Targeted Mimicry Attacks on Touch Input Based Implicit Authentication Schemes

Hassan Khan, Urs Hengartner, Daniel Vogel
Cheriton School of Computer Science, University of Waterloo, Canada
{h37khan, urs.hengartner, dvogel}@uwaterloo.ca

ABSTRACT

Touch input implicit authentication ("touch IA") employs behavioural biometrics like touch location and pressure to continuously and transparently authenticate smartphone users. We provide the first ever evaluation of targeted mimicry attacks on touch IA and show that it fails against shoulder surfing and offline training attacks. Based on experiments with three diverse touch IA schemes and 256 unique attacker-victim pairs, we show that shoulder surfing attacks have a bypass success rate of 84% with the majority of successful attackers observing the victim's behaviour for less than two minutes. Therefore, the accepted assumption that shoulder surfing attacks on touch IA are infeasible due to the hidden nature of some features is incorrect. For offline training attacks, we created an open-source training app for attackers to train on their victims' touch data. With this training, attackers achieved bypass success rates of 86%, even with only partial knowledge of the underlying features used by the IA scheme. Previous work failed to find these severe vulnerabilities due to its focus on random, non-targeted attacks. Our work demonstrates the importance of considering targeted mimicry attacks to evaluate the security of an implicit authentication scheme. Based on our results, we conclude that touch IA is unsuitable from a security standpoint.

1. INTRODUCTION

Implicit authentication (IA) employs behavioural biometrics to continuously and transparently recognize and validate the identity of smartphone users. Several biometrics have been proposed such as device usage behaviour [7, 31], gait behaviour [6, 19], keystroke behaviour [9, 13] and touch input behaviour ("touch IA") [4, 8, 11, 12, 14, 20, 21, 33, 34]. Touch IA schemes rely on the finger movement patterns that are generated when people use their device normally: no special gestures are required. Previous work suggests not only do they provide lower detection delay compared to other behavioural biometrics, but they are secure since studies suggest less than 5% of random mimicry attackers would be successful at bypassing them [4, 12, 14]. Consequently, these touch IA schemes have been called *"... the more natural, unobtrusive future of smartphone biometrics"* [24]. Researchers have proposed to use touch IA as a middle ground for users who do not configure any authentication mechanism due to usability issues [33] or as a second line of defense if primary authentication is compromised [14, 20, 33]. Touch IA is even under consideration as a candidate behavioural biometric in the active authentication project of DARPA [5].

The accuracy evaluation in existing touch IA proposals follows a conventional methodology for classifier evaluation. Touch data is collected from users and a classifier is trained for each user by employing a subset of their data as positive training samples and other users' data as negative samples. To calculate accuracy statistics, the remaining data from other users is used as synthetic attack data. Using random attackers who have no knowledge of their victims' behaviour fails to capture more realistic attack scenarios such as shoulder surfing and offline training where attackers have access to their victims' raw touch data. Only Bo et al. [4] and Frank et al. [14] acknowledge their evaluations omit these realistic attack scenarios. Both argue adversaries could not learn invisible features such as touch pressure and swipe acceleration only by shoulder surfing, and Frank et al. [14] rule out attacks using a victim's raw touch data because malware is needed to gather the data. These arguments have been considered *"sound without doubt"* by others [28] and to the best of our knowledge, these attacks have never been evaluated.

We argue that shoulder surfing and offline training are realistic for malicious insiders like friends, family, and colleagues – insiders that are recognized threats [23]. In a recent study on the security perceptions of IA, potential early adopters voiced their concern regarding shoulder surfing attacks [17]. Unlike random attackers, malicious insiders are able to observe their victims' behaviour, giving them with an advantage for shoulder surfing attacks. Moreover, an attacker may launch sophisticated mimicry attacks after gathering the victim's raw touch data by asking that victim to perform a task on the attacker's device. This eliminates the need for malware or sophisticated logging. Previous work provides no evidence that touch IA protects against these targeted mimicry attack scenarios.

We perform the first evaluation of targeted mimicry attacks on three diverse touch IA schemes: SilentSense [4], Touchalytics [14], and Li et al. [20]. These schemes have hundreds of citations and media coverage [24]. We collect

MobiSys'16, June 25-30, 2016, Singapore, Singapore
 ISBN 978-1-4503-4269-8/16/06...$15.00
DOI: http://dx.doi.org/10.1145/2906388.2906404

raw touch data from 55 users and multi-angle video of 9 users during the data collection step. We recruit 32 attackers and motivate them to launch shoulder surfing and offline training attacks on selected victims. For shoulder-surfing attacks, the attacker studies videos of the victim to observe their touch behaviour before launching a mimicry attack. For offline training attacks, we develop a feedback and training app, `Mimicker`, that analyses and visualizes a victim's raw touch data to train attackers.

Our results show it is surprisingly easy to bypass touch IA schemes. For 128 unique victim-attacker pairs, we observe a bypass success rate of 84% for shoulder-surfing attacks, much higher than the 5% rate using the random attacker model [4, 14, 20]. Furthermore, among successful attackers, 90% only shoulder surfed for two minutes or less and 70% successfully mimicked in their first attempt. Similarly, for 128 unique victim-attacker pairs, we record a bypass rate of 86% for offline training attacks using `Mimicker` with over 80% of successful attackers bypassing IA on their first attempt. Additional experiments show that for shoulder surfing attacks, an attacker has about 70% chance of success without any knowledge of the underlying features of the touch IA scheme. With offline training, the attacker does not need to know the exact IA scheme used by the victim because schemes have many overlapping features. An attacker can train using one scheme, and bypass other schemes with a 76% bypass success rate. These targeted mimicry attacks not only require fewer resources, but are significantly more devastating compared to a non-targeted generic attack [28]. Our findings provide strong evidence to reconsider the evaluation strategy of IA proposals and question the security of touch IA in general. Our main contributions are:

1. We provide the first ever evaluation of targeted mimicry attacks on touch IA schemes and show their susceptibility to these attacks.
2. We show the accepted assumption that shoulder surfing attacks on touch IA are infeasible due to the hidden nature of some features is incorrect.
3. We outline a method and provide the necessary apparatus for malicious insiders to perform offline training attacks without installing a malicious app on their victims' devices.
4. We release shoulder surfing videos, training models, and our open source `Mimicker` Android app for researchers to replicate our experiments and extend our methodology to other IA domains[1].

2. THREAT MODEL AND ATTACKS

We use the standard threat model used for touch IA schemes [4, 20, 33]. An adversary attempts to gain unauthorized access to a victim's device, which employs a touch IA scheme to continuously authenticate the device user. The victim has either not configured a primary authentication scheme (such as a PIN) or the adversary has bypassed it completely through known mechanisms like shoulder surfing or smudge attacks [2]. Furthermore, the adversary is aware of the presence of a touch IA scheme on the victim's device.

Accuracy numbers reported in the IA literature evaluate this threat model against a random attacker model. This means that data from random attackers with no knowledge of their victims' behaviour is used as synthetic attack data. While this tests scenarios where attackers have possession of a stranger's device, it does not cover attacks by malicious insiders seeking to mimic their victim's behaviour. Smartphone users are concerned about insider threats from friends, family members, and colleagues [23]. IA evaluations should also consider this threat. We evaluate two malicious insider mimicry attacks: shoulder surfing and offline training.

