Augmented Reality-based Mimicry Attacks on Behaviour-Based Smartphone Authentication

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

ABSTRACT

We develop an augmented reality-based app that resides on the attacker's smartphone and leverages computer vision and raw input data to provide real-time mimicry attack guidance on the victim's phone. Our approach does not require tampering or installing software on the victim's device, or specialized hardware. The app is demonstrated by attacking keystroke dynamics, a method leveraging the unique typing behaviour of users to authenticate them on a smartphone, which was previously thought to be hard to mimic. In addition, we propose a low-tech AR-like audiovisual method based on spatial pointers on a transparent film and audio cues. We conduct experiments with 31 participants and mount over 400 attacks to show that our methods enable attackers to successfully bypass keystroke dynamics for 87% of the attacks after an average mimicry training of four minutes. Our AR-based method can be extended to attack other input behaviour-based biometrics. While the particular attack we describe is relatively narrow, it is a good example of using AR guidance to enable successful mimicry of user behaviour—an approach of increasing concern as AR functionality becomes more commonplace.

CCS CONCEPTS

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

KEYWORDS

Augmented reality, Mimicry attacks, Authentication, Behavioural biometrics

ACM Reference Format:
Hassan Khan, Urs Hengartner, Daniel Vogel. 2018. Augmented Reality-based Mimicry Attacks on Behaviour-Based Smartphone Authentication. In *MobiSys '18: The 16th Annual International Conference on Mobile Systems, Applications, and Services, June 10–15, 2018, Munich, Germany*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3210240.3210317

1 INTRODUCTION

Although Augmented Reality (AR) research dates back to Sutherland's experiments in the 1960s [46], only recently have advances in computer vision, display technologies, and portable graphics processing power brought it to mass-consumer devices like smartphones [21, 35]. This means we may finally be seeing AR used in fields like architecture, industrial design, health care, marketing, manufacturing, military, education, and entertainment [3, 7, 49]. And regardless of specific vertical applications, the general idea of using mobile AR to provide personal or public information to annotate and visualize the world will likely touch all of our lives [25].

Of course, every new technology also introduces benefits and risks for security and privacy. Security researchers have examined attacks on AR hardware and AR systems [32, 36, 40] and proposed ways for people to use AR for such things as increasing public privacy [47] and hiding device authentication [52]. However, we believe one aspect has not been fully explored — when an attacker uses AR to break conventional security methods like behaviour-based authentication.

Knowledge-based authentication systems on smartphones (e.g., PIN, passwords, and Android's Pattern Lock) are susceptible to shoulder surfing [19]. Researchers have proposed authentication schemes that rely on the input behaviour of users to defend against shoulder surfing attacks like user-specific patterns of swiping and typing [9, 22, 33]. Researchers suggest that these proposals are secure because a user's unconscious behaviour is difficult to steal or mimic by an observer [14, 22, 57, 58].

Other researchers have enumerated ways to steal the input behaviour and developed non-AR interfaces to train an attacker to mimic the stolen behaviour [30, 48]. However, this approach is not effective when the attacker has to mimic multiple features at a milliseconds resolution. We demonstrate this against a recent keystroke behaviour-based scheme ("keystroke dynamics") for smartphones [9]. Our experiments show that the non-AR training interfaces approach is only moderately successful against keystroke dynamics. More specifically, while attackers are able to reproduce the victim's behaviour on the training interfaces, they often fail to reliably reproduce the behaviour when mounting the attack on the victim's device due to the lack of any guidance.

We present a novel AR-based method that enables an attacker to precisely mimic the behaviour of their victim in real-time using a standard smartphone. The attacker captures a sample of the victim's behaviour-based input dynamics (such as convincing them to enter text on the attacker's phone), then later gains temporary possession of the victim's device. The attacker runs our AR-based app on their own smartphone and positions it so that the camera captures a view of their hands as they hold the victim's phone. Using the captured data, our app overlays guidance information on the camera stream enabling the attacker to mimic the victim's input behaviour in real-time. We also design and test a simple physical AR prototype to

MobiSys '18, June 10–15, 2018, Munich, Germany

ACM ISBN 978-1-4503-5720-3/18/06…$15.00
https://doi.org/10.1145/3210240.3210317

simulate perfect tracking. The attacker sticks a transparent film with spatial cues onto the victim's device and uses audio beeps to time their taps. Note our guidance methods do not require specialized hardware, they do not require special software on the victim's device, and tampering is as simple as sticking a piece of tape onto the display. While the challenging aspect of capturing user behaviour makes this attack relatively narrow, it is a good example of using AR guidance to enable successful mimicry of user behaviour. Our attack is of increasing concern as AR functionality becomes more commonplace and as researchers propose more efficient methods to locate user behaviour from a large population [38].

We recruit 31 participants to evaluate over 400 mimicry attacks on keystroke dynamics using our AR techniques. The results show attackers successfully bypass keystroke dynamics for 87% of the attacks after an average mimicry training of four minutes. For 73% of attacks, attackers bypassed keystroke dynamics in their first attempt—a considerable improvement over the 6% bypass success rate for first attempts without AR. Our AR method can be generalized to attack other input behaviour-based methods (e.g., Touchalytics [22]). Our contributions are:

(1) A smartphone AR method and an AR-like audiovisual method to provide real-time input guidance to mimic user behaviour on a smartphone.
(2) An application mounting the first ever successful attack on keystroke dynamics for smartphones. Our methods show that keystroke dynamics is susceptible to these mimicry attacks.
(3) An opensource release of the AR-based app for researchers to replicate our experiments and extend our methodology to attack other input behaviour-based authentication methods[1].

2 RELATED WORK

Researchers have proposed AR-based systems that guide users' movements for purposes like teaching them how to move or dance [2, 6] or play piano [10, 51] or play other musical instruments (see the survey by Santos et al. [41] and references therein). However, to the best of our knowledge, no AR-based system has been proposed that guides users to input in a specific way on a target smartphone. or for breaking conventional security methods including behaviour-based authentication. Therefore, we only discuss related tools that have been proposed to attack behaviour-based authentication systems.

Researchers have used video footage of users to mount shoulder surfing attacks on a swiping input behaviour-based system [30]. Other researchers have demonstrated that such attacks are ineffective against schemes that employ cognitive abilities [1], eye movement patterns [18] or touchscreen gestures [44, 45]. Such attacks were unsuccessful because the attackers failed to mimic the behaviour of the user. However, it is an open question whether a technique like ours, which assists the attackers to precisely mimic the behaviour of the user, will be able to break these systems.

