INTERNATIONAL JOURNAL OF HUMAN-COMPUTER INTERACTION
https://doi.org/10.1080/10447318.2022.2049144

PushPIN: A Pressure-Based Behavioral Biometric Authentication System for Smartwatches

Youngeun Song[a] and Ian Oakley[b]

[a]Department of Biomedical Engineering, Ulsan National Institute of Science and Technology, Ulsan, Republic of Korea; [b]Department of Design, Ulsan National Institute of Science and Technology, Ulsan, Republic of Korea

ABSTRACT

Smartwatches support diverse applications but suffer from security issues due to their limited resources; their small size poorly supports the rich, accurate input required for screen lock authentication. Additionally, traditional approaches to unlocking smart devices, such as Personal identification number, are highly susceptible to attacks such as guessing and video observation. Therefore, we propose PushPIN, a novel scheme that combines knowledge-based and behavioral biometric approaches to increase security. Input symbols are composed of the selection of one of four different targets with one of five different pressure levels, for a total of 20 possibilities. We complement this passcode by capturing behavioral biometric features from screen touches and wrist motion during input. We present two studies to assess the performance of PushPIN. The first assesses both usability and security against a random guessing attack. It shows acceptable usability—recall times of approximately 8 s and no errors—and strong security: equal error rates of 0.51%. The second study examines the resistance of PushPIN against a video observation attack, ultimately revealing that 36.67% of PushPINs could be cracked, performance that represents a substantial improvement over prior work on pressure-based authentication input. We conclude that pressure-based input can increase the security, while maintaining reasonable usability, of smartwatch lock systems.

1. Introduction

The increasing power and sophistication of smartwatches mean they are now capable of supporting a diverse set of applications and features including: performing fitness tracking (Adapa et al., 2018); bio-signal monitoring and logging; accessing and presenting personal messages and data; making and receiving phone calls and; purchasing and other real-world transactions (Nguyen & Memon, 2018). While these services provide substantial value, they also raise security concerns as they involve collecting and retaining private user data or providing access to sensitive services, such as payments. As a result, multiple security technologies and systems (e.g., user access controls, encryption) need to be integrated into new smartwatch platforms (Fortify, 2015).

One fundamental challenge in maintaining device security is ensuring only the legitimate user has access. To achieve this, early authentication systems on smartwatches mainly relied on secure authentication to a paired smartphone. However, as modern smartwatches can perform many functions independently, this approach is no longer sufficient. In this context, it is critical to develop and deploy unlock systems that can be implemented and effectively operated on standalone smartwatches. This is currently achieved by techniques such as Personal Identification Number (PIN), Android Pattern Lock (APL) (Nguyen & Memon, 2017), or biometrics (Buriro et al., 2018; Li & Xie, 2018; Lu et al., 2017; Nguyen & Memon, 2018). However, we note that while such systems are critical, they are unevenly deployed on smartwatches. In 2015, for example, only five of the top 10 most popular smartwatches provided a basic lock system (Fortify, 2015). Furthermore, the dominant lock methods remain PIN or APL (Fortify, 2015), approaches with well-documented weaknesses to various primary attacks including guessing and video observation (Nguyen & Memon, 2018). Smartwatches are susceptible to these attacks after they are removed—an event reported to happen an average of 3.17 times per day (Jeong et al., 2017; Li & Xie, 2018)—and then mislaid, forgotten, or stolen. Furthermore, the small size of smartwatch touchscreens (typically 30mm square) means that performance of traditional smartphone lock input tasks may be impaired by the fat-finger problem (Siek et al., 2005)—the fact that a user's finger will obscure screen contents during input, potentially reducing accuracy and increasing task time.

To address these problems, researchers have begun to explore lock systems explicitly designed for the smartwatch form factor. In terms of usability, the Personal Identification Chord (PIC) (Oakley et al., 2018) sought to address the problems inherent in small smartwatch screens by designing

CONTACT Ian Oakley ian.r.oakley@gmail.com Department of Biomedical Engineering, Ulsan National Institute of Science and Technology, Ulsan, Republic of Korea

a chorded lock system based on four large buttons that can be pressed either individually or in pairs (for a total of 10 possible unique inputs). Similarly, Press Touch Code (PTC) (Ranak et al., 2017) targeted a similar usability problem with the design of a pressure-sensitive lock system based on counting the peaks in intentionally fluctuating pressure values during a single sustained screen touch. A key goal of these systems is to use alternative input channels (e.g., chords, pressure) to increase the number of symbols that can be generated by touches to a small number of large, readily accessible on-screen targets. There have also been complementary efforts to increase the security of existing smartwatch unlock systems. One area of focus has been to increase resistance against guessing and observation attacks by, for example, combining entry of a specific sequence of symbols with detailed behavioral data captured during the input process. TapMeIn (Nguyen & Memon, 2018) exemplifies this approach by combining a temporal sequence of finger taps with features derived from the smartwatch motions they cause; attackers found this input hard to replicate, even given detailed videos of authentication sessions. Similarly, Li and Xie (2018) verified users based on a combination of explicit gestures created by wrist motions and a set of behavioral features calculated from this input.

Building on these ideas, this paper explores behavioral biometrics derived from touch screen and wrist motion data captured during pressure input as a technique to improve unlock systems on a smartwatch. We argue pressure input is an interesting and worthwhile modality to study for several reasons. First and foremost, compared to an input made by physically tapping different targets, pressure input is highly unobtrusive. As such it may increase resistance to observation attack via shoulder surfing or recorded videos (Krombholz et al., 2016). Second, pressure input requires limited screen space—many different pressure values can be specified with a single touch to a single location. As such it is suitable for a wide variety of device form factors, including those with small screens such as smartwatches (Ranak et al., 2017). Lastly, pressure-based input can combine explicit symbolic passcodes (Krombholz et al., 2016; Ranak et al., 2017) with data on the patterns of movements and forces users generate while producing various pressure levels. We argue this data has the potential to serve as a novel behavioral biometric. Reflecting these goals, this paper presents *PushPIN*, a novel multi-factor authentication system for smartwatches. It combines knowledge-based authentication using five-force-level input (rather than binary pressure (Krombholz et al., 2016)) on a 4-key interface, with a behavioral biometric authentication process based on features from wrist motion and screen touch data. It seeks to reduce the impact of fat-finger problems through the use of a reduced number of relatively large on-screen targets (Oakley et al., 2018). It also increases the size of the available password space through the inclusion of multi-level pressure. Finally, it seeks to increase resistance to video observation attacks by integrating pressure input based behavioral biometrics during input passcode entry.

We present a comprehensive analysis of the usability and security of PushPIN in two studies. In the first study ($N = 30$), we collect PushPIN lock codes and input events to measure time and accuracy in set up and recall tasks and to build recognizers for user verification of smartwatch. In the second study ($N = 10$), participants performed a video observation attack on PushPIN users from the first study. The results of the first study indicate PushPIN lock codes achieve acceptable usability with mean set up times of 103.79 second, recall times of $7.99 - 8.07$ second, and very high accuracy—no failed authentication attempts were recorded. Behavioral features derived from this input achieve 0.16% false-positive rate and 7.33% false-negative rate in a simulated random guessing attack, a promising level of performance that suggests that how users perform pressure-based input may be highly unique. This assertion is born out in the video observation attack study. Participants achieve an attack success rate of 36.67%, a substantial improvement over the 97% reported in prior work (Krombholz et al., 2016) on a binary pressure-based PIN system. We conclude that multi-level pressure-based input represents a good solution to increase the security of smartwatch unlock as it provides a rich set of biometric features that can enhance resistance to random guessing and observation attack during a passcode input process that does not unduly burden users: it remains acceptably short and reliable.

2. Related work

There are two modes in which an authentication system can be designed to operate (Teh et al., 2016): identification and verification. The goal of an identification system is to determine which individual in a database is the most similar based on information about the individual. Verification systems, on the other hand, seek to classify a user as genuine or an imposter while only storing data about the genuine user. Due to its relevance to the device unlock scenario, the work in this paper is limited to verification scenarios.

2.1. Knowledge-based authentication on smartwatches

Smartwatches log, store, and provide access to various private user information. This makes the use of lock systems capable of limiting an attacker's access to devices imperative. The dominant approach to this problem involves knowledge based authentication: users verify their identity by entering "something they know," typically a secret code or password. Common smartwatch techniques are adapted from smartphones in the form of PIN and APL. While they are convenient and familiar, these techniques are widely reported to be susceptible to guessing and video observation attacks (Nguyen & Memon, 2018). Furthermore, the small screens of smartwatches may lead to fat-finger problems Siek et al. (2005), which may result in compromised security by reducing the adoption rates of lock systems (Oakley et al., 2018).

To address these issues, several researchers have proposed unlock schemes designed specifically for small watch touch screens. For example, Beat-PIN (Hutchins et al., 2018) and

Table 1. False Positive Rate (FPR, %), False Negative Rate (FNR, %), and Equal Error Rate (EER, %) of random guessing attacks and FPR and EER from observation attacks on existing smartwatch authentication systems based on behavioral biometrics.

Work	FPR (Random)	FNR (Random)	EER (Random)	FPR (Observation)	EER (Observation)
Li & Xie (2018)	0	22	NA	10.6 – 14.6	NA
AirSign (Buriro et al., 2018)	21.65	19.48	NA	NA	NA
VeriNet (Lu et al., 2017)	10.24	20.77	7.17	NA	NA
TapMeIn (Nguyen & Memon, 2018)	0.98	5.3 (Sitting) 9.1 (Walking)	1.3	NA	2.3–4.1

PIC (Oakley et al., 2018) solve the fat finger problem by, respectively, omitting on-screen targets altogether or by presenting a small number of large targets. To support a large set of lock codes, these systems rely on alternative input properties, specifically tapping rhythms and chorded input over multiple buttons. The results of these studies show reasonable usability (e.g., recall times of between 1.7 s (Hutchins et al., 2018) and 3.84 s (Oakley et al., 2018)) and, though studies are small scale, the authors speculate they may also support improved security against vectors such as brute force attack or observation. This work suggests reducing the number of on-screen targets in a smartwatch lock system through the use of a complementary input modality is a viable strategy to maintain usability while also improving security. Accordingly, we adopted this approach in the design of PushPIN.