Shoulder surfing attacks: Malicious insiders may observe their victims' interactions. It is impractical for users to conceal all touch input behaviour by shielding the device screen or holding the device at extreme angles. While the "invisible nature of features" argument [4, 14] is true for a subset of features such as touch pressure, it may not hold for features such as touch location and swipe duration. Thus, adversaries may attempt to mimic these observable features. It is unclear whether mimicking observable features by shoulder surfing provides an advantage.

Offline training attacks: Malicious insiders can gain access to the raw touch data of their victims through various techniques. Since any foreground app is able to capture touch input events, malicious insiders can recommend an instrumented app from the official app store to their victims, which in turn collects and transmits raw touch data to the malicious insiders. Malicious insiders may ask their victims to visit a webpage where HTML5 `TouchEvents` are used to skim raw touch data. TapPrints uses similar methods to infer tap locations and corresponding keystrokes by sensing the accelerometer and gyroscope sensors in the background [22]. Finally, malicious insiders have the convenient option of obtaining the raw touch data of their victims by asking them to perform a task (e.g., read an article or view photos) on the insiders' device. This eliminates the need to install or access anything on the victims' devices. Once the insiders gain access to the raw touch data, they can use it to train and mimic their victims' behaviour.

3. BACKGROUND

Before we describe the three touch IA schemes evaluated in our work, we review metrics used in the IA literature when reporting accuracy statistics. A *true accept* (TA) is when an access attempt by a legitimate user is granted; a *false reject* (FR) is when an access attempt by a legitimate user is rejected by the IA scheme. A *true reject* (TR) is when an access attempt by an adversary is rejected; a *false accept* (FA) is when an access attempt by an adversary is granted by the IA scheme. *Equal Error Rate* (EER) is the operating point where the rate of false accepts equals the rate of false rejects.

With these metrics in mind, we now describe and justify the three IA schemes we evaluated.

3.1 Touchalytics [14]

Touchalytics extracts 31 features from the raw touch data of a swipe. These features capture a user's behaviour using four features for the swipe location, three features for the swipe direction, nine features for the velocity and acceleration of the swipe, six features for the duration and the length of the swipe, four features for the curvature of the swipe, three features for the orientation of the finger and the device, and the touch area and the touch pressure at the midpoint of the swipe. An evaluation of Touchalytics on a

[1]https://crysp.uwaterloo.ca/software/mimicker

dataset of 41 participants shows that with the SVM or the kNN classifier it provides an EER of 4% for a window of eight swipes. We chose Touchalytics because in addition to its low EER, to the best of our knowledge, it captures touch input behaviour using the most extensive feature set.

3.2 LXG [20]

LXG[2] derives following features from a swipe gesture: (1) coordinates of the first touch point; (2) touch area at the first touch point; (3) moving direction at the first touch point; (4) moving distance; (5) duration; (6) average moving direction; (7) average moving curvature; (8) average touch area; and (9) max-area portion. The authors also evaluate the tap gesture but they propose using it only as an auxiliary gesture due to its high EER. The authors evaluate LXG with the SVM classifier and they create separate training models for each of the four swipe directions. Their evaluation on a dataset of 75 participants indicates that LXG provides an EER of 8% with a window of eight swipes. We selected LXG because its small feature set complemented our selection of Touchalytics and enabled us to evaluate the impact of feature set size on the training effort of the attackers.

3.3 SilentSense [4]

SilentSense uses a combination of the touch input behaviour and the device's reaction to the touch input to create a model of touch behaviour. SilentSense uses four touch features: the touch pressure, the finger area, the duration, and the location of the swipe; and two device reaction features: the device vibration and the device rotation using the accelerometer and gyroscope sensors, respectively. For the scenarios where users are walking, the micro-movement patterns are adjusted using four walking features. However, since an attacker can choose to stay stationary during the attack, we omit the walking scenario and the corresponding features from our lab-based evaluations. This omission does not impose non-trivial restrictions on the attackers. An evaluation of SilentSense shows that with an SVM classifier on a dataset of 100 users, it achieves an EER of 1% with a window of three swipes. We chose SilentSense due to its low EER and its use of micro-movement features in addition to the touch features.

4. VICTIM DATA COLLECTION

We collected a dataset of raw touch data and video recordings of "victims." Raw touch data is used to train IA classifiers and train attackers for offline mimicry attacks. Video recordings are used for shoulder surfing mimicry attacks. Existing raw touch datasets [14, 33] do not contain accelerometer and gyroscope data, nor do they have accompanying video recordings. Note that we received approval from our university's IRB for all experiments involving human participants.

4.1 Data collection

We implemented two Android apps to collect raw touch data using the same tasks as Touchalytics [14]: a Wikipedia app collects up and down swipes while users read articles of their choice; and a "spot the differences" app collects left and right swipes while users navigate between two slightly different illustrations. Each participant used these apps on a LG Nexus 5 device while in our lab. No directions were given beyond explaining the basic tasks of reading articles and finding differences. Each participant interacted until at least 150 swipes in each direction were logged, the minimum number of training samples required by the IA schemes.

Logged data: For every touch interaction, we record-ed: time stamp in milliseconds; touch point x and y coordinates; touch pressure; area covered by the finger on the screen; finger orientation; screen orientation; rotation values from the gyroscope sensor across three axes; and acceleration values from the accelerometer sensor across three axes. Accelerometer and gyroscope data were collected in a separate thread up to 100Hz.

Videos for shoulder surfing attacks: We captured video of nine participants while they used the data collection apps. At least ten swipes in each direction were captured in two views, above the device and from the side. Each had an unobstructed view of the participant's finger on the touch screen. All videos were shot in 1080p format (1920x1080 pixels) with a frame rate of 29 FPS. The smartphone occupied 4-5% of the video frame. Given the open-ended task, the videos were between 23 and 44 seconds (avg 31 secs).

Data statistics: We recruited 55 participants (a subset of these also participated in the attacks experiment). On average, the participants took 26 minutes to submit data. In total, we logged about 35,000 swipes comprising over 1.1 million touch points, and over 2.5 million accelerometer and gyroscope sensor readings.

4.2 Parameter value selection

We fix two tunable parameters in our experiments, operating threshold and window size. Operating threshold defines the desired values for negatively correlated FA and FR entries. By increasing the operating threshold, FRs can be decreased at the cost of increased FAs (and vice versa). Theoretically, at lower false accept rates (FAR), it should be difficult to launch successful mimicry attacks. Therefore, we set FAR for an arguably higher false reject rate (FRR) of 20% with corresponding FARs of 0.4% for Touchalytics, 4% for LXG, and 0.2% for SilentSense (see §4.3). The effect of the operating threshold is further investigated in §8.3. Window size defines the number of swipes used to calculate a user's authentication score. Larger window sizes increase confidence against classification scores at the cost of increased detection delay. We set the window size to eight swipes since the IA schemes we evaluate provide reasonable accuracy at eight swipes or less (see § 3).

4.3 Evaluation baseline

To establish a baseline, we use our dataset to evaluate the three touch IA schemes against the random attacker model. We construct non-overlapping training and test sets for each user using negative instances from other users' data. Half of the data is used for training, and the remaining for testing. Figure 1 shows the ROC curves that plot true accept rate (TAR) against FAR using an SVM classifier. The ROC curves show an EER of 4% for Touchalytics, 9% for LXG, and 3% for SilentSense. These low EERs are very similar to the rates reported in the original papers and establish the efficacy of these schemes against the random attacker model.

[2]LXG are the initials of the last names of the authors.