Attackers may be able to mount mimicry attacks by submitting the keystroke behaviour on a PC. Serwadda and Phoha [42] evaluated generative attacks against a keystroke biometric on a PC. They built a generative model by analyzing statistical traits from over 3000 users. They demonstrated that their generative model increased the EER of a keystroke classifier from 28% to 84% (see § 4.1 for a definition of EER). Their model could be leveraged by a malware or bot to submit fake keystroke data on a PC. However, this approach only works for the user-to-website authentication scenario. This approach is ineffective against the user-to-device authentication scenario (e.g., a device protected by a password), where the attacker has to mimic the behaviour on the victim's device. Our techniques can be used to attack both user-to-website and user-to-device authentication scenarios.

An attacker may also leverage a robotic device to reproduce the behaviour of the user. Serwadda and Phoha [43] demonstrated such an attack on a swiping behaviour-based biometric for smartphones. They showed that a robotic device equipped with the generic swiping traits poses a major threat to these schemes as it increases their EER from 5% to 50%. A similar robotic device may be designed to mimic keystroke dynamics. However, this approach may be impractical in scenarios where bringing a robotic device to the premises of the victim may raise suspicion. Our technique is portable since it only requires an off-the-shelf smartphone.

Using specialized non-AR training interfaces for mimicry attacks has been demonstrated for touch input and keystroke biometrics. In earlier work [30], we developed an interface on smartphones that uses raw swiping data to visualize and train attackers to mimic the swiping behaviour of their victim. We recruited 32 attackers to show that by training on our interface, attackers were able to successfully mount attacks for 86% of their attempts. More related to our work is that of Tey et al. [48], who trained attackers to mimic two keystroke features on a PC with a physical keyboard. Their evaluation showed that 14 of their best attackers (out of 84 attackers) were able to achieve a 99% bypass success rate. Our proposed techniques are complementary to these approaches and help the attacker to better mimic their victim by providing real-time input guidance using AR.

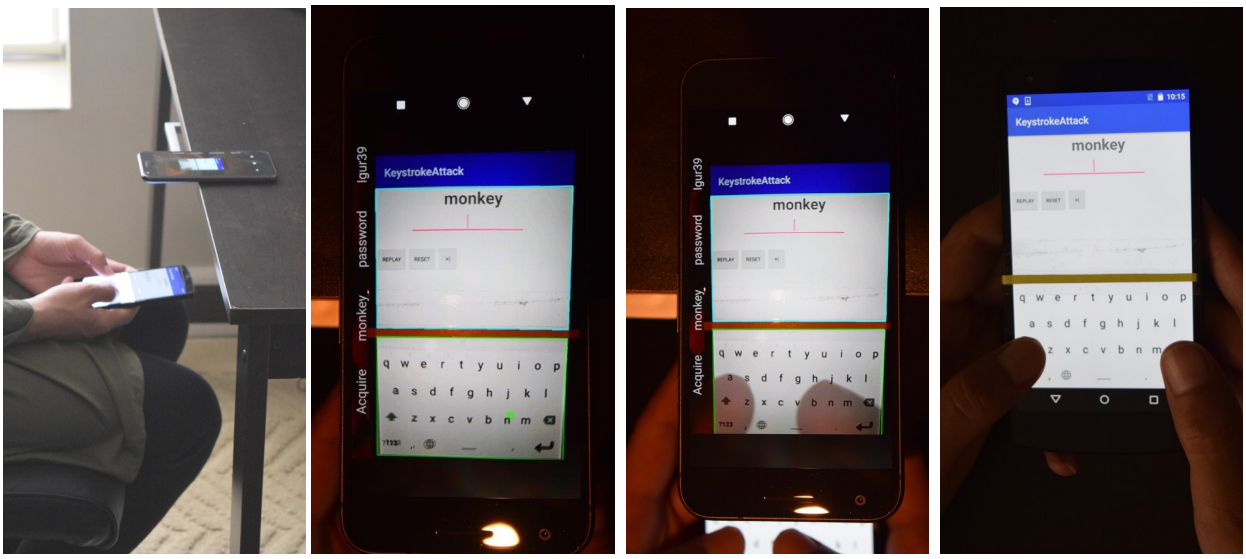
3 AR TO MIMIC BEHAVIOUR BIOMETRICS

Our proposed techniques provide real-time input guidance to mount mimicry attacks on input behaviour-based authentication systems. We exemplify this by attacking keystroke dynamics after an opportunistic acquisition of the unattended device of the victim for a limited time only. Such an acquisition may take place at a location where the attacker has access to limited resources (e.g., at the victim's home or work). We followed the following design principles to satisfy these constraints:

Non-tampering: Attacking a user-to-device authentication system entails that the device is protected through an authentication mechanism and the attacker cannot install or tamper any application that may assist them with the attack. Therefore, our techniques should only require that the attacker has access to device configuration settings that do not require authentication (e.g., the brightness level of the screen) or that the attacker is able to put a removable piece of paper or a transparent film on the device for any assistance.

Portable: The location constraint requires that the guidance techniques should be portable. Approaches that require special equipment (e.g., assistance from a robotic device [43]) are impractical since bringing such equipment to the premises of the victim may

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



(a) Device placement. (b) Attacker's view I. (c) Attacker's view II. (d) Victim's device.

Figure 1: 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.

raise suspicion. Our goal was to develop a technique that does not require any special equipment beyond an off-the-shelf smartphone.

We now provide details of our AR and audiovisual techniques that satisfy these constraints and provide real-time guidance for successful mimicry attacks.

3.1 AR-based Approach

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 1 shows an attacker using our app to get real-time guidance during the attack.

In our setup, the attacker adjusts 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 640x480 pixel frame is captured by the camera and OTSU adaptive thresholding [39] is used to isolate the brightest parts of the image.
(2) Connected-component analysis [15] 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.
(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) [16]. Contours that do not reduce to quadrilaterals (i.e., four points) are removed.
(4) If there are multiple quadrilateral contours remaining, 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) For stable tracking, the four corners of the top quadrilateral are smoothed using the one-euro filter [11].
(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

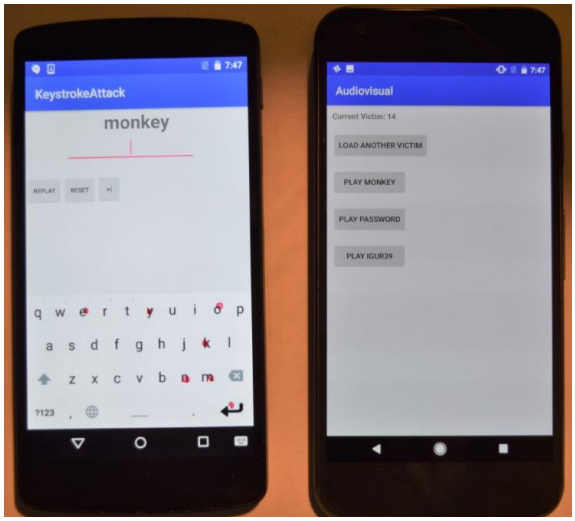


Figure 2: Audiovisual technique. Spatial cues are marked on victim's phone (left). Audio from attacker's device (right) provides cues on inter-stroke interval.

the rectangular representation of the display and mimicry guidance is pasted on top of the keyboard. 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).