2.2. Behavioral biometrics on smartwatches

An alternative approach to smartwatch authentication involves biometrics. While traditional physiological approaches, such as fingerprints, are well-established, they are hard to integrate into watch form factors due to the size and cost of the dedicated sensors required (Nguyen & Memon, 2018). To address this practical problem, researchers have proposed captured and analysing aspects of human behavior rather than physical traits. For example, data relating to brain activity (Saulynas et al., 2018), keyboard typing patterns, speech, hand-writing (e.g., signature production), and gait have all been studied as *behavioral biometrics* (Unar et al., 2014) in order to support user authentication. On mobile smart devices, user behaviors captured by widely deployed sensors such as touchscreens (Sae-Bae et al., 2012) or motion sensors (Buriro et al., 2016, 2019; Li et al., 2021) have also been considered as a source of behavioural biometric data to support user authentication. Behavioral biometric authentication systems for smartwatches have also been implemented with pre-existing built-in sensors (e.g., of motion or touch) (Nguyen & Memon, 2018) to track and detect aspects of user behavior that show high differentiation between individuals. Behavioral biometrics have the advantages that they can support both explicit and continuous authentication (Nguyen & Memon, 2018; Teh et al., 2016), and typically show high resistance against attacks such as observation, especially when used in combination with knowledge-based passwords (Nguyen & Memon, 2018).

We present a review of smartwatch behavioral biometric systems used in conjunction with knowledge-based schemes in Table 1. We report on their performance in terms of *False Negative Rate* (FNR, the rate at which a genuine user's attempts to authenticate are rejected), *False Positive Rate* (FPR, the proportion of non-genuine attempts made by, for example, other users or attackers that are accepted) and the *Equal Error Rate* (EER, the point at which a classifier is tuned such that FNR and FPR are equal). EER is typically viewed as representing the middle ground in the trade off between lenient recognition criteria that accept most genuine user authentication attempts at the cost of also accepting some imposter attempts (i.e., low FNR, elevated FPR) and strict authentication criteria that reject most imposters at the cost of rejecting a higher proportion of genuine user attempts (i.e., low FPR, elevated FNR) (Saad & Djedi, 2017). In general, these systems have been assessed by their resistance to random guessing attacks, situations when an attacker has no information about a user's passcode and guesses at essentially chance levels. Resistance to observation attacks has also been examined in some cases (Li & Xie, 2018; Nguyen & Memon, 2018). This work also relies on a diverse set of modalities including general hand/arm gestures (Li & Xie, 2018), mid-air gestures (Buriro et al., 2018), or data captured during traditional input processes such as entering a PIN (Lu et al., 2017). Performance is diverse and reporting of measures is not fully consistent. Regardless, it is clear that achieving strong performance, in terms of low EERs (or the combination of low FPRs and FNRs) is challenging: a majority of articles report scores of 20% or greater on at least one of these metrics. TapMeIn (Nguyen & Memon, 2018) is an exception to this trend that achieves an EER of 0.98% in response to a random guessing attack and between 3.5% and 4.1% for an observation attack. TapMeIn captured various touch features derived from a bespoke form of authentication: a passcode in the form of a rhythmic tapping pattern. This showcases the potential benefits of capturing biometrics from novel touch behaviors that go beyond standard taps. We argue that the more performative qualities of this type of input may increase the salience of the behavioral biometrics that can be extracted. Based on this intuition, we designed PushPIN to explore the value of the behavioral features extracted from both the touch screen (including touch force data) and wrist motion sensors (accelerometer and gyroscope) while users explicitly performed a rich and dynamic input task: generating specific forces during entry of a pressure based secret passcode.

2.3. Authentication via pressure input

Pressure input has a long history in research (Brewster & Hughes, 2009) with authors claiming it is expressive and precisely controllable. Reflecting this promising performance, highly accurate pressure sensors are now available in

consumer PCs and smartphones, while binary pressure sensors have been implemented in smartwatches. We believe pressure input is particularly useful in small form factor devices, as it can provide extra input capabilities without consuming any of the scarce screen real estate (Ranak et al., 2017). Beyond these general points, pressure input may also be specifically useful for authentication scenarios—as making variations in pressure input does not involve large-scale spatial motions, authors have suggested it may be hard to observe and thus provide resistance to shoulder surfing or video attacks (Krombholz et al., 2016).

This assertion has been explored in a variety of prototypes, with mixed results. For example, ForcePIN (Krombholz et al., 2016) is a PIN enhanced with either soft or hard pushes, result in doubling the input space of possible symbols. ForcePIN showed a reasonable recall time of 3.66 second, but low resistance to video observation attack (Khan et al., 2018); this is because attackers observed more forceful touches took longer and simply used touch time as a proxy for pressure, ultimately cracking over 97% of ForcePINs. PTC (Ranak et al., 2017), a system that authenticates users via counting the number of high-pressure peaks users make while intentionally oscillating between lighter and heavier touches, was also highly susceptible to video observation attacks with video taken at 50 cm: the crack rate was 100%. Although this dropped to 5% at 3 m distance, this result indicates that visual cues relating to the shape or motions of the hand and fingers made it clear when users were exerting binary pressures on the screen.

Pressure has also been integrated into touch-based behavioral biometrics. One important area of this work is enhancing the security of traditional lock input. On smartphones De Luca et al. (2012), for example, collected touch-related features including location, size, speed, duration, and pressure during input of APL and achieved an accuracy of 77% during a user verification task. Salem and Obaidat (2019) showed the strongest classification performance in this area—0.9% EER—in a study using 10 features, including pressure, related to keystroke dynamics while typing eight alphanumeric passwords. We argue these initial results suggest that pressure input tasks can yield data that is suitable for use as a behavioral biometric.

The work in this paper operates at the intersection of this literature. It leverages the idea that pressure input can increase the expressivity of touch input on small screen devices to create an authentication system that supports a large set of possible passcodes on a small input surface. The goal is to maintain reasonable performance in terms of usability while increasing resistance to random guessing attacks and observation attacks. In addition, we seek to capitalize on the prior suggestions that pressure input is hard to observe to increase resistance to an observation attack. We do this by examining how effectively the features generated during pressure input can serve as touch and wrist motion based behavioral biometrics. Our intuition here is that, while prior work has demonstrated that attackers can extract data about pressure input given sufficiently high-quality recordings of user tasks (Khan et al., 2018), they may not be similarly able to extract and attack aspects of the detailed performance of pressure input represented by behavioral features. In this way, an approach based on behavioral biometrics may be able to enhance the observational resistance of pressure-based authentication input.

3. PushPIN system design

This section introduces the threat model underlying this work and presents a high-level overview of PushPIN, including details of the user enrollment and verification processes.

3.1. Threat model

PushPIN primarily seeks to increase resistance against various forms of brute force attack including random guessing and content-aware attacks (Li & Xie, 2018). Random guessing attacks are common and simple. They involve an attacker who has acquired a user's device (e.g., via theft) but has no knowledge about the user or password. Accordingly, this type of attack may exploit known biases in, for example, the distribution of symbols users choose for their security codes. In content-aware attacks, attackers possess personal information about the user (e.g., birth date) which can be used to further inform guesses. In addition, PushPIN was designed to resist video observation attacks on a smartwatch. It closely follows the threat model in Nguyen and Memon (2018), as this targets a similar device form factor and scenario. Attackers, including strangers, record videos of a small number of device unlock processes from a distance and examine and analyze these in detail. After subsequently acquiring a user's device, they attempt to mimic user behavior during authentication to unlock it. In line with Krombholz et al. (2016), we assume attackers are able to "clearly observe all sensitive information and behavior" from the recorded videos. Furthermore, we assume attackers are aware that PushPIN relies on behavioral features derived from finger and wrist movements during authentication. Accordingly, the recorded videos capture not only the screen of the watch but also the finger, hand, and arm movements of the user. Finally, we also assume that attackers are able to practice on their own devices without consequence, but are restricted to a limited number of attempts to unlock a user's device before it bars further attempts.

3.2. PushPIN interface

Pressure input on smartwatches is currently available on several commercial models (e.g., Apple Watch First Generation and above). However, current devices report only binary levels of finger force (e.g., light/strong), precluding use with a systems designed for fine-grained pressure input. To side-step this issue, the PushPIN prototype was implemented on an iPhone X smartphone (iOS 12.1.4)—this device sports a pressure-sensitive touch screen that returns continuous data and this family of devices has been widely used in other work on pressure-based input (Goguey et al., 2018; Krombholz et al., 2016). To create a watch form factor

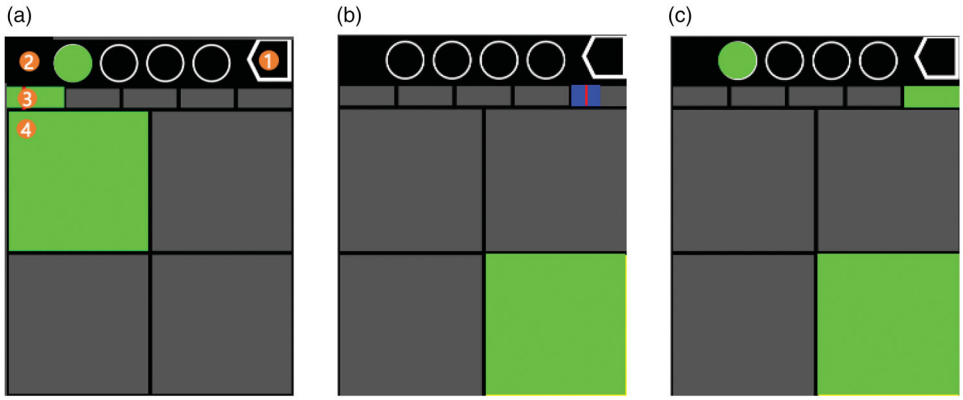

Figure 1. GUI of PushPIN and example series of scenes while input an entry. (a) Four graphic user interface (GUI) components—(1) delete button, (2) entry progress window, (3) touch force gauge, and (4) buttons. (b) Holding entry. (c) Releasing entry.

for the phone, we mounted it on an armband and restricted all input to a 24 by 30 mm region in the center of the screen. Re-purposing a smartphone in this way is a commonly used way of exploring next-generation smartwatch experiences (Gil et al., 2017). Furthermore, while there are clear weight differences between this smartphone (174 g) and a watch (e.g., Apple Watch 6's 30.5 g) we do not believe the will substantially change the key aspects of users' behavior we seek to observe. Studies of arm encumbrance (Knight & Baber, 2007) indicate that significant muscle activity and participant fatigue occur when wearing a wrist-mounted wearable computer of 0.54 kg or more and during tasks involving arm lifts for 10 second or more. Our device is approximately one third of this weight limit and authentication tasks, in general, are relatively short. In the remainder of this paper, we refer to this wrist-mounted device as a "watch."

