2-2013

I Can Be You: Questioning the Use of Keystroke Dynamics as Biometrics

Chee Meng TEY
Singapore Management University, cmtey.2008@smu.edu.sg

Payas GUPTA
Singapore Management University, payas.gupta.2008@smu.edu.sg

Debin GAO
Singapore Management University, dbgao@smu.edu.sg

Citation

TEY, Chee Meng; GUPTA, Payas; and GAO, Debin. I Can Be You: Questioning the Use of Keystroke Dynamics as Biometrics. (2013). *Annual Network and Distributed System Security Symposium 20th NDSS 2013, 24-27 February*. 1-16.
Available at: https://ink.library.smu.edu.sg/sis_research/1699

I can be You: Questioning the use of Keystroke Dynamics as Biometrics

Tey Chee Meng, Payas Gupta, Debin Gao
School of Information Systems, Singapore Management University
{cmtey.2008,payas.gupta.2008,dbgao}@smu.edu.sg

Abstract

Keystroke dynamics refer to information about the typing patterns of individuals, such as the relative timing when the individual presses and releases each key. Prior studies suggest that such patterns are unique and cannot be easily imitated. This lays the foundation for the use of keystroke biometrics in authentication systems. The research effort in this area has thus far focused on novel detection techniques to differentiate between legitimate users and imposters. In this paper, we demonstrate a novel feedback and training interface named Mimesis. Mimesis provides both positive and negative feedback on the differences between a submitted pattern vs. a reference pattern. This allows one person to imitate another through incremental adjustment of typing pattern. We show that even for targets whose typing patterns are only partially known, training with Mimesis allows attackers to defeat one of the best anomaly detection engines using keystroke biometrics. For a group of 84 participants playing the role of attackers and 2 eight-character passwords of different difficulty, the false acceptance rate (FAR) of the easy and difficult password increases from 0.24 and 0.20 respectively (before Mimesis training) to 0.63 and 0.42 respectively (after Mimesis training with partial information of the victim). With full information, the FAR increases to 0.99 for both passwords for the 14 best attackers.

1. Introduction

Biometrics are the oldest form of authentication; people verify each other on telephone based on voice, humans recognize each other by face when they meet. There is a wide variety of candidate characteristics, including facial features, speech patterns, hand geometry [8], fingerprints, iris scans, DNA, typing patterns, signature geometry[1] and mouse dynamics. These biometrics can be classified into two major categories: 1) physiological biometric – a biometric that is based on a physical trait of an individual, e.g., facial features, hand geometry, fingerprints, iris scans, and DNA; and 2) behavioral biometric – a biometric that is based on the behavioral trait of an individual, e.g., speech patterns, typing patterns, signatures, and mouse dynamics.

A central issue[2] of biometrics security concerns the uniqueness of the biometrics feature. In the context of fingerprint biometrics for example, there had been cases where suspects were wrongly identified through fingerprints [10]. For keystroke biometrics, prior literature had shown that although typing patterns between individuals do overlap, and misidentification is possible as in fingerprinting, the error rates are low enough such that typing patterns can be considered unique to each individual [7, 14, 21, 17, 16].

In this paper, we question the uniqueness property of keystroke biometrics. We consider the scenario where attackers are shown the typing pattern[3] of their victims and make a conscious attempt to imitate. If imitation is possible, the error rates of detection engines would become unacceptably high. This means keystroke dynamics would be unsuitable for use as a biometrics feature. The existing commercial security solutions using keystroke biometrics [4, 1, 6, 5, 3, 2] can therefore be attacked.

The majority of literature in this area focused on finding a detection algorithm that best separates the legitimate users from imposters. The only work [18] which resembles ours shows that (based on 21 participants) the provision of feedback shortens the distance between the attacker and victim's typing pattern by 9.7%. While differences can be reduced, Rundhaug et al. suggested that an attack remains difficult

[1] Signature geometry encompasses not just the look of the signature, but also possibly the pen pressure, signature speed, etc.

[2] Another important issue of biometrics security which is outside the scope of this paper, is verifying if the authentication data came directly from the owner. For example, in the case of fingerprint biometrics, if such verification is absent, arbitrary fingerprints may be forged. Geller et al. demonstrated how fingerprints can be forged in a forensic context [11]. Boatwright et al. cited an instance where gelatin created fingerprints were used to gain unauthorized access [8]. Likewise, for keystroke biometrics, if such verification is absent, an automated system may be used to deliver the desired typing pattern to the detection engine.

[3] Scenarios where the typing pattern may be known includes (a) an attacker captures samples of the victim's password typing and (b) information in the biometrics database was leaked.

and proposed larger scale experiments to verify the feasibility of such attacks. In this paper, we demonstrate that it is possible to imitate someone else's keystroke typing if appropriate feedback is provided.

We propose a novel feedback interface Mimesis with the following design goals: (a) The information must be easy to understand with minimal cognitive load required. The latter is for the attackers to focus on their imitation task. (b) The interface should provide specific tips on particular aspects to improve on. (c) Both positive and negative feedback should be provided to the attacker so that she can repeatedly make minor adjustments to her typing pattern to imitate better.

We assembled a group of 84 participants to play the role of attackers against one of the best keystroke biometrics based authentication systems by Araujo et al. [7] (based on the evaluation by Killourhy and Maxion [15]). We evaluate the effectiveness of Mimesis and demonstrate that there exists individuals who can adjust their typing pattern to imitate someone else. We demonstrate this attack using two scenarios (a) when the attacker only has an incomplete model of the victim's typing pattern, such as when only a limited number of victim typing samples are available to infer the model, and (b) when the attacker has complete information.

We show that even for attacks based on an incomplete model, the average false acceptance rate increases from 0.24 to 0.63 for an easy password, and from 0.20 to 0.42 for a harder password. For the best attackers, given a complete model of the victim's keystroke typing, we show that a false acceptance rate of 0.99 can be achieved. Since our results shows that even the best detector can be defeated by imitation with Mimesis, we draw the conclusion that keystroke biometrics is unsuitable as an authentication mechanism.

2. Background

In this section, we describe the commonly followed procedures in the evaluation of a keystroke biometrics authentication system. We first provide an overview of keystroke dynamics in Section 2.1 followed by the information that is categorized for the anomaly detection methodology in Section 2.2. We then describe the training process and the calculation of the anomaly score in Section 2.3. Finally, the computation of the threshold from the training and the anomalous data set is shown in Section 2.4.

2.1. Choice of timing information

Keystroke dynamics refer to information about the typing pattern. For example, pressing and releasing of a keystroke pair (k_a, k_b) results in 4 timings which are of interest to keystroke biometrics systems: (a) key-down time of k_a: $t^{\downarrow}_{k_a}$, (b) key-up time of k_a: $t^{\uparrow}_{k_a}$, (c) key-down time of k_b: $t^{\downarrow}_{k_b}$ and (d) key-up time of k_b: $t^{\uparrow}_{k_b}$

From these absolute time measurements, four relative timings can be derived:

- an inter-keystroke timing between k_a and k_b: $I_{k_a,k_b} = t^{\downarrow}_{k_b}$ - $t^{\downarrow}_{k_a}$.
- hold timing of k_a: $H_{k_a} = t^{\uparrow}_{k_a}$ - $t^{\downarrow}_{k_a}$
- hold timing of k_b: $H_{k_b} = t^{\uparrow}_{k_b}$ - $t^{\downarrow}_{k_b}$
- a key up-down timing between k_a and k_b: $U_{k_a,k_b} = t^{\downarrow}_{k_b}$ - $t^{\uparrow}_{k_a}$

Different anomaly detectors used in keystroke biometrics used different combinations of I, H and U such as I, H and U [7], only I [13, 7], only H [7], only U [7], I and H [7], H and U [9, 7], I and U [7].

2.2. Data vectorization

In the context of this paper, we are interested in the timing information collected for each password in a keystroke biometric based authentication system. These timing information are typically stored in vectors. However, prior research differs in the layout of the vectors. In this paper, as in the case for Araujo et al. [7], we store in each collected vector the timing information of each password, resulting in n vectors of length $2l - 1$ (because we collect $l - 1$ inter-keystroke times and l hold times). For brevity, the remainder of this section assumes the case where only the I and H timing components are collected (see Table 1).

Sample	Inter-Keystroke Time			Hold Time		
# 1	$I^1_{k_1,k_2}$	$\cdots$	$I^1_{k_{l-1},k_l}$	$H^1_{k_1}$	$\cdots$	$H^1_{k_l}$
# 2	$I^2_{k_1,k_2}$	$\cdots$	$I^2_{k_{l-1},k_l}$	$H^2_{k_1}$	$\cdots$	$H^2_{k_l}$
$\vdots$	$\vdots$					
# n	$I^n_{k_1,k_2}$	$\cdots$	$I^n_{k_{l-1},k_l}$	$H^n_{k_1}$	$\cdots$	$H^n_{k_l}$

Table 1. Example of data vectorization

For a password, e.g. 'serndele', each timing information vector $\boldsymbol{z}$ can be represented as

$$\boldsymbol{z} = \overbrace{\underbrace{I_{s,e}, \ldots, I_{l,e}}_{\text{inter-keystroke time}}, \underbrace{H_s, \ldots, H_e}_{\text{hold time}}}^{\text{`serndele'}}$$

The collected vectors are typically divided into 4 sets when evaluating a keystroke biometrics system. For each user of the system, 1 set of normal timing vectors from that user and 1 set of anomalous timing vectors is used for training. 1 additional set each of normal and anomalous timing vectors are used for testing. In an experimental setting, the anomalous timing vectors for each user is typically constructed from the normal timing vectors of all other users in the same authentication system.

2.3. Anomaly detector training and scoring

Once the training data set is collected, the next step is to train the anomaly detector. The purpose of training is to find parameters for the detector corresponding to the particular set of training data. Detectors differ in the choice of parameters. For example, in the papers of Joyce et al. and Cho et al. [13, 9] only the mean vector is needed, whereas Araujo et al. requires both a mean vector and an absolute deviation vector [7]. Once the parameters are determined, a detector can compute an anomaly score for each test vector.

Computation of mean vector The mean vector, denoted by $\bar{\boldsymbol{x}}$ is computed from:

$$\begin{aligned}\bar{\boldsymbol{x}} = &\left(\frac{\sum_{i=1}^{n} I^{i}_{k_1,k_2}}{n}, \ldots, \frac{\sum_{i=1}^{n} I^{i}_{k_{l-1},k_l}}{n}, \frac{\sum_{i=1}^{n} H^{i}_{k_1}}{n}, \ldots, \frac{\sum_{i=1}^{n} H^{i}_{k_l}}{n} \right) \\ =&(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{2l-1})\end{aligned}$$

Computation of absolute deviation vector The absolute deviation $\boldsymbol{d}$ can be computed from:

$$\begin{aligned}\boldsymbol{d} = &\left(\frac{\sum_{i=1}^{n} |I^{i}_{k_1,k_2} - \bar{x}_1|}{n-1}, \ldots, \frac{\sum_{i=1}^{n} |I^{i}_{k_{l-1},k_l} - \bar{x}_{l-1}|}{n-1}, \right. \\ &\left. \frac{\sum_{i=1}^{n} |H^{i}_{k_1} - \bar{x}_l|}{n-1}, \ldots, \frac{\sum_{i=1}^{n} |H^{i}_{k_l} - \bar{x}_{2l-1}|}{n-1} \right) \\ =&(d_1, \ldots, d_{2l-1})\end{aligned}$$

Euclidean distance based anomaly score After the parameters of the detector are computed, the anomaly score for any given test vector can be computed by applying the detection algorithm. Denoting the test vector as $\boldsymbol{ts} = (ts_1, ts_2, \ldots, ts_{2l-1})$, we calculate the Euclidean distance based anomaly score a_e of $\boldsymbol{ts}$ using,

$$a_e = \sqrt{\sum_{j=1}^{2l-1} (ts_j - \bar{x}_j)^2}$$

Note that the calculation of Euclidean distance requires only the mean vector of the victim but not the absolute deviation vector.

Manhattan distance based anomaly score Unlike the Euclidean distance, the Manhattan (scaled) distance requires both the mean and absolute deviation vector. This anomaly score a_s is computed using,

$$a_s = \sum_{j=1}^{2l-1} \frac{|ts_j - \bar{x}_j|}{d_j}$$

2.4. Computation of threshold

Anomaly detectors take a test vector as input and output a single bit of information classifying the input vector as either normal or anomalous. An anomaly score by itself is therefore insufficient. A threshold, or decision criteria is also needed, such that the anomaly score can be mapped to a normal/anomalous range. Setting a strict threshold means that less anomalous vectors are wrongly classified as normal. The percentage of such vectors is known as the false acceptance rate (FAR). A strict threshold, however, also means that more normal vectors are wrongly classified as anomalous. The percentage of such vectors is known as the false rejection rate (FRR). A lenient threshold on the other hand has the opposite effect: FAR increases (worse) but FRR decreases (better).

For both the Euclidean detector and Manhattan (scaled) detector, if the anomaly score is higher than the threshold, the test vector ts_j is classified as anomalous. Conversely, if the anomaly score is lower than the threshold, ts_j is classified as normal. In the context of keystroke biometric based authentication, submission of a ts_j that is classified as anomalous means that the authentication attempt is rejected and similarly, if ts_j is classified as normal then the authentication attempt is accepted. The selection of the threshold therefore involves a tradeoff between FAR and FRR. A common way to set the threshold is to chose it such that the FAR and FRR are equal. The value of the FAR (or FRR) at such a threshold is known as the equal error rate (EER). Once the threshold is computed, an anomaly detector becomes ready for the classification task. Typically, a set of test vectors containing both normal and anomalous vector are used to evaluate the effectiveness of the detector.

3. Experimental design considerations

This paper questions whether it is possible for one person to imitate another's typing pattern. Our approach is to provide feedback, such that the imitator can incrementally adjust her typing pattern to be closer to her target's. We also want to investigate the factors affecting the effectiveness of imitation. In this section, we explain our experimental design considerations.

3.1. Choice of detector and its features

We chose the Manhattan (scaled) anomaly detector by Araujo et al. [7] (the best out of 14 anomaly detectors evaluated by Killourhy and Maxion [15]). In Section 2.3, we have provided a brief description of its computation of anomaly score. Araujo et al. conducted 7 experiments based on different combinations of inter-keystroke timing, hold timing and key down-up timing. The inter-keystroke timing and hold timing are always positive. Key down-up timing refers to the time between releasing the previous key and pressing the next key. It is possible that the next key is pressed before the previous key is released, therefore, the key down-up timing can sometimes be negative. Although only 2 timings are independent[4], the best performing combination used all three timings (choice VII [7]). In our study however, we chose to use a combination of only inter-keystroke timing and hold timing (choice V [7]), which had a FAR and FRR of 5.59% and 1.27% respectively, compared to 1.89% and 1.45% for choice VII [7].

Our reasons for excluding U are: firstly, including U increases the amount of feedback information to show in the feedback interface by about 50%. Our concern is that this may overwhelm the participants. Secondly, I and H timings are rather intuitive and participants should have little issue understanding it. U on the other hand is less intuitive and can even be negative. By excluding it, we avoid the possibility of under-performance due to poor understanding of this parameter.

3.2. Attack scenarios

As a prerequisite for imitation, an attacker must know the typing pattern of her victim. When designing the experiments, we considered 2 possible scenarios whereby the typing pattern may be obtained. In the first scenario, the attacker is able to extract the victim pattern from a compromised biometrics database. From the attacker's point of view, this is the optimum scenario, because it allows her to build an exact replica of the detector with the victim's parameters for her training needs. In the second scenario, the attacker may be able to capture samples of the victim's keystrokes as she is authenticating (e.g. by installing a key-logger). If the attacker is able to capture a large number of samples, she would be able to get a good approximation of the victim parameters.

A question however arises when the attacker is only able to capture a relatively small number of samples. For our chosen detector, there are 2 parameters which are important to the attacker: the mean vector and the absolute deviation vector. It is possible that only one such parameter can be estimated with a small number of samples. An investigation into imitation effectiveness should therefore include an analysis of the extent to which both vectors can be approximated. If only one vector can be approximated, it is useful to measure the imitation effectiveness under such a scenario. We refer to this as the partial information scenario.

3.3. Motivating the participants

We consider motivation as a key factor that decides the outcome of our experiments. For that, we gave special considerations in three aspects. Firstly, the feedback interface must be designed to sustain the participant's interest. Secondly, good imitators should be rewarded for their extra efforts in the form of a performance bonus. Lastly, the duration of the experiments must strike a balance between (1) pushing the participants to try hard enough and (2) not setting it so long that it bores the participants.

Given that there will be multiple experiments, we decided that the first imitation experiment should have a fixed duration of about 30-45 minutes. For subsequent experiments, we assumed that we can identify and select those with high motivation. For these participants, the experiment is designed to be target based. That is, they will be given targets and associated rewards. There is no duration constraint: they can leave anytime or ask for more time.

3.4. Basis for comparison of results

In each experiment, we need to determine how much each attacker has improved and more importantly, their chance of success for the next try if they are sent to attack a system in an actual scenario. If we used all the data in each experiment to determine the improvement, there will be a problem of underestimation in 2 aspects. Firstly, each attacker is likely to spend a good part of her time exploring and fine tuning her keystrokes. The data during this period reflects the trials and errors of each attacker's learning process, but not the outcome of the learning. Secondly, for the fixed duration experiments, boredom may set in after some point. The participant may be just "clocking" their time without trying hard.

We therefore decide that for all experiments, feedback shall include a history of their last 20 tries. Comparison of results across different experiments is also be based on the same set of data. We name the best 20 consecutive tries of each experiment the b20 data set. Our justification for the choice of 20 is that if an attacker achieves a certain target for 20 tries, even if she has only a 50% chance of repeating the feat for the next 20 tries, the probability of success for the *next* try is given by $\sqrt[20]{0.5} = 0.97$. It means that an attacker who has been trained before based on the b20 targets has a significant chance of success in a real life scenario.

[4] The inter-keystroke timing, hold timing and key up-down timing are related: $I_{k_a,k_b} = H_{k_a,k_b} + U_{k_a,k_b}$. Hence, any one of the three can be calculated if the other 2 are known.

3.5. Choice of password

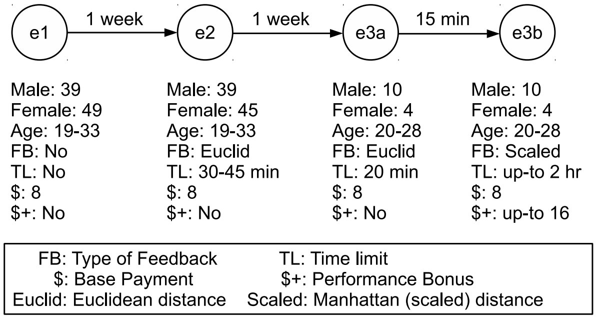
One common problem with password based authentication systems is the prevalence of weak passwords. For example, ‘password’ is the top password choice. Peacock et al. considered keystroke dynamics as an effective low cost countermeasure [17]. The argument is that even if attackers guess the weak password, they cannot imitate the typing pattern. However, weak password tend to be easy to type. If attackers can imitate better as the password weakens, then the effectiveness of keystroke biometrics in mitigating weak passwords is lesser than previously assumed.

For this reason, we decide to have 2 groups of participants. One group practises based on an easy password chosen to minimize finger movements on a standard US keyboard. The other group practices based on a relatively harder password chosen to maximize finger movements and therefore difficulty of typing. We also added a criteria that it must contain mixed case alphabets, at least 1 number and at least 1 punctuation to ensure compliance with the requirements of a strong password. We chose the weak and strong password as ‘serndele’ and ‘ths.ouR2’ respectively. Having two groups allows us to evaluate the effect of password typing difficulty on imitation.

4. Experimental setup

In this section, we describe our experimental setup[5] given the considerations in Section 3. We divide our investigation into 4 experiments, e1, e2, e3a and e3b. In e1, we collect the keystroke dynamics for each participant; e2 and e3a involves imitation training with only the mean vector given (the latter is a repeat of the former, but with one week interval in between); and e3b studies the effectiveness of using Manhattan (scaled) distance as the feedback. Figure 1 shows the experimental structure and demographics.

Timing information was collected using Javascript, by monitoring key-down and key-up events. Although Javascript timing measurements have a granularity of millisecond (via the Date object), the actual timing granularity is affected by the operating system. For example, Windows XP based machines have a scheduling tick quantum of approximately 16 ms. This implies that the Javascript events which we are monitoring occur at the timing of the quantum. The timings collected on such machines therefore are in multiples of approximately 16 ms. In comparison, the timing granularity of keystroke events in the literature varies from 0.2ms [15], 1ms [7] and 10ms [20, 12]. In all the experiments, mistyped samples were discarded. All participants were paid $8 for each experiment they completed.



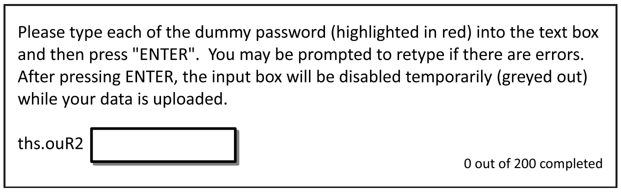
e1	e2	e3a	e3b
Male: 39	Male: 39	Male: 10	Male: 10
Female: 49	Female: 45	Female: 4	Female: 4
Age: 19-33	Age: 19-33	Age: 20-28	Age: 20-28
FB: No	FB: Euclid	FB: Euclid	FB: Scaled
TL: No	TL: 30-45 min	TL: 20 min	TL: up-to 2 hr
$: 8	$: 8	$: 8	$: 8
$+: No	$+: No	$+: No	$+: up-to 16

FB: Type of Feedback	TL: Time limit
$: Base Payment	$+: Performance Bonus
Euclid: Euclidean distance	Scaled: Manhattan (scaled) distance

Figure 1. Experimental structure and demographics

4.1. Exp e1: Training Data Collection

e1 is designed based on the enrolment phase of existing keystroke dynamics based authentication systems, where each user is required to submit a certain number of samples to train the anomaly detector. Figure 2 shows the interface used. 88 participants took part in this experiment. This part of the study was conducted online and the participants were asked to type in the password (provided by us) in an input box via our web interface. Each participant is required to type the same password 200 times without taking any break.


Please type each of the dummy password (highlighted in red) into the text box and then press "ENTER". You may be prompted to retype if there are errors. After pressing ENTER, the input box will be disabled temporarily (greyed out) while your data is uploaded.

ths.ouR2

0 out of 200 completed

Figure 2. User interface for e1

4.2. Exp e2: Imitation using Euclidean distance

In Section 3.2, we mentioned the need to analyst the extent to which the mean and absolute deviations vectors can be approximated by relatively few samples as well as measuring the imitation effectiveness in such a case. Based on a preliminary analysis, we found that the mean can be estimated more accurately than the absolute deviation (see Section 6.1 for the justification). We designed e2 to investigate the imitation effectiveness when only the mean vector is known to the attacker. Without the absolute deviation vector, the actual anomaly score for each vector cannot be calculated. The feedback therefore can only provide an approximation. We chose the Euclidean distance based anomaly score (described in Section 2.3) for this purpose.

[5] The experiments conducted were approved by the Institutional Review Board of the Singapore Management University (IRB approval reference IRB-12-0031-A0039). Data collected from the participants were anonymized and protected according to the procedures described in the corresponding IRB submission documents.

e2 was conducted 1 week after e1. The choice of 1 week was made so that (a) participants have enough time to rest after e1 and (b) we have enough time to compute the parameters of the anomaly detectors needed for the experiment. 84 participants played the role of attackers. Ten victims were chosen randomly from among the participants of e1. Each attacker in e2 is randomly assigned a victim from the set of 10 to imitate his/her typing .

Two constraints apply to the assignment: (a) the attacker and the victim cannot be the same person, (b) the attacker and victim were assigned the same password in e1. Each attacker was given an approximate anomaly score feedback based on the Euclidean distance and required to spend at least 30 minutes. An additional 15 minutes were provided if requested. No performance bonus was offered, but the attackers were told that only the best few will be chosen for e3a and e3b (for which they will be paid up to $28). The feedback interface is more elaborated compared to e1 and is described in Section 5. At the end of e2, participants answered a questionnaire on the imitation experience.

Computation of the threshold for each victim requires a set of anomalous data in addition to the normal data. Following the same procedure as Killourhy and Maxion [15], we build the anomalous data for each victim using the first 5 samples of the passwords typed from all other participants in the same category.

To help evaluate the effects of different input devices on the imitation outcome, some participants were asked to type directly on their own notebook keyboard. Others were asked to use an external keyboard provided by us.

4.3. Exp e3a: Additional imitation session with Euclidean distance

e3a is the second imitation experiment conducted and is very similar to e2. It was conducted 1 week after e2 to allow time for the attackers to rest and reflect, as well as for the researchers to process the data and pick the best attackers. 14 participants were chosen from the attackers of e2 using a subjective gauge of the interest level and aptitude based on (a) the enthusiasm observed during e2, (b) the number of samples submitted, (c) their response to our queries if they would like a second session with more time and (d) the improvement profile (see Figure 5).

The set of attacker-victim assignment, the anomaly score calculation (Euclidean distance) and the feedback interface remains unchanged. Each attacker is required to spend 20 minutes. As in e2, no performance bonus was offered. The purpose of this session is to investigate the effect an additional session has on the imitation outcome.

4.4. Exp e3b: Imitation using Manhattan distance

The goal of this part of the experiment is to analyze the effectiveness of keystroke imitation if an attacker is highly motivated and can obtain the full set of victim's typing pattern parameters. e3b is the final imitation experiment. It was conducted after a break of 15 minutes from e3a, so as to allow the attackers rest and refreshments.

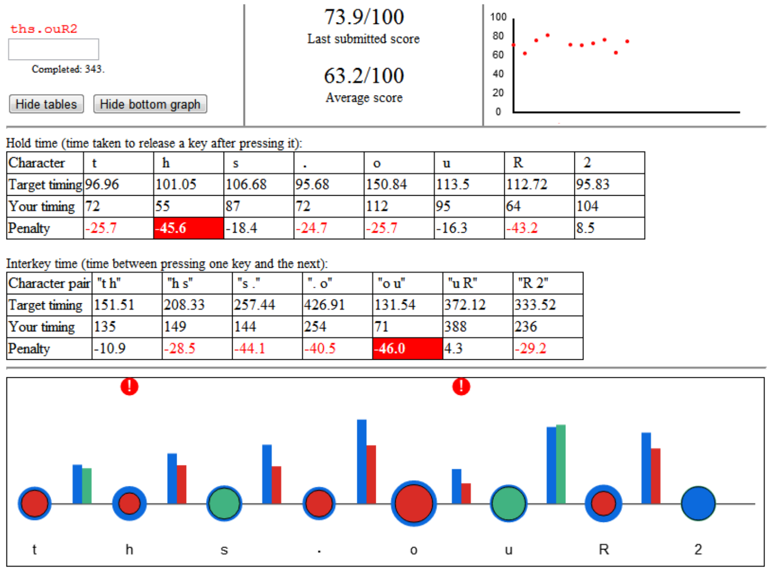
In e3b, the interface was changed to (a) compute the anomaly score based on the Manhattan (scaled) distance, and (b) include information about the absolute deviation vector (see Section 5). Two performance bonuses of equal to and double the base payment rate were offered. The first bonus is given if they can produce a consecutive run of 20 vectors *all of which* are scored better than their best average score in e2. The second bonus increases this difficulty by 10%. The attackers are also offered additional time up to 2 hours. All other experimental settings including the set of attackers and their demographics remain unchanged from e3a.

5. Mimesis

Mimesis is the feedback interface for our imitation experiments. We provide the design goals of Mimesis in Section 1. The design of the feedback is important because the quality of the feedback directly affects the outcome of our imitation experiments. Inadequate or inappropriate feedback may hamper the performance of the attacker. Figure 3 shows the Mimesis interface for the scenario where only partial information of the victim is available. We denote this interface as M_{part}. The interface for the full information scenario (denoted M_{full}) is similar. Mimesis consists of 5 components.

Top-left section This contains the password which the attacker a is trying to imitate and an input box to type in. a can also break up the password into segments and practice on each segment separately. Two buttons here provides participants with the option to hide both the tables in the center section and/or the graphical form of feedback in the bottom section.

Top-center section This contains the attack score (computed from the anomaly score using negative linear scaling and then translated to fit within 0 and 100) for a's last submitted password. It also shows the average of the recent 20 scores. For M_{part}, the scores are derived from the Euclidean distance. For M_{full}, the scores are calculated from the Manhattan (scaled) distance. The scores are only updated when the attacker presses the enter key and only after typing the correct password. For practice sessions with password segments, no score is computed.



ths.ouR2

Completed: 343.

73.9/100

Last submitted score

63.2/100

Average score

Hold time (time taken to release a key after pressing it):

Character	t	h	s	.	o	u	R	2
Target timing	96.96	101.05	106.68	95.68	150.84	113.5	112.72	95.83
Your timing	72	55	87	72	112	95	64	104
Penalty	-25.7	-45.6	-18.4	-24.7	-25.7	-16.3	-43.2	8.5

Interkey time (time between pressing one key and the next):

Character pair	"t h"	"h s"	"s ."	". o"	"o u"	"u R"	"R 2"
Target timing	151.51	208.33	257.44	426.91	131.54	372.12	333.52
Your timing	135	149	144	254	71	388	236
Penalty	-10.9	-28.5	-44.1	-40.5	-46.0	4.3	-29.2

Figure 3. Mimesis interface with Euclidean distance

Top-right section This contains a graphical plot of the attack scores for the recent 20 correctly typed passwords. Our basis of comparison considers the best 20 consecutive vectors (see Section 3.4). This section therefore allows each attacker to easily grasp her past performance.

Center section This section contains two tables corresponding to the hold timing H and inter-keystroke timing I respectively. The tables provides numerical feedback on the victim's mean vector and the last submitted attacker vector. For M_{full}, a weight w computed from the corresponding victim's absolute deviation is also shown so that attackers know the relative importance of each key in calculating the attack score. To help the attackers make their adjustments, we also provide positive and negative feedback in the form of a penalty (the last row of each table) to the score. Penalty is computed based on victim's and attacker's typing. The timing components accounting for the largest differences are highlighted in red as negative feedback. Components that are similar between attacker and victim are highlighted in a different color as a positive feedback.

Bottom section This contains a graphical form of the information shown in the two tables. Circles represent H_{k_a} and vertical bars between the circles represent I_{k_a,k_b}. The larger/smaller the circle is, the longer/shorter H_{k_a} is, respectively. (Circles are chosen for the similarity with finger marks left on glass surfaces; the harder one presses, the bigger the mark.) Similarly, the taller/shorter the vertical bar is, the longer/shorter I_{k_a,k_b} is, respectively. As is the case for numerical feedback, both positive and negative feedback are provided, using different color code. Red color is used as negative feedback when the differences in component timings are large. An additional alert is placed above the component with the most critical difference. Green is used as positive feedback to indicate similarity between attacker and victim's timing components. In the case of M_{full}, a weight w (computed from the victim's absolute deviation vector) is added to the feedback as shown in Figure 4. w shows the relative importance of each component in calculating the attack score.

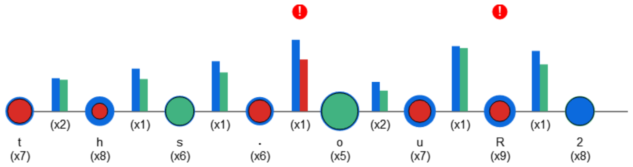
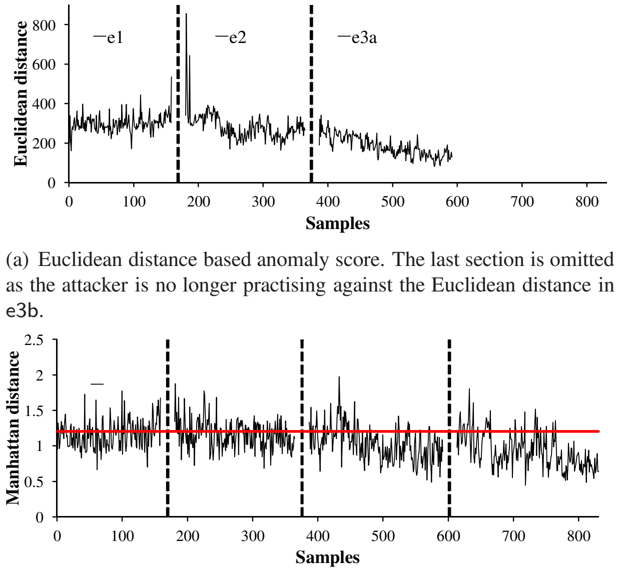


Figure 4. Mimesis interface (bottom section) based on Manhattan (scaled) distance

6. Evaluation

In this section, we present the results from our experiments. We start with the typing profile of a participant playing the role of an attacker as she progressed through each of the 4 experiments. Figure 5 shows the anomaly scores for both Euclidean and Manhattan (scaled) distances against a running index of her submitted timing vectors. *Vertical lines* separate the vectors in each experiment; *gaps* in the running indices are added around each separator line for clarity; *horizontal line* is the equal error rate threshold of the victim; *anomaly scores* below the horizontal line are



(a) Euclidean distance based anomaly score. The last section is omitted as the attacker is no longer practising against the Euclidean distance in e3b.

(b) Manhattan (scaled) distance based anomaly score

Figure 5. The anomaly scores of an attacker across all four experiments.

(falsely) accepted by the authentication system.

In e1, the attacker is required to type the assigned password 200 times. The Euclidean distance between each timing vector collected during this experiment and the victim's mean vector is plotted in the first section of Figure 5(a). Similarly, the Manhattan (scaled) distance is plotted in the first section of Figure 5(b). The data collected in e1 serves as a baseline because it is what an attacker can achieve without any imitation effort. Although we collected 200 vectors for each participant in this experiment, some vectors were discarded[6] and are not shown.

In e2, the attacker is provided with Euclidean distance based feedback using the M_{part} interface. We can observe that the attacker took only less than 100 tries to achieve her best results for this experiment. Note that the improvement is more obvious in the Euclidean distance as compared to the Manhattan (scaled) distance.

In e3a, the attacker was given 20 minutes to repeat e2. After a one week of time along with the prior experience of e2 (learning effect), we can observe that the attacker produced a noticeable improvement in her Euclidean distance based anomaly scores. The corresponding improvement in Manhattan (scaled) distance is less pronounced. This is due to the weak correlation between Euclidean distance and Manhattan (scaled) distance. The coefficient of correlation between Euclidean and Manhattan (scaled) distance for all vectors collected in e1, e2 and e3a is 0.543.

In e3b, the attacker was given (a) feedback based on the Manhattan (scaled) distance, (b) a performance bonus to improve on her previous results and (c) additional time of up to 2 hours. These conditions simulate a motivated attacker operating under optimum conditions. We can see from Figure 5(b) a noticeable improvement towards the end of the experiment.

In the remainder of this section, we present the rest of our experimental findings. In Section 6.1 we investigate the possibility of collision attacks in e1 where the attackers are not provided with any form of feedback. We also evaluate the quality of detector parameter estimation with few samples. In Sections 6.2, 6.3 and 6.4 we present the outcome of e2, e3a and e3b experiments respectively.

6.1. Interesting results from e1

In this experiment, we obtained the timing vectors from 84 attackers who were asked to type their corresponding victim's password 200 times without any feedback. The victim assignment was random. 37 attackers typed the simpler password 'serndele', while the remainder typed the harder password 'ths.ouR2'. From the submitted timing vectors, we evaluate (a) the likelihood of collision attacks and (b) the extent to which anomaly detector parameters can be estimated when few samples are available. The latter results provide the justification for the partial information scenario described in Section 3.2.

6.1.1 Collision attack

In Killourhy and Maxion's evaluation [15], the anomalous timing vectors of each user was constructed from the first 5 vectors submitted by all other users. This simulated attackers who are unfamiliar with typing their victim's password. They raised but did not answer the question of whether the FAR would change if attackers are allowed to practise typing the password. In e1 we attempt to answer this question. We compute the overall FAR based on *all* vectors submitted by each attacker (instead of just the first 5).

Figure 6 shows the overall FAR in e1. Most attackers have an overall FAR of 0.2 or less. However, there exists 1 attacker (the last bar) with an overall FAR of more than 0.8. This implies that even without any imitation training, she has at least an 80% chance of pretending to be the victim successfully. This is an example of a collision attack. In practice, given a target organization with 10 high value targets, if a team of 84 attackers were to be assembled, we expect to find on average, one attacker with the same typing pattern as one of the high value targets.

[6] Only e1 was conducted online. After it was concluded, we found that certain submitted vectors has near zero hold timing and inter-keystroke timing between the first and second characters because of network error. We filtered all such vectors from e1. The remaining experiments were all conducted in our lab and did not have the same issue.

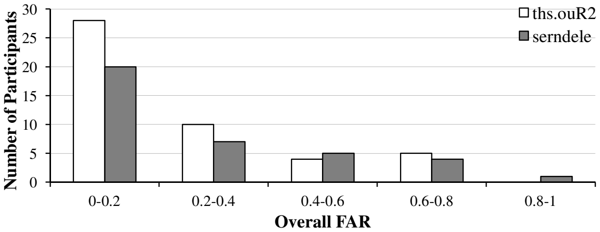


Figure 6. Overall FAR in e1

6.1.2 Estimation of anomaly detector parameters from few samples

We conducted a Monte Carlo simulation based on the timing vectors collected in e1. For each participant in e1, 10 samples were randomly picked from the collected data and the mean and absolute deviation were estimated based on these 10 samples. This is compared against the actual mean and absolute deviation computed using all available samples. We repeat this process 10,000 times for each participant. It was found that on average, the estimated mean is within ±10% of the actual mean 74% of the time. On the other hand, the estimated absolute deviation is only close to the actual absolute deviation 21% of the time. This shows that the scenario where an attacker can infer only the mean vector but not the absolute deviation vector is plausible. This provides the justification for the partial information scenario of e2 and e3a.

6.2. Imitation outcome of e2

In e2, each attacker is provided feedback based on the Euclidean distance assuming the partial information scenario of Section 3.2. 84 participants participated in this experiment. We present the change in FAR (see Section 6.2.1), followed by an analysis of how this change was affected by (a) choice of keyboard (see Section 6.2.2), (b) password difficulty (see Section 6.2.3), (c) attacker typing consistency (see Section 6.2.4). We also analyzed the optimum duration per training session in Section 6.2.5.

6.2.1 Improvement in FAR after imitation training

Figure 7 shows the FAR improvement in e2 as compared to e1. More than two-third of the attackers (56) improved their FAR from e1. However, there were some attackers (12) with no improvement in the FAR, while a small proportion (16) degraded. The last point demonstrates the shortcoming of using the Euclidean distance to approximate the Manhattan (scaled) distance in the partial information scenario. Intuitively, if the attacker decreases her differences in one component of her vector, but increases in another, whether the corresponding Manhattan (scaled) distance increases or decreases depends on the scaling ratio of the two components, which is not known to the attacker. The partial information scenario is plausible, but not ideal.

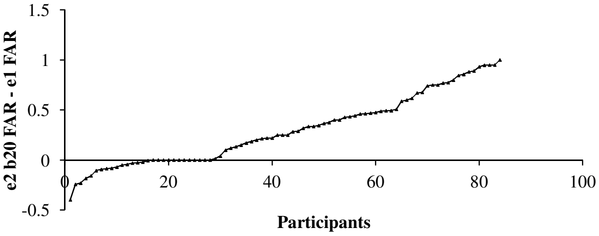


Figure 7. Improvement in FAR in e2 b20 **from** e1

6.2.2 Effect of keyboards

Prior to the experiments, we speculated that external keyboards, compared to notebook keyboards, facilitate the imitation. Feedback from attackers also supports this conjecture. To investigate this, in e2, 40 participants out of 84 participants used an external keyboard provided by us. The remainder used their own notebook keyboard. To verify our conjecture, we used a 2-sample Student's t test assuming unequal variance. The null hypothesis states that there are no differences between the mean of the b20 FAR for attackers using external keyboards, compared to those not using one. We use a two-tailed test as there are no conclusive evidence to support the use of a one-tailed test. The results are shown in Table 2. While there are differences, the p value of 0.227 is not significant enough to conclude that an external keyboard made any difference.

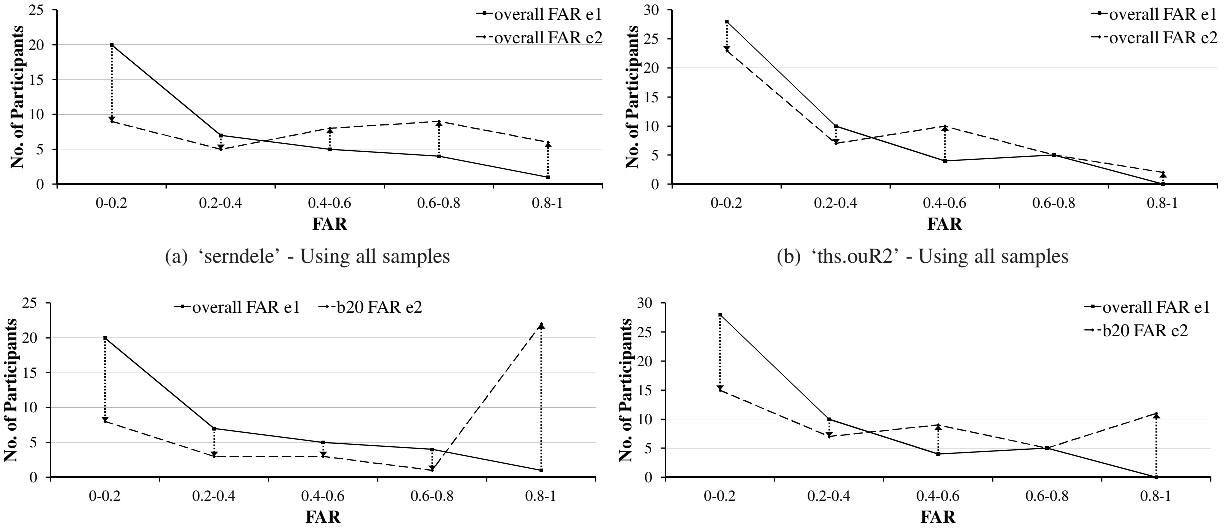
Groups	Mean	Variance	t Stat	P(T≤t)	t Critical
Own	0.564	0.146	1.217	0.227	1.989
External	0.462	0.148			

Table 2. Effect of keyboards

6.2.3 Effect of password difficulty

Figures 8(a) and 8(b) shows the change in overall FAR in e2 for different passwords. In e1, only 1 attacker practising on the easier password had a similar typing pattern as her victim ($0.8 \leq \text{FAR} \leq 1$). The training in e2 increased the number of such attackers to 6. For the harder password, no attacker is similar to her victim in e1. After e2, there were 2 such attackers. Statistical analysis using a 2-sample t-test with unequal variance showed that for the harder password, the change is not statistically significant (see Table 3(b)). In contrast, the change is highly significant for the easier password (see Table 3(a)).

In Section 3.4, we explained why using the overall FAR underestimates the effects of imitation. Therefore, we also



(a) ‘serndele’ - Using all samples

(b) ‘ths.ouR2’ - Using all samples

(c) ‘serndele’ - Using best consecutive 20 samples

(d) ‘ths.ouR2’ - Using best consecutive 20 samples

Figure 8. Improvement in FAR in e2 **from** e1

Groups	Mean	Variance	t Stat	P(T≤t)	t Critical
e1 overall	0.241	0.065	-3.586	< 0.001	1.993
e2 overall	0.471	0.085			

(a) ‘serndele’

Groups	Mean	Variance	t Stat	P(T≤t)	t Critical
e1 overall	0.196	0.050	-1.769	0.081	1.987
e2 overall	0.288	0.075			

(b) ‘ths.ouR2’

Table 3. t-test on overall FAR in e1 **and overall FAR in** e2

Groups	Mean	Variance	t Stat	P(T≤t)	t Critical
e1 overall	0.241	0.065	-5.126	< 0.001	1.998
e2 b20	0.633	0.150			

(a) ‘serndele’

Groups	Mean	Variance	t Stat	P(T≤t)	t Critical
e1 overall	0.196	0.050	-3.678	< 0.001	1.991
e2 b20	0.425	0.131			

(b) ‘ths.ouR2’

Table 4. t-test on overall FAR in e1 **and** b20 **FAR in** e2

evaluate the change in b20 FAR. These are plotted in Figure 8(c) and 8(d) respectively. As expected, the number of attackers similar to their victims show a marked increase to 22 and 11 (as opposed to 1 and 0) for the easier and harder passwords respectively. This suggests that even imitation with partial information (based on M_{part} interface) helps the attacker in her performance significantly.

Statistical analysis using a 2-sample t-test with unequal variance showed that for both passwords, the change is highly significant (see Tables 4(a)and 4(b)). The differences in mean between the easier and the harder password suggest that passwords that are easier to type are also easier to imitate. The implications raised by Section 3.5 is therefore confirmed: the effectiveness of keystroke biometrics in mitigating weak passwords is lesser than previously assumed.

6.2.4 Effect of attacker consistency

Intuitively, if an attacker is more consistent she should be able to exercise better control over her keystrokes, which should lead to better imitation outcome. To investigate the validity of this conjecture, we define a measure of consistency c. Referring to the vector notation of table 1, the standard deviation vector[7] ($\boldsymbol{s}$) can be computed using the following

$$\begin{aligned}\boldsymbol{s} = &\left(\frac{\sum_{i=1}^{n} \sqrt{(I^i_{k_1,k_2})^2 - (\bar{x}_1)^2}}{n-1}, \ldots, \frac{\sum_{i=1}^{n} \sqrt{(I^i_{k_{l-1},k_l})^2 - (\bar{x}_{l-1})^2}}{n-1}, \right. \\ &\left. \frac{\sum_{i=1}^{n} \sqrt{(H^i_{k_1})^2 - (\bar{x}_l)^2}}{n-1}, \ldots, \frac{\sum_{i=1}^{n} \sqrt{(H^i_{k_l})^2 - (\bar{x}_{2l-1})^2}}{n-1} \right) \\ =&(s_1, \ldots, s_{2l-1})\end{aligned}$$

For each component of $\boldsymbol{s}$, the larger its value, the larger the variability and therefore the lesser the consistency score. The consistency score c for each participant is defined as the

[7] Note the difference between the standard deviation vector $\boldsymbol{s}$ vs the absolute deviation vector $\boldsymbol{d}$ of the scaled manhattan detector.

inverse of the average deviation in s:

$$c = \frac{2l-1}{\left(\sum_{j=1}^{2l-1} s_j\right)}$$

The relation between imitation outcome and consistency is shown in Figure 9 which plots the b20 FAR against attacker consistency. Each point in the plot corresponds to one attacker. We can observe that there is no correlation between the imitation outcome (b20) of e2 against attacker's consistency score for both the easy password and the harder password. The coefficient of correlation for the easy password is 0.11 and for the harder password is -0.09. Therefore, there is no evidence to support our intuition that consistent attackers imitate better.

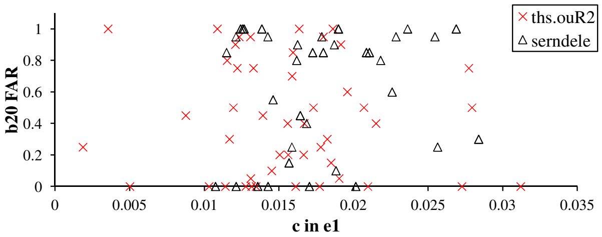


Figure 9. Imitation performance based on consistency in e1

A related and perhaps more interesting question is whether imitation training also improves attacker consistency. Figure 10 compares the consistency score of each attacker in e1 and e2. The attackers who are on the left of the vertical line were assigned the harder password 'ths.ouR2'. Those on the right were assigned the easier password 'serndele'. The attackers are sorted in a descending order according to their consistency score in e1.

Visual inspection of Figure 10 showed (a) there is no correlation between the c in e1 and e2, and (b) imitation training also improves the attacker's typing consistency regardless of the password complexity. (a) is confirmed by the coefficient of correlation, which has a near-zero value of 0.088. To verify (b), we used a 2-sample t-test with unequal variance. The results are shown in Table 5. The p value is less than 0.001 and confirms that the difference in consistency score c between e1 and e2 is highly significant.

Groups	Mean	Variance	t Stat	P(T≤t)	t Critical
e1	0.016	2.880E-05	-9.158	< 0.001	1.975
e2	0.025	5.005E-05			

Table 5. t-test on c in e1 and e2

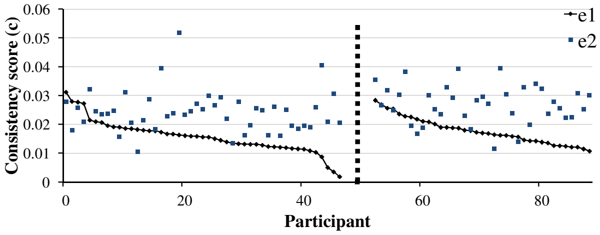


Figure 10. Consistency scores (c) in e1 and e2

6.2.5 Optimum duration per training session

In Figure 11, we show the time required for the attackers to reach their b20 performance in e2. 47 out of 84 (56%) attackers took less than 20 minutes. However, we also saw in Figure 5 that there is further room for improvement when given a second session. This suggests that instead of a single long session, imitation may be more effective when conducted in multiple sessions of shorter duration. A full investigation into the outcome for various combinations of session duration and number of sessions is however out of the scope of this paper and we leave it for future work.

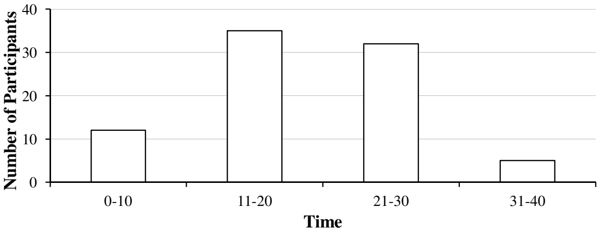


Figure 11. Time required in e2

6.3. Imitation outcome of e3a

After e2, the 14 best attackers (based on their imitation performance and consistency score) were selected and given a week's rest. They were then recalled for a repeat of e2. Based on the findings in e2 we limit the duration of e3a to 20 mins. The question we want to investigate in this section is that under the partial information scenario, do attackers reach their peak performance within the first 30 minutes or they are capable of further improvements when given more time to rest, reflect and repeat their earlier efforts.

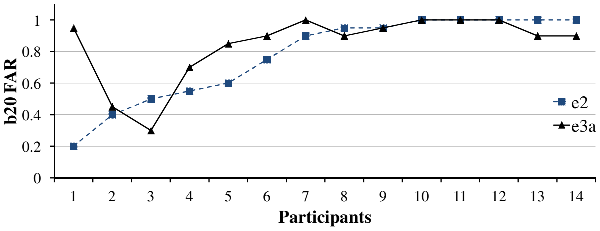
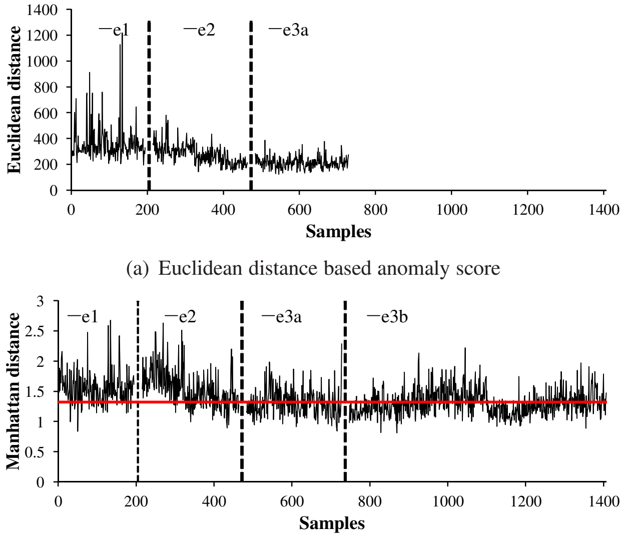


Figure 12. b20 FAR in e2 and e3a

Figure 12 shows the improvement in the b20 FAR between e2 and e3a. We found that there is no significant difference between b20 FAR obtained in e2 and e3a (see Table 6). Out of the 14 attackers, 6 improved their b20 FAR, 4 were unchanged while 4 actually worsened.

Groups	Mean	Variance	t Stat	P(T≤t)	t Critical
e2	0.771	0.073	-0.770	0.448	2.059
e3a	0.842	0.046			

Table 6. t-test on b20 FAR in e2 and e3a



(a) Euclidean distance based anomaly score

(b) Manhattan (scaled) distance based anomaly score

Figure 13. The anomaly scores of the worst performing attacker of e3a

To explain the anomaly where the performance of some participants actually degraded, we examine the profile of one such attacker with the worst e3a FAR (participant no. 3 in Figure 12). Figure 13 shows the typing profile of this participant where the anomaly scores are plotted against a running index of each timing vector. We can observe that the Euclidean score of this attacker actually improves in e3a. However, the improvement in Euclidean score did not translate into a marked improvement in Manhattan (scaled) score for this attacker. The reason is due to the weak correlation between these two distances.

6.4. Imitation outcome of e3b

Experiment e3b was conducted following e3a after a break of 15 minutes. This experiment simulates highly motivated attackers operating under optimum conditions (full victim information, performance bonus and more time). The attackers were told to try to achieve the 2 targets (see Section 4.4 for details). Out of the 14 attackers, 2 managed to achieve the lesser bonus and another 3 achieved the full bonus. (Note that qualifying for the bonus is more difficult than crossing their victim's acceptance threshold.) We can observe from Figure 14 that almost all attackers were able to achieve near perfect imitation of their victims. The results of a 2-sample t-test with unequal variance is shown in Table 7. The p value of 0.022 confirms that the difference in the FAR for e3a and e3b is statistically significant.

Groups	Mean	Variance	t Stat	P(T≤t)	t Critical
e3a	0.842	0.046	-2.594	0.022	2.160
e3b	0.992	3.29E-04			

Table 7. t-test on b20 FAR in e3a and e3b

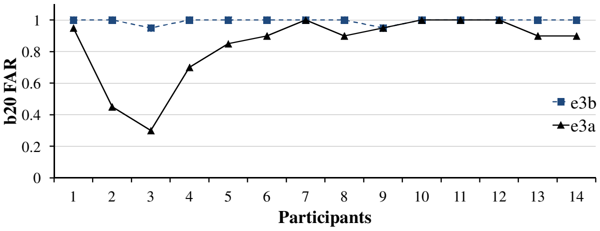


Figure 14. b20 FAR in e3a and e3b for participants of all four experiments

Figure 15 shows for one victim, her original FRR from e1, original FAR from e1 and the FAR of her 2 assigned attackers (*a1* and *a2*) from e3b. During the training phase, the threshold is set at a Manhattan (scaled) distance of 1.2, resulting in an EER of 0.2 for the detector. Imitation training markedly increases the FAR curve for both attackers. If the threshold remains unchanged, FAR increases to 1. This means the detector is unable to differentiate between the attackers and this victim. If the threshold is recalibrated to a distance of 0.75, it results in an unacceptable EER of 0.7.

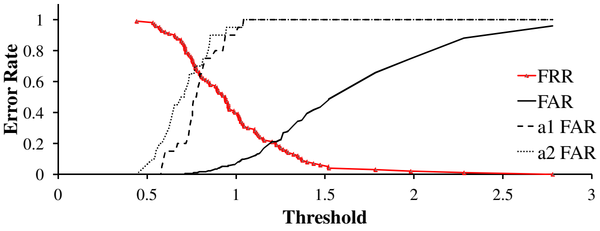


Figure 15. Effect of imitation training on the FRR and FARs

The amount of time taken by each attacker in e3b to reach their b20 performance is shown in Figure 16. For 9 out of 14 (64%) attackers, their performance peaked in 20 minutes or less. This is consistent with our observations in Section 6.2. Two highly motivated participants took nearly 2 hours. One of them achieved both performance bonuses at the end.

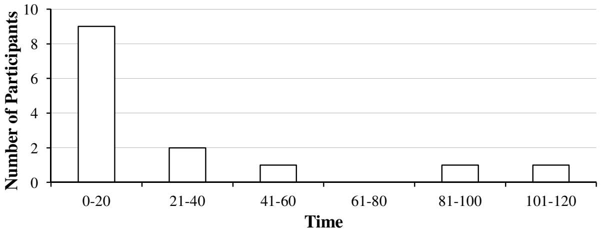


Figure 16. Time required in e3b

7. Discussion

In this section, we discuss the various factors affecting the outcome of the imitation, such as gender (see Section 7.1.1), typing speed (see Section 7.1.2), imitation strategy (see Section 7.1.3) and similarities in typing patterns (see Section 7.1.4). In Section 7.2, we discuss the attackers' interface preferences. Their perception towards the difficulty of hold timings or inter-keystroke timings is discussed in Section 7.3. Finally, we state the limitations in Section 7.4.

7.1. Factors affecting imitation outcome

Various hypotheses were put forward by both researchers and participants in the course of our experiments to account for the differences in imitation outcome. We evaluate each of them in the following discussions.

7.1.1 Gender

In the experiment e2, our participants pool consists of 39 males and 45 females. In Figure 17, we show the average e2 b20 FAR for both genders across the two passwords used. We can observe that male attackers achieved an average FAR of 0.51 and 0.81 for the harder and easier password respectively, compared to 0.33 and 0.51 for the female attackers. Male participants therefore perform significantly better than females in the imitation experiments. Furthermore, for both genders, the easier password to type is also the easier password to imitate, although the difference is more pronounced in males.

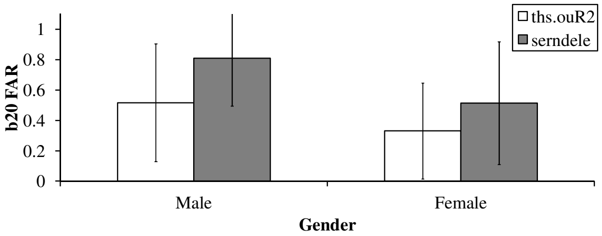


Figure 17. FAR based on gender

We confirm the differences using a 2-sample t-test assuming unequal variance. The null hypothesis states that there are no differences between the mean of the b20 FAR between male and female attackers. The results are summarized in Table 8, which shows that the null hypothesis can be rejected. Since the p value is less than 1%, the test is highly significant.

Groups	Mean	Variance	t Stat	P(T≤t)	t Critical
Female	0.420	0.133	-2.548	0.006	1.664
Male	0.629	0.149			

Table 8. t-test on b20 **FAR in** e2 **on gender**

7.1.2 Typing speed

During the experiment, feedback from certain attackers indicated that they believed slower victims are easier to imitate. To evaluate this, a measure of their typing latency is required. For each attacker and victim, we compute the average timing of all components in each participant's mean vector $\bar{\boldsymbol{x}}$ as a measure of their latency:

$$v = \frac{\sum_{j=1}^{2l-1} \bar{x}_j}{2l-1}$$

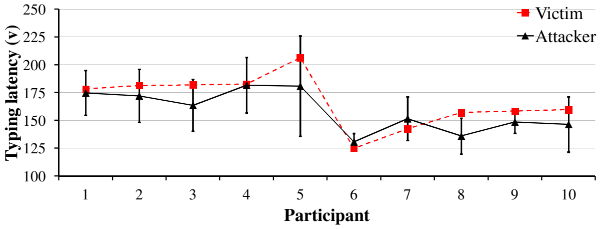


Figure 18. Typing latency of each victim and their attackers

Figure 18 shows the latency profile of all 10 chosen victims using the typing keystrokes from e1. Each victim was assigned 8 to 9 attackers. The attackers' latencies are shown on the same plot as vertical lines. The midpoint of each line indicates the mean attacker latency. The top and bottom of each line are one standard deviation away from the mean. The first 5 victims and their corresponding attackers were assigned the harder password 'ths.ouR2' and the last 5 practised on the easier password 'serndele'. The assigned attackers are generally spread out, with a mix of faster, equal and slower attackers in typing relative to the victims.

To investigate whether it is easier for a faster attacker to imitate a slow victim, the relative latency of each attacker w.r.t. her victim is computed. The coefficient of correlation between the b20 FAR in e2 and the relative latency is 0.02 and -0.12 for the harder and easier passwords respectively. Therefore, contrary to participant's feedback, there exists no correlation between the typing speed and the imitation outcome. This is shown in Figure 19.

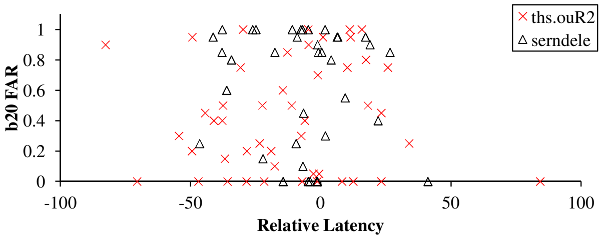


Figure 19. Relative latency v/s b20 FAR in e2

7.1.3 Number of trials per minute

During our experiments, we notice that participants vary greatly in the number of samples submitted. Different attackers adopt different strategy to get to their best high score. Certain participants would submit samples after samples, while others spend more time studying and reflecting on the feedback mechanism. For example some would pause and tap on their palm to grasp the rhythm. We want to know which approach is better and whether there is any effect on the FAR.

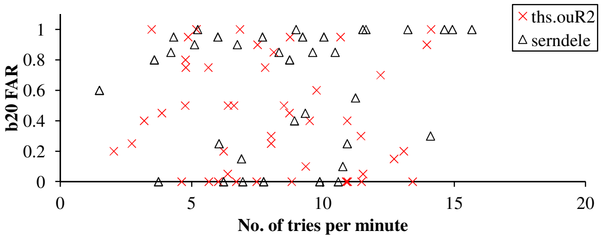


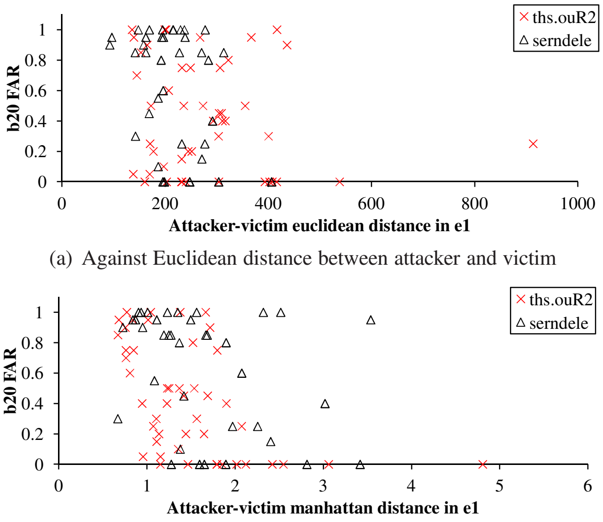
Figure 20. Tries per minute in e2

Figure 20 shows the b20 FAR in e2 against the number of tries per minute. The coefficient of correlation is 0.069. We found no correlation between how each attacker go about improving their imitation and the acceptance rate. This also suggests that there is no standard way to perform better in imitating someone else.

7.1.4 Initial typing similarity with the victim

If an attacker's typing pattern is already similar to the victim's before imitation training, intuitively, one would expect that even a slight improvement would result in a noticeable change in the FAR. In Figures 21(a), we show the b20 FAR in e2 of each attacker against the Euclidean distance between her mean vector and that of the victim's. In Figure 21(b), we show the b20 FAR in e2 of each attacker against the Manhattan (scaled) distance of the attacker's mean vector from the victim's mean vector (with the scaling based on the victim's deviation vector).

The coefficients of correlation are -0.26 and -0.38 for Figures 21(a) and 21(b) respectively. Therefore there exists a weak correlation between the e2 imitation outcome and the similarity between the attacker and victim's typing pattern. From Figure 7, we can observe that the correlation is weak because the extent of improvement varies for different attackers.



(a) Against Euclidean distance between attacker and victim

(b) Against Manhattan distance between attacker and victim

Figure 21. Correlation between b20 FAR in e2 and the typing similarity between attacker and victim

7.2. Mimesis interface

In Section 5, Mimesis provided feedback in both a table of numerical timings as well as a graph. Figure 22 shows the preference among attackers for these 2 feedback options. There are 51 participants who relied predominantly on the graph, while 23 relied on the raw data shown in the tables. 6 used both and another 4 used neither. For the latter, this implies that they relied only on the attack score and the coloring scheme adopted. Among participants who used the table and/or graph, feedback indicated that the reliance is only during the initial part of the experiment to get the rhythm correct. Thereafter, participants usually rely on their gut feeling to imitate.

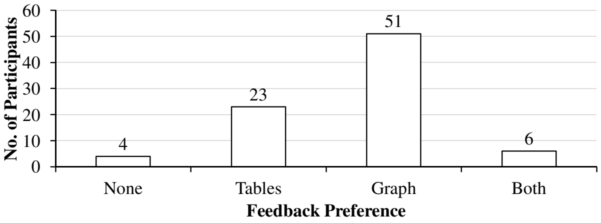


Figure 22. Preferences based on the feedback

7.3. Imitating hold v/s inter-keystroke timing

In this section we compare whether it is easier to imitate hold timing or inter-keystroke timing. From Figure 23, there are 42 attackers who find hold timing easy to imitate and inter-keystroke timing difficult. On the other hand, 30 attackers found hold timing difficult and inter-keystroke timing easy. 3 attackers found both are easy to imitate, while 9 think that both are equally difficult.

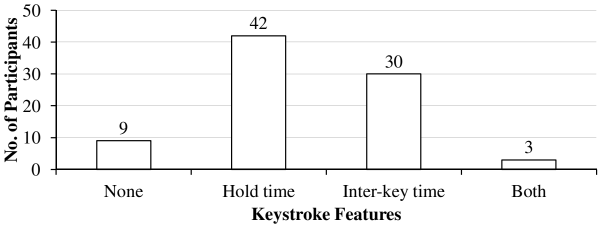


Figure 23. Easier to imitate

	Notebook keyboard	External keyboard
H easy, *I* difficult	19	23
H difficult, *I* easy	20	10

Table 9. Effect of the keyboard in hold timing and inter-keystroke timing

It turned out that the type of keyboard made a difference. Refer to Table 9. Of the 72 attackers who found one timing easy and the other difficult, 39 used their notebook's keyboard while 33 used an external keyboard. We found that among those using an external keyboard, the number of people who find hold timing easy to imitate is significantly more than those who do not. The reason could be because pressing and releasing of a key is more apparent in an external keyboard as compared to the notebook's keyboard.

7.4. Limitations

In Section 3, it was stated that we deliberately exclude the key up-down timing as a precaution that it may be difficult to understand and therefore affects the experimental results. A question therefore arises on whether it is really difficult to imitate key up-down timing. If true, it implies that a simple countermeasure against imitation would be to include and give a greater weight to key up-down timing in the anomaly score calculations. We did not address this issue in this paper and leave it as future work.

8. Related Work

Cho et al. [9] and Hwang et al. [19] explored the use of artificial rhythms and cues to improve the quality of typing samples. These include: (a) Breaking up a password into multiple segments and inserting pauses in between the segments when typing. (b) The use of 'tune, chant or rooting' to guide the password typing. (c) Minimizing the hold time in a "staccato" style [23]. (d) Maximizing the hold time in a "legato" style [22]. (e) Maximizing the inter-keystroke time in a "slow tempo" style. With such cues, users produced timing vectors that are more consistent and unique. We considered the use of audio feedback when designing the experiments, but found that in the context of imitation, it is hard to infer precise information just by listening.

The work by Rundhaug et al. [18] is similar to ours in terms of intent and general approach. They provided the feedback to a team of attackers in three ways: a simple accept/reject feedback, a distance score feedback and full feedback, where the attackers are able to examine each component of their timing vectors. For each attacker, three different passwords were used, each paired with a feedback mechanism. Each attacker goes through 3 imitation sessions, one for each password. For the full feedback, a graph with 3 lines was plotted. The first line plots the victim's mean vector plus 1 standard deviation, serving as an upper boundary. The second line plots the victims's mean vector minus 1 standard deviation, serving as the lower boundary. The third line plots the attacker's timing vector. Each attacker tries to modify their typing pattern to fit their timing line within the upper and lower boundaries of the victims. The author concluded that differences in anomaly scores before and after training is statistically significant and imitation is therefore possible. However, they also concluded that imitation is difficult and 'can indicate that keystroke dynamics is a very secure authentication method when combined with a password'.

9. Conclusions

This paper shows that contrary to the beliefs of prior studies, when a victim's typing pattern is known, imitation is possible. If the attacker has an incomplete model of the victim's typing pattern, her success rate is around 0.52 after imitation training. For the best attackers, imitation training increases the FAR to nearly 1, rendering keystroke biometrics based authentication systems unusable. Furthermore, when the number of attackers and victims are sizeable, the chance of a natural collision in typing pattern (without any imitation training) is significant.

Among the key factors affecting the imitation, we found that the easier the password, the easier the imitation. Males were also found to be better at imitation compared to females. On the other hand, various factors such as use of external keyboard, typing consistency, typing speed, imitation strategy and similarities in typing patterns were found to have much less influence on the imitation outcome.

References

[1] Admit one security. http://www.admitonesecurity.com/.

[2] Id control. http://www.idcontrol.net.

[3] imagicsoftware. http://www.imagicsoftware.com/.

[4] Intensity analytics. http://www.intensityanalytics.com/.

[5] keytrac. http://www.keytrac.de/.

[6] plurilock security solutions inc. http://www.plurilock.com/.

[7] L. Araujo, L. Sucupira, Jr., M. Lizarraga, L. Ling, and J. Yabu-Uti. User authentication through typing biometrics features. *Trans. Sig. Proc.*, 53(2):851–855, Feb. 2005.

[8] M. Boatwright and X. Luo. What do we know about biometrics authentication? In *Proceedings of the 4th annual conference on Information security curriculum development*, InfoSecCD '07, pages 31:1–31:5, New York, NY, USA, 2007. ACM.

[9] S. Cho, C. Han, D. H. Han, and H. il Kim. Web based keystroke dynamics identity verification using neural network. *Journal of Organizational Computing and Electronic Commerce*, 10:295–307, 2000.

[10] S. A. Cole. More than zero: Accounting for error in latent fingerprint identification. *Journal of Criminal Law and Criminology*, 95(3), 2005.

[11] B. Geller, J. Almog, P. Margot, and E. Springer. A chronological review of fingerprint forgery. *Journal of forensic sciences*, 44:963–968, 1999.

[12] S. Haider, A. Abbas, and A. Zaidi. A multi-technique approach for user identification through keystroke dynamics. In *IEEE International Conference on Systems, Man and Cybernetics*, SMC 2000, pages 1336–1341, 2000.

[13] R. Joyce and G. Gupta. Identity authentication based on keystroke latencies. *Commun. ACM*, 33(2):168–176, Feb. 1990.

[14] K. S. Killourhy. *A Scientific Understanding of Keystroke Dynamics*. Dissertation, Carnegie Mellon University, 2012.

[15] K. S. Killourhy and R. A. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *DSN*, pages 125–134, 2009.

[16] F. Monrose and A. D. Rubin. Keystroke dynamics as a biometric for authentication. *Future Gener. Comput. Syst.*, 16(4):351–359, Feb. 2000.

[17] A. Peacock, X. Ke, and M. Wilkerson. Typing patterns: A key to user identification. *IEEE Security and Privacy*, 2(5):40–47, Sept. 2004.

[18] F. E. N. Rundhaug. Keystroke dynamics can attackers learn someone's typing characteristics. Master's thesis, Gjovik University College, 2007.

[19] S. seob Hwang, H. joo Lee, and S. Cho. Improving authentication accuracy using artificial rhythms and cues for keystroke dynamics-based authentication. *Expert Systems with Applications*, 36(7):10649 – 10656, 2009.

[20] D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on ssh. In *Proceedings of the 10th conference on USENIX Security Symposium - Volume 10*, SSYM'01, pages 25–25, Berkeley, CA, USA, 2001. USENIX Association.

[21] D. Umphress and G. Williams. Identity verification through keyboard characteristics. *International Journal of Man-Machine Studies*, 23(3):263–273, Sept. 1985.

[22] Wikipedia. Legato — Wikipedia, the free encyclopedia, 2012. [Accessed 02-August-2012].

[23] Wikipedia. Staccato — Wikipedia, the free encyclopedia, 2012. [Accessed 02-August-2012].