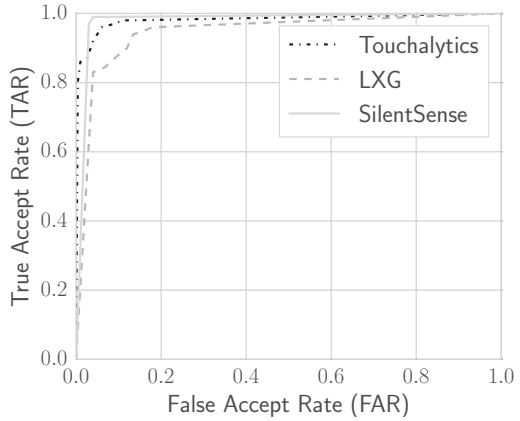


Figure 1: Accuracy of the IA schemes against the random attacker model with a window size of eight swipes.

5. ATTACK DESIGN

In this section, we describe apps and tasks used for offline training attacks and for attack evaluation.

5.1 `Mimicker` for offline training attacks

Our `Mimicker` app trains an attacker to mimic a victim's behaviour using feedback and visualizations generated from that victim's raw touch data. The three main components of `Mimicker` are the target swipe selection module, the training interface, and the feedback module. The target swipe selection module chooses an optimal *target swipe* from actual victim swipes. The training interface displays the target swipe so the attacker can swipe along a similar path (the *mimic swipe*). If the mimic swipe is rejected by the IA scheme, the feedback module displays instructions about a single behavioural aspect to bring the mimic swipe closer to the target swipe (e.g., move start point towards right). The attacker continues adjusting their swipe based on these instructions until their mimic swipe is accepted.

Target swipe selection: Any victim's swipe classified as a true accept can be used as a target swipe. However, our goal is to present the attackers with an optimal swipe to increase their chances of success. We do this by selecting a victim's true accepted swipe with the highest similarity score with the rest of the victim's true accepted swipes. This simple heuristic provides an advantage to the attackers since their mimicry attempts can focus on the hypothesis space with the maximum concentration of true accepted swipes (i.e., the victim's most typical swipes). The target swipe selection module accomplishes this as follows: It creates a dataset, D, with positive samples from the victim's swipes and negative samples from other users' swipes. It uses the kNN classifier ($k = 5$) to find a subset of true accept swipes, S_{TA}, from D. For $n = |S_{TA}|$, it identifies the n nearest neighbours from D for every swipe that belongs to S_{TA}. The similarity score is the number of swipes in the n nearest neighbours that belong to S_{TA}. The similarity score of each swipe is recorded in the similarity rank table and the swipe with the highest similarity score is returned as the target swipe. We also evaluate using non-optimal target swipes in §8.4.

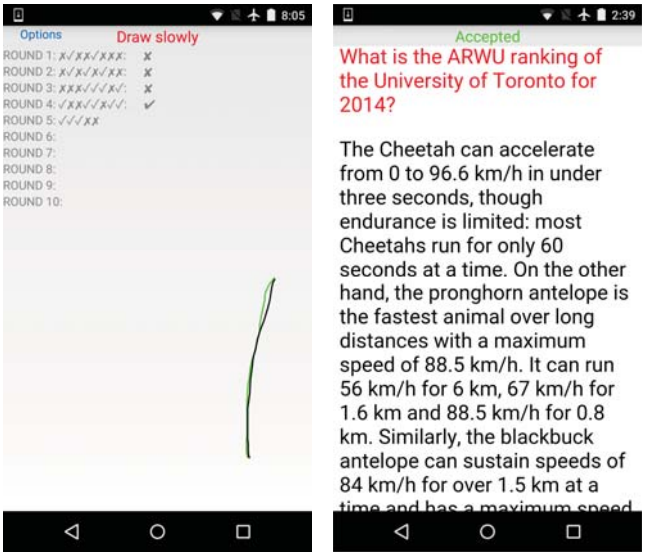
Training interface: Figure 2a provides a screenshot of the `Mimicker` user interface. The target swipe is displayed in green and the last mimic swipe is displayed in black. If the mimicry attempt is successful, "Accepted" is displayed at the top in green, otherwise the instruction from the feedback module is displayed in red. In the upper-left, the success or failure of recent attempts are displayed at swipe- and window-level. The success criteria for window-level are defined in §6.

Feedback module: The feedback module is responsible for comparing the mimic swipe with the target swipe to generate feedback for the attackers. The most obvious way to achieve this is to give feedback based on the classifier feature with the largest difference between target swipe and mimic swipe. However, this would not consider that some features are easier to mimic (e.g., first touch location is easier to mimic compared to midpoint velocity). To address this, the feedback module selects the first feature with absolute difference greater than a threshold from feature categories ordered by ease of adjustment: swipe location, swipe length, swipe duration, touch pressure, touch area, device rotation, device vibration, and swipe curvature. The feedback module focuses on features that are more adjustable for attackers and can be visualized or explained in simple instructions. Some features, like 50%-percentile pairwise velocity and median velocity at the last three points are hard to adjust for and difficult to comprehend. We acknowledge this means our bypass success rates form a lower bound by focusing on low-effort attackers.

5.2 Apparatus for attack evaluation

Personal data on smartphones can be broadly categorized as textual (e.g. emails, texts) and multimedia (e.g. images, videos). Our attack tasks reflect these categories. To further simulate a realistic mimicry attack scenario, the attacker has to multi-task by searching for interesting data while mimicking the victim. We introduce two tasks that capture the multi-tasking nature of real-world mimicry attacks. For attacks on textual data, the attacker is presented with a browser like interface with a collection of paragraphs from Wikipedia where each paragraph discusses a different topic. A question precedes the paragraphs and the attacker has to swipe up or down to find the paragraph that contains the answer to the question and then find the answer within that paragraph. For multimedia data, the attacker is provided with several feline images along with a numeric label for each image in an image viewer app. The attacker is then provided with a description of a feline (e.g., a white kitten) and is asked to swipe left or right to report the numeric label of the image that matches the description. While the attacker has to tap the target app icon to launch it, we do not consider the tap gesture since it provides too few features to be discriminative and has a high EER [14, 20].

At launch time, the apps are trained using the SVM classifier on the victim's training model constructed using positive samples from the victim and negative samples from other users. The apps provide no feedback for individual swipes; however, in case of a reject, the apps display a popup to inform the attackers of their failure (simulating the point when an explicit authentication method should appear in a deployed IA scheme). If the attackers are successful, the apps allow them to complete the task. Finally, the apps record the raw touch, accelerometer and gyroscope data of the attackers along with the result of their mimicry attempt.



(a) Tracing phase (b) Pseudo-attack phase

Figure 2: Screenshot of `Mimicker` interface.

6. ATTACK PROTOCOL

We now describe the protocol used to conduct attacks. The attack protocol was shaped by a pilot study with three volunteers from our lab. Relevant results from the pilot with subsequent changes to the final protocol are noted where applicable.

6.1 Participant recruitment and motivation

We recruited participants to be attackers in September of 2015 through a university-wide mailing list and using Craigslist and Kijiji under the "other jobs" section. The title of the advertisement was "Participate in a research study on mimicry attacks on a novel authentication scheme for smartphones" and it stated adults who owned and used a smartphone for over six months could participate. Each participant completed a demographic survey and was invited to our lab for the study.

In a real attack, malicious adversaries are motivated to snoop the devices of their victims to find valuable information. For our experiments, we motivate participants to mount best effort attacks with performance-based monetary rewards. All participants were paid $10, but they could earn another $6 based on performance. If they mounted a successful mimicry attack on the chosen victim in their first attempt, they received $0.75. If they mounted a successful attack on the second or subsequent attempt, they received $0.50.