Individual PushPIN passcode items combine a touch to a specific on-screen button with a specific force applied during that touch. Figure 1(a) shows the current implementation, based on four large (12 mm) square buttons that can be selected with any one of five pressure levels, therefore supporting a total of twenty distinct inputs. We selected four targets as this design is used in prior work on smartwatch authentication that also seeks to present users with large, readily selectable targets (Oakley et al., 2018). While these larger buttons may improve usability, we do not believe they will impact security against observation—current systems such as PIN or APL already show very low resistance against this attack (Hutchins et al., 2018). Rather, we sought to improve resistance to observation via isometric pressure input, which may be harder to observe as it does not entail gross physical motion. We selected five pressure levels following Goguey et al. (2018), where participants self-report this scheme (used in a text selection task) enabled them to achieve high levels of accuracy. In addition, we further increase the resistance against observation by including behavioral biometrics-based authentication. We argue that touch and wrist motion behavior during controlled pressure input will be a rich source of information capable of verifying an individual's identity.

The five pressure levels we used were distributed over the full scale of values reported by the iPhone but were not equally sized; boundaries were determined manually during system design to facilitate accurate selection. Specifically, the sizes of ranges used for the lightest and two heaviest pressure levels were reduced compared to the second and third levels. This is because we found heavier and, in particular, edge pressure levels (i.e., lightest/heaviest) were simpler to select than the central levels—any light touch will select the initial pressure level and, similarly, any strong touch will select the final pressure level, even if it exceeds the maximum detectable force. In this way, the two boundary regions are "infinitely deep," a quality typically reported to facilitate accurate selection (Accot & Zhai, 2003). The iPhone reports pressure in arbitrary units (0–6.66); we measured the pressure levels corresponding to the five levels in our system in gram-force using an electronic scale: 0–54.85; 54.85–167.50; 167.50–280.16; 280.16–333.17 and; 333.17–392.81.

The PushPIN interface supports target selection by simple green button highlights and pressure input with an interactive feedback gauge (Figure 1) that visualizes the current pressure applied (a red line), the boundaries demarcating each of the five selectable pressure levels, and the currently selected pressure level. Pressure levels are selected by remaining within the same pressure level for 300 ms—a technique and threshold taken from prior work (Goguey et al., 2018) that is akin to "pressure dwell." As users apply pressure to enter a new pressure level, they see an expanding blue highlight that begins to cover the corresponding region of the pressure gauge. After 300 ms, the gauge region is completely covered and the highlight turns green to signify selection. Any prior selection of a pressure level remains active until a new level is selected. This design ensures that users do not inadvertently select lighter pressure levels during screen release—the inevitably varying pressures that occur during intentional finger lifts are brief and do not result in inadvertent selections of undesired pressure levels. During user input, PushPIN records the following touch screen data at 100 Hz: touchpoint (x/y), touch radius, and touch force. Additionally, it records the following motion

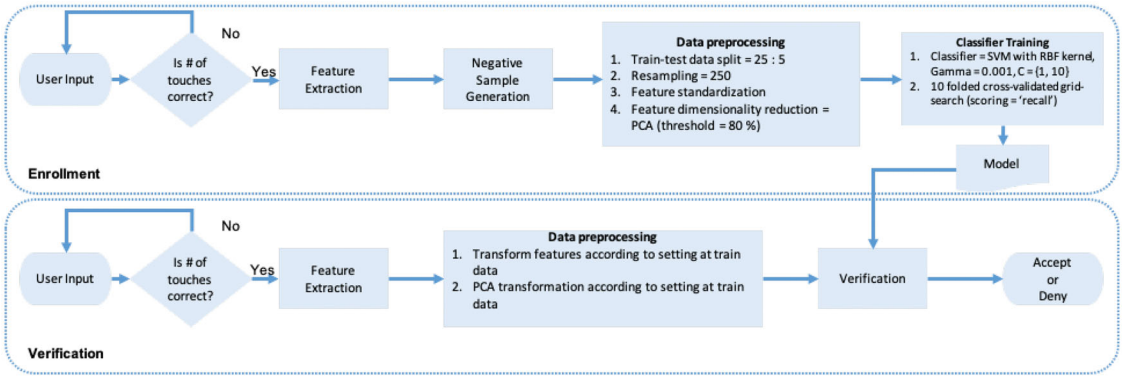

Figure 2. Overview of PushPIN system applied for the video observation attack study.

sensor data at 250 Hz: acceleration and rotational velocity (i.e., from a gyroscope) in x, y, and z axes.

PushPIN passcodes are composed of four sequential passcode items, thus enabling a total possible input space of 20^4, or 160,000, different unique passcodes. The PushPIN interface also features standard feedback and controls to support this data entry task: a panel at the top indicates the number of PushPIN items currently entered and includes buttons to both delete the previously entered item and to clear the currently entered sequence of items.

3.3. System overview

As with other biometrics authentication techniques, PushPIN is composed of two separate processes: *Enrollment* and *Verification*. These are illustrated in Figure 2 and described below.

Enrollment: Users enter a four-item PushPIN passcode and a feature vector is calculated. A set of negative samples is then synthesized based on the statistical distribution of the features observed in a previously collected data set of PushPIN enrollments. All samples are then pre-processed (scaled, re-sampled, and reduced in dimensionality) and split into training and test sets. Finally, a recognizer is trained for verification.

Verification: Users unlock their devices by entering their four-item PushPIN passcodes. Feature vectors are generated and the trained model is used to determine whether or not the user is genuine.

Feature vectors are derived by calculating summary statistics from 12 variables from four different behavioral traits during each PushPIN passcode item entry—force, touch (x position, y position, and radius), acceleration(x, y, z, and magnitude), and rotational velocity(x, y, z, and magnitude). Specifically, for each variable, we calculate the minimum, maximum, range, mean, standard deviation, and, after applying zero-padded fast Fourier transformations (FFT), the top four amplitudes and, for the latter three amplitudes, the frequencies at which they occur. The top frequency did not vary (it was always zero), so was excluded. Furthermore, we calculated skewness and kurtosis for all variables except touchpoint and radius, as these variables showed minimal variation in these metrics. Finally, we also included three features related to the input timing trait—the touch duration in ms and the number of samples logged for both touch and motion data. These served as proxies for touch duration. Ultimately, this led to 165 features for each PushPIN item: three-timing features; three touch metrics by 12 features; one force metric, four rotational velocity metrics, and four acceleration metrics, each with 14 features. Consequently, each four-item PushPIN had 660 features. Data for all features were normalized within training sets.

PushPIN uses a binary classifier trained using the combination of each users' genuine data (captured during enrolment) and imposter data synthesized from feature distributions derived from a data set of PushPIN users. In comparison to alternative approaches such as the use of a one-class classifier (Buriro et al., 2018), this reduces the number of samples each user needs to provide during enrolment. We opted for this approach as prior work has suggested users are reluctant to provide a large number of samples during enrolment (Nguyen & Memon, 2018). Furthermore, the use of synthesized imposter data means that no genuine data from other users needs to be stored or used to train the system. Specifically, we synthesize PushPIN feature vectors from a data set containing 900 PushPIN entries captured from 30 different users (30 entries each), including the genuine user. For each possible combination of pressure level and button, we calculate the mean and standard deviation of each of the 165 features. We then calculate z-scores for this data, sample 3480 individual instances from the feature distributions, and combine them into 870 complete four-item PushPIN passcode feature vectors.

In addition to these methods, we explored three different variables that impact the optimization of PushPIN classifiers. These are:

Feature dimensionality reduction. A large number of features may inflate classification performance; to mitigate this, we apply dimensionality reduction techniques (Teh et al., 2016). We first culled constant or quasi-constant features that had a variance of less than 1%. We then applied five automatic techniques to the remaining feature set: Principal Component Analysis (PCA); Neighborhood Component Analysis (NCA); Linear Discriminant Analysis (LDA); cross-validated recursive feature elimination (RFE)

with a linear Support Vector Machine (SVM) estimator and; a Gini importance based feature selector.

Train and test set size. We first split the data into differently sized train and test sets that each maintain the original ratio between the genuine user and synthesized imposter data. While splitting the data, we also maintained the entry order of genuine user data as this best represents the realistic context of a genuine lock scheme enrolment session. We study the impact of training set size by assessing performance with between 3 and 25 genuine user entries. The goal of this process is to derive an appropriate minimum number of samples to capture from users in the enrolment stage. Keeping the number of samples as low as possible is desirable as it reduces the burden on the user during enrolment (Nguyen & Memon, 2018).

Re-sampled set size. After splitting the data, we adjust the amount of re-sampling used for instances of both genuine user and imposter for minimizing the impact of imbalances between classes of train data on classifier performance—we up-sample genuine user data and down-sample imposter data to create matched sets via random re-sampling. We explore performance from 100 to 800 in steps of 50 samples to identify an optimal size for the training set. We note re-sampling was only applied to the training sets.

We explore the use of three different binary classifiers, each frequently used in research in behavioral biometrics (Teh et al., 2016): SVM, K-Nearest Neighbor (KNN), and Random Forest (RF). A 10 fold cross-validated grid-search is used to tune hyperparameters of each classifier (Scikit-learn, 2021).

The verification process checks whether a new passcode that has been entered matches the profile of the genuine user or not. This process involves a simple match on the equality of the passcodes in terms of the sequence of targets and pressure levels entered as well as an assessment of whether the behavioral measures match the original user. Before matching, entered data is subjected to the same processes of feature extraction, normalization, and dimensionality reduction used during enrolment.