3.2 Audiovisual Approach

We also propose an audiovisual approach, which is a low-tech physical AR prototype with perfect tracking but limited static visuals. This approach provides guidance through audio and visual channels (see Figure 2). The attacker prints spatial cues on a transparent film (or a smartphone screen protector) and overlays it over the victim's device. Alternatively, the attacker may also mark spatial cues using a wet erasable marker. The attacker then uses an audio signal from their own device for cues on the inter-stroke interval of the victim (i.e., a beep indicates time to move to the next key).

4 MIMICRY OF KEYSTROKE DYNAMICS

We demonstrate the efficacy of our techniques by mounting mimicry attacks on keystroke dynamics. Our choice of keystroke dynamics is motivated by the availability of commercial keystroke dynamics solutions for financial institutions [5] and also by the keystroke dynamics' ability to capture the behaviour through multiple features at a milliseconds resolution. In this section, we first provide the necessary background and outline the threat model. We then provide the details of the dataset that we used for our experiment and the features that we target for mimicry attacks. These details are essential to understand the evaluation of the AR techniques.

4.1 Background

In this section, we provide an overview of various input behaviour-based proposals and provide details on keystroke dynamics.

4.1.1 Input Behaviour-Based Schemes. Researchers have proposed several authentication schemes that rely on the input behaviour of users with their smartphones to defend against shoulder surfing attacks. The touch input behaviour-based proposals rely on the unique swiping or typing input of smartphone users [9, 22, 33]. The touchscreen gesture-based proposals rely on the precise manner in which a user draws a free-form or predefined shape on the touchscreen, where the shape is not a secret [14, 44, 45]. Finally, some proposals capture the unique device waving and shaking behaviour of the user through the on-board accelerometer and gyroscope sensors [26, 55, 58]. Researchers have argued that these proposals are secure because the input behaviour is difficult to steal or mimic for an observer [14, 22, 57, 58].

4.1.2 Keystroke Dynamics. Before we provide an overview of keystroke dynamics and the scheme that we evaluate, we define various accuracy metrics that are used in the literature. 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 *equal error rate (EER)* is the operating point where the rate of FA is equal to the rate of FR. 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.

Keystroke dynamics has been widely studied for physical keyboards on a PC (see Banerjee and Woodard [4]). Features that are derived from physical keyboards include the key hold and inter-stroke intervals. 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 75–90%) [13, 27, 34]. Note that the accuracy numbers provided in the literature are not directly comparable since different schemes use different numbers of keystrokes to calculate the authentication score.

Modern smartphones use soft keyboards, a software-based keyboard rendered on the touchscreen. In addition to temporal features, researchers have employed contact features, like touch pressure and area, to improve the accuracy [17, 20]. Giuffrida et al. [24] 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. [9] proposed spatial features and showed that they reduced EER by up to 23% when compared to temporal features only. Their scheme employs the most extensive feature set. For a single bigram resulting from two keystrokes (K_1 and K_2), it constructs the 24 features listed in Table 1 (also see a subset illustrated in Figure 3). Their evaluation on data from 28 users against a zero-effort attacker model using an SVM classifier achieved 3.3% EER. They showed that spatial features outperform temporal and contact features. They

Table 1: Buschek et al.'s keystroke features. K_1 and K_2 are two keys of a bigram. ↓ indicates key press; ↑ indicates key release; ^{t,s,c} indicate temporal, spatial, and contact features, respectively.

Feature(s)	Description
key hold interval^t	interval between $\downarrow K_1$ and $\uparrow K_1$
inter-stroke interval^t	interval between $\uparrow K_1$ and $\downarrow K_2$
up-up^t	interval between $\uparrow K_1$ and $\uparrow K_2$
down-down^t	interval between $\downarrow K_1$ and $\downarrow K_2$
down & up pressure^c	touch pressure at $\downarrow K_1$ and $\uparrow K_1$
down & up area^c	touch area at $\downarrow K_1$ and $\uparrow K_1$
down & up axis^c	ellipses axis at $\downarrow K_1$ and $\uparrow K_1$
down x & y^s	x & y coordinate at $\downarrow K_1$
up x & y^s	x & y coordinate at $\uparrow K_1$
offset x & y^s	tap offset x & y from key centre
jump x & y^s	x & y distance between K_1 and K_2
drag x & y^s	x & y drag between $\downarrow K_1$ and $\uparrow K_1$
jump & drag angles^s	jump & drag angles
jump & drag dist.^{s}	jump & drag distances

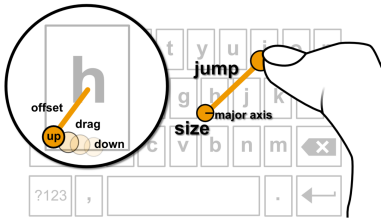


Figure 3: A subset of keystroke features for the bigram "hi". (Illustration from Buschek et al.)

demonstrated their scheme's accuracy superiority by comparing with proposals for physical [13] and soft keyboards [17]. 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.

4.2 Threat Model and Attack

We use the same threat model as Buschek et al. An adversary attempts to gain 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) or for user-to-website authentication (e.g., banking). With password hardening, in case the password is compromised, the difference in typing behaviour of the attacker is expected to flag the compromise.

The adversary has shoulder surfed the password or obtained it through known mechanisms [37]. Furthermore, the adversary is aware that the victim's device is protected using keystroke dynamics. In order to mount a mimicry attack, an attacker must know the typing behaviour of the victim. Tey et al. [48] argue an attacker may learn this behaviour using a compromised biometric database or key loggers. For smartphones, a cloned banking or email app may also serve the purpose [23]. In addition, malicious insiders (i.e., family, friends, and colleagues) may gain access to raw keystroke data of their victims through various techniques. They can ask their victims to type carefully crafted text (e.g., take a note or type a URL) on the attacker's device, which contains the same bigrams as the victim's password. This method eliminates the need to install anything on victims' devices. Malicious insiders can also recommend an instrumented app from the official app store, which in turn collects and transmits raw keystroke data generated in the app to the attacker [37, 54]. Once the insiders obtain raw keystroke data, they can use it to train and mimic their victims' behaviour.