6.2 Study procedure

The procedure began with each participant submitting raw touch data using the collection apps described in §4.1. This data forms a baseline to measure adjustments made by the participant during the attacks. The experimenter then briefed the participant (using a script and visual aids) with an explanation of touch IA, comprehensible features of their target IA schemes, the apparatus, tasks, and the performance based rewards. We investigate the scenarios where the attackers have limited knowledge of touch IA in §8.2.

Each participant mounted shoulder surfing and offline training attacks as explained below. Attack type order was counterbalanced across participants. For each attack type, the participant was assigned four victims to mimic up and left swipes. In the pilot study, each attacker was assigned only two victims and mimicked four swipe directions, but there were no significant differences in attack success. By reducing to up and left swipes, we could double the number of victims for each attacker. Testing more attacker-victim pairs is most relevant to our work, and up and left swipes are predominant directions for viewing new content. All eight victims assigned to an attacker were unique to avoid carry-over effects.

The four victims for each attack type were split into two groups: two were protected using either Touchalytics or LXG, and two were protected using SilentSense. We decided to assign either Touchalytics or LXG to make it easier for the attackers to remember their target IA schemes' features. The assignment of Touchalytics or LXG, and the order of the target IA schemes, were both counter balanced across the attackers. Attackers trained and mounted their attack on one victim swipe direction at a time using the assigned touch IA scheme and attack type. Swipe direction and the corresponding attack task order was counterbalanced.

Shoulder surfing attacks: The shoulder surfing attack had two parts: watching videos of the victim and attacking by mimicking the victim's swipes while completing the attack tasks (see §5.2). The attacker was shown the victim's shoulder surfing video clips on a 50" television. The attacker was not allowed to hold a device while watching the clips. They were informed about the camera angles and told they could watch the clips from either angle as many times as they wanted. Once the participant indicated they were prepared, the video was closed and they were given the device to mimic the victim's swipes while performing the attack task. Participants were told that if their attack failed, they could watch the clips again before mounting another attack. In §8.1, we evaluate a scenario where the attack occurs one week after shoulder surfing.

Offline training attacks: The offline training attack had two parts: training using the `Mimicker` app and attacking by mimicking the victim's swipes while completing an attack task. Training and attack were performed on two different LG Nexus 5 devices to simulate switching to a victim's device in a real attack.

Training was completed in two phases, a *tracing phase* (Figure 2a) with feedback and the target swipe and a *pseudo-attack phase* (Figure 2b) with only feedback overlaid on the attack task. The participant was informed that they had to bypass the IA scheme for two consecutive windows during each training phase before proceeding to the actual attack. If they were not successful, they had to continue the current training phase for at least ten windows before given another opportunity to bypass the IA scheme. During the attack, no feedback was provided. Attackers had to set the device down in between the tracing and pseudo-attack phases.

The pseudo-attack phase was introduced after the pilot study where 25% of the attacks failed despite successful completion of training. This appeared to be caused by the abrupt change between the tracing phase and the attack task: the attackers did not always memorize the location

Gender:	56% Male
	44% Female
Occupation:	31% Employed
	63% Grad student
	6% Undergrad student
Age group:	41% 18 - 25 years
	31% 26 - 30 years
	28% 31 - 35 years
IT experience:	53% Studied/worked in IT

Table 1: Demographics of the participants (n=32).

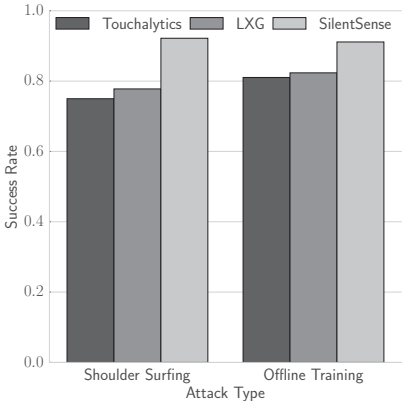


Figure 3: Bypass success rate for the three IA schemes across all attacker-victim pairs.

of the target swipe during training since `Mimicker` displayed the swipe; switching the device after training disrupted their device holding posture; and, unlike training, the attacks involved performing a task in addition to mimicking. The pseudo-attack phase increases training quality for attackers: we argue that a real attacker can leverage similar training mechanisms by approximating the task they plan to attack. For the purpose of our experiment, no feedback was provided for one window (*experiment window*) between the two phases to measure the efficacy of introducing the pseudo-attack phase.

7. EVALUATION

The study was completed by 32 participants (demographics provided in Table 1). In total, 512 attacks were logged (256 for each attack type), which were mounted by 256 unique attacker-victim pairs in up and left directions. We logged 3656 mimic swipes for shoulder surfing and 2984 mimic swipes for offline training attacks. During training, 17,064 swipes were logged.

7.1 Attacker success

We measure the efficacy of attacks on IA schemes through the bypass success rate at the victim-level and the TRR at the window-level. The bypass success rate is defined as the ratio of successful attacks to all attacks of a particular attack type or a particular direction.

Figure 3 shows bypass success rate against each IA scheme for both attack types across all attacker-victim pairs. For shoulder surfing attacks, 75%, 78%, and 92% of the attacks successfully bypassed against Touchalytics, LXG, and SilentSense, respectively. For offline training attacks, 81%, 82%, and 91% of the attacks successfully bypassed against Touchalytics, LXG, and SilentSense, respectively. An independent samples t-test for bypass success rates between shoulder surfing and offline training attacks for each scheme indicates no significant differences. We note that the bypass success rate may be over-reported due to naturally occurring false accepts; however, for the random attacker model, our chosen operating threshold has negligible FAR. Compared to the random attacker model (Figure 1) at a FRR of 20%, we observe about 20,000%, 2000%, and 45,000% increase in FAR for Touchalytics, LXG, and SilentSense, respectively.

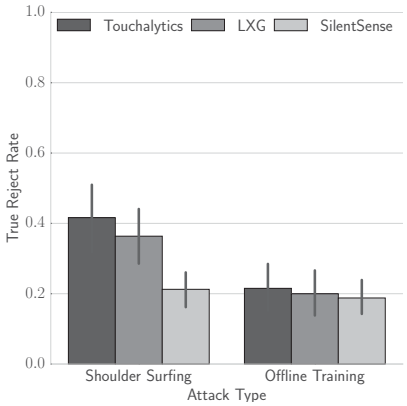


Figure 4: Average within window TRR for the three IA schemes across across all attacker-victim pairs. Error bars represent 95% confidence intervals.

To understand the performance of each scheme against mimicry attacks at the window-level, we calculate the average TRR across each attack window for all victim-attacker pairs for both attack types in Figure 4. Figure 4 shows that significantly lower TRRs are observed for offline training attacks for Touchalytics ($t = 2.12$, $p = 0.03$) and LXG ($t = 3.28$, $p = 0.001$). Lower TRRs are expected for offline training attacks because the attacks were mounted after the attackers received training. On the other hand, there are no significant differences between shoulder surfing and offline training attacks for SilentSense ($t = 0.56$, $p = 0.57$). Since the device reaction features of SilentSense rely on the device holding posture, the attackers were able to observe and mimic it during shoulder surfing. Consequently, the attackers performed better for SilentSense for shoulder surfing attacks when compared with the other schemes.

In terms of swipe direction, the bypass success rate for shoulder surfing attacks was 86% and 82% for up and left swipes. For offline training attacks, the bypass success rates were 85% and 87% for up and left swipes. An independent samples t-test indicates no significant differences between attack type for up ($t = -0.36$, $p = 0.71$) and left swipes ($t = 1.03$, $p = 0.3$).