4. Study 1. Data collection and random attack

We conducted a user study to collect data during PushPIN enrolment and verification to establish user performance on basic usability metrics and provide a data set with 900 data points to both generate the required binary classifiers and also to assess classification performance through simulated guessing attacks. The study was approved by the local institutional review board (IRB).

4.1. Method

4.1.1. Participants

A total of 30 participants were recruited (mean age = 23.07, $\sigma = 2.66$) from posts to local university social media groups. In total, 18 were male and 12 female. Participants were screened to exclude left-handedness to increase the homogeneity of captured data. Three identified as both-handed and one indicated a preference for wearing a watch on their right wrist. We surveyed their experience with mobile and wearable technology and pressure input using 5-point Likert scales. They reported high familiarity with smartphones ($\mu = 4.47$, $\sigma = 1.31$) but not smartwatches ($\mu = 1.33$, $\sigma = 0.76$). In terms of pressure input, they reported moderate experience with this technology on smartphones ($\mu = 2.27$, $\sigma = 1.51$) and low experience with it on smartwatches ($\mu = 1.27$, $\sigma = 0.69$). Additionally, 21 reported they were familiar with devices powered by Google's Android platform, while eight were familiar with Apple's iOS and one was familiar with both.

Participants were compensated for the study with approximately 5 USD in local currency. Additional compensation was available for the second session but was contingent on performance: participants were informed they would receive 5 USD for recalling their PushPIN correctly so long as it did not overlap with the PushPIN of any other study participant who attended the experiment on the same day. This compensation structure was designed to encourage participants to select both memorable and unique passcodes.

4.1.2. Apparatus

Two study apps were implemented on the watch introduced in section 3.3. The first was a demonstration app that showcased the targets and interactive feedback. It was used to familiarize participants with PushPIN. The second app executed the study procedures and collected data as described below.

4.1.3. Procedure

The study was conducted in a quiet office environment with participants seated in a chair with no armrests. The study ran over two days and each participant completed the following phases:

Day 1—Instructions: The study began with participants reading instructions and agreeing to and signing consent forms. Participants were able to ask questions to clarify any uncertainties. They were also requested to complete all input tasks both rapidly and accurately, to keep their watch arm in free-space (e.g., not resting on a surface) during study tasks and freely and comfortably rest at any time between study tasks. They were mandated to take a 5-second break between successive PushPIN entries to minimize fatigue. They were not explicitly informed of the behavioral biometric aspects of PushPIN during this study. There were given a series of guidelines to avoid large-scale within-subject variability between different PushPIN entries: they should not rest or support their arm, they should not perform any extra activities such as speaking should try to minimize fidgeting or adjusting their body pose. While participants were aware they would be asked to recall their PushPIN after one day, we did not explicitly restrict (or encourage) them from marking down or

otherwise externally storing their created passcode. This protocol follows closely related studies Oakley et al. (2018) and reflects the fact that such note-taking is a common real world password recall strategy Anwar and Imran (2015). Observing how it is deployed during PushPIN use may help shed light on how challenging participants expected it to be to recall their PushPINs.

Day 1—Experience: The participants used a dedicated app to become accustomed to PushPIN input. The app graphically highlighted a target button and pressure level with yellow; participants needed to select this target and pressure level. Participants practiced PushPIN input using this application until they were satisfied they were familiar with it. Overall, they spent on average 3.88 min ($\sigma = 1.62$) on this process.

Day 1—Creation: The participants created a four-item PushPIN passcode by consecutively selecting targets and pressure levels on the watch. We applied a mandatory selection policy that restricted the re-use of pressure levels—each of the four items entered needed to use a different pressure level. We applied this policy to ensure we collected diverse pressure levels in the course of the study to best support training. During passcode creation, participants were also able to delete or clear entered items at any time. After four items were entered, they selected a button to move to the next phase.

Day 1—Confirmation: Participants were required to re-enter their four-item PushPIN passcode, using a similar interface to the creation phase. This confirmation passcode was checked to determine if targets and pressure levels matched the created passcode. If they did not, participants moved back to the creation phase. If they matched, the PushPIN was set and could not be modified at any further point in the study. During the confirmation phase, it was also possible to tap a cancel button to return to the creation phase.

Day 1—First Practice: Participants used the same interface to correctly enter their PushPIN ten times. Incorrect entries were logged and also needed to be repeated.

Day 1—Distractor: Participants completed distractor tasks to erase their short-term memory of their PushPIN—they first played a simple web puzzle game "Free Brain Age"[1] for 3 min. After playing the game, participants filled in a demographic questionnaire.

Day 1—Recall: Participants were asked to correctly recall their PushPIN by entering an identical sequence of targets and pressure levels. Trials were counted as failed only if participants entered wrong button/force combinations. We did not consider any behavioral biometrics features during the data collection phases of this study. They were given five attempts to do this. A failure to accurately enter their PushPIN within five attempts led to the termination of the study. Participants were not able to look at notes or other material (if they had made any) during this task. Participants who successfully entered their PushPIN completed a short questionnaire about their usage of primary unlock systems on smart devices and their perceptions of the usability of PushPIN.

Day 1—Second Practice: Participants completed a session structured identically to the first practice. It collected additional ten correctly matched PushPIN passcodes and was the final task on the first-day of the study.

Day 2—Recall: The second day of the study took place between 24 and 72 hr after the first day. It started with a recall test, structured identically to the Day 1—Recall session. Furthermore, participants reported memorability and any technique used to remember their passcodes in this session (section 4.1.4).

Day 2—Third Practice: Participants completed a third and final practice session, structured identically to the prior two sessions. It was the final task of the study.

4.1.4. Measures

We logged captured the following objective metrics during PushPIN creation: *PushPIN passcodes*, the sequences of button—pressure level combinations created by participants; *Setup time*, the moment from the start of the creation phase through to the end of the confirmation phase and the successful registration of a PushPIN passcode; *Setup cancels*, the number of times participants canceled the creation process during creation or confirmation phases; *Setup deletions*, the number of times participants deleted passcode items during creation or confirmation phases and; *Setup mismatches*, the number of times participants' creation and confirmation passcodes did not match. Also, we captured the following objective metrics during each recall session: *Input time*, the period between tapping first entry and final entry of a correct passcode; *Recall rate*, the proportion of participants who correctly enter their PushPIN passcodes within five attempts and; *Recall attempts*, the number of attempts participants required to correctly enter their PushPIN passcodes.

We also captured subjective measures at various points. After each recall phase, we captured the *System usability scale (SUS)* (Bangor et al., 2009), an instrument that measures the perceived usability of the system via 10 questions with 5-point Likert scales, and the *NASA task load index (TLX)* (Hart & Staveland, 1988), a widely used tool to assess workload. At the end of Day 2—Recall, participants rated the difficulty of memorizing their PushPIN passcode on a 5-point Likert scale and answered a free-text question inquiring about any remembrance techniques they used for their PushPIN passcode. Finally, to support future analysis all practice sessions for Day 1 and Day 2 were recorded by a camera (either a Nikon D610 or a GoPro Hero 4) positioned 50 cm in front of participants. This captured a full view of the input task included the watch screen as well as the participant's fingers and entire upper body: see Figure 3(a) for an example.

4.2. Results

We present results describing PushPIN in terms of both objective and subjective usability, and an analysis of its security based on both the created passcodes and the

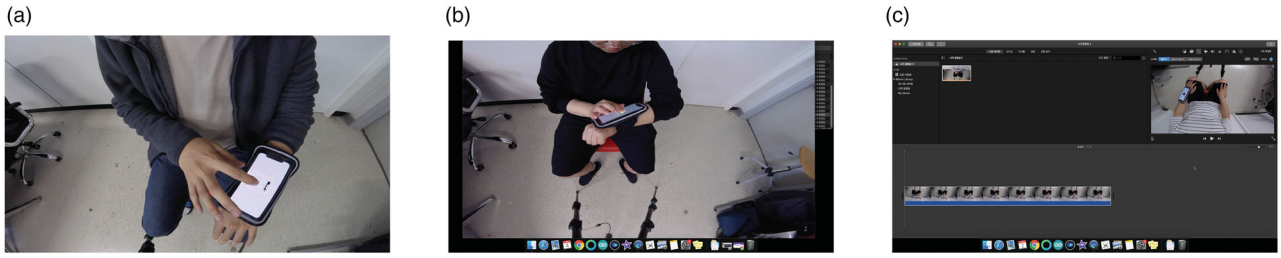

Figure 3. Example scenes of recorded video during the data collection study and screen log of attack study. (a) Recorded video of victim. (b) An attacker explores the recorded video of the victim via the video player. (c) An attacker modifies the recorded video of the victim via editor software.

Table 2. Summary of usability data for PushPIN Setup and Day 1 and Day 2 Recall sessions.

		Setup						Day 1				Day 2			
Time		Cancels		Deletions		Mismatches		Input Time		Recall Atts.		Input Time		Recall Atts.	
μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
103.79	62.34	0.6	1.07	0.43	1.01	0.07	0.37	7.99	2.58	1	0	8.07	2.65	1.33	0.8

"Recall Atts." is a number of recall attempts during single recall session. All times are in seconds, while all other data are counts. (μ: mean, σ: standard deviation).

viability of the pressure-based input it supports as a biometric. We report data from the full set of 30 participants across both days of the study—no participants dropped out, nor failed to complete recall tasks.