We note that while a malicious insider may easily collect the typing behaviour, other adversaries are required to mount targeted social engineering attacks to collect it. Social engineering attacks from strangers may not always be successful, thereby limiting the potential impact of these attacks. We also 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 schemes (e.g., composing a formal email vs. texting friends) [28, 29]. This improves an attacker's chance of success. Therefore, our proposed attack is effective against both scenarios.

4.3 Data Collection

We need a keystroke dynamics dataset for our evaluations. Buschek et al. made their dataset publicly available. We were interested in using it as victims' data for our experiment; however, it was collected using a custom Android keyboard. We contacted the authors but were unable to obtain their keyboard layout. Therefore, we use their dataset to analyze features and collect an independent dataset for evaluating our attacks. Note that we received approval from our university's IRB for all experiments involving human participants.

4.3.1 Data collection setup. Similar to Buschek et al., we implemented an Android keyboard and an app 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, at both key press and key release events, we logged the key, the timestamp in milliseconds, the x and y coordinates, the touch pressure and the touch area, 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).

Buschek et al. evaluated passwords of different strengths. We chose a subset of passwords from their set to validate our dataset's accuracy against theirs and to limit the study session length. We chose a 6-character dictionary password ("monkey"), an 8-character dictionary password ("password") and a 6-character complex password ("Igur39"). Users were asked to enter each password 20 times. If a user made a mistake during the password entry, they were prompted to repeat the entry.

4.3.2 Statistics of collected data. We approached graduate students through word-of-mouth for data collection. The necessary condition for participation was that the participant owned an Android device for over six months. An LG Nexus 5 device was used to collect data in a lab setup. Participants were asked to enter the password in their natural way (i.e., both thumbs or either thumb

Table 2: Evaluation of Buschek et al's approach against collected data 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

or either index finger to enter the passwords). Participants were allowed to take breaks, but were not allowed to change the posture.

We collected data from 18 participants (8 females): six were between 21–30 years old and the rest were between 31–40 years old. All participants entered passwords using both thumbs. We collected 304 keystrokes across 27 unique bigrams per subject ("Enter" and "Shift" were considered a part of bigram). In total, we collected 12,240 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 (SD = 40ms). The slower cluster had eight victims and an average speed of 521ms per character (SD = 74ms).

4.3.3 Evaluation of keystroke dynamics on collected data. 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 false accept rate (FAR) and true accept rate (TAR). FAR is the proportion of the adversary's attempts that are incorrectly classified as those from a legitimate user and TAR is the proportion of the legitimate user's attempts that are incorrectly classified as those from an adversary. 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)) [12]. If the binary outcome of at least 60% bigrams of the password was "accept", 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. 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). 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.

4.4 Analyzing Keystroke Behaviour

Buschek et al.'s approach captures keystroke behaviour through 24 features. However, mimicking 24 features against each keystroke on smartphones is challenging. To reduce this complexity, we analyzed Buschek et al.'s dataset to investigate keystroke behaviour at the feature level. We withheld our dataset from this analysis to ensure that our findings are evaluated on a non-overlapping dataset. The goal of our analysis was to identify highly correlated features and the key specific or key independent nature of features. This analysis helps to reduce features required for training by removing the majority of highly correlated features. Similarly, the identification of key specific or key independent features helps to build cleaner interfaces for the one time training of the key independent features. The feature analysis details are not a required reading and readers may jump to the findings of this analysis in § 4.4.4.

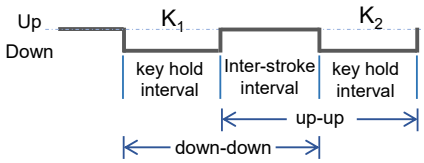


Figure 4: Relationship among temporal features.

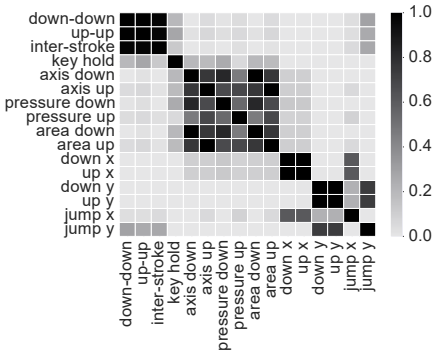


Figure 5: Correlation matrix for selected features (darker colour indicates larger correlation coefficient).

4.4.1 Temporal features. Figure 4 shows the relationship among temporal features. Up-up for K_2 comprises of the inter-stroke interval between K_1 and K_2 and the key hold interval of K_2. Similar composition exists for down-down. To identify redundant features, in Figure 5, we plot the correlation coefficients (an absolute value between −1 and +1, where a higher value indicates a strong positive correlation and vice versa) against all pairs of the four temporal features. It shows a strong positive 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 positive correlation with down-down and up-up. This suggests that both up-up and down-down are redundant for training.

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

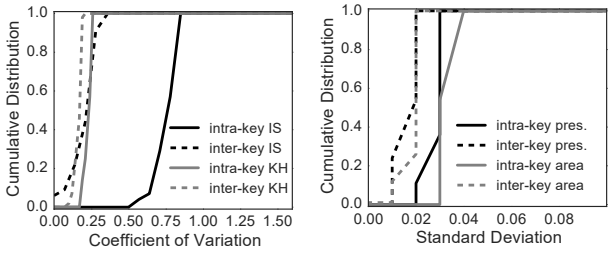


Figure 6: Intra- and inter-key differences: (a) For inter-stroke (IS) and key hold (KH) intervals. (b) For touch pressure (pres.) and area at key press.

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 6a shows that the C_V difference between intra-key and inter-key inter-stroke intervals always exceeds 0.35. On the other hand, 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 mimic only one key hold interval for a victim.

4.4.2 Contact features. Figure 5 shows that a strong positive linear relationship exists between feature values at key press and key release events for touch area and ellipses axis (≥ 0.75). For the touch pressure, it shows a moderate positive relationship between the two events (≥ 0.5). Due to this relationship 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 5 also shows a correlation among touch pressure, touch area and ellipses axis. A strong positive linear relationship between touch pressure and touch area for the key down event also exists (≥ 0.75). Similar moderate to strong positive linear relationships exist among touch pressure, touch area and ellipses axis for the key release event (≥ 0.63). Due to the perfect positive 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 6b shows the cumulative distribution of intra- and inter-key standard deviations between touch pressure and touch area for the key press event. It shows small differences between intra- and inter-key scores for these features. The low variation of these features across different keys for individual users allows us to train attackers to mimic one behaviour across different keys for these features for a victim.

4.4.3 Spatial features. Spatial features are inherently key specific due to the unique location of each key (drag features are an exception), 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 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 5 shows a strong positive 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 among their feature set and under a single feature evaluation, these features provide an average EER of 0.45 (SD $<$ 0.01) (compared to an average EER for the remaining features of 0.31 (SD = 0.02)). Therefore, we choose to only train for down x and down y.

4.4.4 Discussion. Our analysis identifies six target features: key hold interval, 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 once for a victim and not for each key.

5 EVALUATION

In this section, we establish a baseline for our evaluations by evaluating the approach of using non-AR training interfaces for mimicry attacks. The baseline evaluations also enable us to understand the limitations of this approach and motivate the need for a guidance method. We then provide details of the experimental setup and evaluation results for the proposed techniques.

5.1 Non-AR Training Interfaces Baseline

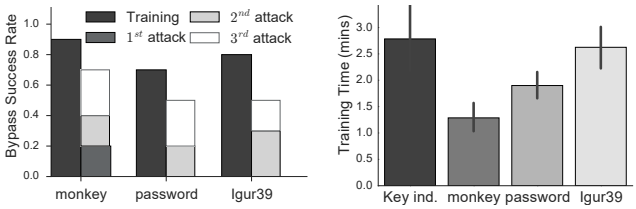
To establish a baseline, similar to Tey et al., we chose to provide sufficient training to attackers so that they could learn the typing behaviour of their victims. During attacks, they were expected to reproduce the typing behaviour from their learning experience (i.e., no guidance was available to the attacker during the attacks). For the rest of this paper, we use the term "unguided attacks" for these attacks. We now provide details on the non-AR training interfaces that we developed.

5.1.1 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, the interface 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 7a shows the training interface for key specific features. The target location is provided as a blue coloured target. The inter-stroke timing is communicated by moving the target to the next

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

Figure 7: 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 colour coded for feedback.



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

Figure 8: Evaluation baseline using unguided attacks. (a) Bypass success. (b) Training effort on key independent and key specific interfaces for successful attacks.

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 7b shows the feedback provided to the attacker. After a mimicry attempt, all targets and the attacker's attempts are displayed 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 one (i.e., go slow), whereas attempts coded green suggest decreasing the interval. Attempts coded light blue indicate that no change is required. The attacker is also able to see the outcome of the SVM classifier (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.

5.1.2 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 (§ 5.2.6). We asked attackers to undergo training on an 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 victim 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. 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.

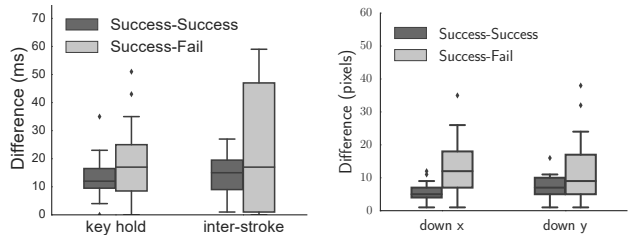
Each participant was asked to mimic the three passwords (in order from weakest to strongest) from a single victim. We chose the top five fastest and top five slowest typists from our 18 victims' dataset and randomly assigned them to the participants. Participants were allowed to spend up to ten minutes to train for a password. If they failed to complete training in ten minutes, 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.

5.1.3 Participant recruitment and motivation. We used our university's graduate students mailing list to recruit participants for a 30-minute session. We restricted participation to those who owned an Android device for over six months. Ten people participated (six females): five were between 21–30 years old and the rest were between 31–40 years old. To motivate participants, we offered a performance-based reward in addition to $5 for participation. For a successful mimicry, we offered $1 reward for "monkey" and $2 each for "password" and "Igur39".

5.1.4 Results for unguided attacks. We report on attackers' success and the amount of effort required to mount successful attacks. Figure 8a shows the overall bypass success rate of attacks. Since attackers retrained in case of a failure during attacks, we show their success separately for different attempts. We note that while the bypass success rate may be over-reported due to the non-zero FAR, our chosen threshold has very low FAR for the zero-effort attacker model.

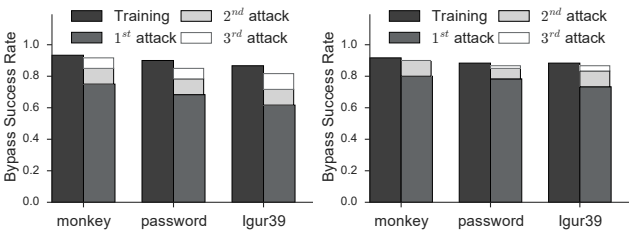
Figure 8a shows that 90% of attackers used the training interface to mimic (six times consecutively) "monkey". 70% and 80% of attackers were also able to mimic "password" and "Igur39", respectively. However, the majority of successful attackers failed to mimic passwords in their first attempt for the attacks. Only two attackers were able to correctly mimic "monkey" in their first attempt. 20–30% of the attackers were able to mimic these passwords in their second attempt and another 20–30% required three attempts. Finally, 20%, 20% and 30% of the attackers were able to complete training but unable to mimic their victim in three attempts for "monkey", "password" and "Igur39", respectively. z-tests (Bonferroni corrected) on the binary outcome of bypass success for passwords of different strengths indicate no significant differences *(*all $p > 0.05$*)*.

Figure 8b shows 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 (min = 1.5; max = 4.4; SD = 1) to learn to mimic key independent features. For "monkey", the successful attackers spent an average of 1.3 minutes (min = 0.8; max = 2.1; SD = 0.4) on training. The successful attackers spent an average



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

Figure 9: 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 failed attacks.



(a) Augmented reality. (b) Audiovisual.

Figure 10: Overall bypass success for proposed techniques.

of 1.9 minutes (min = 1.6; max = 2.4; SD = 0.3) and 2.6 minutes (min = 1.9; max = 3.4; SD = 0.60) on training for "password" and "Igur39", respectively. We conduct t-tests (Bonferroni corrected) to compare the attacker's effort for passwords of different strengths. Only the t-test comparing the attacker's effort between "monkey" and "Igur39" shows significant differences *($p < 0.01$)*.

5.1.5 Discussion. Our experiment shows that while more than 70% of the attackers were able to successfully mimic their victim during training, 29% of the successful trainees failed to mimic during the attacks. To investigate the high rate of failure during attacks, 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 9. For comparison, we also plot the difference between the attackers' mimicked values for the last two successful training attempts ("Success-Success").

Figure 9a shows the differences for the two temporal target features. For the inter-stroke interval, it shows an average difference of 15ms (SD = 7ms) for Success-Success, but 24ms (SD = 44ms) for Success-Fail. Furthermore, 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. A similar trend is observed for the key hold interval (Figure 9a), the two spatial target features (Figure 9b), and down area (results omitted due to space constraints) but not for down pressure. Our analysis suggests that the performance gets worse during attack attempts in the absence of guidance for all features except down pressure. These results show that in the absence of guidance, the margin of error for attackers increases despite the training.

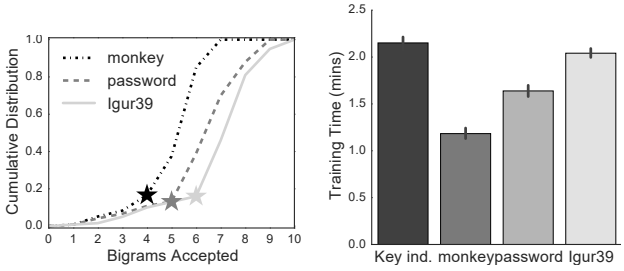
5.2 Evaluation of AR Techniques

We now evaluate the efficacy of our AR techniques to provide real-time mimicry guidance during the attacks.

5.2.1 Attack protocol and study procedure. The recruited participants were invited to our lab and were briefed about the setup. 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. We followed the same procedure for attackers' data collection and training as for the unguided attacks (§ 5.1.2). However, during attacks, attackers were provided real-time guidance through one of the guidance methods. For the AR-based method, we used a Google Pixel device to provide guidance. For the audiovisual method, we rendered the spatial cues on the victim's device through an app. The rendered cues were the same as the cues used for training. Due to time constraints, rendering cues was favored over printing them on a screen protector or drawing them with an erasable marker. However, since an attacker could print or draw cues that are visually similar to the rendered cues, this minor deviation is unlikely to affect our research findings. Each participant was asked to mimic the three passwords (weakest to strongest) 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.

5.2.2 Participant recruitment and motivation. The recruitment procedure and restrictions were the same as for the unguided attacks (§ 5.1.3). 30 people participated in this experiment (16 females): 20 were between 21–30 years old and the rest were between 31–40 years old. 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 participation, 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, they received the remaining $4.

5.2.3 Attacker's success. We report results against 360 attacks (30 attackers x 2 victims x 2 guidance methods x 3 passwords) and cover only the first round. (See § 5.2.5 for the second round results.) 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.



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

Figure 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.

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 10a) and audiovisual method (Figure 10b), respectively. A Fisher's Exact test on the binary outcome of the attack for the two methods indicates no significant differences *($p > 0.05$)*. 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 Fisher's Exact test on the binary outcome of the first attack attempt indicates significant differences between the two methods *($p = 0.04$)*. 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 11a, 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 11a 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 for 62%, 60%, and 53% of all attacks, the attackers were able to successfully mimic 80% or more bigrams for "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).

5.2.4 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 (min = 2; max = 4.6; SD = 0.9).

Figure 11b 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 3.3–4.1 minutes to complete their training for different passwords. t-tests (Bonferroni corrected) between the attacker's effort for different passwords show significant differences between all three pairs *(*all $p < 0.001$*)*.

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

We use t-tests (Bonferroni corrected) to compare the attacker's effort between unguided and guided attacks. A t-test for the attacker's effort to train the key independent features between unguided (avg = 2.8 minutes) and guided attacks (avg = 2.1 minutes) indicates significant differences *($p < 0.01$)*. t-tests also indicate significant differences *(*all $p < 0.001$*)* for the attacker's effort to train "password" and "Igur39" between unguided (avg = 1.9 and 2.6 minutes, respectively) and guided attacks (avg = 1.6 and 2 minutes, respectively). While the attackers used the same training interfaces as in the unguided attacks, the reduction in the training time is a result of the higher success rate for the first attempt (thereby requiring fewer retrainings.)

5.2.5 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 7/19 attacks they successfully adapted their behaviour with practice. We also note that for 12/19 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 *($p > 0.05$)*. These results suggest the possibility that mimicry success is affected by the typing proficiency of attackers or victims (examined further in § 5.2.6).

5.2.6 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 Fisher's Exact test on the binary outcome of the attack for the two typing speeds indicates no statistically significant differences *($p > 0.05$)*.

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 Fisher's Exact test on the binary outcome of the attack and between the two typing speed groups. A Fisher's Exact test indicates statistically significant differences between the two groups of attackers in terms of their success *($p < 0.05$)*. These results show that attackers with faster typing speeds are more likely to mount a successful attack than attackers with slower typing speeds.

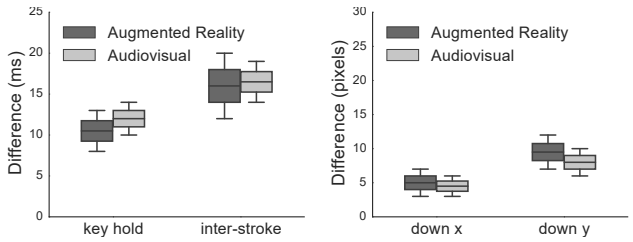
6 DISCUSSION

In this section, we discuss how effective the AR guidance techniques were compared to the evaluation baseline (i.e., unguided attacks) and to provide guidance against different features. We ground this discussion in additional analysis of our attack data. We then discuss how the proposed techniques can be extended to attack other behaviour-based biometrics.

6.1 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 the first successful attack attempt with each technique in Figure 12. Figure 12a shows that on average the AR-based approach enabled attackers to mimic key hold intervals within 10ms (SD = 3ms) of the target values. The audiovisual approach enabled attackers to mimic key hold intervals within 14ms (SD = 5ms) of the target values. Similarly, the AR and audiovisual approaches enabled attackers to mimic the inter-stroke



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

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

intervals within 16ms (SD = 6ms) and 23ms (SD = 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 ($p < 0.05$) for the difference of target and mimicked values for both features between the two approaches. This indicates that the AR approach enabled the attackers to more accurately mimic the temporal features.

Figure 12b shows that on average the AR and audiovisual approaches enabled attackers to mimic the down x feature within 5 pixels (SD = 3 pixels) and 4 pixels (SD = 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 (SD = 5 pixels) and 8 pixels (SD = 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 no significant differences ($p < 0.05$) for the difference of target and mimicked values for both features between the two approaches.

6.2 Extending our Attack Techniques

Several input behaviour-based authentication proposals (e.g., those discussed in § 4.1) are presumed secure with the assumption that the attacker is unable to steal and 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. In the wake of a recent attack on Android's Pattern Lock [56], where computer vision was used to process the video footage to crack the authentication secret, the "unstealable" property of input behaviour-based biometrics is questionable. Similarly, our findings strongly suggest that security researchers must consider the threat posed by novel methods that precisely reproduce the behaviour of the user.

While the audiovisual approach is limited by static visuals, our AR technique can be easily extended to attack swiping behaviour-based authentication schemes [22, 33, 53]. Among other features, these schemes rely on the location, curvature and speed of the swipe. These features can be easily depicted using a cursor similar to the one used by our AR prototype. Touchscreen gesture-based approaches that rely on similar features [44] may potentially be mimicked if the swiping behaviour is known. A limitation of the current AR prototype is its inability to communicate features like pressure. However, Serwedda et al. [43] demonstrated that a large number of users are clustered around a narrow band of values. One

possibility is to mimic these values by suggesting various levels of the target pressure (e.g., low, medium or high pressure). More experiments need to be conducted to determine the viability of this approach.

Schemes that rely on how users shake or wave their device [26, 55, 58] are also susceptible to AR-based attacks. Anderson et al. [2] 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 [6]) might be extended to attack gait behaviour-based authentication systems [8, 31].

7 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.

We used the same device for data collection and attacks. If an attacker collects data using a device with a different form factor than the victim's, appropriate transformation would need to be applied to the features. Furthermore, we used the same device for training and attacks. If the training device has a different form factor or manufacturer than the attack device, the attacker may not be able to reproduce the trained behaviour due to the underlying differences in device sizes or the pressure values returned by different manufacturers.

We chose a subset of passwords from Buschek et al.'s work. This allowed us to validate our dataset's accuracy against theirs and limit the study session length (to keep attackers motivated and avoid fatigue). While our experiments do not investigate more complex passwords, Zezschwitz et al. found that simple passwords are more widely used on smartphones due to their ease of input [50]. Despite this limitation, our work calls attention to the well-established assumption that weaker passwords can be hardened using keystroke dynamics.

We allowed participants to spend up to ten minutes to train for a given password. This ensured that attackers did not get overtired and provided 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 sessions. However, Buschek et al. showed that the classifier's accuracy reduces for datasets collected across sessions. This made our attacks more challenging for the attackers.

The number of participants for the unguided attacks may seem low; however, the data collected was sufficient to draw statistically significant conclusions on the efficacy of the training interfaces, and motivate the need for attack guidance mechanisms.

8 CONCLUSION AND FUTURE WORK

We present AR smartphone methods that provide real-time input guidance to enable an attacker to attack behaviour-based authentication systems. We evaluate our methods on keystroke dynamics to demonstrate that they enable an attacker to precisely mimic multiple behavioural features at a milliseconds resolution. Our methods only require an off-the-shelf smartphone and do not require tampering with the software on the victim's device. Potential future work includes an evaluation of our AR-based method for mimicry attacks on other input behaviour-based systems. Another interesting research avenue that will result in wider applicability of these attacks is exploring advancements in computer vision to obtain the input behaviour of users through their video footage.

9 ACKNOWLEDGEMENTS

Thanks to the anonymous reviewers and our shepherd, Mary Baker, for their valuable comments. We gratefully acknowledge the support of NSERC for grants RGPIN-2014-05499 and Discovery Grant 2018-05187.

REFERENCES

[1] Asadullah Al Galib and Reihaneh Safavi-Naini. 2014. User Authentication Using Human Cognitive Abilities. In *18th International Conference on Financial Cryptography and Data Security*.
[2] Fraser Anderson, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2013. YouMove: Enhancing Movement Training with an Augmented Reality Mirror. In *26th Annual ACM Symposium on User Interface Software and Technology*. ACM.
[3] Ronald T Azuma. 1997. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments* 6, 4 (1997), 355–385.
[4] Salil P Banerjee and Damon L Woodard. 2012. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research* 7, 1 (2012), 116–139.
[5] BehavioSec. 2017. A supplement to Authentication in an Internet Banking Environment. https://www.behaviosec.com/financial-services/. (June 2017). Last accessed: 08/2017.
[6] Karissa Bell. 2017. New ARKit iPhone app will help your learn to be a better dancer. (July 2017). https://mashable.com/2017/07/09/dance-reality-arkit-app
[7] Mark Billinghurst, Adrian Clark, Gun Lee, et al. 2015. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction* 8, 2-3 (2015), 73–272.
[8] Matthew Boyle, Avraham Klausner, David Starobinski, Ari Trachtenberg, and Hongchang Wu. 2011. Poster: Gait-based Smartphone User Identification. In *9th International Conference on Mobile Systems, Applications, and Services*. ACM.
[9] Daniel Buschek, Alexander De Luca, and Florian Alt. 2015. Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices. In *33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM.
[10] Ozan Cakmakci, François Bérard, and Joëlle Coutaz. 2003. An augmented reality based learning assistant for electric bass guitar. In *10th International Conference on Human-Computer Interaction*.
[11] 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 *SIGCHI Conference on Human Factors in Computing Systems*. ACM.
[12] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 3 (2011), 27.
[13] Nathan L Clarke and SM Furnell. 2007. Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security* 6, 1 (2007), 1–14.
[14] Alexander De Luca, Alina Hang, Frederik Brudy, Christian Lindner, and Heinrich Hussmann. 2012. Touch me once and I know it's you!: implicit authentication based on touch screen patterns. In *Annual Conference on Human Factors in Computing Systems*. ACM.
[15] Michael B Dillencourt, Hanan Samet, and Markku Tamminen. 1992. A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM (JACM)* 39, 2 (1992), 253–280.
[16] 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: The International Journal for Geographic Information and Geovisualization* 10, 2 (1973), 112–122.
[17] 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.
[18] Simon Eberz, Kasper B Rasmussen, Vincent Lenders, and Ivan Martinovic. 2015. Preventing Lunchtime Attacks: Fighting Insider Threats With Eye Movement Biometrics. In *22nd Network and Distributed System Security Symposium*.

[19] 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 *35th Annual ACM Conference on Human Factors in Computing Systems*. ACM.

[20] Tao Feng, Xi Zhao, Bogdan Carbunar, and Weidong Shi. 2013. Continuous Mobile Authentication Using Virtual Key Typing Biometrics. In *12th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE.

[21] Jon Fingas. 2018. Android's official augmented reality toolkit is available to the public. (Feb. 2018). https://www.engadget.com/2018/02/23/google-launches-arcore-1-0/

[22] 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 Transactions on Information Forensics and Security* 8, 1 (2013), 136–148.

[23] Clint Gibler, Ryan Stevens, Jonathan Crussell, Hao Chen, Hui Zang, and Heesook Choi. 2013. Adrob: Examining the landscape and impact of android application plagiarism. In *11th Annual International Conference on Mobile Systems, Applications, and Services*. ACM.

[24] 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.

[25] Tobias Höllerer and Steve Feiner. 2004. Mobile augmented reality. *Telegeoinformatics: Location-Based Computing and Services. Taylor and Francis Books Ltd., London, UK* 21 (2004).

[26] Feng Hong, Meiyu Wei, Shujuan You, Yuan Feng, and Zhongwen Guo. 2015. Waving Authentication: Your Smartphone Authenticate You on Motion Gesture. In *33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM.

[27] Seong-seob Hwang, Sungzoon Cho, and Sunghoon Park. 2009. Keystroke dynamics-based authentication for mobile devices. *Computers & Security* 28, 1 (2009), 85–93.

[28] Hassan Khan, Aaron Atwater, and Urs Hengartner. 2014. Itus: an implicit authentication framework for Android. In *20th Annual International Conference on Mobile Computing & Networking*. ACM.

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

[30] Hassan Khan, Urs Hengartner, and Daniel Vogel. 2016. Targeted Mimicry Attacks on Touch Input Based Implicit Authentication Schemes. In *14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM.

[31] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. 2010. Cell phone-based biometric identification. In *4th IEEE International Conference on Biometrics: Theory Applications and Systems*. IEEE.

[32] Kiron Lebeck, Kimberly Ruth, Tadayoshi Kohno, and Franziska Roesner. 2017. Securing augmented reality output. In *IEEE Symposium on Security and Privacy*. IEEE.

[33] Lingjun Li, Xinxin Zhao, and Guoliang Xue. 2013. Unobservable reauthentication for smart phones. In *20th Network and Distributed System Security Symposium*.

[34] Emanuele Maiorana, Patrizio Campisi, Noelia González-Carballo, and Alessandro Neri. 2011. Keystroke dynamics authentication for mobile phones. In *Symposium on Applied Computing*. ACM.

[35] Caitlin McGarry. 2018. What Is Apple's ARKit? (2018). https://www.tomsguide.com/us/apple-arkit-faq,review-4636.html

[36] Richard McPherson, Suman Jana, and Vitaly Shmatikov. 2015. No escape from reality: security and privacy of augmented reality browsers. In *24th International Conference on World Wide Web*.

[37] Emiliano Miluzzo, Alexander Varshavsky, Suhrid Balakrishnan, and Romit Roy Choudhury. 2012. Tapprints: your finger taps have fingerprints. In *10th International Conference on Mobile Systems, Applications, and Services*. ACM.

[38] Parimarjan Negi, Prafull Sharma, Vivek Jain, and Bahman Bahmani. 2018. K-means++ vs. Behavioral Biometrics: One Loop to Rule Them All. In *25th Network and Distributed System Security Symposium*.

[39] Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9, 1 (1979), 62–66.

[40] Franziska Roesner, Tadayoshi Kohno, and David Molnar. 2014. Security and privacy for augmented reality systems. *Commun. ACM* 57, 4 (2014), 88–96.

[41] Marc Ericson C Santos, Angie Chen, Takafumi Taketomi, Goshiro Yamamoto, Jun Miyazaki, and Hirokazu Kato. 2014. Augmented reality learning experiences: Survey of prototype design and evaluation. *IEEE Transactions on Learning Technologies* 7, 1 (2014), 38–56.

[42] Abdul Serwadda and Vir V Phoha. 2013. Examining a large keystroke biometrics dataset for statistical-attack openings. *ACM Transactions on Information and System Security (TISSEC)* 16, 2 (2013), 8.

[43] Abdul Serwadda and Vir V Phoha. 2013. When kids' toys breach mobile phone security. In *ACM SIGSAC Conference on Computer & Communications Security*. ACM.

[44] 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 *19th Annual International Conference on Mobile Computing & Networking*. ACM.

[45] Michael Sherman, Gradeigh Clark, Yulong Yang, Shridatt Sugrim, Arttu Modig, Janne Lindqvist, Antti Oulasvirta, and Teemu Roos. 2014. User-generated Free-form Gestures for Authentication: Security and Memorability. In *12th Annual International Conference on Mobile Systems, Applications, and Services*. ACM.

[46] Ivan E Sutherland. 1968. A head-mounted three dimensional display. In *Fall Joint Computer Conference, Part I*. ACM.

[47] Robert Templeman, Mohammed Korayem, David J Crandall, and Apu Kapadia. 2014. PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces.. In *21st Network and Distributed Security Symposium*.

[48] Chee Meng Tey, Payas Gupta, and Debin GAO. 2013. I can be You: Questioning the use of Keystroke Dynamics as Biometrics. 20th Network and Distributed System Security Symposium.

[49] DWF Van Krevelen and Ronald Poelman. 2010. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality* 9, 2 (2010), 1.

[50] Emanuel Von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. 2014. Honey, I shrunk the keys: influences of mobile devices on password composition and authentication performance. In *8th Nordic Conference on Human-Computer Interaction*. ACM.

[51] Matthias Weing, Amrei Röhlig, Katja Rogers, Jan Gugenheimer, Florian Schaub, Bastian Könings, Enrico Rukzio, and Michael Weber. 2013. PIANO: enhancing instrument learning via interactive projected augmentation. In *ACM Conference on Pervasive and Ubiquitous Computing*. ACM.

[52] Christian Winkler, Jan Gugenheimer, Alexander De Luca, Gabriel Haas, Philipp Speidel, David Dobbelstein, and Enrico Rukzio. 2015. Glass unlock: Enhancing security of smartphone unlocking through leveraging a private near-eye display. In *33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM.

[53] Hui Xu, Yangfan Zhou, and Michael R Lyu. 2014. Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones. In *Symposium On Usable Privacy and Security*. Usenix.

[54] Zhi Xu, Kun Bai, and Sencun Zhu. 2012. TapLogger: Inferring User Inputs on Smartphone Touchscreens Using On-board Motion Sensors. In *5th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM.

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

[56] Guixin Ye, Zhanyong Tang, Dingyi Fang, Xiaojiang Chen, Kwang In Kim, Ben Taylor, and Zheng Wang. 2017. Cracking Android pattern lock in five attempts. In *24th Network and Distributed System Security Symposium*.

[57] Nan Zheng, Kun Bai, Hai Huang, and Haining Wang. 2014. You are how you touch: User verification on smartphones via tapping behaviors. In *22nd International Conference on Network Protocols*. IEEE.

[58] Hongzi Zhu, Jingmei Hu, Shan Chang, and Li Lu. 2017. ShakeIn: Secure User Authentication of Smartphones with Single-Handed Shakes. *IEEE Transactions on Mobile Computing* 16, 10 (2017), 2901–2912.