Note that IA provides continuous authentication so attackers must continue mimicking a victim swipe after swipe. While our experiments require attackers to bypass their victim only once, after learning their victims' behaviour and bypassing IA for one window, they can mimic the behaviour on subsequent windows. This is shown in offline training,

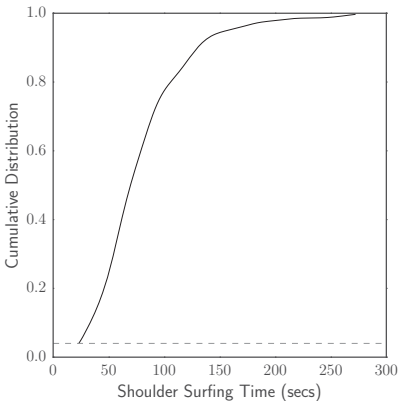


Figure 5: Shoulder surfing time for successful attacks.

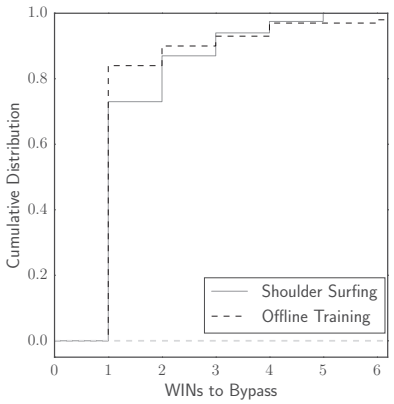


Figure 7: Windows until bypass for successful attacks.

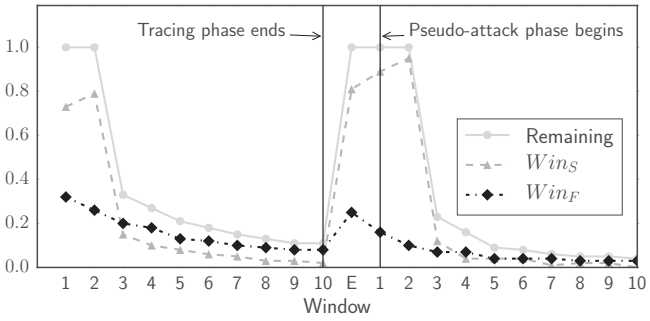


Figure 6: Proportion of attacker-victim pairs who required training for a window (Remaining) and their distribution across successful (Win_S) and failed (Win_F) windows. Window ('E') is the experimental window.

where attackers must bypass IA for three consecutive windows (two pseudo-attack windows and one attack window).

7.2 Attacker effort

To estimate attacker effort, we define three measures: shoulder surfing time (i.e., time spent viewing videos) captures attack preparation effort for shoulder shoulder surfing attacks; the number of windows used during training captures attack preparation effort for offline training attacks; and the number of windows until bypass (number of attempts required to mount a successful attack) captures attack execution effort for both types of attacks.

Shoulder surfing time: Figure 5 shows the cumulative distribution of time spent shoulder surfing before successful attacks. For 15% of successful attacks, the attackers were able to estimate the victims behaviour by observing them only for half a minute. Similarly, 40% and 90% of the successful attackers required less than a minute and less than two minutes of shoulder surfing time. These results indicate attackers require trivial shoulder surfing time for successful attacks.

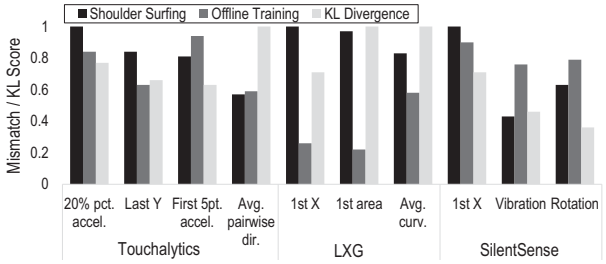
Number of training windows: Figure 6 shows the proportion of remaining attacker-victim pairs who still required training in a particular window and their distribution across successful (Win_S) and failed windows (Win_F). Recall that our protocol requires attackers to successfully mimic two consecutive windows in each phase. This means that 100% of all attackers will have at least two training windows in both phases in addition to the experiment window between the two phases. As an example for interpreting Figure 6, consider window 3 of the tracing phase. Here 33% of attacker-victim pairs remain because they were unsuccessful at mimicking the target swipe in both window 1 and window 2. Moreover 15% of attacker-victim pairs were successful at mimicking the target swipe while 19% failed in window 3.

Overall, 67% and 77% attacker-victim pairs only required two windows to complete the tracing and the pseudo-attack phase, respectively. A consistent decrease in the proportion of remaining attacker-victim pairs and Win_F from window 3 to window 7 in the tracing phase shows that `Mimicker` feedback was effective. However, after window 7, only 4% are able to successfully complete the training. Some attacker-victim pairs were unable to complete training after ten windows: 11% did not proceed past the tracing phase and 4% did not proceed past the pseudo-attack phase.

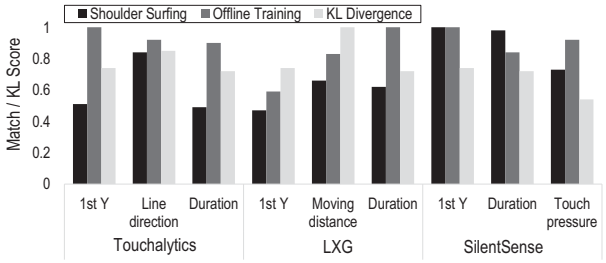
Regarding the efficacy of our pseudo-attack training phase, observe that the special experiment window ('E') with only the attack task and no feedback has a 25% increase in Win_F. This window simulates the same abrupt jump from training to attack, and the result is the same as the 25% attack failure rate in the pilot. However, attacker-victim pairs corrected their behaviour when they received feedback: Win_F drops to 16% in Window 1 then to 10% in Window 2.

Number of windows until bypass: Figure 7 shows the cumulative distribution of the number of windows until bypass. This captures the attack execution effort in terms of number of attempts to successfully mount each type of attack. Recall attackers were allowed to retry in case of a failed attack attempt and they could optionally shoulder surf or retrain before their next attempt. For the shoulder surfing attacks, about 73% of the successful attackers only required a single window and 93% required three windows or less to bypass the IA scheme. For offline training attacks, about 85% of the successful attackers bypassed the IA scheme in their first attempt and 90% required two attempts or less to gain access to the victim's device. There were no significant differences across the number of attempts to bypass for IA schemes or attack types.

Attacker effort and success: For shoulder surfing attacks, a Pearson correlation test indicates a slightly negative correlation between success rate of an attacker (the ratio of successful attacks among all attacks of a particular attack



(a) Difficult to mimic features



(b) Easy to mimic features

Figure 8: Difficult and easy to mimic features across failed and successful attempts, respectively.

type or direction executed by this attacker) and the amount of time they shoulder surfed *(r = -0.07)*. We also observe a negative correlation between the success rate and the number of retries *(r = -0.19)* for the shoulder surfing attacks. These results indicate that the failed attempts are not due to the lack of effort by the attackers. A similar analysis cannot be performed for offline training attacks since `Mimicker` requires the attackers to attempt training for at least ten windows before giving up and only the attackers who successfully complete training mount attacks.

7.3 Difficult or easy to mimic features

To measure how difficult or easy it was for the attackers to mimic IA schemes, we calculate mismatch and match scores for individual features for successful and failed attempts, respectively. We first calculate the absolute differences between individual features of the target and mimic swipes. The target swipe is explicit in offline attacks, but not when shoulder surfing. For shoulder surfing, the victims' swipe that is the nearest neighbour of the mimic swipe is selected as the target swipe. Mismatch and match scores are calculated by computing the normalized distribution of the occurrence of features in the top three most divergent and similar features, respectively. The Kullback-Leibler (KL) divergence [18] score for features is also calculated for the attacker-victim pairs on the raw input data from the device usage dataset. The KL divergence score provides a baseline for feature similarity and indicates the extent of adjustments made by the attackers for a feature. A higher KL divergence score indicates that the feature values are different across the participants.

Figure 8 shows features with high match or mismatch scores (features with low scores omitted due to space constraints). Features related to swipe curvature, velocity, and acceleration have higher KL divergence and mismatch score for both attack types (see Figure 8a). `Mimicker` does not provide feedback for acceleration or velocity related features since they are hard to comprehend, so we cannot fully conclude they are hard to mimic for offline training attacks. However, we can conclude these features are difficult to mimic for shoulder surfing attacks.

Figure 8b shows that swipe duration and touch pressure are easier to mimic. Some location based features (first and last touch coordinates) fall in both difficult and easy to mimic feature groups, especially for shoulder surfing attacks. We suspect that this is because some attackers are more critical observers than others.

Figure 8a shows that for almost all the features, the mismatch score is lower for offline training attacks. Since `Mimicker` provides feedback, attackers know how to adjust their behaviour to improve performance. This is quite pronounced for the location-related features such as '1st X', '1st Y', and 'Last Y' because `Mimicker` renders the target swipe.

There is an anomaly between LXG and SilentSense mismatch scores for the '1st X' location feature for offline training attacks. We suspect the higher mismatch score for SilentSense is due to its small feature set size and low KL divergence scores for vibration and rotation features. This indicates an overlap in behaviour across participants, increasing the chance that '1st X' is in the top three mismatched features.

8. DISCUSSION

In this section we discuss constrained attack scenarios and the effect of key parameters.

8.1 Basic shoulder surfing attacks

In our evaluation of shoulder surfing attacks, we assume attackers can record a video of their victims and watch it immediately before launching the attack. We were curious about the performance of a basic shoulder surfing attack using direct observation with hand written notes and a delay between observation and attack. To investigate this scenario, we repeated the shoulder surfing portion of the experiment with ten of the participant attackers, but inserted a one week delay between watching the videos and the attack. Each attacker viewed shoulder surfing videos of one victim they did not encounter in the main experiment. They were encouraged to make notes. One week later, the attacker returned to mount the attack on the victim. They were encouraged to consult their notes.

Table 2 compares mean bypass success rates across attackers for shoulder surfing attacks with delay and without delay (taken from their performance in the main experiment). Delayed attacks had a bypass success rate of 80%, 90%, and 90%, which an independent samples t-test did not find significantly different compared to attacks without delay. To gain insight into what attackers felt was important and what features they were mimicking, the notes were collected and their content categorized. Eight attackers drew the device screen with the corresponding location and curve of the swipe. Six of these attackers also noted the holding posture of the victims by drawing the holding hand and the location of the fingers of their victims. Two attackers wrote textual notes (such as: *"right bottom on the edge"*), seven noted the swipe speed (five noted it as slow/medium/fast; one marked a location on a continuous scale from slow to fast, and one wrote

Attack follow up	Success Rate		
	Touchalytics	LXG	SilentSense
Immediate	80%	90%	95%
Delayed	80%	80%	90%

Table 2: Effect of introducing one week delay between shoulder surfing and the attacks.

"one-mississippi"). Three attackers noted of the swiping finger or thumb of the victim. This small experiment suggests basic shoulder surfing is surprisingly effective, even when behavioural characteristics are mimicked at a coarse level.

8.2 Attacks with limited knowledge

In our evaluation, we assume the attacker has full knowledge regarding the victim's IA scheme. We were curious how performance is affected if the attacker has limited knowledge about the IA scheme or its features. In a scheme-oblivious offline attack scenario, the attacker has the victim's raw data and a training app like `Mimicker`, but they do not know the exact IA scheme used by the victim (e.g., whether it is Touchalytics or LXG). For the feature-oblivious shoulder surfing attack scenario, the attacker has no knowledge of any of its features. We evaluate the performance of attacks for these scenarios using the data from our main evaluation.

8.2.1 Scheme-oblivious offline attacks

We simulate this attack scenario by mounting attacks on a different IA scheme than the one used for offline training. For example, if an attacker trained and successfully attacked a victim using `Mimicker` for LXG, would the attacker have been successful if they trained for LXG but that victim actually used Touchalytics? We accomplish this by replaying attack swipes logged during successful attacks in the main experiment on the same victim protected with a different IA scheme. Note there is some inherent overlap in schemes since they share some features, or capture touch behaviour across limited dimensions like location, pressure, area and speed. However, the specific model built by the scheme classifier could still be very different.

Table 3 shows that after training on Touchalytics, bypass success rates increased by 3% when the victim was actually protected by LXG and decreased by 8% when they used SilentSense. After training for LXG, bypass success decreased by 6% for Touchalytics and 11% for SilentSense. Finally, after training SilentSense, bypass success decreased by 17% for Touchalytics and 14% for LXG. There appears to be a greater drop when attackers train on Touchalytics/LXG and attack SilentSense (and vice versa). This is due to less overlap between Touchalytics/LXG and SilentSense (more touch features in the former, more device reaction features in the latter). Overall, 70% or higher bypass success rate for scheme-oblivious attacks indicates attackers may not even need to know the exact scheme used by their victims.

8.2.2 Feature-oblivious shoulder surfing attacks

We evaluate the success rate for feature-oblivious shoulder surfing attacks with an experiment using ten participants from our data collection (they were not attackers in the main experiment). The main experiment shoulder surfing attack protocol was followed except attackers were provided no details about touch IA and were simply told they would attack a device protected using a scheme that employs the touch input and device holding behaviour.

Training Scheme	Scheme for Attacks		
	Touchalytics	LXG	SilentSense
Touchalytics	81%	84%	73%
LXG	76%	82%	71%
SilentSense	75%	78%	92%

Table 3: Bypass success rates for offline training attacks when attack attempts are replayed for different schemes.

	FRR			
	20%	**25%**	**30%**	**35%**
Shoulder surfing	84%	81%	78%	77%
Offline training	86%	85%	83%	83%

Table 4: Bypass success rates for the attackers for different operating thresholds.

The results show a bypass success rate of 60%, 50% and 80% for Touchalytics, LXG, and SilentSense, respectively. Unsuccessful attackers were briefed on the features after they ended their attempted attacks. These attackers viewed the videos again and mounted attacks on the victims they failed to mimic. For these retry attempts after a briefing, bypass success rates were 75%, 80% and 85% for Touchalytics, LXG and SilentSense, respectively. These results indicate that attackers have a 50% or higher chance of defeating these schemes without any knowledge of the underlying features, but the chances of success increase with feature knowledge.

8.3 Effect of operating threshold

In the main experiment, we chose a FRR of 20%, which created FAR of 0.4%, 4%, and 0.2% for Touchalytics, LXG and SilentSense, respectively. To understand the effect of different operating thresholds on mimicry attacks, we evaluate attacks efficacy at FRRs from 20% to 35% and corresponding FARs. Note that a FRR of 35% is impractical since the IA scheme would reject the device owner for a third of their swipes making it quite unusable. We perform an offline evaluation by replaying the attack data through the IA schemes with different operating thresholds and log the bypass success rates for each threshold. A limitation with this simulated evaluation is that offline attackers are likely to perform better if they trained at the tested FRR. However, these results do provide a lower bound for bypass success rate.