4.2.1. Usability

Table 2 shows objective performance data during PushPIN use for both setup and recall phases on day 1 and day 2 of the study. Recall rate is not shown as all participants correctly recalled their PushPINs during all stages of the study: it is 100% throughout. Beyond this high success rate, the most notable data are the temporal data—both setup and input times are relatively long compared to figures reported for more standard authentication techniques such as PIN. For example, Oakley et al. (2018) report PIN setup times on a smartwatch to be 11.1 second and input times to be 1.34 second, figures that are between 10% and 17% of the PushPIN data captured in the current study. Furthermore, ForcePIN (Krombholz et al., 2016), a system that used two pressure levels during PIN entry on a smartphone reports a mean input time of 3.66 second (45.81% of our data). While novelty effects no doubt account for some of these differences, particularly in terms of setup time, we note the temporal performance reported in this study is relatively substantial compared to these baselines. However, data from PushPIN compares more favorably to prior authentication systems that seek to resist an observation attack. For instance, Undercover (Sasamoto et al., 2008), a system in which haptic cues transmit visually hidden signals during authentication, reports median recall times of between 32 and 45 second. Similarly, Spinlock, Colorlock, and Timelock, systems that rely on haptic or audio cues, in conjunction with input actions such as dwell, to resist observation (Bianchi et al., 2012) show prolonged authentication times of between 8.03 to 20.09 second. While much of this literature lacks comparison points for setup times, we believe this data is sufficient to suggest that PushPINs usability performance is acceptable in the context in which it was designed: to resist video observation attacks.

In contrast to these extended times, input actions to correct errors, and the occurrence rate of errors themselves were very low. In the Day 1 Recall session, no participant inaccurately entered a PushPIN passcode. While performance decreased on Day 2, it remained acceptable; the mean number of recall attempts was 1.33 (SD 0.8) and all participants completed the recall task within the allowed five attempts. A Mann-Whitney test (as the data was not normally distributed) revealed the recall attempt count was not significantly different between the two study days. We also note that the increased number of incorrect entries in Day 2 did not result in longer input times or ultimately result in recall failures. We conclude that although input tasks were prolonged with PushPIN, they remain acceptable with respect to prior systems with similar security objectives (i.e., increasing resistance to shoulder surfing as in Bianchi et al. (2012); Sasamoto et al. (2008)), and participants are capable of completing them with a very high degree of accuracy. We believe this data supports the viability of PushPIN as an observation-resistant smartwatch authentication technique. The costs to usability that it incurs are reasonable with respect to prior systems with similar objectives.

4.2.1.1. Workload and perceived usability. Table 3 shows a summary of SUS and TLX scores for PushPIN for both Day 1 and Day 2. Shapiro-Wilk tests showed both SUS and TLX data were normally distributed, so we ran *t*-tests on the scores between Day 1 and Day 2. SUS showed no significant difference, suggesting usability between setup and recall processes was similar. It had an overall mean of 60.5 which can be interpreted as corresponding to an acceptable level of usability (Bangor et al., 2009). On the other hand, TLX scores varied more significantly between the two days ($p < 0.001$), hovering around the mid-point on the scale on Day 1 and showing a marked drop on Day 2. This suggests that while the PushPIN setup tasks were moderately taxing,

Table 3. TLX and SUS questionnaire data from Day 1 and Day 2 (μ: mean, σ: standard deviation).

	TLX														SUS	
	Mental demand		Physical demand		Temporal demand		Performance achieved		Effort expended		Frustration experienced		Overall workload			
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Day 1	10.07	5.43	10.83	5.73	8.13	4.67	6.83	5.74	8.67	5.21	6.67	4.85	51.20	20.83	58.67	15.48
Day 2	5.33	4.74	6.57	5.42	5.90	5.33	5.67	6.13	6.73	4.14	5.03	4.54	35.23	21.93	62.42	13.42

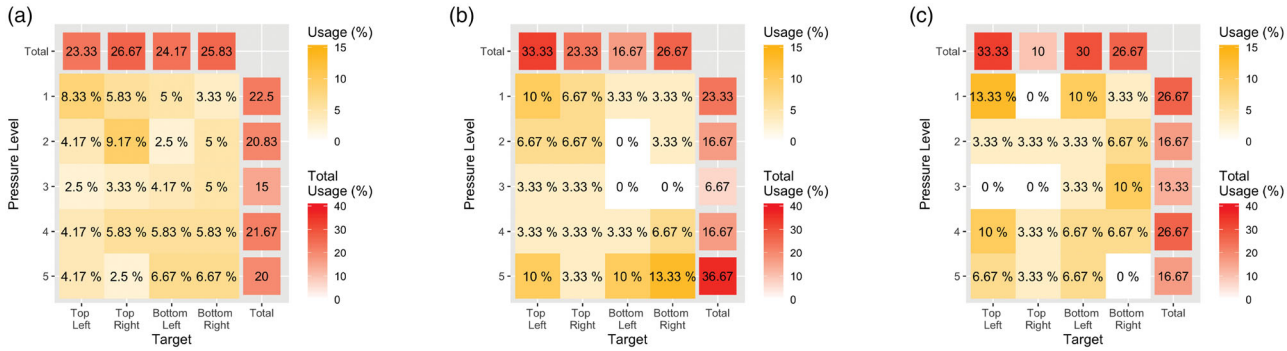

Figure 4. Ratio of each PushPIN item used (%). (a) Overall. (b) First PushPIN item. (c) Final PushPIN item.

the shorter and simpler recall task on Day 2 placed low demands, in terms of workload, on participants.

4.2.1.2. Memorability and remembrance techniques.

Participants reported few difficulties in memorizing their PushPINs ($\mu = 2.27$, $\sigma = 1.01$). They used a range of different remembrance techniques. Six participants indicated they noted down their PushPINs, four used digital copies on their mobile devices, one sent it to him/herself in a messenger application and one wrote it down on a sticky note. None reported referring to their notes during the recall tasks. Besides, nine participants stated they used spatial patterns such as a shape (e.g., "N") or sequence (e.g., clockwise) to remember buttons, and one reported use of existing personal information (their phone number). Pressure levels were selected to be a sequence of integers (four participants), a musical rhythm (four participants), a shape (two participants) or to reflect personal information (three participants), or simply the ease of entering pressure levels (one participant). The relatively low rate with which participants used personal information to compose their PushPINs (4/30 participants) is promising, as this remembrance technique has been previously identified as highly exploitable (Bonneau et al., 2012).

4.2.2. Security

We considered the security of PushPIN based on an analysis of PushPIN passcodes generated by participants and via an assessment of the security of using the pressure-based input as a behavioral biometric.

4.2.2.1. PushPIN frequency analysis.

Biases in PushPIN item selection could reduce its security. To make a preliminary assessment of how this issue might impact PushPIN, we first calculated the usage frequency of each of the 20 PushPIN items—see Figure 4. The data show a relatively even use of the available input symbols: the data for all codes hovers around 25% for the selected target and 20% for the selected pressure level, values which indicate an even distribution (see the row and column totals in the figure). There may be a bias to select targets and pressure levels as "matched pair" as, for example, the top-left target is commonly used with the lightest pressure level (8.33%) and the top-right target with the second-lightest pressure level (9.17%). This may correspond to the use of what participants perceive to be the "first" and "second" targets with the "first" and "second" pressure levels. Some biases also appear in the data from the initial PushPIN items, with participants showing a marked tendency to start with the heaviest pressure level (36.67%) and, to a lesser extent, the top-left target (33.33%). In the final PushPIN item data, the top-right target was very infrequently chosen (10%). In the first and last item data, we also note further evidence for use of "matched" combinations of targets and pressure levels—the bottom-right target with the heaviest pressure level (initial item, 13.33%) and top-left target with the lightest pressure level (final item, 13.33%). In general, we conclude that the distribution of selected PushPIN symbols was relatively even, but that the tendency for participants to match pressure levels and target locations is a trend that attackers may be able to exploit during guessing attacks. We also note this tendency may have been implicitly encouraged by our policy of not allowing participants to select the same pressure level twice. As such it may reduce the resistance of PushPIN to guessing attacks and would not be recommended for use in any real system.

4.2.2.2. PushPIN pattern analysis.

As with other authentication systems, such as PIN, users can opt to generate patterns that repeat items, typically to improve memorability. Such patterns can be exploited by attackers. The policy used in this study mandated the use of unique pressure levels for

Table 4. The number of PushPINs that contain an item used twice or more.

Patterns of button	XX??	X?X?	?XX?	?X?X	??XX	XXX?	XX?X	X?XX	?XXX	XXXX	Total
Count	2	2	1	1	1	0	0	0	0	0	7
Ratio (%)	6.67	6.67	3.33	3.33	3.33	0	0	0	0	0	23.33

"X" represents an item that is used more than twice in a given PushPIN, and "?" represent any other item.

Table 5. Random attack performance using optimal recognizer configuration (PCA feature reduction, training set of 25, re-sampled set size of 250, RBF-SVM classifier).

Accuracy		FPR		FNR		EER	
μ	σ	μ	σ	μ	σ	μ	σ
99.6	0.6	0.16	0.35	7.33	13.37	0.51	1.14

The unit is % for all metrics. μ is mean, σ is the standard deviation.

each item and so prevented the use of these strategies for pressure. Accordingly, we examined the PushPIN passcodes for this behavior in terms of target selection only. We did this by counting the occurrence of a wide range of repeating patterns—see Table 4. In total, seven participants (23.3%) employed one of these patterns, with no more than two participants employing the same pattern. This contrasts well with prior work reporting much higher rates of repeated pattern use in PINs (e.g., 46.67% (Oakley et al., 2018)). On the other hand, we note this indicates that 23 participants used all four items in their PushPIN, a high rate that attackers may also be able to exploit. Perhaps reflecting the prevalence of this pattern, one PushPIN was created by two participants. Additionally, there were seven patterns of target selections that repeated a mean of 2.71 times and two repeating pressure selection patterns that occurred a mean of 2.5 times. This suggests that as with other knowledge-based authentication systems (Oakley et al., 2018), PushPIN suffers from biases in the way users select passcodes; policies, such as mandated initial selections (Cho et al., 2017), which can alleviate the tendency for participants to commence their lock codes in predictable ways, would likely need be used to mitigate these.

