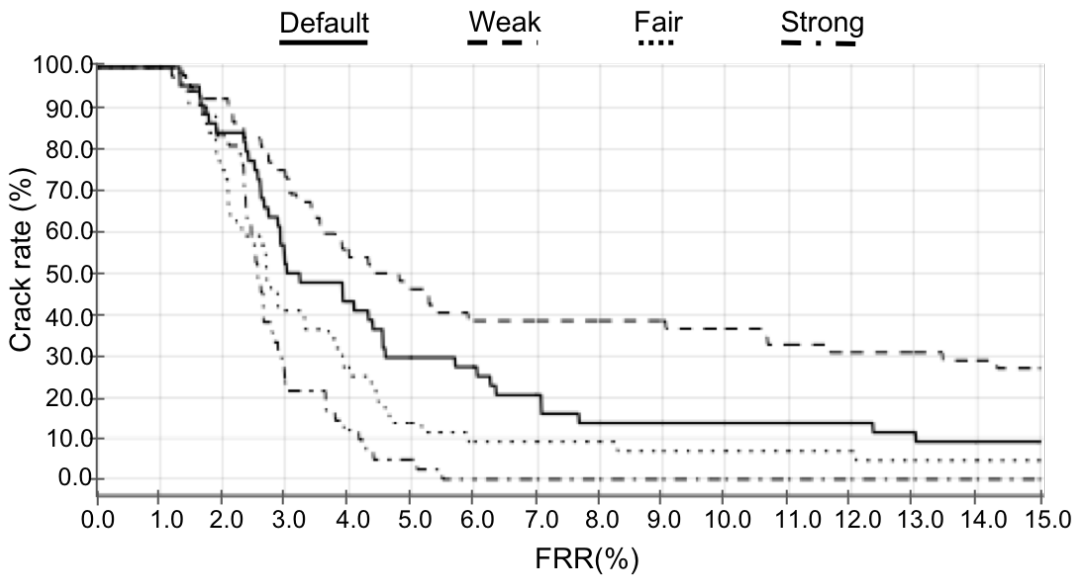


Figure 11. Proportion of cracked gestures with dictionaries at FRRs from 0%- 15% with DTW recognizers in the second study.

V. DISCUSSION AND CONCLUSION

5.1 Discussion

5.1.1 Implications from the first study

From the first study, we set multiple security assessments for gesture passwords and showed how a large proportion of users actually chose vulnerable gestures as their passwords against online guessing attack. However, limitations in this study exists: 1) representativeness of user-chosen gestures collected with online study [38] and 2) further discretization methods that can improve n-gram Markov models. To solve the proposed limitations, future work may focus on collecting real-world gesture passwords such as application usage setting and understanding further n-gram Markov model features that can improve both stroke discretization and representation.

5.1.2 Selection of Password Meter

To successfully increase security of gesture passwords, we proposed four password strength meters that vary minimum compliance policies. We observed security improved as users were given with more strict compliance policies, and that there was a usability cost to these changes, as indicated by prior studies [20].

Weak compliance did not improve security where a user had to fill at least one fourth of the score bar policy. Online dictionary attack derived from the first study successfully guessed 59.62% of *weak* compliance gestures at system threshold and this feature was not significant different from the first study. Furthermore, partial guessing analysis also revealed these gestures have lower guessing entropy in small alpha values ($\alpha < 0.4$) than other compliance gesture sets. We conclude from these results that mandating minimum compliance does not improve security for gesture password strength meter.

Although stronger compliances such as *fair* and *strong* policies significantly improved security - crack rates of 36.36% for *fair*, and 21.43% for *strong* - where users had to fill at least two fourths and three fourths respectively, it negatively impacted usability reducing recall rate 8% to 10% ($p < 0.05$) with

increased set up & recall time ($p< 0.01$). We note that mandating compliance more than half of the total range of score significantly affect usability despite advantages in security.

We recommend the *default* policy over any compliance policy for the following reasons: 1) mere cost in usability - no significant difference in recall rate with prior rate of 98.9% ($p > 0.05$) shorter setup time compared to *strong* compliance policy ($p < 0.005$) and more engagement with high setup cancel over other compliance policies ($p < 0.05$), 2) improvement in security - higher partial guessing entropy over diverse α values than compliance policies and no significant difference in crack rate in high FRR threshold values (FRR > 8.27%) with a stronger compliance policy ($p > 0.05$).

5.1.3 Gesture samples of the second study

We observe a major difference in demographics between the first and the second study. Most of the participants were white (56.32%) followed by Asian (18.47%) in the first study, while the majority of participants were Asian (66.5%) followed by white (28%) in the second study. We need to collect more comparable data set for the second study with larger sample numbers to minimize the impact of this difference in future studies.

Because of reduced number of samples in the second study, limitations in security and usability analysis exist. We do not perform affinity propagation algorithm independently for each compliance policies in the security analysis. It is worth running the technique for the future work to explore how many clusters are generated with larger samples (n > 1000), what is general inter-cluster feature and how cluster centers appear. Another benefit of running a large scale meter study will be collecting reasonable amount of day-2 or week-1 performance data to further examine the usability of gestures.

5.1.4 Design of password strength meter

Although general bar and text feedback is applied identically over different compliance policies, visual factors will be considered in the future work as prior study [10] explored diverse condition of designs to optimize the best performing model. Moreover, it is intriguing to figure out whether interaction effects exist between compliance conditions and diverse designs. In this way, we can further optimize our gesture meter.

5.1.5 Gesture implications

Even though users typically chose simple and homogeneous gestures as their passwords in the first study, security improvement with application of password strength meter suggests user-chosen gesture passwords can be strong if guided by an appropriate system and policy.

5.2 Conclusion

We perform a large scale online study of gesture passwords. We propose a framework analyzing gesture password security and determine key vulnerabilities. We propose novel meter designs to improve gesture passwords by diversifying mandated compliances: *default*, *weak*, *fair*, and *strong*. While the *weak* compliance policy reduces security against online dictionary attack, *fair* and *strong* policies improve security with significant cost in usability. The *default* policy improves security to a similar level of the *fair* compliance policy with an acceptable cost in terms of usability. We conclude gesture passwords are a promising authentication technique for phone lock if supported by well-designed selection policies. We plan to perform a large-scale study of gesture password meters in the future.

References

[1] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz, “Quantifying the security of graphical passwords: The case of android unlock patterns,” in Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, ser. CCS ’13. New York, NY, USA: ACM, 2013, pp. 161–172.

[2] H. Kim and J. H. Huh, “PIN selection policies: Are they really effective?” Computers & Security, vol. 31, no. 4, 2012.

[3] M. Sherman, G. Clark, Y. Yang, S. Sugrim, A. Modig, J. Lindqvist, A. Oulasvirta, and T. Roos, “User-generated free-form gestures for authentication: Security and memorability,” in Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, ser.MobiSys’14. New York, NY, USA: ACM, 2014, pp.176– 189.

[4] T. Nguyen and N. Memon, “Tap-based user authentication for smartwatches,” Computers & Security, vol. 78, pp. 174 – 186, 2018.

[5] C. Liu, G. D. Clark, and J. Lindqvist, “Where usability and security go hand-in-hand: Robust gesture-based authentication for mobile systems,” in Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ser. CHI ’17. New York, NY, USA: ACM, 2017, pp. 374–386.

[6] C. Liu, G. D. Clark, and J. Lindqvist, “Guessing attacks on user- generated gesture passwords,” Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 1, no. 1, pp. 3:1–3:24, Mar. 2017.

[7] J. Bonneau, “The science of guessing: Analyzing an anonymized corpus of 70 million passwords,” in Proceedings of the 33rd IEEE Symposium on Security and Privacy, May 2012, pp. 538–552.

[8] G. Cho, J. H. Huh, J. Cho, S. Oh, Y. Song, and H. Kim, “Syspal: System- guided pattern locks for android,” in 2017 IEEE Symposium on Security and Privacy (SP). IEEE, May 2017, pp. 338–356.

[9] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman, “Of passwords and people: Measuring the effect of password-composition policies,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI ’11. New York, NY, USA: ACM, 2011, pp. 2595–2604.

[10] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor, “How does your password measure up? the effect of strength meters on password creation.” In Proceedings of the 21st USENIX conference on Security symposium, ser. Security'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 5-5.

[11] S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley. “Does my password go up to eleven?: the impact of password meters on password selection.” In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI '13. New York, NY, USA: ACM, 2013, pp. 2379-2388.

[12] N. L. Clarke, S. M. Furnell, P. M. Rodwell, P. L. Reynolds, “Acceptance of Subscriber Authentication Methods For Mobile Telephony Devices”, Computers & Security, vol. 21, pp. 220 - 228, 2002.

[13] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, “Biometric-rich gestures: A novel approach to authentication on multi-touch devices,” In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI ’12. New York, NY, USA: ACM, 2012, pp. 977.

[14] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, “Touch me once and i know it’s you!: Implicit authentication based on touch screen patterns,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI ’12. New York, NY, USA: ACM, 2012, pp. 987–996.

[15] R. Giusti and Gustavo E. A. P. A. Batista, “An Empirical Comparison of Dissimilarity Measures for Time Series Classification.” in Proceedings of the Brazilian Conference of Intelligent Systems, ser. BRACIS ’13. Washington, DC, USA: IEEE, 2013, pp. 82 – 88.

[16] E. M. Taranta II, A. Samiei, M. Maghoumi, P. Khaloo, C. R. Pittman, and J. J. LaViola Jr., “Jackknife: A reliable recognizer with few samples and many modalities,” in Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ser. CHI ’17. New York, NY, USA: ACM, 2017, pp. 5850–5861.

[17] Y. Li, “Protractor: A fast and accurate gesture recognizer,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI ’10. New York, NY, USA: ACM, 2010, pp. 2169–2172.

[18] Y. Yang, G. D. Clark, J. Lindqvist, and A. Oulasvirta, “Free-form gesture authentication in the wild,” in Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ser. CHI ’16. New York, NY, USA: ACM, 2016, pp. 3722–3735.

[19] A. Sahami Shirazi, P. Moghadam, H. Ketabdar, and A. Schmidt, “Assessing the vulnerability of magnetic gestural authentication to video-based shoulder surfing attacks,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI ’12. New York, NY, USA: ACM, 2012, pp. 2045–2048.

[20] R. Shay, S. Komanduri, P. G. Kelley, P. Leon, Michelle L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor, “Encountering stronger password requirements: user attitudes and behaviors.” In Proceedings of Symposium on Usable Privacy and Security ser. SOUPS '10. New York, NY, USA: ACM, 2010, Article 2

[21] P. Inglesant and M. A. Sasse, “The true cost of unusable password policies: Password use in the wild.” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI ’10. New York, NY, USA: ACM, 2010, pp 383 - 392.

[22] B. Ur, F. Alfieri, M. Aung, L. Bauer, N. Christin, J. Colnago, L. F. Cranor, H. Dixon, P. E. Naeini, H. Habib, N. Johnson, and W. Melicher. “Design and Evaluation of a Data-Driven Password Meter.” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI '17. New York, NY, USA: ACM, 2017, pp 3775-3786.

[23] C. Sun, Y. Wang, J. Zheng, “Dissecting pattern unlock: The effect of pattern strength meter on pattern selection”, Journal of Information Security and Applications, vol.19, pp. 308-320, 2014.

[24] G. D. Clark, J. Lindqvist, and A. Oulasvirta, “Composition policies for gesture passwords: User choice, security, usability and memorability,” in 2017 IEEE Conference on Communications and Network Security (CNS). IEEE, Oct 2017, pp. 1–9.

[25] D. Wang, D. He, H. Cheng and P. Wang, "fuzzyPSM: A New Password Strength Meter Using Fuzzy Probabilistic Context-Free Grammars," in 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Toulouse, 2016, pp. 595-606.

[26] Xavier de Carné de Carnavalet and Mohammad Mannan, “From very weak to very strong: Analyzing password-strength meters.” in 2014 Network and Distributed System Security (NDSS), 2014.

[27] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, and T. Vidas, “Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms,” in 2012 IEEE Symposium on Security and Privacy (SP). IEEE, May 2017, pp. 523–537.

[28] C. Castelluccia, M. Dürmuth, and D. Perito, “Adaptive Password-Strength Meters from Markov Models.” in 2012 Network and Distributed System Security (NDSS), 2012.

[29] S. Houshmand and S. Aggarwal, “Building better passwords using probabilistic techniques” in Proceedings of the Annual Computer Security Applications Conference, ser. ACSAC '12. New York, NY, USA: ACM, 2012, pp. 109–118.

[30] J. Galbally, I. Coisel and I. Sanchez, "A New Multimodal Approach for Password Strength Estimation—Part I: Theory and Algorithms," in IEEE Transactions on Information Forensics and Security, vol. 12, pp. 2829-2844, Dec. 2017.

[31] L. Anthony and J. O. Wobbrock, “$n-protractor: A fast and accurate multistroke recognizer,” in Proceedings of Graphics Interface 2012, ser. GI ’12. Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 2012, pp. 117–120.

[32] J. Ma, W. Yang, M. Luo and N. Li, "A Study of Probabilistic Password Models," 2014 IEEE Symposium on Security and Privacy, San Jose, CA, 2014, pp. 689-704.

[33] D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” Cartographica: The International Journal for Geographic Information and Geovisualization, vol. 10, no. 2, pp. 112–122, 1973.

[34] W. Shi and C. Cheung, “Performance Evaluation of Line Simplification Algorithms for Vector Generalization”, The Cartographic Journal, vol. 43, no. 1, pp. 27-44, 2006

[35] V. Satopaa, J. Albrecht, D. Irwin and B. Raghavan, "Finding a Needle in a Haystack: Detecting Knee Points in System Behavior," 31st International Conference on Distributed Computing Systems Workshops, Minneapolis, MN, pp. 166-171, 2011.

[36] B. J. Frey and D. Dueck, “Clustering by passing messages between data points,” Science, vol. 315, no. 5814, pp. 972–976, 2007.

[37] S. Cha, S. Kwag, H. Kim, and J. H. Huh, “Boosting the guessing attack performance on android lock patterns with smudge attacks,” in Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ser. ASIA CCS ’17. New York, NY, USA: ACM, 2017, pp. 313–326.

[38] S. T. Haque, M. Wright, and S. Scielzo, “A study of user password strategy for multiple accounts,” in Proceedings of the Third ACM Conference on Data and Application Security and Privacy, ser. CODASPY ’13. New York, NY, USA: ACM, 2013, pp. 173–176.

[39] E. von Zezschwitz, A. De Luca, and H. Hussmann. 2013. “Survival of the shortest: A retrospective analysis of influencing factors on password composition.” in Proceedings of Human-Computer Interaction – INTERACT, vol. 8119, Springer, Berlin, Heidelberg, 2013

[40] L. Qiu, A. De Luca, I. Muslukhov, and K. Beznosov, “Towards understanding the link between age and smartphone authentication,” in Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, ser. CHI ’19. New York, NY, USA: ACM, 2019, pp. 163:1–163:10.

Acknowledgement

I had a great time during the last 2 years at UNIST. Without guidance and support from many people I met during these years, I could never be where I am today.

Most importantly, I would like to express my deepest gratitude to Professor Ian Oakley for his kind guidance and supervision. Professor Ian Oakley guided me in right academic direction and stayed patient with my struggles during the two years. I was always excited to work with Professor Ian Oakley.

I also would like to sincerely appreciate Professor Sung-phil Kim and Professor Kyungho Lee as the thesis committee members. Professor Sung-phil Kim and Professor Kyungho Lee provided me invaluable interests and advice for my thesis work.

I would like to thank my lab members and intern: Hyunmi, Doyoung, Youngeun, Mintra, Hyunjae, Hongmin, Youryang, Rasel, Yonghwan, and Suhwan for their encouragement. They were good mentors, friends, and co-workers during the two years.

I would like to acknowledge my parents, Jiyong, and Jihyang for supporting me as always.

Appendix

Model Parameters					*Model Performance*		
Len	Ang	Phase	Smoothing	Excl.	CR	SM	CP
2	6	offset	add-1	all	12.99%	45.53%	98.81%
2	6	offset	add-1	dual	12.68%	45.53%	98.81%
2	6	offset	add-1	single	12.95%	45.53%	98.81%
2	6	offset	add-1/n	all	12.99%	45.53%	98.81%
2	6	offset	add-1/n	dual	12.72%	45.53%	98.81%
2	6	offset	add-1/n	single	12.95%	45.53%	98.81%
2	6	offset	Good-Turing	all	12.99%	45.53%	98.81%
2	6	offset	Good-Turing	dual	12.68%	45.53%	98.81%
2	6	offset	Good-Turing	single	12.95%	45.53%	98.81%
2	6	aligned	Good-Turing	single	10.25%	42.25%	99.40%
2	8	offset	add-1	dual	16.96%	58.98%	95.83%
2	8	offset	add-1/n	dual	16.92%	58.98%	95.83%
2	8	offset	Good-Turing	dual	16.92%	58.98%	95.83%
2	8	aligned	add-1	dual	12.49%	46.88%	97.22%
2	8	aligned	add-1	single	12.37%	46.88%	97.22%
2	8	aligned	add-1/n	dual	12.49%	46.88%	97.22%
2	8	aligned	add-1/n	single	11.95%	46.88%	97.22%
2	8	aligned	Good-Turing	dual	10.18%	46.88%	97.22%
2	8	aligned	Good-Turing	single	12.26%	46.88%	97.22%
2	10	offset	add-1	dual	16.65%	61.14%	94.55%
2	10	offset	add-1	single	17.58%	61.14%	94.55%
2	10	offset	add-1/n	dual	16.54%	61.14%	94.55%
2	**10**	**offset**	**add-1/n**	**single**	**17.77%**	**61.14%**	**94.55%**
2	10	offset	Good-Turing	dual	16.54%	61.14%	94.55%
2	10	offset	Good-Turing	single	17.77%	61.14%	94.55%
2	10	aligned	add-1	dual	12.76%	57.75%	94.55%
2	10	aligned	add-1/n	dual	13.03%	57.75%	94.55%
2	10	aligned	add-1/n	single	10.06%	57.75%	94.55%
2	10	aligned	Good-Turing	dual	14.65%	57.75%	94.55%
2	10	aligned	Good-Turing	single	10.45%	57.75%	94.55%
2	12	offset	add-1	dual	14.15%	64.26%	92.63%
2	12	offset	add-1	single	13.61%	64.26%	92.63%
2	12	offset	add-1/n	dual	14.42%	64.26%	92.63%
2	12	offset	add-1/n	single	13.76%	64.26%	92.63%
2	12	offset	Good-Turing	dual	14.34%	64.26%	92.63%
2	12	offset	Good-Turing	single	13.80%	64.26%	92.63%
2	12	aligned	add-1	dual	12.84%	60.68%	93.11%
2	12	aligned	add-1	single	11.49%	60.68%	93.11%
2	12	aligned	add-1/n	dual	12.41%	60.68%	93.11%
2	12	aligned	add-1/n	single	11.57%	60.68%	93.11%
2	12	aligned	Good-Turing	dual	12.99%	60.68%	93.11%
2	12	aligned	Good-Turing	single	11.64%	60.68%	93.11%
2	14	offset	add-1	dual	12.57%	63.69%	91.19%
2	14	offset	add-1	single	13.22%	63.69%	91.19%
2	14	offset	add-1/n	dual	12.07%	63.69%	91.19%
2	14	offset	add-1/n	single	13.15%	63.69%	91.19%
2	14	offset	Good-Turing	dual	12.11%	63.69%	91.19%
2	14	offset	Good-Turing	single	13.15%	63.69%	91.19%
2	14	aligned	add-1	dual	11.33%	63.61%	90.95%
2	14	aligned	add-1	single	10.18%	63.61%	90.95%
2	14	aligned	add-1/n	dual	11.26%	63.61%	90.95%
2	14	aligned	add-1/n	single	10.68%	63.61%	90.95%
2	14	aligned	Good-Turing	dual	10.91%	63.61%	90.95%
3	6	offset	add-1	dual	13.38%	53.16%	94.72%
3	6	offset	add-1	single	11.84%	53.16%	94.72%
3	6	offset	add-1/n	dual	13.38%	53.16%	94.72%
3	6	offset	add-1/n	single	11.84%	53.16%	94.72%
3	6	offset	Good-Turing	dual	13.38%	53.16%	94.72%
3	6	offset	Good-Turing	single	11.84%	53.16%	94.72%
3	8	offset	add-1	dual	14.49%	70.70%	89.74%
3	8	offset	add-1/n	dual	14.96%	70.70%	89.74%
3	8	offset	Good-Turing	dual	14.88%	70.70%	89.74%
3	8	aligned	add-1	single	10.72%	55.86%	92.31%
3	8	aligned	add-1/n	dual	11.68%	55.86%	92.31%
3	8	aligned	add-1/n	single	10.45%	55.86%	92.31%
3	8	aligned	Good-Turing	single	10.95%	55.86%	92.31%
3	**10**	**offset**	**add-1**	**dual**	**15.27%**	**73.59%**	**88.65%**

Model Parameters					*Model Performance*		
Len	Ang	Phase	Smoothing	Excl.	CR	SM	CP
3	10	offset	add-1	single	11.95%	73.59%	88.65%
3	10	offset	add-1/n	dual	15.23%	73.59%	88.65%
3	10	offset	add-1/n	single	11.26%	73.59%	88.65%
3	10	offset	Good-Turing	dual	15.23%	73.59%	88.65%
3	10	offset	Good-Turing	single	11.83%	73.59%	88.65%
3	10	aligned	add-1	dual	14.69%	70.51%	88.13%
3	10	aligned	add-1/n	dual	13.26%	70.51%	88.13%
3	10	aligned	Good-Turing	dual	14.46%	70.51%	88.13%
3	12	offset	add-1	dual	13.80%	76.56%	82.75%
3	12	offset	add-1	single	10.06%	76.56%	82.75%
3	12	offset	add-1/n	dual	13.92%	76.56%	82.75%
3	12	offset	add-1/n	single	10.25%	76.56%	82.75%
3	12	offset	Good-Turing	dual	13.99%	76.56%	82.75%
3	12	offset	Good-Turing	single	10.18%	76.56%	82.75%
3	12	aligned	add-1	dual	11.84%	73.05%	85.75%
3	12	aligned	add-1/n	dual	11.22%	73.05%	85.75%
3	12	aligned	Good-Turing	dual	12.14%	73.05%	85.75%
3	14	offset	add-1	dual	14.80%	78.10%	80.30%
3	14	offset	add-1	single	10.52%	78.10%	80.30%
3	14	offset	add-1/n	dual	12.95%	78.10%	80.30%
3	14	offset	Good-Turing	dual	13.07%	78.10%	80.30%
3	14	aligned	add-1	dual	11.80%	77.49%	79.65%
3	14	aligned	add-1/n	dual	10.56%	77.49%	79.65%
3	14	aligned	Good-Turing	dual	12.34%	77.49%	79.65%
4	6	offset	add-1	dual	14.11%	59.33%	90.38%
4	6	offset	add-1	single	11.68%	59.33%	90.38%
4	6	offset	add-1/n	dual	14.15%	59.33%	90.38%
4	6	offset	add-1/n	single	11.60%	59.33%	90.38%
4	6	offset	Good-Turing	dual	14.15%	59.33%	90.38%
4	6	offset	Good-Turing	single	11.60%	59.33%	90.38%
4	8	offset	add-1	dual	10.87%	77.06%	84.56%
4	8	offset	add-1/n	dual	11.33%	77.06%	84.56%
4	8	offset	Good-Turing	dual	11.45%	77.06%	84.56%
4	8	aligned	add-1	single	10.79%	62.88%	86.21%
4	8	aligned	add-1/n	single	10.25%	62.88%	86.21%
4	8	aligned	Good-Turing	single	10.56%	62.88%	86.21%
4	10	offset	add-1	dual	12.03%	81.46%	80.24%
4	10	offset	add-1	single	14.57%	81.46%	80.24%
4	10	offset	add-1/n	dual	11.87%	81.46%	80.24%
4	10	offset	add-1/n	single	13.76%	81.46%	80.24%
4	10	offset	Good-Turing	dual	11.80%	81.46%	80.24%
4	10	offset	Good-Turing	single	14.15%	81.46%	80.24%
4	10	aligned	add-1	dual	12.72%	78.60%	80.65%
4	10	aligned	add-1/n	dual	10.53%	78.60%	80.65%
4	10	aligned	Good-Turing	dual	12.03%	78.60%	80.65%
4	**12**	**offset**	**add-1**	**dual**	**15.46%**	**84.70%**	**73.67%**
4	12	offset	add-1	single	11.60%	84.70%	73.67%
4	12	offset	add-1/n	dual	12.72%	84.70%	73.67%
4	12	offset	add-1/n	single	11.29%	84.70%	73.67%
4	12	offset	Good-Turing	dual	14.00%	84.70%	73.67%
4	12	offset	Good-Turing	single	11.45%	84.70%	73.67%
4	12	aligned	add-1	dual	11.91%	82.81%	75.13%
4	12	aligned	add-1	single	11.03%	82.81%	75.13%
4	12	aligned	add-1/n	dual	11.30%	82.81%	75.13%
4	12	aligned	add-1/n	single	10.95%	82.81%	75.13%
4	12	aligned	Good-Turing	dual	10.87%	82.81%	75.13%
4	12	aligned	Good-Turing	single	10.95%	82.81%	75.13%
4	14	offset	add-1	dual	12.95%	86.39%	69.74%
4	14	offset	add-1	single	10.95%	86.39%	69.74%
4	14	offset	add-1/n	dual	12.53%	86.39%	69.74%
4	14	offset	add-1/n	single	11.57%	86.39%	69.74%
4	14	offset	Good-Turing	dual	12.45%	86.39%	69.74%
4	14	offset	Good-Turing	single	11.72%	86.39%	69.74%
4	14	aligned	add-1	dual	11.26%	85.89%	69.33%
4	14	aligned	add-1	single	10.14%	85.89%	69.33%
4	14	aligned	add-1/n	dual	10.06%	85.89%	69.33%
4	14	aligned	Good-Turing	dual	10.18%	85.89%	69.33%

Table 6. Subset of 134 *n*-gram Markov models generated from the first study achieving crack rates greater than 10%. Metrics of Crack Rate (CR), Similarity (SM), and Completeness (CP) are calculated for each model. Selected models for optimization are highlighted in bold.