The results in Table 4 show only a 7% decrease in success rate when the FRR is increased from 20% to 35% for shoulder surfing attacks. For offline attacks, the decrease in success rate is only 3%. There is a sublinear decrease in bypass success rate for a linear increase in FRRs, which is similar to the sublinear increase in FAR for linear increase in FRR (see Figure 1). We believe the lower relative decrease in the bypass success rate for offline training attacks is a result of the low intra-window TRR (see Figure 4).

8.4 Effect of different target swipes

The `Mimicker` target swipe selection module selects the 'best' swipe from a set of 150 swipes of the victim (§5.1). We were curious how an attacker's bypass success rate was affected if they could not collect a large number of their

victim's swipes and had to train using a suboptimal target swipe. To evaluate this, ten volunteers from the main experiment participated in a smaller experiment spanning three ten-minute sessions of offline training attacks, with each session spaced one week apart (to mitigate carry over effects). Each attacker was assigned two victims; one victim was protected with LXG and the other with SilentSense. For each session, the attackers performed offline training attacks using the best, average, and worst swipes (the order was counter balanced across attackers). The best, average and worst swipes correspond to the top, middle and bottom locations of the similarity rank table, respectively (see §5.1). These well defined categories of swipes avoided a possible confound when randomly picking a target swipe.

The bypass success rates for the best, average and worst target swipes for LXG were 90%, 100% and 60%, respectively. For SilentSense, the bypass success rates were 100%, 100% and 80% for best, average, and worst swipes, respectively. A one way between subjects ANOVA score indicates a significant effect of target swipe on the intra-window TRR for the three target swipe types *($F(2, 27) = 9.18$, $p = 0.0009$)*. Post hoc comparisons using the Tukey HSD test indicated that the mean for the intra-window TRR for the best *($M = 0.05$, $SD = 0.07$)* and average *($M = 0.02$, $SD = 0.04$)* swipes were significantly different than the worst swipe *($M = 0.47$, $SD = 0.44$)*. However, the intra-window TRR for the best swipe did not significantly differ from the average swipe. These results indicate that while there is a reasonable chance ($\geq 70\%$) of success using any true accepted swipe as the target swipe, attackers can increase their chances of success by collecting more raw data to mine for an optimal target swipe.

8.5 Attacker- or victim-bound success?

Previous work has shown the performance of a generic attack against touch IA schemes is victim dependent [28], meaning some victims are easier to attack than others. We investigate if the same is true for mimicry attacks. A Kruskal-Wallis test comparing bypass success rates found no significant effect across victims, indicating no victims were easier to bypass than others. The same test was used to compare bypass success rates across attackers. No significant effect was found, indicating no attackers were better at mounting attacks than others.

9. LIMITATIONS

Like most human subjects studies, the scope is limited to people willing to participate. We discuss more specific limitations below.

We used similar devices for data collection and attacks. If an attacker collects a victim data using a device with a different form factor than the victim's device, a transformation would need to applied to the features.

We used the same tasks to collect victim data and evaluate attacker performance. This setup represents a resourceful adversary able to obtain a victim's data using an app similar to the target. Even if an adversary used apps with different input behaviour, the results will likely be similar (previous work reports 5-8% increase in EER due to significantly different apps [16]).

The participants who volunteered for the additional experiments (reported in §8) did not receive remuneration for their participation. Consequently, these participants may had less motivation, which may have affected their performance.

We omitted difficult to comprehend features from briefings and the feedback module of `Mimicker`. We note that dedicated attackers may have increased their chances of success by training on the omitted features. However, our experimental setup demonstrates the vulnerability of these schemes against relatively effortless attacks.

Our protocol trained an attacker to mimic their victim's swipes one direction at a time excluding multi-touch gestures like pinch to zoom [12]. This simplified the attack tasks and expedited attacker training.

It might be difficult for the attacker to mimic multi-touch gestures or multiple swipe directions simultaneously. However, swipe is the predominant form of gesture and an attacker can view content using one directional swipe for most target applications (email and messaging apps using up swipes while gallery app using left swipes). Due to their infrequent use, Touchalytics also ignores multi-touch gestures. While efficacy of mimicry attacks on multi-touch gestures is an open research question, we do not address it in this paper to conform to our realistic attack scenario. In terms of mimicry attacks on multiple swipe directions, given the trivial shoulder surfing and offline training time required, the attacker can leverage `Mimicker` or observational notes to train for different directions in real-time to mount these attacks.

10. RELATED WORK

Since the focus of our work is targeted mimicry attacks on touch IA, we do not discuss spoofing attacks on physiological biometrics (such as fingerprint and facial recognition). In this section, we discuss attacks on behaviour-based authentication systems.

Generative algorithms based attacks employ general population statistics and have been proposed for handwriting recognition, keystroke and touch IA. For handwriting recognition, Ballard et al. [3] show that a generative model based on concatenative synthesis exceeds the effectiveness of forgeries rendered by skilled humans. Serwadda and Phoha [27] analyze keystroke data from over 3000 PC users to observe statistical traits and then feed it to their generative algorithm to increase the EER of a keystroke classifier from 28% to 84%. More related to our work is a generative algorithm based attack on touch IA by Serwadda and Phoha [28]. They show that a robotic device equipped with generic traits across touch data poses a major threat to touch IA schemes as it increases their EER from 5% to 50%. While, unlike the targeted attacks discussed in this work, an attacker mounting their generic attack does not require shoulder surfing or the collection of the raw data of the victims, the attacker requires a mechanical robot to mount the attack, which may be impractical or suspicious in a work environment. Furthermore, the evaluations of the generic attack show that it failed for up to 40% of the victims because their touch behaviour was distinct. Targeted mimicry attacks can be leveraged to target any victim with more devastating effect.

Crowd sourcing based targeted mimicry attacks have been demonstrated for speaker and gait verification systems. Panjwani and Prakash [25] propose a method to crowdsource search for candidate mimics for speakers in a given target population. They show that while the probability of finding a successful match is only 3%, MTurk workers are easier and

cheaper to locate and recruit than mimicry artists. Gafurov et al. [15] used a database of 760 gait sequences from 100 subjects to show that while trained forgery attacks were unsuccessful for the gait biometric, closest matching subjects from the database could be used to increase the EER up to 80%. Given that an inter-user overlap exists across touch behaviour [28], similar crowd sourcing attacks might be possible on touch IA schemes. However, our evaluations show that the attackers can effortlessly adjust their behaviour to mimic their victims thereby eliminating the need to locate and recruit users with similar touch behaviour.

Shoulder surfing attacks have been evaluated for IA proposals that employ cognitive abilities [1] and eye movement patterns [10]. Shoulder surfing attacks have also been evaluated for explicit authentication schemes that employ user defined touch gestures [26, 29, 30]. These research endeavors indicate that shoulder surfing attacks are not a threat for their respective proposals. We suspect that user defined gesture based schemes are resilient because of the smaller intra-user divergence for a gesture. We evaluate touch IA and show its vulnerability to shoulder surfing attacks.

Finally, targeted mimicry attacks that train the attackers to mimic their victims have been demonstrated for handwriting and keystroke biometrics. For handwriting recognition, Ballard et al. [3] show that some users – who are better forgers than others – can be trained using a naive method to successfully attack the handwriting biometric. Tey et al. [32] use the keystroke data of the victim to train attackers to mimic two keystroke features on PCs. Their evaluations show that with the full knowledge of the keystroke patterns of the victims, 14 of their best attackers (out of 84 attackers) were able to achieve a 99% bypass success rate. While our offline training attacks are inspired by the work of Tey et al., we perform the first ever evaluation of offline training attacks on touch IA.

11. CONCLUSION & FUTURE WORK

We evaluate two simple attacks by malicious insiders that effectively circumvent touch IA; showing for the first time that it is unsuitable from the security standpoint. We show that the widely accepted assumption that shoulder surfing attacks on touch IA are infeasible due to the hidden nature of many touch input features is incorrect. We also demonstrate how dedicated attackers can use an app like `Mimicker` to train themselves to mimic victims offline with very high success. Moreover, mimicry attacks appear practical even if the attacker has limited knowledge of the victim's IA scheme or limited logged examples of the victim's touch behaviour.

As future work, an extended evaluation of offline training attacks considering all features (including difficult to comprehend features) is a logical next step for evaluating mimicry attack efficacy. A more hopeful focus for future research is the identification of a set of mimicry-resilient features that bolster IA security.

12. ACKNOWLEDGMENTS

Thanks to the anonymous reviewers and Jakob Eriksson for their valuable comments. We thank Google and NSERC for their support. We also thank members of the CrySP lab for submitting raw data and permitting us to make videos for shoulder surfing attacks.

13. REFERENCES

[1] A. Al Galib and R. Safavi-Naini. User authentication using human cognitive abilities. In 18th International Conference on Financial Cryptography and Data Security. Springer, 2015.

[2] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. In 4th Usenix Conference on Offensive Technologies. Usenix Association, 2010.

[3] L. Ballard, D. Lopresti, and F. Monrose. Forgery quality and its implications for behavioral biometric security. IEEE Transactions on Systems, Man, and Cybernetics, 37(5):1107–1118, 2007.

[4] C. Bo, L. Zhang, X.-Y. Li, Q. Huang, and Y. Wang. Silentsense: silent user identification via touch and movement behavioral biometrics. In 19th Annual International Conference on Mobile Computing & Networking. ACM, 2013.

[5] Board Agency Announcement. DARPA-BAA-13-16 Active Authentication. https://www.fbo.gov/utils/view?id=0a869cb811991d73b143bd5e050d1a4b, Mar. 2016.

[6] M. Boyle, A. Klausner, D. Starobinski, A. Trachtenberg, and H. Wu. Poster: Gait-based smartphone user identification. In 9th International Conference on Mobile Systems, Applications, and Services. ACM, 2011.

[7] S. Buthpitiya, A. K. Dey, and M. Griss. Soft authentication with low-cost signatures. In International Conference on Pervasive Computing and Communications. IEEE, 2014.

[8] 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 Annual Conference on Human Factors in Computing Systems. ACM, 2012.

[9] B. Draffin, J. Zhu, and J. Zhang. Keysens: Passive user authentication through micro-behavior modeling of soft keyboard interaction. In Mobile Computing, Applications, and Services. Springer, 2014.

[10] S. Eberz, K. B. Rasmussen, V. Lenders, and I. Martinovic. Preventing lunchtime attacks: Fighting insider threats with eye movement biometrics. In 22nd Annual Network & Distributed System Security Symposium, 2015.

[11] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen. Continuous mobile authentication using touchscreen gestures. In Symposium on Technologies for Homeland Security. IEEE, 2012.

[12] T. Feng, J. Yang, Z. Yan, E. M. Tapia, and W. Shi. TIPS: Context-aware implicit user identification using touch screen in uncontrolled environments. In 15th Workshop on Mobile Computing Systems and Applications. ACM, 2014.

[13] T. Feng, X. Zhao, B. Carbunar, and W. Shi. Continuous mobile authentication using virtual key typing biometrics. In 12th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, 2013.

[14] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of

touchscreen input as a behavioral biometric for continuous authentication. IEEE Transactions on Information Forensics and Security, 8(1):136–148, 2013.

[15] D. Gafurov, E. Snekkenes, and P. Bours. Spoof attacks on gait authentication system. IEEE Transactions on Information Forensics and Security, 2(3):491–502, 2007.

[16] H. Khan and U. Hengartner. Towards application-centric implicit authentication on smartphones. In 15th Workshop on Mobile Computing Systems and Applications. ACM, 2014.

[17] H. Khan, U. Hengartner, and D. Vogel. Usability and security perceptions of implicit authentication: Convenient, secure, sometimes annoying. In 11th Symposium on Usable Privacy and Security, 2015.

[18] S. Kullback and R. A. Leibler. On information and sufficiency. The Annals of Mathematical Statistics, 22(1):79–86, 1951.

[19] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Cell phone-based biometric identification. In 4th IEEE International Conference on Biometrics: Theory Applications and Systems. IEEE, 2010.

[20] L. Li, X. Zhao, and G. Xue. Unobservable reauthentication for smart phones. In 20th Annual Network & Distributed System Security Symposium, 2013.

[21] C.-C. Lin, C.-C. Chang, and D. Liang. A novel non-intrusive user authentication method based on touchscreen of smartphones. In International Symposium on Biometrics and Security Technologies. IEEE, 2013.

[22] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury. Tapprints: your finger taps have fingerprints. In 10th International Conference on Mobile Systems, Applications, and Services. ACM, 2012.

[23] I. Muslukhov, Y. Boshmaf, C. Kuo, J. Lester, and K. Beznosov. Know your enemy: the risk of unauthorized access in smartphones by insiders. In 15th International Conference on Human Computer Interaction with Mobile Devices and Services. ACM, 2013.

[24] New Scientist. Touchscreen phones know it's you from taps and swipes. http://www.newscientist.com/article/dn24193-touchscreen-phones-know-its-you-from-taps-and-swipes.html, Mar. 2016.

[25] S. Panjwani and A. Prakash. Crowdsourcing attacks on biometric systems. In 10th Symposium On Usable Privacy and Security. Usenix Association, 2014.

[26] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon. Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In Annual Conference on Human Factors in Computing Systems. ACM, 2012.

[27] A. Serwadda and V. V. Phoha. Examining a large keystroke biometrics dataset for statistical-attack openings. ACM Transactions on Information and System Security, 16(2):8, 2013.

[28] A. Serwadda and V. V. Phoha. When kids' toys breach mobile phone security. In 20th ACM Conference on Computer & Communications Security. ACM, 2013.

[29] M. Shahzad, A. X. Liu, and A. Samuel. Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it. In 19th Annual International Conference on Mobile Computing & Networking. ACM, 2013.

[30] 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 12th Annual International Conference on Mobile Systems, Applications, and Services. ACM, 2014.

[31] E. Shi, Y. Niu, M. Jakobsson, and R. Chow. Implicit authentication through learning user behavior. In Information Security. Springer, 2011.

[32] C. M. Tey, P. Gupta, and D. GAO. I can be you: Questioning the use of keystroke dynamics as biometrics. In 20th Annual Network & Distributed System Security Symposium, 2013.

[33] H. Xu, Y. Zhou, and M. R. Lyu. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In 10th Symposium On Usable Privacy and Security. Usenix Association, 2014.

[34] X. Zhao, T. Feng, and W. Shi. Continuous mobile authentication using a novel graphic touch gesture feature. In 6th International Conference on Biometrics: Theory, Applications and Systems. IEEE, 2013.