4.2.2.3. Recognizer performance. We sought to select a classifier, dimensionality reduction method, and the minimum number of samples to support good performance for distinguishing a genuine user from an attacker during PushPIN input. We examined performance via the metrics of: FPR, FNR, and EER. We selected the best model as the one achieving the lowest EER: 0.51%. This used PCA feature dimensionality reduction (threshold = 80%, leading to a reduction of the number of features to a mean of 51.7 ($\sigma = 2.61$)), a training set size of 25, re-sampled set size of 250, and an RBF-SVM recognizer (gamma = 0.001, and C set to either 1 (28 cases) or 10 (2 cases)). The algorithm to generate this model is illustrated in Figure 2. Full details of the model performance on the test data of each participant are presented in Table 5. We argue this level of performance suggests PushPIN is a viable unlock technique: it combines a low FNR, authenticating genuine users in almost all cases, with a fair FPR, suggesting it provides reasonable security against attackers who are aware of the lock code but lack details or recordings of how users actually perform the physical input. Performance in terms of the key metric of FPR contrasts well with closely related watch authentication schemes such as TapMeIn Nguyen and Memon (2018): 0.98% versus 0.16% in the current work.

To arrive at this final model, we simply selected the feature selector (PCA) and classifier (RBF-SVM) that yielded the best performance. Selecting suitable enrollment and re-sampled set sizes was more challenging. Figure 5(a) illustrates how the EER varies along with the range of values we considered. The performance was optimal with an enrollment set size of 25, the maximum supported by our study design; given the general downward trend in this chart, it is possible that larger sets would further improve performance. Increases in re-sampled set size show more modest improvements; we selected 250 (generated from an enrolment set size of 25) as the size required to achieve reasonable performance.

4.2.2.4. Trait ablation analysis result. We performed trait ablation to investigate the contribution of each sensor channel—see Table 6 for a list of channels—to the user verification performance we report (Beyan et al., 2021). We did this by the simple expedient of introducing a new initial step in our data processing pipeline: we remove all data from a sensor channel. We then create and test new recognizers following otherwise identical procedures. We calculate the importance of a given sensor channel by subtracting the verification accuracy of each ablated recognizer from that achieved by the original recognizer. This data is reported in Table 6 for all sensor channels. The results indicate that timing features provided the highest contribution to recognizer performance. This is in line with prior work on behavioral biometrics on smartwatches (Nguyen & Memon, 2018) which is entirely reliant on this trait. We note force features provided the second most prominent contribution, suggesting they may be an effective complement for timing features. In addition, motion features made quite limited contributions. This suggests that participants' arm motions during PushPIN input had limited variability and salience.

5. Study 2. Video observation attack

We conducted a study to measure the security of the PushPIN prototype against a video observation attack. This study involved a new set of participants, acting as attackers, who were informed about the operation of the system then watched videos of the participants in the data collection study and attempted to enter their PushPINs. The study was approved by the local IRB.

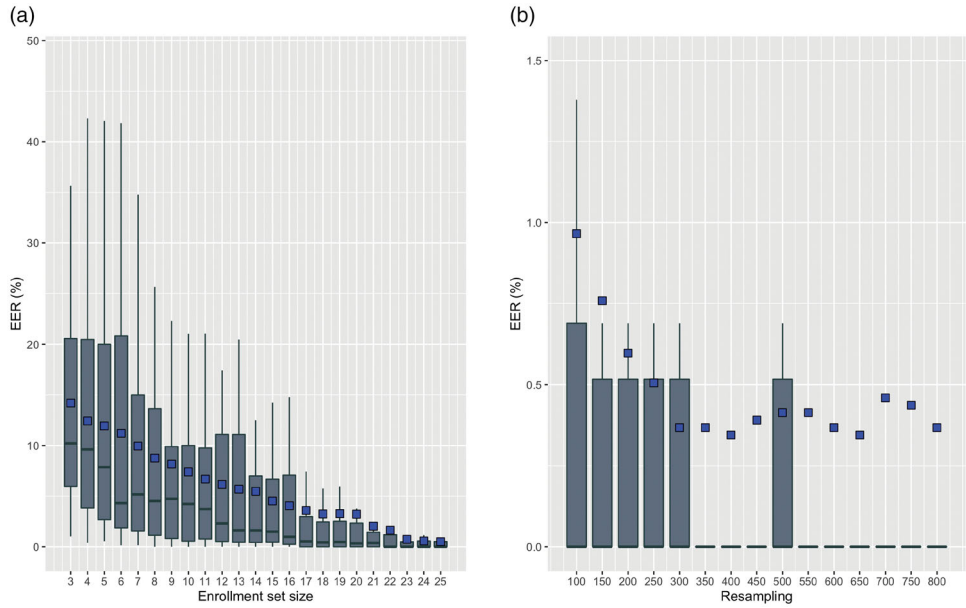

Figure 5. Box plots of the effect by enrollment set and re-sample size on EER—Blue squares are mean EER of each box plot. (a) The effect by enrollment set size on EER. (b) The effect by resampling size on EER.

Table 6. Result of feature ablation analysis. μ is mean, σ is the standard deviation.

Traits	Acceleration	Rotation Velocity	Touch	Force	Timing
μ	1.40	2.11	3.87	6.76	29.64
σ	1.18	1.87	2.36	3.39	12.01

5.1. Method

5.1.1. Participants

We recruited ten new participants to serve as attackers in this study via social media channels. All were students at the local institution, five were female and they had an average age of 20.9 ($\sigma = 1.52$). Participants were also screened for right-handedness to ensure homogeneity with the original study. Nine of the participants reported wearing a watch on their left wrist; the other wore a watch interchangeably on either wrist. All participants frequently used smartphones and five indicated they frequently used pressure input on their phones. Only one participant had previously used a smartwatch and had not experienced pressure input on the watch. Three participants stated that they had previously engaged in shoulder-surfing of phone lock codes: one had attacked APL, one PIN, and the final one had previously attacked both of these schemes. Finally, one attacker was familiar with photo or vector image editing software, but none had experience with movie editing software. They were compensated with approximately 10USD for participation and received additional compensation of 5USD for each PushPIN they successfully cracked. Each participant had the chance to crack three PushPINs.

5.1.2. Apparatus

The study took place in the same environment, and participants used the same furniture, as in the data collection study. The watch used in this study was also identical to the data collection study. Three different applications were used in this study; one to provide familiarization with PushPIN, a second to freely practice attacks, and a final one on which attacks against PushPINs created by the participants of the data collection study were realized. Participants were also provided with a PC (laptop with macOS version 10.14.6) and a 22-inch monitor on which to view, edit, examine and explore the videos of the PushPIN unlock they were asked to crack. A video player (Quick Time), editor (iMovie) and basic tools to capture and edit screenshots (Preview) were available on the PC. Figure 3(b,c) are example scenes of using those tools for attacking. The PC was also used to fill in the study questionnaires.

This study used the videos of successful PushPIN input collected in the data collection study. For each of the original participants, we selected two representative PushPIN entry trials which show clear depictions of the screen and input, the touching fingers, and the participants' upper body pose and motion. We limited the number of recordings to two per participant as it would be challenging to observe multiple unlocks in the real world.

5.1.3. Procedure

Each attacker was allocated three Push PINs to attack, such that 30 attacks were made in total—one for each PushPIN created in the data collection study. This study took approximately 1 hr to complete. It started with instructions that explained the purpose of the study and the operation of PushPIN and then moved on to a practice phase where the attacker could practice with the PushPIN system until they were comfortable with its operation. They then received additional information on the operation of PushPIN, included a detailed explanation of the touch and wrist motion behavioral biometrics data captured, the summary statistics generated and the features used. We also informed

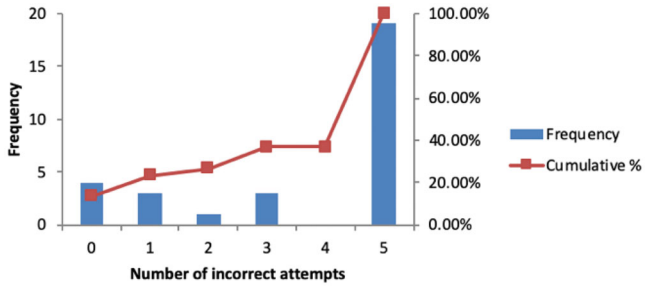

Figure 6. Histogram of the number of incorrect attempts.

participants that success in cracking PushPINs would likely be achieved by careful observation and imitation of the input shown in the videos. Next, they practiced an observation attack by examining a video of an experimenter entering a PushPIN with the image and video playback and manipulation tools provided. They practiced entry of the observed PushPIN in the practice app and finally attempted to crack it in the authentication app.

After completing these instructional and practice activities, participants began to crack their assigned PushPINs, following a similar process to the practice attack. They viewed, captured, and manipulated video of successful PushPIN entries, freely practiced replicating the input they observed and, when satisfied, attempted to authenticate with their targets' PushPINs. They were given a maximum of 15 min and five attempts to crack each PushPIN. From these activities, we logged the attack success rate or proportion of PushPINs that were cracked, the number of attempts made to crack each PushPIN, the time taken for each attack, and self-ratings (collected on a 5-point Likert scale) of the difficulty of each attack. A post-study interview asked participants about their strategies for, and experience of, attacking PushPIN.

5.2. Results

In total, 11 PushPINs (36.67%) were cracked; one attacker failed in all attacks, seven attackers had a single success, and two attackers each successfully attacked two PushPINs. While this indicates that attackers were able to glean sufficient information about PushPINs to support an attack, it also suggests the process is challenging—almost two-thirds of attacks failed. This contrasts to near-perfect observation attack rates for prior pressure-based authentication input techniques. Khan et al. (2018) report more than 97% of the lock codes in their system, which involved input with two pressure levels and no use of behavioral features or analysis, were cracked when attackers were provided with a top-down video of a successful authentication attempt. Figure 6 illustrates the histogram of failed attack attempts: an overall mean of 3.63 ($\sigma = 1.97$). It suggests, as with many prior input schemes, that some PushPINs were relatively weak and easy to enter (i.e., those cracked on the first attempt) while the majority remained more challenging. We note the difficulty in cracking PushPINs was predominantly due to difficulties in mimicking the behavior of the observed users: of the 109 failed attacks recorded in the study, only 4 (3.67%) involved a failure to enter the correct sequence of targets and pressure levels and 105 (96.33%) were denied based on a failure to match the behavioral features. This suggests that although PushPIN codes are readily observable in a video attack, the biometric behaviors captured during PushPIN input are relatively hard to accurately observe or precisely imitate.

5.2.1. Attacker effort

Overall, attackers spent a mean of 8.57 min ($\sigma = 2.89$) to attack the PushPINs of each victim. During each attack task, the attackers observed videos a mean of 9.87 times ($\sigma = 7.26$) and used the rewind controls to re-examine subsections a mean of 13.63 times ($\sigma = 9.68$). They practiced a mean of 7.57 times ($\sigma = 6.27$) prior to initially attempting to attack each PushPIN. These efforts were viewed as challenging. Attackers rated the perceived difficulty of completing an attack as a mean of 4.13/5.0 ($\sigma = 1.11$). We also compare the means of these measures recorded for successful attacks versus unsuccessful attacks. Completion time and the number of observations were normally distributed so we applied t-tests. Mann-Whitney tests were used for other measures. The only significant difference between successful attacks and unsuccessful attacks was in perceived difficulty; a mean value of 4.73/5.0 when the attack failed attack and 3.09/5.0 when it succeeded ($p < 0.001$). The high difficulty ratings for failed attacks highlight the challenges participants experienced in mimicking and cracking the majority of PushPINs.

To support their tasks, attackers used several tools to clearly observe the behavior of victims and take notes, including paper to record each of the victim's PushPIN (all attackers). Eight attackers also used QuickTime player to freely play, rewind, and rotate the videos while the remaining two attackers used iMovie to play, but also crop and magnify sections of the videos. Their description of the strategies they used for video observation generally focused on the specific features they carefully observed and later mimicked during attacks. Nine attackers directly mentioned touch features such as force (six attackers), centroid location (five), touch time (five), and touch radius (two). Four attackers mentioned observing motions of the watch such as tilting or the posture of arm or hands.

6. Discussion

6.1. Security of PushPIN

Results from the data collection study suggest that PushPIN has the potential to achieve a high level of security. Metrics in the random guessing attack show very high levels of performance can be achieved—the EER is 0.51% and the optimal FPR is 0.16%. This compares well to similar attacks conducted on closely related systems. TapMeIn (Nguyen & Memon, 2018), for example, involves knocking patterns on a smartwatch and was reported to result in an optimal FPR of 0.98% and EER of 1.3%. PushPIN's non-conventional symbolic format, in which single inputs involve specifying

two separate values simultaneously, may also support or encourage the creation of diverse secret codes. Evidence to support this assertion comes from the data on PushPIN item selection. Participants used all items fairly evenly suggesting PushPIN may be resilient to brute force attacks based on item usage frequency. One caveat to this suggestion is the presence of biases in the selection of first and last PushPIN items. Further evidence for the strength of PushPIN against guessing attacks comes from the low use of repeated patterns. Prior work studying smartwatch PIN entry (Oakley et al., 2018) has suggested the use of repeated patterns was prevalent—46.67% of PINs involved common repeating patterns compared to just 23.33% with PushPIN. We also note that just 4 (13.33%) of study participants indicated relying on, or reusing, personal information (such as birth dates or phone numbers) in their PushPINs. This again stands in contrast to the relatively high rates with which these insecure practices have been reported to occur in studies of standard watch PINs (Oakley et al., 2018). While considerably larger studies would be needed to reliably determine the actual distribution of PushPIN symbols that users opt to select, we believe the limited data provided in the current study are promising. Taken together, we believe this evidence supports the claim that PushPIN shows the potential to achieve good security against brute force attacks.

We also report on performance against the video observation attack. PushPIN shows greatly improved performance over closely related prior work. For example, ForcePIN (Khan et al., 2018), a PIN system that uses binary pressure levels, was reported to be highly susceptible to observation attack, with over 97% of codes being cracked. This stands in stark contrast to the 36.67% reported in the current study. We attribute these differences to both the increased number of pressure levels in PushPIN and, most critically, to the inclusion of behavioral features in its recognizer. The diversity of features that contribute to the recognizer's performance include, in order of importance, timing-related features, followed by pressure, touch, and wrist motion-related features. The high proportion of crack attempts (96.33%) that included the correct button and pressure selections but failed on the behavioral features suggests that observing and imitating this diversity of features was extremely challenging for attackers.

In addition, we note that attackers spent considerable time preparing their attacks (8.57 min) and self-reported attacking PushPIN to be challenging (4.13/5). This, in combination with the high quality of material provided to support attackers (videos clearly showing input processes as well as full information on the operation of the scheme and opportunities for practice), suggest that while observation attacks on PushPIN are clearly possible, they would be difficult to conduct in real life. We believe the results of the security study suggest PushPIN offers improved security versus observation when compared to both standard authentication techniques (e.g., PIN) or prior research prototypes (Hutchins et al., 2018; Zhao et al., 2017) for smartwatches.

6.2. Usability of PushPIN

We assessed the usability of PushPIN from a wide range of perspectives including setup and recall times, error rate, perceived memorability, and the subjective measures of SUS and TLX. While data remain reasonable throughout, some key differences deserve discussion. Perhaps most notably, both setup (104 s) and recall (8 s) times were lengthy compared to closely related prior work such as ForcePIN (authentication time is 3.66 s) (Krombholz et al., 2016) or Beat-PIN (setup time is 12.3 s and login time is 1.7 s) (Hutchins et al., 2018). While there are numerous possible explanations for this, the most likely of these is a combination of the increased complexity of multi-level pressure input and participants' lack of familiarity with this modality. It may be interesting for future work to examine whether performance in recall task time decreases with longer periods of use. On the positive side, we note that PushPIN input times are better aligned with various prior attempts to obfuscate password or PIN entry through the use of input that is hard to observe such as Undercover (32 or 45 s depending on condition) (Sasamoto et al., 2008) or work by Bianchi et al. (2012) (8–20 second). Designing input to protect against an observation attack typically results in the kind of prolonged input processes we observed with PushPIN. In addition, PushPIN recorded no failed authentications within allowed maximum of five attempts, a level of performance which is uncharacteristically high for work in this area: Undercover, for example, resulted in error rates of between 26% and 52%. This suggests that participants were able to use the extended input times of PushPIN to achieve highly accurate input, something that has not been possible with many prior observation-resistant schemes. Taken together, this combination of relatively long but accurate authentication input makes PushPIN most suitable for cases in which user authentication is required only occasionally. This scenario is a good fit for a wearable device, where authentication typically only occurs at the moment when the device is donned (Nguyen & Memon, 2017).

A final important aspect of our usability results is the challenges participant's reported in terms of PushPIN's memorability: we recorded a somewhat low score of 2.27 out of 5. Our investigations of the remembrance strategy participants employed provide material to make design suggestions for feedback that may help boost this. For example, we could display the input pressure level numerically within the highlighted input button during PushPIN creation; this multi-modal presentation may help participants as many reported memorizing button entries spatially and pressure entries numerically or rhythmically.

In sum, we believe the usability results reported in this work, while mixed, are sufficient to support further investigations of PushPIN or related systems that rely on pressure-based input for authentication.

6.3. Limitation and future work

Despite these positives, there are some limitations to this work. Firstly, we used a smartphone rather than a real

smartwatch for collecting fine-grained pressure data. We did this as current commercial smartwatches do not support accurate multi-level pressure input. However, we believe there is value in examining pressure on the watch form factor as this input modality has strong potential for extending the expressiveness of touch input on small screens—pressure input is well matched to wearables and likely to be included in next-generation devices. We also believe that the ergonomic differences between a watch and a phone are unlikely to have affected the task performance we recorded—a phone is a handheld device and, as such, it is sufficiently light to be mounted on the wrist (Knight & Baber, 2007). In addition, we note that motion features, which would be those most likely to be impacted by variations in device weight or size, contributed weakly to user verification performance (see Table 6). We believe this suggests that the performance we report would be replicated on a genuine smartwatch. Future studies on actual smartwatches will be required to formally test these suggestions and measure other important metrics such as energy consumption and model training time.

A more significant limitation is that, in order to achieve optimal security performance, our recognizer required a training set of 25 PushPIN entries. This would be laborious for users to create and is somewhat larger than that reported for prior smartwatch authentication techniques (Nguyen & Memon, 2018). To mitigate this problem, future work should develop an adaptable recognizer that can incrementally learn a user's profile over samples captured over different unlocks. In this way, a user would not be overburdened by the need to provide a large number of samples during the initial lock setup. Also, gathering samples over time would likely increase the diversity of the samples captured ultimately leading to a better representation of genuine user behavior. Furthermore, this method will also enable researchers to explore various forms of learning effect. For example, as users become more familiar with PushPIN, entry times likely decrease and input accuracy likely increases. There is some evidence to support the presence of these trends in our current data: in the enrollment session in the current study, participants completed 30 PushPIN entries and exhibited a modest (8 s–6 s) but significant (p_i0.05) learning effect in PushPIN entry time that plateaued after making 16.79 entries (calculated by applying the linear-plateau model). This may have impacted the classifier performance we report as initial examples provided by novice users poorly match later, and more practiced and fluent, ones from experienced users. It may be that initial PushPIN entries are best excluded from classifier training sets. Furthermore, a prolonged enrollment process may enable study of the more realistic password memorization and recall experiences that occur over a protracted period. This will complement and extend the data on compressed sessions reported in this article. Such work should also revisit the issue of PushPIN recall strategies explicitly. In the current study, participants were not barred from using notes and a minority (20%) reported using such aids. While we can draw few conclusions from these behaviors in the current study, future work should more formally control them (e.g., bar or balance) to reduce confounds and more fully understand PushPIN memorability.

It would also be worth exploring variations to the specific design studied in this work. For example, to further increase resistance to observation attacks, the graphical feedback presented during recall tasks could be redesigned to be less explicit. Participants in our study were provided with graphical feedback on the currently selected pressure level via both a highlight and a gauge. This reflected the fact participants were unfamiliar with pressure input and intended to support accurate performance of input tasks. More experienced participants may be able to precisely enter pressure input in the absence of this feedback, and doing so would have the advantage of greatly obfuscating input from potential observers. Future work should explore how much pressure level feedback experienced users require to maintain system usability. In addition, this project used a fixed set of four buttons and five pressure levels to input PushPIN items. While these choices were grounded in literature, future work could seek to develop an optimal combination of the number of buttons and pressure levels used by systematically varying these properties. For example, it may be that a reduced number of pressure input levels, combined with a larger number of input buttons (e.g., three pressure levels on six buttons) can lead to improvements in both usability (reduced recall times) and security (reduced EERs)—only further studies can identify the best values for these parameters. Furthermore, the current work did not consider the fact that behavioral biometric data may vary depending on posture or activity; future studies that examine how poses impact the results reported here need to be conducted.

7. Conclusion

This paper introduces PushPIN, an authentication system that uses touch and wrist motion features derived from the input of multiple pressure levels to build a usable and secure primary unlock scheme for a smartwatch. Our studies show PushPIN provides resilience against random guessing, rule-based guessing, informed guessing attacks using personal information, and video observation attack. Moreover, while it shows elevated setup and recall times, input error rates are very low. We suggest this combination of improved security and high accuracy at a modest cost to efficiency make PushPIN a viable possible candidate for unlocking systems on smartwatches, a device category on which unlock events are relatively rare. Future research should develop the classifiers used in this work further (to support incremental learning or retraining), capture a larger sample of user PushPINs (possibly without applying policies that restrict item selection), examine whether performance with pressure input increases over time, and explore other attack vectors, such as smudge (Ranak et al., 2017). By continuing this work, we hope to show that pressure-based unlock schemes are a viable approach to increasing the security and accuracy of user unlock tasks on small screen wearable devices.

Note

1. http://freebrainagegames.com/recall.html

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT [2020R1F1A1070699].

ORCID

Youngeun Song http://orcid.org/0000-0001-9213-4310
Ian Oakley http://orcid.org/0000-0001-5834-8577

Data availability statement

Due to the nature of this research, participants of this study did not agree for their data to be shared publicly, so supporting data is not available.

References

Accot, J., & Zhai, S. (2003). Refining fitts' law models for bivariate pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 193–200). Association for Computing Machinery. https://doi.org/10.1145/642611.642646

Adapa, A., Nah, F. F.-H., Hall, R. H., Siau, K., & Smith, S. N. (2018). Factors influencing the adoption of smart wearable devices. *International Journal of Human–Computer Interaction*, *34*(5), 399–409. https://doi.org/10.1080/10447318.2017.1357902

Anwar, M., & Imran, A. (2015). A comparative study of graphical and alphanumeric passwords for mobile device authentication. In M. Glass & J. H. Kim (Eds.), *Proceedings of the 26th Modern AI and Cognitive Science Conference 2015, Greensboro, NC, USA, April 25–26, 2015* (Vol. 1353, pp. 13–18). CEUR-WS.org. http://ceur-ws.org/Vol-1353/paper_11.pdf

Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *The Journal of Usability Studies*, *4*(3), 114–123. https://dl.acm.org/doi/10.5555/2835587.2835589

Beyan, C., Zunino, A., Shahid, M., & Murino, V. (2021). Personality traits classification using deep visual activity-based nonverbal features of key-dynamic images. *IEEE Transactions on Affective Computing*, *12*(4), 1084–1099. https://doi.org/10.1109/TAFFC.2019.2944614

Bianchi, A., Oakley, I., & Kwon, D. S. (2012). Counting clicks and beeps: Exploring numerosity based haptic and audio PIN entry. *Interacting with Computers*, *24*(5), 409–422. https://doi.org/10.1016/j.intcom.2012.06.005

Bonneau, J., Preibusch, S., & Anderson, R. (2012). A birthday present every eleven wallets? The security of customer-chosen banking pins. In A. D. Keromytis (Ed.), *Financial cryptography and data security* (pp. 25–40). Springer Berlin Heidelberg.

Brewster, S. A., & Hughes, M. (2009). Pressure-based text entry for mobile devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. Association for Computing Machinery. https://doi.org/10.1145/1613858.1613870

Buriro, A., Crispo, B., & Conti, M. (2019). ANSWERAUTH: A bimodal behavioral biometric-based user authentication scheme for smartphones. *Journal of Information Security and Applications*, *44*, 89–103. https://doi.org/10.1016/j.jisa.2018.11.008

Buriro, A., Crispo, B., Delfrari, F., & Wrona, K. (2016). Hold and sign: A novel behavioral biometrics for smartphone user authentication. In *2016 IEEE Security and Privacy Workshops*. IEEE.

Buriro, A., Van Acker, R., Crispo, B., & Mahboob, A. (2018). Airsign: A gesture-based smartwatch user authentication. In *2018 International Carnahan Conference on Security Technology (ICCST)* (pp. 1–5). IEEE. https://doi.org/10.1109/CCST.2018.8585571

Cho, G., Huh, J. H., Cho, J., Oh, S., Song, Y., & Kim, H. (2017). Syspal: System-guided pattern locks for android. In *2017 IEEE Symposium on Security and Privacy (SP)* (pp. 338–356). IEEE. https://doi.org/10.1109/SP.2017.61

De Luca, A., Hang, A., Brudy, F., Lindner, C., & Hussmann, H. (2012). Touch me once and i know it's you! implicit authentication based on touch screen patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 987–996). Association for Computing Machinery. https://doi.org/10.1145/2207676.2208544

Fortify, H. (2015). *Internet of things security study: Smartwatches*. Technical report, HP. https://www.ftc.gov/system/files/documents/public_comments/2015/10/00050-98093.pdf

Gil, H., Lee, D., Im, S., & Oakley, I. (2017). Tritap: Identifying finger touches on smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 3879–3890). Association for Computing Machinery. https://doi.org/10.1145/3025453.3025561

Goguey, A., Malacria, S., & Gutwin, C. (2018). Improving discoverability and expert performance in force-sensitive text selection for touch devices with mode gauges. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1–12). Association for Computing Machinery. https://doi.org/10.1145/3173574.3174051

Hart, S. G., Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload* (Vol. 52, pp. 139–183). North-Holland. https://www.sciencedirect.com/science/article/pii/S0166411508623869

Hutchins, B., Reddy, A., Jin, W., Zhou, M., Li, M., & Yang, L. (2018). Beat-pin: A user authentication mechanism for wearable devices through secret beats. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security* (pp. 101–115). Association for Computing Machinery. https://doi.org/10.1145/3196494.3196543

Jeong, H., Kim, H., Kim, R., Lee, U., & Jeong, Y. (2017). Smartwatch wearing behavior analysis: A longitudinal study. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(3), 1–31. https://doi.org/10.1145/3131892

Khan, H., Hengartner, U., & Vogel, D. (2018). Evaluating attack and defense strategies for smartphone pin shoulder surfing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1–10). Association for Computing Machinery. https://doi.org/10.1145/3173574.3173738

Knight, J. F., & Baber, C. (2007). Assessing the physical loading of wearable computers. *Applied Ergonomics*, *38*(2), 237–247. https://doi.org/10.1016/j.apergo.2005.12.008

Krombholz, K., Hupperich, T., & Holz, T. (2016). Use the force: Evaluating Force-Sensitive authentication for mobile devices. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)* (pp. 207–219). USENIX Association. https://www.usenix.org/conference/soups2016/technical-sessions/presentation/krombholz

Li, C., Jing, J., & Liu, Y. (2021). Mobile user authentication-Turn it to unlock. In *2021 6th International Conference on Mathematics and Artificial Intelligence (ICMAI 2021)* (pp. 101–107). Association for Computing Machinery. https://doi.org/10.1145/3460569.3460577

Li, Y., & Xie, M. (2018). Understanding secure and usable gestures for realtime motion based authentication. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 13–20). IEEE. https://doi.org/10.1109/INFCOMW.2018.8406912

Lu, C. X., Du, B., Kan, X., Wen, H., Markham, A., & Trigoni, N. (2017). Verinet: User verification on smartwatches via behavior biometrics. In *Proceedings of the First ACM Workshop on Mobile Crowdsensing Systems and Applications* (pp. 68–73). Association for Computing Machinery. https://doi.org/10.1145/3139243.3139251

Nguyen, T., & Memon, N. (2018). Tap-based user authentication for smartwatches. *Computers & Security*, *78*, 174–186. https://doi.org/10.1016/j.cose.2018.07.001

Nguyen, T., & Memon, N. D. (2017). Smartwatches locking methods: A comparative study. In Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017). USENIX Association. https://www.usenix.org/conference/soups2017/workshop-program/way2017/nguyen

Oakley, I., Huh, J. H., Cho, J., Cho, G., Islam, R., & Kim, H. (2018). The personal identification chord: A four button authentication system for smartwatches. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security* (pp. 75–87). Association for Computing Machinery. https://doi.org/10.1145/3196494.3196555

Ranak, M. S. A. N., Azad, S., Nor, N. N. H. B. M., & Zamli, K. Z. (2017). Press touch code: A finger press based screen size independent authentication scheme for smart devices. *PLoS One*, *12*(10), e0186940. https://doi.org/10.1371/journal.pone.0186940

Saad, N., & Djedi, N. (2017). Recognition of 3d faces with missing parts based on sift and lbp methods. In R. Jiang, S. Al-maadeed, A. Bouridane, P. D. Crookes, & A. Beghdadi (Eds.), *Biometric security and privacy: Opportunities & challenges in the big data era* (pp. 273–297). Springer International Publishing. https://doi.org/10.1007/978-3-319-47301-7_12

Sae-Bae, N., Ahmed, K., Isbister, K., & Memon, N. (2012). Biometric-rich gestures: A novel approach to authentication on multi-touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 977–986). Association for Computing Machinery. https://doi.org/10.1145/2207676.2208543

Salem, A., & Obaidat, M. S. (2019). A novel security scheme for behavioral authentication systems based on keystroke dynamics. *Security and Privacy*, *2*(2), e64. https://doi.org/10.1002/spy2.64

Sasamoto, H., Christin, N., & Hayashi, E. (2008). Undercover: Authentication usable in front of prying eyes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 183–192). Association for Computing Machinery. https://doi.org/10.1145/1357054.1357085

Saulynas, S., Lechner, C., & Kuber, R. (2018). Towards the use of brain–computer interface and gestural technologies as a potential alternative to pin authentication. *International Journal of Human–Computer Interaction*, *34*(5), 433–444. https://doi.org/10.1080/10447318.2017.1357905

Scikit-learn (2021). *3.2. