Mimicry Attacks on Smartphone Keystroke Authentication

HASSAN KHAN, School of Computer Science, University of Guelph
URS HENGARTNER and DANIEL VOGEL, Cheriton School of Computer Science, University of Waterloo

Keystroke behaviour-based authentication employs the unique typing behaviour of users to authenticate them. Recent such proposals for virtual keyboards on smartphones employ diverse temporal, contact, and spatial features to achieve over 95% accuracy. Consequently, they have been suggested as a second line of defense with text-based password authentication. We show that a state-of-the-art keystroke behaviour-based authentication scheme is highly vulnerable against mimicry attacks. While previous research used training interfaces to attack physical keyboards, we show that this approach has limited effectiveness against virtual keyboards. This is mainly due to the large number of diverse features that the attacker needs to mimic for virtual keyboards. We address this challenge by developing an augmented reality-based app that resides on the attacker's smartphone and leverages computer vision and keystroke data to provide real-time guidance during password entry on the victim's phone. In addition, we propose an audiovisual attack in which the attacker overlays transparent film printed with spatial pointers on the victim's device and uses audio cues to match the temporal behaviour of the victim. Both attacks require neither tampering or installing software on the victim's device nor specialized hardware. We conduct experiments with 30 users to mount over 400 mimicry attacks. We show that our methods enable an attacker to mimic keystroke behaviour on virtual keyboards with little effort. We also demonstrate the extensibility of our augmented reality-based technique by successfully mounting mimicry attacks on a swiping behaviour-based continuous authentication system.

CCS Concepts: • **Human-centered computing** → *Visualization systems and tools*; • **Security and privacy** → **Biometrics**; **Spoofing attacks**;

Additional Key Words and Phrases: Mimicry attacks, authentication, behavioural biometrics, spoofing attacks, augmented reality

ACM Reference format:
Hassan Khan, Urs Hengartner, and Daniel Vogel. 2020. Mimicry Attacks on Smartphone Keystroke Authentication. *ACM Trans. Priv. Secur.* 23, 1, Article 2 (February 2020), 34 pages.
https://doi.org/10.1145/3372420

1 INTRODUCTION

Text-based passwords are the most popular form of authentication and they are likely here to stay [Bonneau et al. 2012]. They are widely used on smartphones for user-to-device authentication

Part of this work appeared in ACM MobiSys'18 [Khan et al. 2018].
We gratefully acknowledge the support of NSERC for Grants No. RGPIN-2019-05120, No. RGPIN-2014-05499, and No. RGPIN-2018-05187.
Authors' addresses: H. Khan, Reynolds Building, School of Computer Science, University of Guelph, Guelph, Ontario, Canada; email: hassan.khan@uoguelph.ca; U. Hengartner and D. Vogel, Davis Center, Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada; emails: {urs.hengartner, dvogel}@uwaterloo.ca.

2471-2566/2020/02-ART2 $15.00
https://doi.org/10.1145/3372420

and to authenticate with banking, email, and social networking services [Egelman et al. 2014]. However, passwords are subject to shoulder surfing, weak passwords are vulnerable to guessing, and password database leaks are not uncommon [Alsaleh et al. 2012; Eiband et al. 2017]. Password hardening through keystroke behavioural cues ("keystroke dynamics") has been proposed as a second line of defense [Buschek et al. 2015; Monrose et al. 2002]. With password hardening, in case the password is compromised, the difference in typing behaviour of the attacker is expected to thwart the compromise.

Keystroke dynamics schemes for smartphones with physical keyboards have relied on key hold and inter-stroke intervals [Clarke and Furnell 2007; Maiorana et al. 2011]. Modern smartphones use virtual or software-based ("soft") keyboards, where users input data by tapping on a keyboard rendered on the touchscreen. This provides additional spatial (e.g., key press offset from the key centre) and contact features (e.g., touch pressure and area) [Buschek et al. 2015]. Several recent keystroke dynamics proposals that employ these features claim over 95% accuracy [Buschek et al. 2015; Draffin et al. 2014]. Commercial products that rely on keystroke dynamics are available for financial institutions [BehavioSec 2017].

The accuracy evaluation in existing keystroke dynamics proposals assumes an adversary with no knowledge of the victim's behaviour. Keystroke 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 the accuracy, the remaining data from other users is used as synthetic attack data. This results in a "zero-effort attacker model," where the attacker has no knowledge of their victims' behaviour and spends no effort to bypass keystroke dynamics. A determined adversary may obtain keystroke data of their victim through a seemingly benign app or social engineering (by asking the victim to perform a text entry task on the attacker's device). The adversary can then leverage this data to their advantage.

Tey et al. [2013] demonstrated the susceptibility of keystroke dynamics to such an attack on personal computers, where they developed an interface to train the attacker to mimic two features—the key hold and inter-stroke intervals—on a physical keyboard. Attacking keystroke dynamics for soft keyboards is significantly more challenging, since these schemes employ a larger feature set, with up to 22 more features per bigram, and a more diverse feature set that uses spatial and contact features in addition to temporal features. As a result, features that are not considered with physical keyboards (e.g., the tap location on the key) are important factors for soft keyboards. Another related challenge is training attackers for a larger and more diverse feature set on a small smartphone screen. Given these challenges, it is unclear whether keystroke dynamics on smartphones is susceptible to mimicry attacks.

We perform the first evaluation of mimicry attacks on a state-of-the-art keystroke dynamics scheme in the context of password hardening for soft keyboards. We analyze publicly available touchscreen data from 28 users as they type to develop insights at the feature level. Our analysis identifies features that should be targeted by an attacker. Like Tey et al.'s attack on physical keyboards, we develop interfaces to train attackers to mimic their victims. However, our experiments show that while moderately successful, this approach has limitations. More specifically, while attackers are able to reliably re-produce the victim's behaviour on the training interfaces, they often fail when mounting the attack on the victim's device. This is due to the large number of touchscreen behavioural cues an attacker is expected to learn and mimic. Tey et al. did not observe this effect in their experiment, because their attackers only trained to mimic the victim's behaviour and never tested on a separate interface (which was not required in their threat model focusing on user-to-website authentication on personal computers). However, this is a challenge for user-to-device authentication on smartphones.

We address this challenge by developing an augmented reality-based app for an attacker's smartphone to provide real-time guidance during password entry on a victim's phone. Under our proposed attack, when an attacker comes into the possession of the victim's device that is protected using keystroke dynamics, they position their smartphone camera to capture the victim's keyboard. The app analyzes the captured video stream, and overlays graphical cues in real-time (i.e., where to tap) to guide the attacker to enter the password like their victim. We also evaluate a simpler audiovisual guidance method in which spatial cues are printed on a transparent film to be overlayed on the victim's device, and audio cues played from the attacker's smartphone provide temporal cues (i.e., a beep indicates when to move to the next key). Neither method requires specialized hardware, tampering with the victim's device, or installing software on the victim's device.

We recruit 30 participants to evaluate over 400 mimicry attacks on keystroke dynamics using our training interfaces and real-time guidance methods. Our results show that for 90% of the attacks, the attackers used our training interfaces to successfully learn to mimic the victim's behaviour in 3–4 minutes. Our guidance methods further enabled 98% of the attackers who successfully completed their training to mount the attack on the victim's device in at most three attempts. For 74% of the successful attacks, the attackers were able to mount the attack in their first attempt. We also evaluate the logged attack data to show that attackers with faster typing speeds are more likely to mount a successful attack than attackers with slower typing speeds.

We conduct additional experiments to demonstrate the efficacy of our approach under constrained attack scenarios. These new findings were not part of our previous paper, which focused on the augmented reality system [Khan et al. 2018]: (i) Using new behavioural data collected with more complex passwords, we demonstrate that our approach can be used to effectively mimic keystroke behaviour against complex passwords; (ii) we show that an attacker can collect related bigrams from their victim through social engineering methods to reconstruct and mimic the keystroke behaviour of their victim's password, and we also explore cross-device behavioural differences with 15 participants on three different devices to understand the efficacy of the outlined social engineering methods; (iii) we show the efficacy of the proposed attacks when the operating threshold of the keystroke classifier is tuned to make such attacks more difficult to execute; and (iv) we show that the augmented reality system also enables attackers to successfully mimic a touch input behaviour-based biometric, Touchalytics, which relies on the swiping behaviour of the user [Frank et al. 2013]. Finally, we also provide the source code for the augmented reality system so researchers can replicate our methodology, and extend it to other behavioural biometrics.[1]

Overall, the main contributions of this work are:

(1) Novel attacks to bypass keystroke authentication on smartphones that do not require tampering of the victim's device.
(2) A demonstration of the susceptibility of keystroke authentication to these attacks against the password hardening scenario for two simple and four complex passwords.
(3) A demonstration of the threat posed by bigram splicing, where an attacker combines a victim's keystroke data gathered using social engineering means to defeat keystroke dynamics. A demonstration of our attack methods to target other touch input interaction-based behavioural biometrics (e.g., Touchalytics [Frank et al. 2013]).

2 BACKGROUND

In this section, we first define accuracy metrics that are used in the literature and statistical tests used in this work. We then provide an overview of keystroke dynamics on smartphones and the scheme that we evaluate in this work.

[1] https://crysp.uwaterloo.ca/software/AR-keystroke-attacks/.

2.1 Accuracy Metrics and Statistical Tests

A *true accept (TA)* is when a real-time usage pattern of a device owner is correctly classified. A *true reject (TR)* is when a real-time usage pattern of a non-owner is correctly classified. A *false accept (FA)* is when a real-time usage pattern of a non-owner is incorrectly classified. A *false reject (FR)* is when a real-time usage pattern of a device owner is incorrectly classified. The *false accept rate (FAR)* is the proportion of the adversary's attempts that are incorrectly classified as those from a legitimate user. The *true accept rate (TAR)* is the proportion of the legitimate user's attempts that are correctly classified. The *equal error rate (EER)* is the operating point where the rate of true accepts is equal to the rate of true rejects. The *accuracy* of a scheme is defined as the ratio of the sum of TA and TR outcomes to the total number of outcomes of an experiment.

For test statistics, we use a t-test when comparing between two conditions (e.g., guided vs. unguided attacks). We use a one-way ANOVA when comparing more than two conditions (e.g., three passwords). For comparisons between three conditions, multiple post hoc t-tests are used only if the related ANOVA test is significant. In all cases, a $p < 0.05$ critical value is used for statistical significance. For multiple comparisons of the same data category, we apply Bonferroni correction to p-values (and set the significance cut-off at α/n, where n is the number of multiple comparisons [Holm 1979]). We use a chi-square test when comparing two nominal variables.

2.2 Keystroke Dynamics on Smartphone

Keystroke dynamics has been widely studied for physical keyboards on personal computers (see Banerjee and Woodard [2012] and references therein). Features that are derived from physical keyboards include key hold interval and inter-stroke interval. The key hold interval is the interval between a key press event and the corresponding key release event. The inter-stroke interval is the interval between a key release event and the next key press event. Researchers have deployed these schemes on smartphones with physical keyboards to achieve varying levels of accuracy (between 75 and 90%) [Clarke and Furnell 2007; Hwang et al. 2009; Maiorana et al. 2011]. Note that the accuracy numbers provided in the literature are not directly comparable, since they are reported against different window sizes (where the window size is the number of keystrokes used to calculate the authentication score).

Modern smartphones use soft keyboards, a software-based keyboard rendered on the touch-screen. In addition to temporal features, researchers have employed contact features, like touch pressure and area, to improve the accuracy [Draffin et al. 2014; Feng et al. 2013]. Giuffrida et al. [2014] added features derived from accelerometer and gyroscope sensors, which capture the force of the key press and showed that these features were effective. In addition to temporal and contact features, Buschek et al. [2015] proposed spatial features for soft keyboards and showed that spatial features reduced EER by upto 23% when compared to temporal features. We discuss their proposal in detail in Section 2.3.

Another text input method on smartphones, swipe-based predictive typing or gesture keyboards, allow users to enter words by sliding a finger across the individual letters of the word (from the first letter to the last). Burgbacher and Hinrichs [2014] utilized spatio-temporal features from typing gestures to authenticate the device user. We do not evaluate gesture keyboards, since they require a dictionary with word frequencies and a model for word order, aspects that are not generally applicable to password entry.

2.3 Choosing a Keystroke Dynamics Scheme for Evaluation

Buschek et al. [2015] proposed a keystroke dynamics scheme, which employs the most extensive feature set. For a single bigram resulting from two keystrokes (K_1 and K_2), their scheme constructs

Table 1. Keystroke Features Proposed by Buschek et al. [2015]
K_1 and K_2 are the Two Keys of a Single Bigram

Category	Feature	Description
Temporal	key hold interval	Interval between key press and key release
	inter-stroke interval	Interval between K_1 release and K_2 press
	up-up	Interval between release of K_1 and K_2
	down-down	Interval between press of K_1 and K_2
Contact	down and up pressure	Touch pressure at key press and release
	down and up area	Touch area at key press and release
	down and up axis	Ellipses axis at key press and release
Spatial	down x and y	x and y coordinate at key press
	up x and y	x and y coordinate at key release
	offset x and y	Tap offset x and y from key centre
	jump x and y	x and y distance between K_1 and K_2
	drag x and y	x and y drag between key press and release (see Figure 1)
	jump angle	Angle between x-axis and K_1 and K_2 touch points
	drag angle	Angle between x-axis and key press and release points
	jump distance	Distance between K_1 and K_2 touch points
	drag distance	Distance between key press and release points

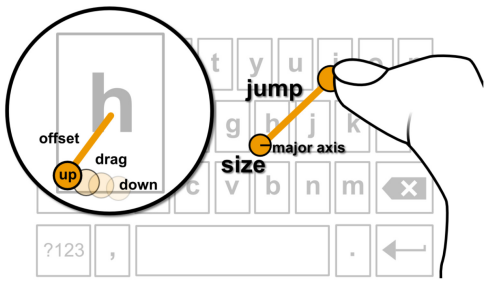


Fig. 1. A subset of keystroke features on a soft keyboard for the bigram "hi." (Illustration from Buschek et al. [2015].)

24 features listed in Table 1 (also see a subset illustrated in Figure 1). For their evaluations, they considered a scenario where keystroke dynamics is employed for password hardening. They collected data from 28 users across different sessions and different postures.

Their evaluation against a zero-effort attacker model using an SVM classifier achieved 3.3% EER. They also showed that spatial touch features outperform temporal features by further reducing EER between 14 and 23%. Furthermore, they showed that spatial touch features also outperform contact specific features. They compared their scheme with previous approaches for physical [Clarke and Furnell 2007] and soft keyboards [Draffin et al. 2014] and showed that their scheme provided better accuracy. Our choice of evaluating Buschek et al.'s scheme is due to its extensive and more effective feature set, and its accuracy advantage over other keystroke dynamics proposals. We describe these features in detail when we discuss our method to analyze them in Section 5.

3 THREAT MODEL

We use the same threat model as Buschek et al. An adversary attempts to gain unauthorized access to a victim's device, which employs keystroke dynamics for password hardening for user-to-device

authentication (i.e., a password-based primary authentication on the device). The adversary has shoulder surfed the password or obtained it through known mechanisms [Miluzzo et al. 2012], which only leaves the second line of defence (i.e., password hardening through keystroke dynamics). Furthermore, the adversary knows about the presence of a keystroke dynamics scheme on the victim's device and has obtained the typing behaviour of the victim using one of the mechanisms described later in this section.

We note the possibility of keystroke dynamics being used for continuous authentication of free text to monitor all keystrokes from the user. Previous research has shown that application specific differences reduce the accuracy of behaviour-based classifiers (e.g., composing a formal email versus texting friends) [Feng et al. 2014; Khan and Hengartner 2014]. This improves the chances of an attacker to mount a successful attack. However, continuous authentication requires mimicry over a longer period of time and not just during password entry. Therefore, while our proposed approach enables an attacker to effectively mimic the keystroke behaviour for a shorter period of time, it remains to be seen whether the attackers can use this setup against continuous authentication.

Behavioural biometrics are non-secret. Authentication methods that rely on secret knowledge, such as PINs or passwords, assume that the secret is only known to the legitimate user. Schneier [2009] classifies physiological and behavioural biometrics (including "typing patterns") as non-secret. Physiological biometrics like fingerprints are left on surfaces that users touch and behavioural biometrics like keystroke dynamics are submitted to websites that users visit. Attackers can steal or collect these biometrics and mount spoofing or mimicry attacks. The security of biometrics hence either requires that the biometric is not spoofable or that samples are submitted through a mechanism that provides some assurance of being "tightly bound" to the user present. For biometric installations at physical locations (e.g., a top secret research lab), a guard can make sure that the person authenticating is not using a photo or false finger made of glycerine. For physiological biometrics on smartphones, such as facial and fingerprint recognition, approaches like liveness detection are used to ensure a user's presence and defend against spoofing attacks due to their non-secret nature [Bao et al. 2009]. However, for behavioural biometrics (e.g., keystroke dynamics), only demonstrating their non-mimicable nature can mitigate the limitations that arise due to their non-secret nature. This non-mimicability is evaluated in this work.

Collecting victim's keystroke behaviour. Adversaries interested in mounting mimicry attacks may have different capabilities. As a result they may leverage different strategies to acquire their victim's keystroke behaviour.

First, we consider adversaries that have no control over which apps their victims install. Tey et al. [2013] argue that such adversaries may learn their victim's behaviour through compromised biometric databases; similar to the compromise of the Biostar 2 database that leaked one million fingerprints [The Register 2019]. Leakage of keystroke dynamics databases is a possibility for smartphones, too. Malicious insiders (i.e., friends, family members, and colleagues) can also direct their victims to an attacker controlled web page that collects keystroke data by rendering a keyboard that looks similar to that on the victim's smartphone. The recent HTML5 Touch API allows web developers to collect raw touch data from the browser, including x and y location, touch area, and touch pressure [W3C 2019]. While this technology is in the experimental stage, it is being adopted by different browsers, including Chrome for Android [Mozilla 2019]. By exploiting a compromised keystroke dynamics database or by collecting raw keystroke data using a standard HTML5 library and a rendered keyboard, adversaries do not have to tamper or install anything on their victim's device. Malicious insiders can also use other social engineering means to gain access to raw keystroke data of their victim. They can ask their victim to type carefully crafted text (e.g., take a note or type a URL) on the attacker's device. This crafted text contains the

same bigrams as the victim's password. This method also eliminates the need to install anything on victims' devices. In Section 8.2, we show the efficacy of extracting password entry behaviour from the same bigrams when input in other texts. If the attacker collects data on a device that is different than the victim's, then appropriate transformations need to be applied to the data before the mimicry attacks. We discuss this scenario in Section 8.4.

Second, we enumerate possibilities that may be adopted by an adversary who can control the apps that their victims install on their devices. We note that the following strategies require tampering with the victim's device by installing an app on the victim's device, and we enumerate these options only for completeness. On smartphones, an app can collect behavioural data generated within that app without requiring any special permissions (i.e., it is a non-secret). This allows opportunistic app developers to harvest the keystroke behaviour of their users at scale, which can then be obtained by the adversary. Malicious insiders may recommend an instrumented app from the official app store to their victims, which collects and transmits raw keystroke data generated in the app to the attacker. We note that the generic keystroke data collected using a malicious app may not contain the same bigrams as the victim's password. However, the collected data can be used to reconstruct the password entry behaviour of the victim using the method described by Negi et al. [2018]. Researchers have uncovered ways in which apps can mount phishing attacks to skim passwords [Bianchi et al. 2015], such approaches could be used to skim the password along with its entry behaviour. Once the insiders gain access to raw keystroke data, they can use it to train and mimic their victims' behaviour.

4 DATA COLLECTION

Buschek et al. made their dataset of raw keystrokes from 28 participants publicly available. We were interested in using it for our experiment; however, it was collected using a custom Android keyboard. We contacted the authors but were unable to obtain the keyboard layout used during data collection. Therefore, we use their dataset to analyze features and collect an independent dataset for evaluating our attacks.

We received approval from our university's IRB for all experiments involving human participants.

4.1 Data Collection Setup

Our data collection setup is similar to Buschek et al. We implemented an Android app and an Android keyboard to collect raw keystroke data. The app presented the password that the user had to enter along with their progress on the task. For each keystroke, we logged the key, the timestamp in milliseconds at the key press and release events, the x and y coordinates at both key press and key release events, the touch pressure and the touch area at both key press and key release events, the x and y offset from the key centre, and the ellipses axis (the length of the major axis of an ellipse that describes the size of the contact surface) at both key press and key release events. For touch pressure, touch area and ellipses axis features, we use the raw floating point value returned by the Android SDK.

Buschek et al. studied the impact of passwords of different strengths. We evaluated against the same passwords as Buschek et al. to validate our dataset's accuracy against theirs. For the main attack experiments, we use a subset from their passwords including: a 6-character dictionary password ("monkey"), an 8-character dictionary password ("password") and a 6-character complex password ("Igur39"). We evaluated against a subset to limit the study session length. In Section 8, we evaluate more complex and random passwords from their set including "Bedufo20," "12hsVi," and "s5mqde3A." We also asked users to enter the words "money" and "banker" to gather useful bigrams to evaluate the conditions discussed in Section 8. Users were asked to enter each password

Table 2. Baseline Under Zero-effort Attacker Model

	Our dataset		Buschek et al.'s	
	FAR	**TAR**	**FAR**	**TAR**
monkey	0.03	0.78	0.08	0.82
password	0.004	0.92	0.03	0.99
Igur39	0.002	0.95	0.01	0.98

We chose a lower FAR to make attacks more challenging.

20 times. If a user made a mistake (or corrected one) during the password entry, then they were prompted to repeat the entry.

4.2 Statistics of Collected Data

To collect data, we approached graduate students through word-of-mouth. The necessary condition for participation was that the subject owned an Android device for six months or more. For data collection, an LG Nexus 5 device was used in a lab setup. Subjects were asked to use the app in their natural posture (i.e., both thumbs or either thumb or either index finger to enter the passwords). Subjects were allowed to take breaks whenever they wanted, but they were not allowed to change the posture in between entry attempts.

We collected data from 18 subjects. Eight participants identified themselves as female and the rest identified themselves as male. Six subjects were between 21 and 30 years old and the rest were between 31 and 40 years old. All subjects entered passwords using both thumbs. We collected 304 keystrokes across 56 unique bigrams per subject ("Enter" and "Shift" keys were considered a part of bigram). In total, we collected 24,480 keystrokes. To study the effect of the victim's typing speed, we used k-means clustering to create two clusters of victims based on their typing speed. The cluster with faster typists had ten victims and an average typing speed of 290ms per character (sdev = 40ms), whereas the slower cluster had eight victims and an average speed of 521ms per character (sdev = 74ms).

4.3 Establishing a Baseline for Evaluation

We evaluated our dataset against the zero-effort attacker model to establish the accuracy of Buschek et al.'s approach. We constructed non-overlapping training and test sets for each user using negative instances from other users' data. Half of the data was used for training, and the remaining for testing. A critical parameter is the operating threshold, which defines the desired values for the correlated FAR and TAR. By increasing the operating threshold, FAR can be reduced at the cost of a lower TAR (and vice versa). Theoretically, at a lower FAR, it should be difficult to launch successful mimicry attacks. For our evaluations, we chose a low FAR (corresponding to a lower TAR).

Buschek et al. found that SVM and kNN provided the lowest EER within a single session. Therefore, we chose an SVM classifier with Radial Basis Function (RBF) kernel (parameter C at 1.0 and γ at 0.04 (1/number of features)) [Chang and Lin 2011]. If the binary outcome of at least 60% bigrams of the password was "accept," then the outcome of the password was treated as an "accept." Table 2 shows the results of the accuracy evaluation under the zero-effort attacker model against the chosen operating threshold. It also shows FAR and corresponding TAR for the password set used in this work on Buschek et al.'s dataset for the same conditions (i.e., typing with both thumbs). It shows that on our dataset, for the three passwords, we achieve an average FAR of 0.01 with a corresponding average TAR of 0.89. On both datasets, it shows relatively high error rates for "monkey" but FARs of 0.01 or lower and TARs of 0.92 or higher for the other two passwords. A contributing

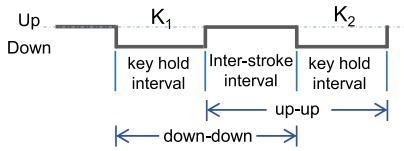


Fig. 2. Relationship among temporal features.

factor to the high error rates for "monkey" is our requirement that at least 60% bigrams of the password should be "accept" for the outcome of the password to be an "accept." "monkey," "password," and "Igur39" contain seven, nine, and ten bigrams, respectively. Due to round-off errors, this results in an uneven distribution of the number of acceptable false negatives at the bigram-level: two (out of seven), three (out of nine), and four (out of ten) bigrams for "monkey," "password," and "Igur39," respectively. In Section 8, we explore the effect of different operating thresholds on mimicry attacks.

5 ANALYZING KEYSTROKE BEHAVIOUR

Tey et al. [2013] mounted an attack on keystroke dynamics on personal computers by using an interface to train attackers to mimic only two temporal features (the key hold interval and inter-stroke interval) against each keystroke. However, training an attacker to mimic Buschek et al.'s [2015] 24 features against each keystroke on smartphones is a more challenging task. To reduce the training complexity, we analyzed the dataset from Buschek et al. to investigate keystroke dynamics at the feature level. We withheld our dataset from this analysis to ensure that our findings are evaluated on a non-overlapping dataset.

To meet our objective of reducing the training complexity, we performed statistical analysis to identify highly correlated features. We compute the correlation coefficient [Duda et al. 2012] between all pairs of features to identify correlated feature pairs and then reduce features by removing one of the correlated feature from the pair. The correlation coefficient is a value between −1 and +1, where a higher positive value indicates a strong positive correlation, a higher negative value indicates a strong negative correlation, and a value of zero indicates no linear correlation. Since we are interested in the absolute value of the correlation (i.e., both strong positive or negative correlations are of equal interest to us), we report the absolute values in Figure 3 to better distinguish between strong or weak correlation. We also use statistical techniques to identify the key specific or key independent nature of features. The identification of key specific or key independent nature of features helps to build cleaner interfaces for the one time training of key independent features. We now provide details on how we achieved this objective for the three feature categories.

5.1 Temporal Features

Figure 2 shows the relationship among temporal features. The "up-up" feature for K_2 comprises of the inter-stroke interval between K_1 and K_2 and the key hold interval of K_2. Similarly, the "down-down" feature for K_2 comprises of the inter-stroke interval between K_1 and K_2 and the key hold interval of K_1. To better quantify features that provide redundant information, in Figure 3, we plot the correlation coefficients against all pairs of the four temporal features. It shows a strong linear relationship (0.98 or higher) between all pairs of "down-down," "up-up," and the inter-stroke interval. For the key hold interval, it shows a weak correlation with "down-down" and "up-up." This suggests that both "up-up" and "down-down" features are redundant for training.

Next, we explore whether users' behaviour for the inter-stroke and key hold interval features varies between different keys. To this end, we use the coefficient of variation (C_V) metric, which is defined for a probability distribution as the ratio of the standard deviation (σ) to the mean (μ).

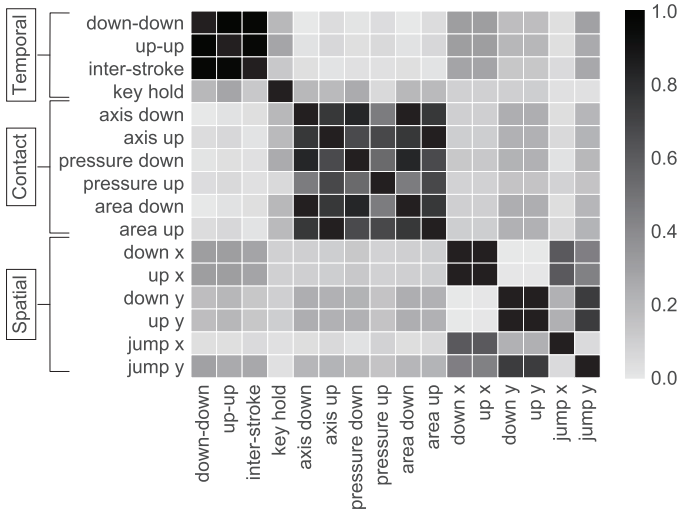


Fig. 3. Correlation matrix for selected features. Darker color indicates larger absolute correlation coefficient.

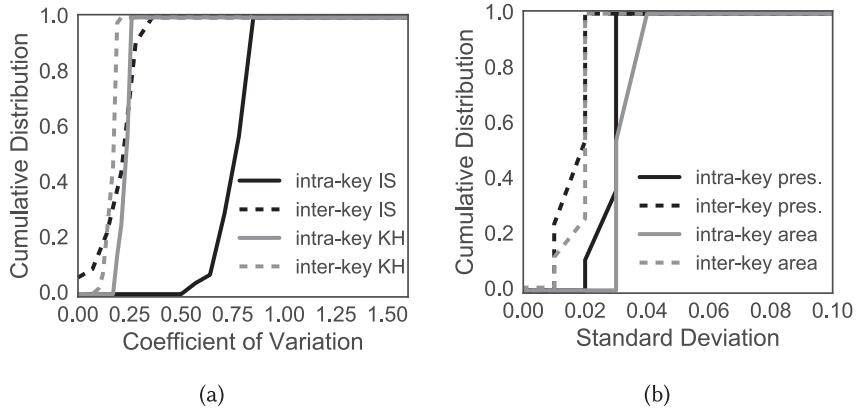


Fig. 4. Intra- and inter-key differences: (a) For inter-stroke ("IS") and key hold ("KH") intervals. (b) For touch pressure ("pres.") and area at key press.

Our choice of using C_V enables us to perform relative standard deviation comparison between two distributions with different spread. For each user, the *intra-key* C_V for a feature is computed over all values of that feature for each key. For contrast, we compute the *inter-key* C_V. For each user, the inter-key C_V for a feature is computed over all values of that feature for all keys. We then plot the cumulative distribution of the C_V for the feature for both intra- and inter-key differences across all users. A relatively low intra-key difference indicates that for a majority of users, the feature value does not vary significantly across different keys for individual users.

Figure 4(a) shows that the C_V difference between intra-key and inter-key inter-stroke intervals always exceeds 0.35. For over 90% of inter-stroke intervals, we note a C_V difference of 0.4 or higher between intra- and inter-key distributions. However, for over 90% of key hold intervals, the C_V difference between intra-key and inter-key never exceeds 0.1. This shows that while the inter-stroke interval changes across keys for a user, the key hold interval does not vary much for a user

across different keys. Consequently, an attacker likely needs to successfully mimic only one key hold interval for a victim.

5.2 Contact Features

We plot the correlation coefficients against all pairs of the six contact features. Figure 3 shows that a strong linear relationship exists between values at key press and key release events for both touch area and ellipses axis features ($\geq$0.75). For touch pressure, it shows a moderate relationship between key press and key release events ($\geq$0.5). Due to the moderate to high relationship between corresponding key press and key release events for these features, the attacker can choose to focus on an effective mimicry at either of the two events. We choose the key press event, since applying pressure or tweaking the touch area is more natural during this event.

Figure 3 also shows a correlation among touch pressure, touch area and ellipses axis. More specifically, for both key press and key release events, ellipses axis has a perfect linear relationship with the touch area. A strong linear relationship between touch pressure and touch area for the key down event also exists ($\geq$0.75). Similar moderate to strong linear relationships exist among touch pressure, touch area and ellipses axis for the key release event ($\geq$0.63). Due to the perfect linear relationship between ellipses axis and touch area, we choose to train for touch area. We choose touch area as it is easier to comprehend than ellipses axis.

Figure 4(b) shows the cumulative distribution of intra- and inter-key standard deviations between touch pressure and touch area for the key press event. It shows that the difference in standard deviation between intra- and inter-key scores never exceeds 0.01. It shows similar low difference ($\leq$0.02) in standard deviation between intra- and inter-key touch area. The low variation of these features across different keys for individual users allow us to train attackers to mimic one behaviour across different keys for these features for a victim.

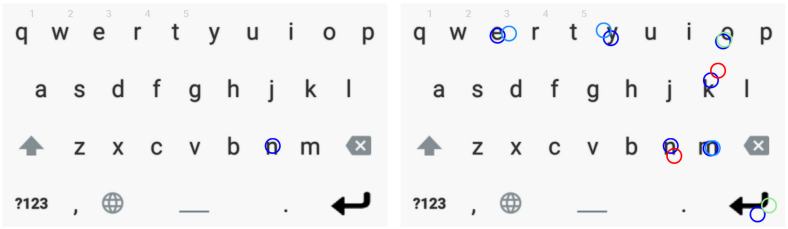
5.3 Spatial Features

Spatial features are inherently key specific due to the unique location of each key (with the exception of drag features), therefore we only investigate the correlation aspect. Table 1 shows that all spatial features are a function of four base spatial features: down x, down y, up x, and up y. There is a strong correlation between offset features and location coordinate features, because the former are a function of the latter and a constant value (the centre of the key). Similarly, drag features are derived from the difference between the down and up coordinates. Finally, jump features are derived from the difference between the the coordinates for K_1 and K_2. Therefore, an attacker only needs to successfully mimic the four base spatial features.

For the four base spatial features, Figure 3 shows a strong linear relationship between values at key press and key release events for both x and y coordinates (>0.99). Despite their high correlation, we need coordinate values at both events to compute drag features. However, Buschek et al. show that drag features are the worst performing features among their feature set. They show that under a single feature evaluation, drag features provide an average EER of 0.45 (sdev <0.01) (compared to an average EER for the remaining features of 0.31 (sdev = 0.02)). Therefore, we choose to only train for down x and down y features.

5.4 Discussion

Our feature analysis identifies six target features: the key hold interval, the inter-stroke interval, down pressure, down area, down x, and down y. Furthermore, key hold interval, down pressure, and down area features are key independent and the attacker needs to train for these features once for a victim and not for each key.



(a) Key specific cues. (b) Feedback.

Fig. 5. Cropped screenshot of keyboard during training for the password "monkey." (a) Target (on "n") moves across keys with same delay as inter-stroke interval. (b) All targets and attacker's attempts shown. Latter are colour coded to provide feedback on desired inter-stroke behaviour.

6 UNGUIDED ATTACKS

Similar to Tey et al., our goal was to provide sufficient training to attackers so that they could learn the typing behaviour of their victims. During attacks, they were expected to reproduce the typing behaviour from their learning experience (i.e., no guidance was available during attacks). We now provide details on the training interfaces that we developed.

6.1 Training Interfaces

We designed separate interfaces for training key independent and key specific features. For key independent features, the interface displays target feature values for the key hold interval, down pressure, and touch area features. Every time the attacker presses a key, it displays the attacker's feature values next to the target values. It also provides feedback to the attacker on how to adapt their behaviour (e.g., increase pressure). Once the attacker is able to mimic features such that the generated feature values fall in the inter-quartile range of the feature values of the victim, the attacker proceeds to the next interface.

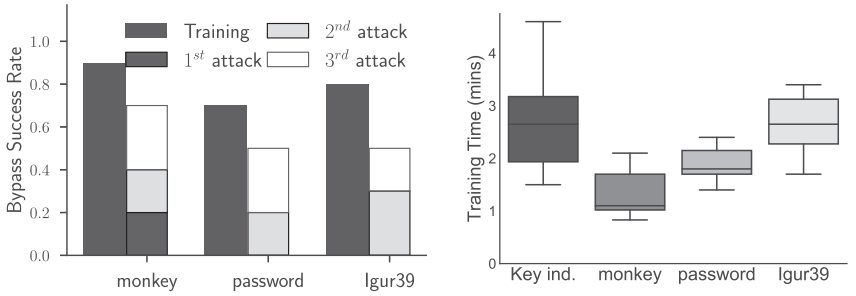
Figure 5(a) shows the interface used to provide cues for key specific features. The target location (down x and down y features) is provided as a blue coloured target. The inter-stroke timing is communicated by moving the target to the next key with the same delay as the inter-stroke interval. The attacker is expected to tap on the target as soon as it appears.

Figure 5(b) shows the feedback provided to the attacker. After a mimicry attempt, all targets and the attacker's attempts are plotted on the screen to show the attacker's accuracy for location features. The attacker's attempts are colour coded to provide feedback on the desired inter-stroke behaviour. Attempts coded red suggest increasing the interval between this key and the previous (i.e., go slow), whereas attempts coded green suggest decreasing the interval. Attempts coded light blue indicate that no change is required. In addition to the feature specific feedback, the attacker is also able to see the outcome of the SVM classifer (accept/reject) at the same operating point that is used during attacks. In case an attacker fails to mimic multiple key independent features, they are directed to the previous interface for retraining.

6.2 Attack Protocol and Study Procedure

We invited recruited participants to our lab and briefed them about keystroke dynamics, training interfaces and the attack protocol. We asked participants to type some text to determine their typing speed. This enabled us to study the effect of attackers' typing proficiency on successful mimicry (see Section 8 for results).

We asked attackers to undergo training on a LG Nexus 5 device. We asked them to remember the victim's behaviour so that they could reproduce it later. Once they successfully mimicked the



(a) Bypass success. (b) Training effort.

Fig. 6. Evaluation results for unguided attacks. (a) Bypass success. (b) Training effort on key independent and key specific interfaces for successful attacks.

victim for six times consecutively on the training interfaces and indicated that they were confident to attempt the attack, they were asked to mount the attack. We introduced a 30-second delay between training and attack, and during this period, we did not allow the attacker to hold the device. This delay was introduced to capture the real-world scenario, where the attacker performs training on their own device and then switches to the victim's device to mount the attack. For attacks, the attacker was asked to successfully mimic the victim's behaviour twice consecutively. In case the attacker failed to do so, they were provided the option to retrain before mounting the attack again.

Each participant was asked to mimic the three passwords (in order from weakest to strongest) from a single victim. Victims were counterbalanced across the faster and slower typist pool (Section 4). Participants were allowed to spend up to ten minutes to train for a given password. If they failed to complete training within ten minutes, then they were asked to proceed to the next password. They were given the option to retry the password that they failed to train later. This was done to ensure that the participants did not get overtired or bored for the remaining experiment. Participants were encouraged to take breaks between passwords.

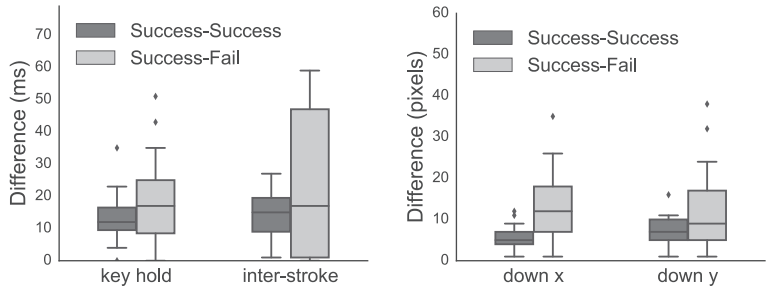
6.3 Participant Recruitment and Motivation

We used our university's graduate students mailing list to recruit participants for a 30 minutes session. We restricted participation to those who owned an Android device for six months or more. Ten subjects participated in this experiment: five subjects were between 21 and 30 years old and the rest were between 31 and 40 years old. Six participants identified themselves as female and the rest identified themselves as male. To motivate participants, we offered a performance-based reward in addition to $5 for participation. We offered $1 as a bonus for the successful mimicry of "monkey," and for "password" and "Igur39," we offered $2.

6.4 Results

We report on attackers' success and the amount of effort required to mount successful attacks. Figure 6(a) shows the overall bypass success rate of attackers during training and during attacks for the three passwords. Since attackers were allowed to retrain in case of a failure during attack, we show their success separately for different attempts. We note that the bypass success rate may be over-reported due to the non-zero FAR for the classifier. However, for the zero-effort attacker model, our chosen operating threshold has very low FAR (0.01).

Figure 6(a) shows that nine (out of ten) attackers used the training interface to mimic (six times consecutively) the password "monkey." Seven and eight attackers were also able to mimic the



(a) Temporal subset. (b) Spatial subset.

Fig. 7. Differences between feature values for last two successful training attempts ("Success-Success") and last successful training attempt and first failed attack attempt ("Success-Fail") for attackers who failed to bypass.

passwords "password" and "Igur39," respectively. However, the majority of successful attackers failed to mimic the passwords in their first attempt for the attacks. Only two attackers were able to correctly mimic the password "monkey" in their first attempt. We note that another two or three attackers are able to mimic these passwords in their second or third attempts. Finally, for all three passwords, two or three attackers were able to complete training but unable to mimic their victim in three attempts or less.

In Figure 6(b), we report the attacker's effort for successful attacks in terms of the amount of time spent on training (including retraining). It shows the attacker's training effort separately for the interface for key independent features (which requires only per victim training, not per password). It shows that the successful attackers spent an average of 2.8 minutes (sdev = 1) to learn to mimic key independent features. For the password "monkey," the successful attackers spent an average of 1.3 minutes (sdev = 0.4) on training. The successful attackers spent an average of 1.9 minutes (sdev = 0.3) and 2.6 minutes (sdev = 0.60) on training for the password "password" and "Igur39," respectively. We analyze the effect of password strength on the success rate of the attackers and the attacker's effort in Section 7.

6.5 Discussion

Our experiment shows that while the attackers were able to successfully mimic their victim during training for 23 out of 30 attack training attempts, only 2 of those 23 successful attempts were first attempts. Furthermore, in 7 of the 23 successful training attempts, the attacker then failed to mimic their victim during the attack. To investigate the high rate of failure during attacks, we examine the logged raw keystroke data during training and attack attempts. For each feature, we plot the difference between the attackers' mimicked values for the last successful training attempt and the first failed attack attempt ("Success-Fail") in Figure 7. For comparison, we also plot the difference between the attackers' mimicked values for the last two successful training attempts ("Success-Success").

Figure 7(a) shows the differences for the two temporal target features. For the inter-stroke interval, it shows an average difference of 15ms (sdev = 7ms) between Success-Success whereas the difference between Success-Fail is 24ms (sdev = 44ms). We also note that for Success-Fail, the difference is more dispersed for the third quartile than in Success-Success. This indicates that for half of the attacker's attempts (the third quartile), the difference in mimicked value is higher than Success-Success. It shows a similar trend for the key hold interval feature. Figure 7(b) shows

the difference for the two spatial target features. For the down x feature, it shows an average difference of 6 pixels (sdev = 3 pixels) and 14 pixels (sdev = 9 pixels) between Success-Success and Success-Fail, respectively. It shows more dispersion for the third quartile of Success-Fail than Success-Success. We observe a similar trend for down y and down area features (we omit quantitatively similar results for down area). While we do not observe this trend for down pressure, our analysis shows that performance becomes worse between training and attack attempts in the absence of guidance for all features except down pressure.

These results show that with guidance, attackers are able to perform better, and when the guidance is removed, the margin of error for attackers increases despite training. Tey et al. did not observe this effect in their experiment, because their attackers only trained to mimic the victim's behaviour and never tested on a separate interface (which was not required in their threat model focusing on user-to-website authentication on personal computers). Since attacking keystroke dynamics on a victim's smartphone requires mimicking of the behaviour on the victim's device, testing on a separate interface is crucial in our setup. Therefore, to enable the attacker to bypass the keystroke dynamics protection in the first few attempts, we suggest providing guidance to the attacker during attacks.

7 GUIDED ATTACKS

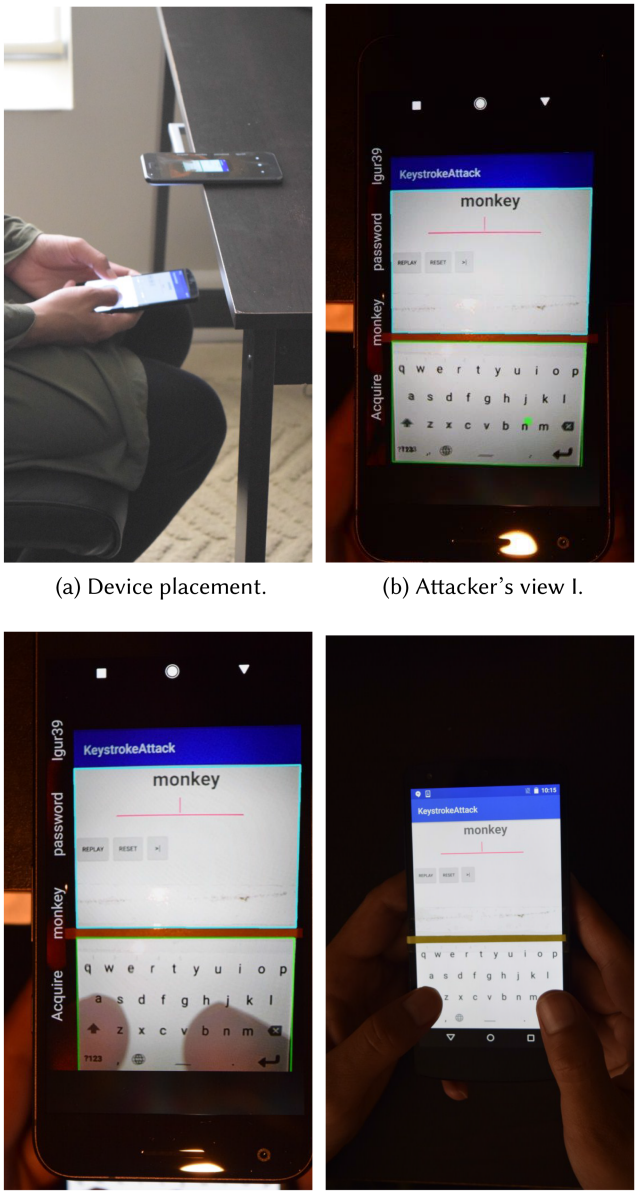
The results from the previous section emphasize the need to provide guidance to the attacker during the attacks. However, providing guidance on the victim's device is challenging, since the attacker has limited control over it. For instance, if the attacker is attacking the password used for user-to-device authentication, they cannot install or tamper any software on the victim's device. We propose two methods that circumvent this limitation to provide real-time guidance during the attacks. Both methods do not require any special equipment and assume only that the attacker has an off-the-shelf smartphone.

7.1 Augmented Reality-based Attack

We developed an Android app that runs on the attacker's device and superimposes mimicry guidance on the victim's device using mobile augmented reality in a controlled context. The app captures the keyboard of the victim's device through the rear camera of the attacker's smartphone, overlays the targets on the captured frame, and displays the frame on the screen of the attacker's phone. Figure 8 shows an attacker using our app to get real-time guidance during the attack. We assume the attacker can control environment lighting and the display brightness on the victim's device to ensure that the display is the brightest part of the frame. (Android allows brightness to be adjusted on locked phones.) Our app needs a stable, un-occluded shape to track the victim's phone. This is achieved by attaching a thin piece of paper to the victim's device just above the keyboard to divide the display into a bright top and bright bottom. The app further assumes the victim's device is entirely in the rear camera angle of the attacker's phone, and oriented in a known way so the bright top can be used for tracking.

Our app uses the OpenCV 2.4 library with Android KitKat native code wrappers. The following pipeline runs at more than 60 FPS on a Google Pixel smartphone.

(1) A 640×480 pixel frame is captured by the camera and OTSU adaptive thresholding [Otsu 1979] is used to isolate the brightest parts of the image.
(2) Connected-component analysis [Dillencourt et al. 1992] is used to find the contour polygons of each bright part. The area of each contour is calculated and contours with an area less than 25% of the camera frame are removed.



(a) Device placement. (b) Attacker's view I.

(c) Attacker's view II. (d) Victim's device

Fig. 8. Augmented reality-based technique applied to attack keystroke dynamics. (a) Attacker places their device on table with rear camera hanging over edge. Attacker types on victim's device held beneath. (b) Attacker looks through their device to see keyboard of victim's device and superimposed mimicry guidance on it (target on "n"). (c) See attacker's fingers partially occluding keyboard. (d) See placement of paper on victim's device to divide display.

(3) The remaining large contours are reduced to similar contours with fewer points using the Douglas-Peucker algorithm (with an epsilon of one-tenth of the original contour perimeter length) [Douglas and Peucker 1973]. Contours that do not reduce to quadrilaterals (i.e., four points) are removed.

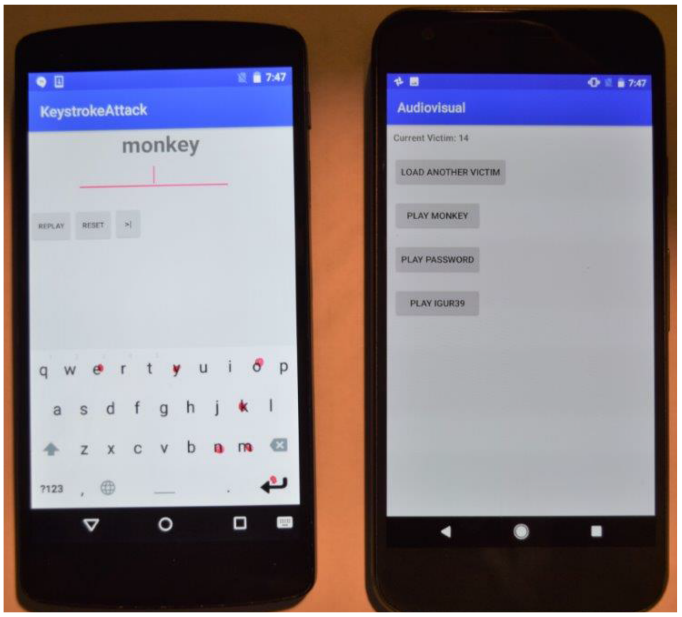


Fig. 9. Audiovisual guidance method. Spatial cues are marked on the keyboard of the victim's phone (left) using a wet erasable marker (for simplicity), and the attacker uses their own device (right) to generate audio signals for cues on inter-stroke interval.

(4) If there are multiple quadrilateral contours remaining, then only the top one is retained for tracking (based on the centroid y-component). This avoids using the bottom contour around the keyboard for tracking, since fingers will occlude it while typing.
(5) To make the tracking more stable, the four corners of the top tracking quadrilateral are smoothed using the one-euro filter [Casiez et al. 2012].
(6) The four corners are then used to find a homography to transform between a known rectangular representation of the entire display (including keyboard area) and the perspective distorted display tracked in the frame. Transforming to the rectangular representation is called rectification, and transforming to the perspective distorted frame is called warping.
(7) A clean image of the keyboard is rectified and extracted from the first few frames when fingers are not yet occluding it. The keyboard image is pasted into the correct portion of the rectangular representation of the display and mimicry guidance is pasted on top of the keyboard. Similar to the training, mimicry guidance is provided in the form of a target that is moving across keys with the same delay as the inter-stroke interval.
(8) The keyboard image with mimicry guidance is warped into the position of the display in the frame and blended to appear partially transparent over the fingers when typing (using 50% alpha transparency).

7.2 Audiovisual Attack

We propose another low-tech approach to provide real-time guidance to the attacker. This approach provides guidance through audio and visual channels (see Figure 9). The attacker prints target pointers (or spatial cues) on a transparent film (or a smartphone screen protector) and carefully overlays it over the victim's device. The attacker then uses an audio signal from their own device for cues on the inter-stroke interval (i.e., a beep indicates that it is time to move to the next

key). We choose an audio cue for inter-stroke interval, since the attacker needs to maintain visual focus on the spatial markers on the victim's keyboard. Simultaneously monitoring a secondary visual cue on the attacker's device for inter-stroke interval would be very difficult.

7.3 Attack Protocol and Study Procedure

The recruited participants were invited to our lab and were briefed about keystroke dynamics, training interfaces, guidance methods, and the attack protocol. The attackers required training on the training interfaces introduced for the unguided attacks, since the guidance methods can neither provide feedback to the attacker nor communicate the victims' behaviour against key specific features (i.e., key hold interval, down pressure, and down area). We followed the same procedure for attackers' data collection and their training as for the unguided attacks (Section 6.2). However, during attacks, attackers were provided real-time guidance through one of the guidance methods. For the augmented reality-based method, we used a Google Pixel device to provide guidance to the attacker. Due to time constraints, for the audiovisual method, we rendered the spatial cues on the victim's device through an app instead of printing them on a screen protector or drawing them with an erasable marker. Each participant was asked to mimic the three passwords for two victims (one from each typing speed pool) for each type of the guidance method. The orders of the two victims and guidance methods were counterbalanced across attackers. The passwords were presented in order from weakest to strongest.

7.4 Participant Recruitment and Motivation

The recruitment procedure and restrictions were the same as for the unguided attacks (Section 6.3). Thirty people participated in this experiment: 20 were between 21 and 30 years old and the rest were between 31 and 40 years old. 16 participants identified themselves as female and the rest identified themselves as male. Eight participants from the unguided attacks experiment also participated in this experiment. Two participants from the unguided attacks experiment were unable to participate due to scheduling conflicts.

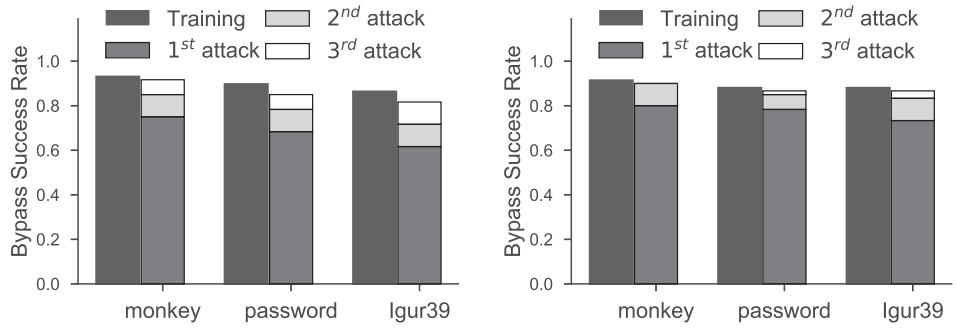
In addition to $10 for base remuneration, participants could receive up to $10 in performance-based rewards. In a first round, the attackers were offered $0.5 for successfully mimicking a password (up to $6). In a second round, they were offered up to $4 if they successfully mimicked up to two victims ($2 per victim) that they had failed to mimic in the first round. This round was introduced to measure the effect of attackers' consistency on mimicry outcome. In case a participant failed to mimic more than two victims in the first round, we chose victims with fast typing speed. Ties were broken in favor of strongest passwords. If a participant successfully mimicked all victims in the first round, then they received the remaining $4.

7.5 Results

We report results for attacker's success and training effort against 360 attacks (30 attackers $\times$ 2 victims $\times$ 2 guidance methods $\times$ 3 passwords). These results cover only the first round. The second round results are provided in Section 7.5.3.

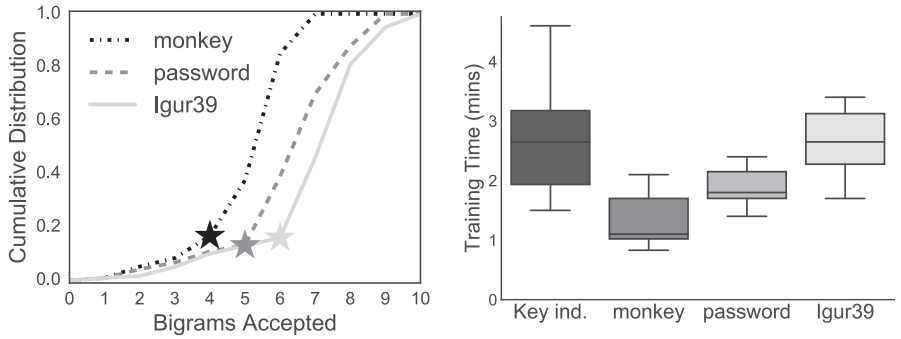
7.5.1 Attacker's Success. For 323 of the 360 attacks, the attackers successfully completed the training. Figure 10 shows that for both guidance methods, 97% of the successful training sessions also resulted in a successful attack in at most three attempts. Furthermore, for 81% of these attacks, the attackers were able to mount the attack in their first attempt.

We also compare the two guidance methods. Our results show that after successful training, only seven and three attacks failed for the AR-based (Figure 10(a)) and audiovisual method (Figure 10(b)), respectively. A chi-square test on the binary outcome of the attack for the two methods



(a) Augmented reality-based guidance. (b) Audiovisual guidance.

Fig. 10. Overall bypass success rate during training and during attacks for guided attacks.



(a) Number of accepted bigrams. (b) Training effort.

Fig. 11. Attackers' success at bigram level and effort. (a) Distribution of number of accepted bigrams. Threshold set at star. (b) Training effort on key independent and key specific interfaces for successful attacks.

indicates no significant differences ($\chi^2(1, N = 323) = 1.62, p = 0.2$). We also observe a difference in the proportion of successful attacks in the first attempt between the two methods. For the audiovisual method, we note that 88% of the attacks only required one attempt as compared to 76% of the attacks for the AR-based method. A chi-square test on the binary outcome of the first attack attempt indicates significant differences between the two methods ($\chi^2(1, N = 312) = 4.8, p < 0.02$). We suspect that these differences were observed for the AR-based method, because it required typing while watching the keyboard through a smaller view window on the attacker's smartphone.

In Figure 11(a), we plot the cumulative distribution of the number of accepted bigrams for different passwords for 48 failed (i.e., part of curve below the star) and 312 successful attack attempts (i.e., part of curve above the star). (Failed attempts also include 37 training failures.) For both failed and successful attempts, we considered attempts with the largest number of accepted bigrams. Recall that a successful attack requires successful mimicry of over 60% of the bigrams. Figure 11(a) shows that for only 5% of the attacks, the attackers were unable to mimic more than two bigrams for all three passwords. It also shows that the attackers were able to successfully mimic 80% or more bigrams for 62%, 60%, and 53% of all attacks against "monkey," "password," and "Igur39," respectively. Finally, for 10% or more of all attacks, the attackers were able to successfully mimic all bigrams. While a key level analysis of mimicry outcome (e.g., character vs. numeral keys) may

provide interesting insights, it is difficult to conduct, because a feature vector is derived from a bigram (comprising of two keys).

7.5.2 Attacker's Effort. We report the attacker's effort for successful attacks in terms of the amount of time spent on training (including retraining). For the AR-based method, attackers practiced typing random text through the smaller view window. This was a one time training where the attackers spent an average of 3.2 minutes (sdev = 0.9).

Figure 11(b) shows the attacker's effort separately for the key independent features (required per victim, not per password) and key specific features. It shows that the successful attackers spent on average between 3.3 and 4.1 minutes to complete their training for different passwords. A one-way ANOVA indicates a significant effect of different passwords on attacker's effort ($F_{2,227} = 236.90, p < 0.01$). Post hoc comparisons using pairwise t-tests (Bonferroni corrected) between the attacker's effort for different passwords show significant differences between all three pairs (all $p < 0.01$). While the effect of attack type (guided vs. unguided) on attacker's effort is of possible interest, we do not have sufficient samples from the unguided attack experiment to perform a conclusive analysis.

7.5.3 Effect of Attacker Consistency. For the second round, attackers were offered a reward if they successfully mimicked up to two victims that they had failed to mimic in the first round. 14 attackers successfully mimicked all victims and were not a part of this experiment. Three attackers declined to participate when they were informed that the victims were the ones that they had failed to mimic during the first round. Their reason was that they had already tried their best and more practice would not change the outcome. Therefore, 13 attackers participated in the second round. Seven of these attackers had to mimic only one victim and the rest had to mimic two victims.

For the second round, eight attackers were able to mimic at least one of the assigned victims. Furthermore, while all attackers were initially unable to mimic different features, for 7 out of 19 attacks they successfully adapted their behaviour with practice. We also note that for 12 out of 19 attacks, attackers were unable to mimic despite practice of up to six minutes. For the successful attacks, the attackers spent an average of 3.8 minutes on training whereas they spent an average of 2.4 minutes on training for the failed attacks. A t-test for the training time between successful and failed attacks indicates no significant differences ($t(17) = 1.74, p > 0.05$). These results suggest the possibility that mimicry success is affected by the typing proficiency of attackers or victims (examined below).

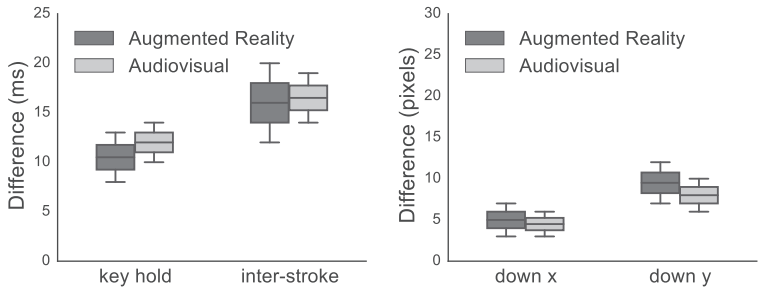
7.5.4 Effect of Victim's and Attacker's Typing Speed. Each participant was assigned two victims of different typing speed. While all victims in our guided attacks experiment were successfully mimicked by one or more attackers, we investigate whether there is a relation between the typing speed of a victim and the bypass success against that victim. For this evaluation, we only use data from the top six fastest and top six slowest victims. We show the bypass success rates against each typing speed in Table 3. A chi-square test on the binary outcome of the attack for the two typing speeds indicates no statistically significant differences ($\chi^2(1, N = 108) = 0.28, p = 0.59$).

The attackers in our experiment had varying levels of typing proficiency. We use the attacker's typing speed to characterize their proficiency. This speed is calculated over the typing data that attackers submitted before training. For this evaluation, we only consider data from the top ten fastest attackers and top ten slowest attackers. The average speed for faster typists was 282ms per character (SD = 45ms), whereas it was 477ms per character (SD = 71ms) for slower typists.

Table 3 shows the bypass success rates against both groups of attackers. To determine whether a relationship exists between the typing speed and the bypass success rate of an attacker, we use a chi-square test on the binary outcome of the attack and between the two typing speed groups. This

Table 3. Effect of Typing Speed on Mimicry Attacks

		Bypass Success Rate		
		monkey	password	Igur39
Victim	Slower	0.94	0.89	0.89
	Faster	0.89	0.89	0.83
Attacker	Slower	0.86	0.82	0.82
	Faster	0.96	0.96	0.93



(a) Temporal subset. (b) Spatial subset.

Fig. 12. Differences between target feature values and mimicked values for the proposed techniques.

test indicates statistically significant differences between the two groups of attackers in terms of their success ($\chi^2(1, N = 120) = 7.65, p < 0.01$). These results show that attackers with faster typing speeds are more likely to mount a successful attack than attackers with slower typing speeds.

7.5.5 Mimicry Accuracy of Proposed Techniques. Our evaluations show that the guidance provided by the proposed techniques improved the bypass success rate against keystroke dynamics. More specifically, compared to the at most 70% bypass success rate in three or fewer attempts for unguided attacks, 97% attempts were successful with our techniques. Similarly, compared to the 6% bypass success rate for the first attempt of unguided attacks, 81% of the attacks were successful in the first attempt with our techniques. For the eight attackers that had previously mounted unguided attacks, 96% of the successful trainings ended up in a successful attack (compared to 70% without our techniques). Similarly, 87% of these successful trainings resulted in the first attempt success for the attack with the guidance (compared to 6% without the guidance). These results demonstrate that the proposed techniques provide a significant improvement over the unguided approach.

We also quantify the mimicry accuracy of attackers for the temporal and spatial features that they received guidance against. To this end, for each feature, we plot the difference between the target feature value and attackers' mimicked values for each key for the first successful attack attempt with each technique in Figure 12. Figure 12(a) shows that on average the AR-based approach enabled attackers to mimic key hold intervals within 10ms (sdev = 3ms) of the target values. The audiovisual approach enabled attackers to mimic key hold intervals within 14ms (sdev = 5ms) of the target values. Similarly, the AR and audiovisual approaches enabled attackers to mimic the inter-stroke intervals within 16ms (sdev = 6ms) and 23ms (sdev = 7ms), respectively, of the target values. We conduct t-tests (Bonferroni corrected) to compare the mimicry accuracy for the two features for both techniques. t-tests indicate significant differences for the difference of target and mimicked values for the key hold interval ($t(418) = 9.94$, $p < 0.025$) and the inter-stroke

interval ($t(418) = 11.02$, $p < 0.025$) between the two approaches. This indicates that the AR approach enabled the attackers to more accurately mimic the temporal features.

Figure 12(b) shows that on average the AR and audiovisual approaches enabled attackers to mimic the down x feature within 5 pixels (sdev = 3 pixels) and 4 pixels (sdev = 3 pixels), respectively, of the target values. Similarly, on average, the AR and audiovisual approaches enabled attackers to mimic the down y feature within 10 pixels (sdev = 5 pixels) and 8 pixels (sdev = 4 pixels), respectively, of the target values. We conduct t-tests (Bonferroni corrected) to compare the mimicry accuracy for the two spatial features for both techniques. t-tests indicate significant differences for the difference of target and mimicked values for the down x feature ($t(418) = 3.41$, $p < 0.025$) and the down y feature ($t(418) = 4.52$, $p < 0.025$) between the two approaches.

7.5.6 Discussion. Our evaluation of the proposed real-time guidance techniques shows that attackers can use our techniques to successfully bypass keystroke dynamics for 87% of the attack attempts. Furthermore, once the attackers have the typing behaviour of their victims, they only require less than five minutes for mimicry training. Although our results show that attackers with slow typing speeds are not as successful as faster typists, they still pose a threat to keystroke dynamics. Our findings show that while keystroke dynamics provide protection against a zero-effort attacker, it may fail to provide protection against more capable adversaries (e.g., intelligence firms). Moreover, the inability of some attackers to bypass keystroke dynamics does not affect such adversaries, since they may have a staff member capable of mounting this attack reliably.

8 ATTACK INFLUENCE FACTORS

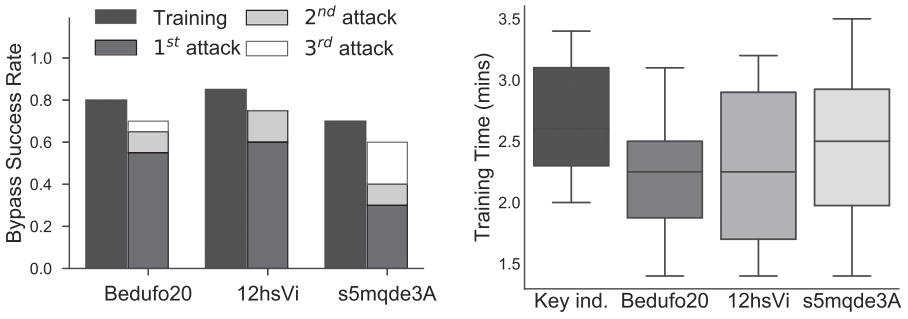
In this section, we discuss constrained attack scenarios and factors that affect attackers' success. For the experiments conducted in this section, we follow the experiment protocol from the guided attacks along with audiovisual guidance.

8.1 Effect of Password Complexity

In Section 7, we evaluate our scheme on 6- and 8-character dictionary words ("monkey" and "password") and a complex password with a combination of upper and lower case characters and numbers ("Igur39"). We conduct further experiments to establish the efficacy of our method with more complex and random passwords. To this end, we choose "Bedufo20," "12hsVi," and "s5mqde3A" from the password set of Buschek et al. We collected data from 16 subjects (nine males and seven females): eight subjects were between 21 and 30 years old and the rest were between 31 and 40 years old. We used the same procedure for data collection as outlined in Section 4.

We also evaluated collected data under the zero-effort attacker model to establish a baseline using the same procedure outlined in Section 4.3. For "Bedufo20," we note a TAR of 0.94 against a FAR of 0.002. We log a TAR of 0.92 against a FAR of 0.007 for "12hsVi". For "s5mqde3A," we note a TAR of 0.95 against a FAR of 0.002 under the zero-effort attacker model. Similar to our original experiment, FARs are considerably low for our evaluation settings.

To evaluate attacks on more complex passwords, we invited a subset of ten participants from the guided attacks experiment. Each participant was asked to mimic the three passwords for two victims. (These victims were new to the attackers.) The passwords were presented to the attackers in the following order: "Bedufo20," "12hsVi," and "s5mqde3A." Figure 13(a) shows the bypass success of the attackers against more complex passwords. It shows that the attackers successfully completed training for 78% of the attacks. 87% of the successful training sessions (68% of the overall attack attempts) also resulted in successful attacks. Across different passwords, we note training success rates of 80%, 75%, and 70% for "Bedufo20," "12hsVi," and "s5mqde3A," respectively. Furthermore, 87%, 88%, and 85% of the successful training sessions for these passwords resulted in



(a) Bypass success. (b) Training effort.

Fig. 13. Evaluation results for attacks on more complex passwords. (a) Bypass success. (b) Training effort on key independent and key specific interfaces for successful attacks.

successful attacks, respectively. Figure 13(b) shows the amount of training effort required for successful attacks. The successful attackers spent an average of 2.2 minutes of training time to bypass "Bedufo20" and "12hsVi" and 2.5 minutes to bypass "s5mqde3A." These results demonstrate that our system can effectively defeat this type of password hardening defense for more complex passwords.

We also compare the success of attackers between simpler (i.e., "monkey," "password," and "Igur39") and more complex passwords (i.e., "Bedufo20," "12hsVi," and "s5mqde3A"). Since we only used the audiovisual guidance method for more complex passwords, we limit our analysis to results from audiovisual experiments only. A chi-square test was conducted to compare the frequency of success between the two sets of passwords. The test indicated that significantly more participants successfully mounted attacks on simpler passwords than more complex passwords ($\chi^2(1, N = 240) = 12.96, p < 0.01$). These results indicate that while an increase in the complexity of passwords decreases the success rate of the attackers, for an alarmingly large 68% of the attacks on complex passwords, our system enabled the attackers to perform successful mimicry.

8.2 Bigram Splicing

In Section 3, we outlined the possibility of an attacker using social engineering means to acquire keystroke data for the same bigrams as the ones in the victim's password. The attacker later splices the keystroke data for the collected bigrams to construct the victim's keystroke behaviour for their password. We conducted an experiment to examine whether this approach will result in successful mimicry attacks. We trained the keystroke classifier used during the attacks phase using raw keystroke data submitted by the victim against "monkey." However, we use the victim's bigram data from the words "money" and "banker" during the training phase. More specifically, this data was used to train the classifier during the training phase and also to determine the inter-quartile range against each target feature.

We invited a subset of ten attackers from the guided attacks experiment to mount this attack. Each attacker was asked to attack two victims. (These victims were new to the attackers.) For 20 bigram splicing attacks, all but two attacks were successfully executed by the attackers in three attempts or less. Eleven attacks were successful in the first attempt, six were successful in the second attempt, and one required three attempts. Two attackers failed the training for one victim but successfully attacked the other victim. These results show that bigram splicing works effectively and poses a real threat to keystroke dynamics. We further establish this similarity in the input behaviour of users by comparing their input behaviour for "monkey" with "money" and

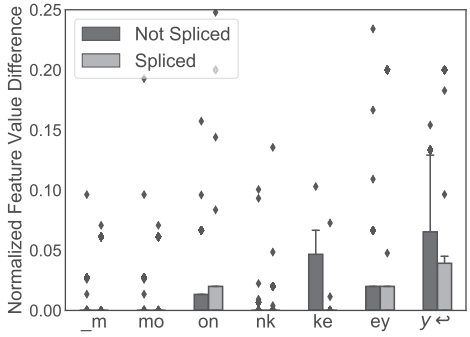


Fig. 14. The difference between the sum of normalized feature values for seven common bigrams between different input attempts of users (↩ denotes the enter key). Data labeled "Not spliced" show the difference between users' input attempts of "monkey" with other input attempts of "monkey." Data labeled "Spliced" show the difference between input attempts of "monkey" with "money" and "banker." (Error bars represent 95% confidence intervals and dots represent outliers.)

Table 4. Effect of Operating Threshold

	Zero-effort		Attackers' success	
	FAR	TAR	Replay	Attack
monkey	0.024	0.73	0.72	—
password	0.002	0.86	0.77	0.85
Igur39	0.002	0.94	0.68	—

"Zero-effort" shows accuracy of classifier under new operating threshold. "Replay" shows outcome when previous attack data (against 0.01 FAR) is replayed on keystroke classifier with average FAR of 0.009. "Attack" shows outcome when attackers train at new operating threshold.

"banker" at the bigram level (i.e., for the shared bigrams "NO_KEYm," "mo," "on," "nk," "ke," "ey," and "yENTER"). In Figure 14, we plot the difference between the sum of normalized feature values between users' input attempts of "monkey" with other input attempts of "monkey" (not spliced) and with spliced bigrams from "money" and "banker." It shows that users exhibit similar behaviour for common bigrams when they are typing "monkey," "money," and "banker."

8.3 Effect of Operating Threshold

In the main experiment, we chose a low average FAR of 0.01, which had a corresponding average TAR of 0.89. This was obtained by thresholding the outcome of the SVM classifier for each bigram by taking a majority score. We tune the γ and C parameters of the SVM classifier to a fixed value and threshold the classifier outcome for individual bigrams of a password, since it enables us to flexibly compute bypass success against a different threshold. To understand the effect of different operating thresholds on the efficacy of attacks, we tune our classifier to a lower FAR, where theoretically it should be difficult for the attackers to mount a mimicry attack. The column titled "Zero-effort model" in Table 4 provides the baseline under the zero-effort attacker model for the new operating threshold. It shows that for the three passwords, we achieve an average FAR of 0.009 with a corresponding average TAR of 0.84. Note that a further decrease in FAR results in a corresponding FRR of 0.27, which is impractical, since the classifier would reject the legitimate user for every fourth attempt, making keystroke dynamics unusable.

We simulate the attacks by replaying the attack data through the keystroke classifier configured at the new operating threshold and log the bypass success rates. A limitation of this approach is that attackers would likely perform better if they trained at the tested FAR. Therefore, in addition to simulation (which provides a lower bound for the bypass success rate), we invited a subset of ten participants from the main experiment to mount this attack. Each participant was asked to attack two victims for the password "password." (These victims were new to the attackers.)

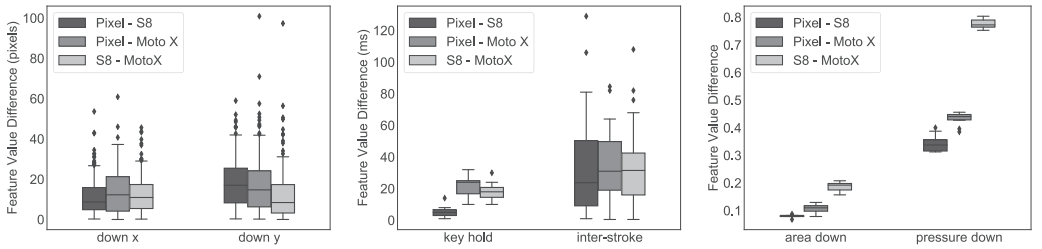
Table 4 shows the results for this experiment. For the simulated evaluation, we note an average bypass success rate of 72%. For the ten attackers who trained with the new operating threshold, we observe that all but three attacks were successful (13 and 4 attacks successful in the first and second attempts, respectively). This shows that even at an unreasonably high FRR and corresponding low FAR, 70% or more attacks are still successful.

8.4 Effect of Device Specific Differences During Victim Behaviour Collection

One of the ways a malicious insider can collect the behaviour of their victim is by social engineering, the victim is tricked into typing using an instrumented application on the attacker's device (see Section 3.) However, if the attacker's device is different than the victim's (such as display resolution and size, physical size or shape, capacitive sensitivity), then the collected data may not effectively capture the behaviour of the victim as performed on their own device. Wang et al. [2017] demonstrated the possibility of transferring behavioural models that were generated on one type of mobile device (smartphones) to another (tablets) to bootstrap continuous authentication. However, they only investigate this transformation for touch input behaviour-based authentication (i.e., swiping behaviour) and not keystroke dynamics. Therefore, we conduct experiments to investigate whether the attacker can collect keystroke dynamics data on their smartphone and mount a mimicry attack using the learned behaviour on the victim's phone when the smartphones have different manufacturer and form factor.

For this experiment, we use use three smartphones of different specifications: (i) a 5.5" Motorola Moto X Play device with a screen resolution of 1080 × 1920 pixels (2015); (ii) a 5" Google Pixel device with a screen resolution of 1080 × 1920 pixels (2016); and (iii) a 5.8" Samsung Galaxy S8 device with a screen resolution of 1440 × 2960 pixels (2017). For this experiment, we recruited participants through word-of-mouth advertising. We collected typing data from 15 participants (nine males and six females): seven subjects were between 21 and 30 years old and the rest were between 31 and 40 years old. For typing data, we asked participants to enter "monkey," "password," and "Igur39." Each participant was asked to enter one password on three devices, take a break for five minutes and then repeat the procedure for the other two passwords. Similar to the data collection setup in Section 4, users were asked to enter each password 20 times on a device before switching to the next device and repeat an entry if they made a mistake (or corrected one). We counter-balanced the device order for participants.

We investigate user behavioural differences across the three different devices for the six key features that were identified in Section 5. To compare behavioural differences, key specific features are compared for each bigram and key independent features are compared across all bigrams. Figure 15(a) shows the differences between the median feature values for the two spatial features from the same user for different device pairs. The spatial feature values were adjusted by performing appropriate scaling to account for differences in the device resolution (i.e., spatial features for the Samsung device were scaled down to 1080 × 1920 pixels) and the size of the rendered keyboard. For the down x feature, Figure 15(a) shows a middle quartile at 15 or fewer pixels for all pair of devices. Similarly, for the down y feature, we note the middle quartile at 17 or fewer pixels. For context, we report the average range of these features on the publicly available dataset from Buschek et al. [2015], which was collected on an LG Nexus 5 device with the resolution of



(a) Spatial features. (b) Temporal features. (c) Contact features.

Fig. 15. Behavioural differences across different devices.

1080 × 1920 pixels. We also remove the outliers (more than two standard deviations from the mean) from the dataset for a more meaningful comparison. The average range across all users for the down x feature is 58 pixels (sdev = 14) and for the down y feature is 66 pixels (sdev = 16). Since the difference at the middle quartile between all device pairs is within the observed range of the feature values, an attacker can estimate their victim's behaviour for these features by collecting data on a different device.

Figure 15(b) shows the differences between the median feature values for the two temporal features without any device-specific adjustments. For the key hold feature, it shows little difference (less than 10ms) between the Google Pixel and Samsung Galaxy S8 devices but more (with middle quartiles at 22ms or less) for the other two pairs of devices. Similarly, for the inter-stroke feature, we note the middle quartile at 28ms or less for all device pairs. In comparison, the average range for the key hold feature across all users is 64ms (sdev = 18) and for the inter-stroke interval feature it is 391ms (sdev = 256) on Buschek et al.'s [2015] dataset. Similar to the spatial features, we note that the difference at the middle quartile between all device pairs is within the observed range of feature values, which suggests the possibility of collecting temporal behaviour on a different device using social engineering means.

Figure 15(c) shows the differences between the median feature values for the two contact features without any device-specific adjustments. (Note that these features take an absolute value between 0 and 1). We note the middle quartile at 0.1 or more for the area feature and at 0.35 or more for the pressure feature between all pairs of devices. Furthermore, we note that these differences are specific to the device make and manufacture. For instance, for the down area feature, the range of values for the Pixel device is 0.196–0.247; for the Galaxy S8 device it is 0.031–0.039; and for the Moto X Play device the down area value is always 0.117 for the same set of users. Similarly, for the same set of users, the down pressure range of values for the Pixel device is 0.196–0.247; for the Galaxy S8 device the down pressure value is always 1; and for the Moto X Play device it is 0.6–0.687. These differences show that different smartphones may return completely different values for area and pressure for the same gesture.

Our results show that while an attacker may be able to capture a users' temporal and spatial behaviour on a smartphone with a different model and manufacturer than the victim's, it will be more difficult for the attacker to do so for the contact features. Therefore, an attacker who is using social engineering to collect their victim's data on their own device should ideally collect it on a device with the same model and manufacturer as their victim's.

9 EXTENDING OUR AR ATTACK TECHNIQUE

While the audiovisual approach is limited by static visuals, our AR technique can be easily extended to attack other behaviour-based authentication schemes. In this section, we demonstrate

Table 5. Touchalytics Features

Feature Category	Features
Spatial	start and stop coordinates, direct end-to-end distance, mean resultant length, length of trajectory, direction, average direction, direction of end-to-end line, largest deviation from end-to-end line, deviation from end-to-end line at 20%-, 50%-, and 80%-percentile, ratio end-to-end distance and length of trajectory
Temporal	stroke duration, inter-stroke time, pairwise velocity at 20%-, 50%-, and 80%-percentile, median velocity at last three points, average velocity, median acceleration at first five points, pairwise acceleration at 20%-, 50%-, and 80%-percentile
Contact	mid-stroke pressure, mid-stroke touch area

For a description of these features, see Frank et al. [2013].

it against a touchscreen swiping behaviour-based biometric and discuss other possible extensions of our AR technique.

9.1 Attacking Swiping Behaviour-based Schemes

Swiping behaviour-based schemes rely on the finger movement patterns that are generated when people use their smartphones normally. For instance, Touchalytics [Frank et al. 2013], a swiping behaviour-based scheme, extracts 31 features from the raw touch data of a swipe. These features capture a user's behaviour using the swipe location, swipe direction, velocity and acceleration of the swipe, duration and length of the swipe, curvature of the swipe, orientation of the finger and the device, and the touch area and the touch pressure of the swipe (see Table 5). Researchers have demonstrated that under the zero-effort attacker model, Touchalytics and similar schemes provide less than 5% FAR [Bo et al. 2013; Frank et al. 2013; Li et al. 2013]. Note that unlike password hardening, swiping input behaviour-based continuous authentication relies on normal usage of the device (i.e., not on a specific gesture to unlock the device). For attacks on swiping input behaviour, we assume the same threat model: the attacker has access to the swiping input behaviour of the user. This behaviour can be obtained using strategies similar to the ones outlined in Section 3.

We demonstrate the efficacy of our AR technique by attacking Touchalytics. 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. In our previous work, we recruited 32 participants to demonstrate two attacks on Touchalytics: shoulder surfing and offline training attacks [Khan et al. 2016]. For shoulder surfing attacks, our previous work demonstrated that attackers are able to observe the swiping behaviour of their victims and then successfully mimic it for 75% of the attack attempts. For offline training attacks, the attackers trained on an interface to learn their victim's behaviour and then successfully mimicked it on the victim's device for 81% of the attack attempts. These attack techniques were unguided, since the attackers had no aid available during the attacks.

9.1.1 AR Technique to Attack Touchalytics. The features that Touchalytics uses to capture the swiping behaviour are extracted from the raw time series data containing x and y coordinates of the touch point, touch pressure, and touch area. Our existing strategy for guided attacks conveys the raw time series containing x and y coordinates of the touch point (by moving the target across points). However, it does not convey touch area and touch pressure to the attacker. Our decision to not convey these factors was to reduce the visual cues the attacker had to track for keystrokes that require resolution in tens of milliseconds. Swiping behaviour is different. An average swipe

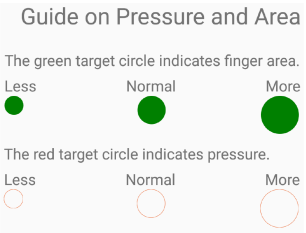


Fig. 16. Screenshot of mapping between the target circles and different pressure and area values shown to attackers as a guide (image cropped).

takes hundreds of milliseconds (e.g., 783ms is the average swipe duration in the publicly available Touchalytics dataset). Furthermore, a swipe requires less dexterity, since unlike typing, the user is not required to perform successive touch screen interactions in a short amount of time. Therefore, we update our AR technique to communicate touch pressure and touch area to the attacker when attacking Touchalytics.

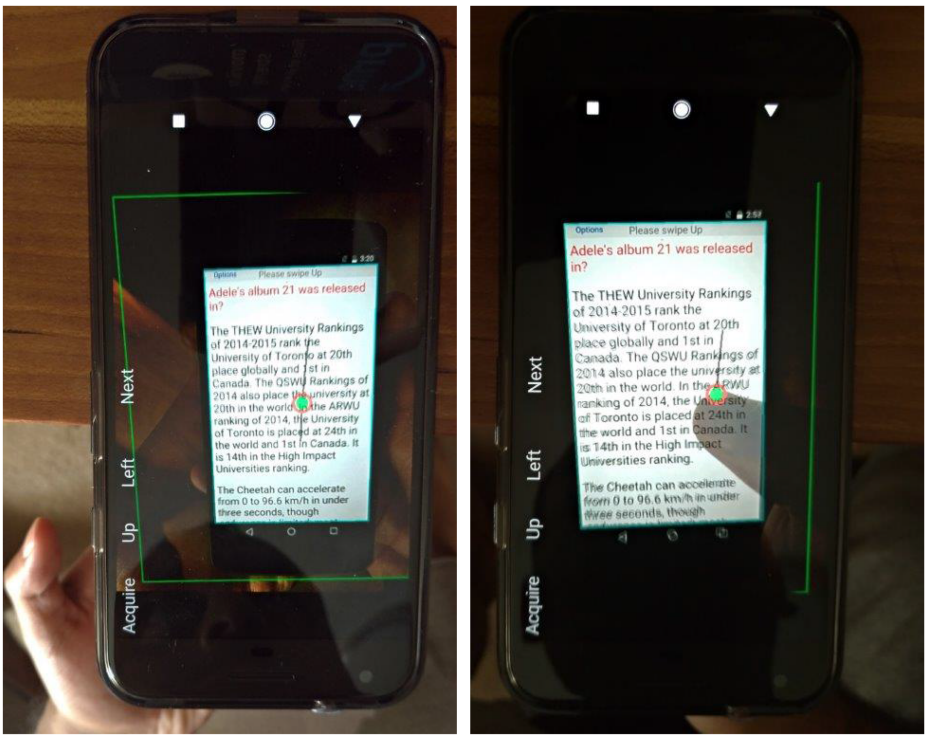
Figure 17(a) shows the AR interface to convey swiping behaviour to the attacker. We communicate the touch pressure value by using a red outline circle around the green target. The more touch pressure the attacker needs to apply, the larger the radius of the circle. To communicate the touch area, we proportionally increase the size of the green target (i.e., the attacker sees a larger target if they need to mimic a larger touch area). The raw touch pressure and touch area are continuous values between 0 and 1, and it is difficult for an attacker to distinguish between close values (e.g., between 0.25 and 0.35). However, since swiping behaviour is used for continuous authentication, the intra-user variability in the model for these features is large. Consequently, instead of reproducing the exact value, the attacker only needs to reproduce values in a certain range to reproduce their victim's behaviour. We also show the attacker a mapping between the target circles and different pressure and area values that they can use as a guide for their swipes before the attack (see Figure 16). Figure 17(b) shows an attacker reproducing a swipe through our modified AR technique.

9.1.2 Data for Attacks and Evaluation Baseline. For our experiments, we used the training models, classifiers and data from our previous work [Khan et al. 2016]. Note that these resources are publicly available through our previous work.[2] A brief description of the data and evaluation baseline follows and interested readers are directed to our previous work for more details.

The data was collected from 55 participants through two Android apps on an LG Nexus 5 device. A Wikipedia app was used to collect up and down swipes while users were reading articles of their choice. A "spot the differences" app was used to collect left and right swipes while users were navigating between two slightly different illustrations. The dataset contains at least 150 swipes in each direction from each participant. In total, it contains over 35,000 swipes comprising over 1.1 million touch points.

For our AR-based guidance technique, we use the same parameters as the previous work. The window size was set to eight swipes (i.e., the authentication score of a user was calculated using eight swipes). The operating point of the SVM classifier was set at a higher FRR of 0.2 corresponding to a FAR of 0.04. For our evaluation baseline, we use the previously reported bypass success rates against the two types of mimicry attacks.

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



(a) Attacker's view I. (b) Attacker's view II.

Fig. 17. Augmented reality-based technique applied to attack Touchalytics. (a) Attacker looks through their device to see the target swipe superimposed on the interface. The size of the green target communicates the touch area and the thickness of the red outline circle communicates the pressure. (b) Attacker's fingers partially occluding screen.

9.1.3 Attack Protocol and Study Procedure. We used the same attack tasks as in our previous work. The first task presents the attacker with a browser like interface with a question preceding a collection of paragraphs from Wikipedia. Each paragraph discusses a different topic 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. The second task displays 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. Previous work shows that attackers who can mimic up and left swipes have no difficulty mimicking down and right swipes [Khan et al. 2016]. Therefore, we only consider mimicry of up and left swipes, which also helps to keep the experiment duration reasonable.

For this experiment, we recruited participants through word-of-mouth advertising. Participants were introduced to the Touchalytics scheme, our AR-based attack interface, and the attack tasks. We used a Google Pixel device to provide mimicry guidance to the attackers and an LG Nexus 5 device as the victim's device. Each participant was assigned three victims and they were instructed to reproduce eight up and eight left swipes using the AR-based interface provided in Figure 17(a) (eight is the size of the window). Pressing the *up* or *left* label in this interface would move the target with the same speed as the victim's swipe. The attacker was provided feedback

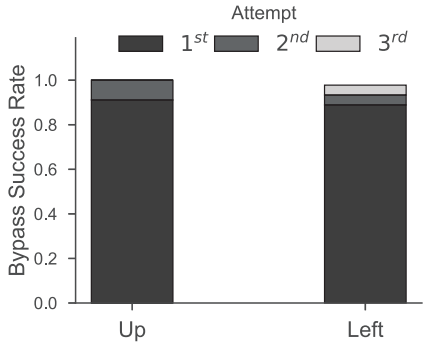


Fig. 18. Overall bypass success rate for attacks on Touchalytics [Frank et al. 2013] using augmented reality-based guidance.

regarding success or failure of their attempt only after eight swipes (i.e., a window). The attacker was allowed to observe the moving target (without tracing it) as many times as they desired. If the attacker failed to succeed after one window, then they were allowed up to two more windows.

9.1.4 Results. 15 participants were invited to the lab for a 30-minute experiment (nine males and six females): seven subjects were between 21 and 30 years old and the rest were between 31 and 40 years old. Unlike previous approaches to attack Touchalytics, our augmented reality-based technique requires no training, and we only report attackers' success.

Figure 18 shows the results for 90 attacks (15 attackers × 3 victims × 2 directions) on Touchalytics using our augmented reality-based guidance method. It shows that except one attack on aleft swipe, all attacks were successful. This 99% bypass success rate is a considerable improvement over the baseline of 81% bypass success rate for offline training attacks and 75% bypass success rate for shoulder surfing attacks. Our augmented reality-based technique also enables attackers to bypass Touchalytics with fewer failed attempts—only 9% of attackers required more than one attempt. In comparison, 27% and 15% attackers required more than one attempt for shoulder surfing and offline training attacks, respectively. The improvement over the baseline approaches were due to the ability of our augmented reality-based technique to provide real-time guidance to the attacker for Touchalytics features. These results show that our augmented reality-based technique enables attackers to mount mimicry attacks on diverse touch input-based biometrics.

9.2 Future Directions

Several input behaviour-based authentication proposals (e.g., those discussed in Section 2) are presumed secure with the assumption that the attacker is unable to reproduce the behaviour of the user. However, advances in computer vision suggest that novel attack vectors must be considered during the security evaluation of these proposals. Our findings strongly suggest that security researchers must consider the threat posed by novel methods that precisely reproduce the behaviour of the user.

Touchscreen gesture-based approaches [Shahzad et al. 2013] that rely on features similar to Touchalytics may potentially be mimicked if the swiping behaviour is known. Continuous authentication schemes that rely on swipe-based predictive typing behaviour have been proposed [Burgbacher and Hinrichs 2014]. These schemes also rely on spatio-temporal features similar to Touchalytics and may be vulnerable to mimicry attacks with real-time guidance if the user behaviour is known. Schemes that rely on how users shake or wave their device [Hong et al. 2015; Yang et al. 2015; Zhu et al. 2017] are also susceptible to AR-based attacks. Anderson et al. [2013]

developed an AR system that allows users to record and reproduce movement sequences. This approach can be used to reproduce device shaking and waving behaviour. AR-based attacks may be used to attack other behaviour-based authentication systems. For instance, iOS apps that teach users how to dance by dictating where to put the feet (e.g., Dance Reality [Bell 2017]) might be extended to attack gait behaviour-based authentication systems [Boyle et al. 2011; Kwapisz et al. 2010].

10 LIMITATIONS

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

For data collection, we used a subset of passwords from Buschek et al.'s study. Like Buschek et al., and other similar password entry studies [Serwadda and Phoha 2013a; Tey et al. 2013], the bigrams that we used were likely not frequently entered by the participants on their devices (i.e., like their passwords) and their behaviour may be different than the frequently entered bigrams. However, this limitation was unavoidable for a time-constrained, lab-based experiment. Despite this limitation, our experiments show that our techniques enable attackers to reproduce diverse typing behaviour of their victims.

We allowed participants to spend up to ten minutes to train for a given password. In addition to ensuring that attackers did not get overtired or bored, this restriction was necessary to get comparable data across attackers. A real-world attacker might be highly motivated to train for longer periods to mount a successful attack. Nevertheless, our experiments establish a lower-bound on attackers' success.

We did not collect data from victims across different sessions. However, Buschek et al. showed that the keystroke classifier achieves lower EERs within session than across sessions. This made the mimicry attack more challenging for the attackers.

We did not offer remuneration to the participants who volunteered for the additional experiments (reported in Section 8 and Section 9). Consequently, these participants may had less motivation, which may have affected their performance. For the data collection task (Section 4 and Section 8.4) and the extended attacks experiment (Section 9), we used word-of-mouth for participant recruitment instead of a mailing list. We chose word-of-mouth for these tasks, since we were able to easily recruit volunteers without remunerating them for small experiments. While the pool of participants was different, our analysis in Section 7.5.4 shows that these participants had sufficiently diverse typing speeds for the data collection experiments. Similarly, while these participants may have resulted in an upper bound on attacker's performance for the attack experiments, they are a good representative of a capable adversary against such schemes.

11 RELATED WORK

Biometrics can be broadly classified into behavioural and physiological (e.g., facial and fingerprint recognition). Since our work investigates mimicry attacks on a behavioural biometric, we do not discuss spoofing attacks on physiological biometrics. In this section, we discuss attacks on behavioural biometrics.

Researchers have proposed crowd sourcing to attack behavioural biometrics. Under this approach, the goal is to search for a candidate whose natural behaviour matches that of the victim. Panjwani and Prakash [2014] demonstrated crowd sourcing attacks on a speaker verification system. They used Amazon MTurk to locate and recruit candidate mimics with a successful match rate of 3%. Gafurov et al. [2007] demonstrated crowd sourcing attacks on gait biometric. They used a database with gait sequences from 100 subjects to show that closest matching subjects against a set of physical characteristics could increase the EER up to 80%. While similar attacks might be

possible on keystroke dynamics, our evaluations show that the attackers can effortlessly adjust their behaviour to mimic their victims. This eliminates the need to locate and recruit users with similar keystroke behaviour.

Behaviour biometrics are also vulnerable to generative algorithms-based attacks. These attacks employ general population statistics to infer the victim's behaviour. Serwadda and Phoha [2013b] demonstrated a generative algorithm-based attack on touch input behaviour-based biometric for smartphones. Their attack increases the EER of touch input biometric from 5% to 50%. Serwadda and Phoha [2013a] evaluated generative attacks against keystroke biometric on personal computers. They built a generative model by analyzing statistical traits from over 3,000 users. They demonstrated that their generative model increased the EER of a keystroke classifier from 28% to 84%. Their model could be leveraged by a malware or bot to submit generate fake keystroke data on a PC (user-to-website authentication). While our collected data was insufficient to determine whether statistical traits exist across a substantial population for keystroke behaviour on soft keyboards, it is a potential avenue for future research. If statistical traits exist, then an attacker can learn the generic behaviour and use our setup to attack a victim for which raw keystroke data is unavailable.

Mimicry attacks that train attackers to mimic their victims have been demonstrated for touch input and keystroke biometrics. Khan et al. [2016] developed an interface on smartphones that uses raw swiping data to visualize and train attackers to mimic the swiping behaviour of their victim. They recruited 32 attackers to show that by training on their interface, attackers were able to successfully mount attacks for 86% of their attack attempts. In Section 9, we show that our technique enables the attackers to successfully mimic the swiping behaviour of their victims. More related to our work is that of Tey et al. [2013]. Tey et al. trained attackers to mimic two keystroke features on personal computers. Their evaluation showed that 14 of their best attackers (out of 84 attackers) were able to achieve a 99% bypass success rate. While our unguided training attacks are inspired by the work of Tey et al., we perform the first ever evaluation of mimicry attacks on keystroke dynamics for smartphones and propose novel methods to provide real-time guidance during attacks.

12 CONCLUSION

We evaluate the susceptibility of keystroke dynamics against password hardening on smartphones to mimicry attacks. We perform feature analysis on a publicly available dataset and use our findings to build interfaces to train users to mimic their victim's keystroke behaviour. We design two methods that enable an attacker to obtain real-time guidance during their mimicry attack. Our setup effectively circumvents keystroke dynamics against a variety of passwords. We also conduct experiments to show that malicious insiders may carefully collect keystroke data through social engineering means and then use the collected data to reconstruct the victim's behaviour. Our findings suggest a careful reconsideration of the security provided by keystroke dynamics when the attacker has access to the behavioural profile of the victim.

REFERENCES

Mansour Alsaleh, Mohammad Mannan, and Paul C. van Oorschot. 2012. Revisiting defenses against large-scale online password guessing attacks. *IEEE Trans. Depend. Secure Comput.* 9, 1 (2012), 128–141.

Fraser Anderson, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2013. YouMove: Enhancing movement training with an augmented reality mirror. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. ACM.

Salil P. Banerjee and Damon L. Woodard. 2012. Biometric authentication and identification using keystroke dynamics: A survey. *J. Pattern Recogn. Res.* 7, 1 (2012), 116–139.

Wei Bao, Hong Li, Nan Li, and Wei Jiang. 2009. A liveness detection method for face recognition based on optical flow field. In *Proceedings of the International Conference on Image Analysis and Signal Processing*. IEEE, 233–236.

BehavioSec. 2017. A supplement to Authentication in an Internet Banking Environment. Retrieved rom https://www.behaviosec.com/financial-services/.

Karissa Bell. 2017. New ARKit iPhone app will help your learn to be a better dancer. Retrieved from https://mashable.com/2017/07/09/dance-reality-arkit-app.

Antonio Bianchi, Jacopo Corbetta, Luca Invernizzi, Yanick Fratantonio, Christopher Kruegel, and Giovanni Vigna. 2015. What the app is that? Deception and countermeasures in the android user interface. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE.

Cheng Bo, Lan Zhang, Xiang-Yang Li, Qiuyuan Huang, and Yu Wang. 2013. SilentSense: Silent user identification via touch and movement behavioral biometrics. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*. ACM, 187–190.

Joseph Bonneau, Cormac Herley, Paul C. Van Oorschot, and Frank Stajano. 2012. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE.

Matthew Boyle, Avraham Klausner, David Starobinski, Ari Trachtenberg, and Hongchang Wu. 2011. Poster: Gait-based smartphone user identification. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*. ACM, New York, NY.

Ulrich Burgbacher and Klaus Hinrichs. 2014. An implicit author verification system for text messages based on gesture typing biometrics. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*. ACM.

Daniel Buschek, Alexander De Luca, and Florian Alt. 2015. Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM.

Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1-Euro filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2527–2530.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2, 3 (2011), 27.

Nathan L. Clarke and S. M. Furnell. 2007. Authenticating mobile phone users using keystroke analysis. *Int. J. Info. Secur.* 6, 1 (2007), 1–14.

Michael B. Dillencourt, Hanan Samet, and Markku Tamminen. 1992. A general approach to connected-component labeling for arbitrary image representations. *J. ACM* 39, 2 (1992), 253–280.

David H. Douglas and Thomas K. Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: Int. J. Geogr. Info. Geovisual.* 10, 2 (1973), 112–122.

Benjamin Draffin, Jiang Zhu, and Joy Zhang. 2014. KeySens: Passive user authentication through micro-behavior modeling of soft keyboard interaction. In *Mobile Computing, Applications, and Services*. Springer, 184–201.

Richard O. Duda, Peter E. Hart, and David G. Stork. 2012. *Pattern Classification*. John Wiley & Sons.

Serge Egelman, Sakshi Jain, Rebecca S Portnoff, Kerwell Liao, Sunny Consolvo, and David Wagner. 2014. Are you ready to lock? In *Proceedings of the ACM SIGSAC Conference on Computer & Communications Security*. ACM.

Malin Eiband, Mohamed Khamis, Emanuel von Zezschwitz, Heinrich Hussmann, and Florian Alt. 2017. Understanding shoulder surfing in the wild: Stories from users and observers. In *Proceedings of the 35th Annual ACM Conference on Human Factors in Computing Systems*. ACM.

Tao Feng, Jun Yang, Zhixian Yan, Emmanuel Munguia Tapia, and Weidong Shi. 2014. TIPS: Context-aware implicit user identification using touch screen in uncontrolled environments. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. ACM.

Tao Feng, Xi Zhao, Bogdan Carbunar, and Weidong Shi. 2013. Continuous mobile authentication using virtual key typing biometrics. In *Proceedings of the 12th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 1547–1552.

Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. 2013. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Trans. Info. Forensics Secur.* 8, 1 (2013), 136–148.

Davrondzhon Gafurov, Einar Snekkenes, and Patrick Bours. 2007. Spoof attacks on gait authentication system. *IEEE Trans. Info. Forensics Secur.* 2, 3 (2007), 491–502.

Cristiano Giuffrida, Kamil Majdanik, Mauro Conti, and Herbert Bos. 2014. I sensed it was you: Authenticating mobile users with sensor-enhanced keystroke dynamics. In *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 92–111.

Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* 6, 2 (1979), 65–70.

Feng Hong, Meiyu Wei, Shujuan You, Yuan Feng, and Zhongwen Guo. 2015. Waving authentication: Your smartphone authenticate you on motion gesture. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM.

Seong-seob Hwang, Sungzoon Cho, and Sunghoon Park. 2009. Keystroke dynamics-based authentication for mobile devices. *Comput. Secur.* 28, 1 (2009), 85–93.

Hassan Khan and Urs Hengartner. 2014. Towards application-centric implicit authentication on smartphones. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. ACM.

Hassan Khan, Urs Hengartner, and Daniel Vogel. 2016. Targeted mimicry attacks on touch input-based implicit authentication schemes. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM.

Hassan Khan, Urs Hengartner, and Daniel Vogel. 2018. Augmented reality-based mimicry attacks on behaviour-based smartphone authentication. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM.

Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. 2010. Cell phone-based biometric identification. In *Proceedings of the 4th IEEE International Conference on Biometrics: Theory Applications and Systems*. IEEE, 1–7.

Lingjun Li, Xinxin Zhao, and Guoliang Xue. 2013. Unobservable reauthentication for smart phones. In *Proceedings of the 20th Network and Distributed System Security Symposium*, Vol. 13.

Emanuele Maiorana, Patrizio Campisi, Noelia González-Carballo, and Alessandro Neri. 2011. Keystroke dynamics authentication for mobile phones. In *Proceedings of the Symposium on Applied Computing*. ACM, 21–26.

Emiliano Miluzzo, Alexander Varshavsky, Suhrid Balakrishnan, and Romit Roy Choudhury. 2012. Tapprints: Your finger taps have fingerprints. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*. ACM, 323–336.

Fabian Monrose, Michael K. Reiter, and Susanne Wetzel. 2002. Password hardening based on keystroke dynamics. *Int. J. Info. Secur.* 1, 2 (2002), 69–83.

Mozilla. 2019. MDB Browser compatibility data. https://github.com/mdn/browser-compat-data. Last accessed: 07/2019.

Parimarjan Negi, Prafull Sharma, Vivek Jain, and Bahman Bahmani. 2018. K-means++ vs. behavioral biometrics: One loop to rule them all. In *Proceedings of the 25th Network and Distributed System Security Symposium*.

Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybernet.* 9, 1 (1979), 62–66.

Saurabh Panjwani and Achintya Prakash. 2014. Crowdsourcing attacks on biometric systems. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS'14)*. USENIX Association.

Bruce Schneier. 2009. Schneier on Security: Biometrics. Retrieved from https://www.schneier.com/blog/archives/2009/01/biometrics.html.

Abdul Serwadda and Vir V. Phoha. 2013a. Examining a large keystroke biometrics dataset for statistical-attack openings. *ACM Trans. Info. Syst. Secur.* 16, 2 (2013), 8.

Abdul Serwadda and Vir V. Phoha. 2013b. When kids' toys breach mobile phone security. In *Proceedings of the ACM SIGSAC Conference on Computer & Communications Security*. ACM, 599–610.

Muhammad Shahzad, Alex X. Liu, and Arjmand Samuel. 2013. Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*. ACM, 39–50.

Chee Meng Tey, Payas Gupta, and Debin Gao. 2013. I can be you: Questioning the use of keystroke dynamics as biometrics. In *Proceedings of the 20th Annual Network & Distributed System Security Symposium*.

The Register. 2019. Not very Suprema: Biometric access biz bares 27 million records and plaintext admin creds. Retrieved from https://www.theregister.co.uk/2019/08/14/biostar_2_suprema_database_exposed_27m_records/.

W3C. 2019. Touch Events: Draft Community Group Report. Retrieved from https://w3c.github.io/touch-events/.

Xiao Wang, Tong Yu, Ole Mengshoel, and Patrick Tague. 2017. Towards continuous and passive authentication across mobile devices: An empirical study. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 35–45.

Lei Yang, Yi Guo, Xuan Ding, Jinsong Han, Yunhao Liu, Cheng Wang, and Changwei Hu. 2015. Unlocking smart phone through handwaving biometrics. *IEEE Trans. Mobile Comput.* 14, 5 (2015), 1044–1055.

Hongzi Zhu, Jingmei Hu, Shan Chang, and Li Lu. 2017. ShakeIn: Secure user authentication of smartphones with single-handed shakes. *IEEE Trans. Mobile Comput.* 16, 10 (2017), 2901–2912.

Received February 2019; revised August 2019; accepted November 2019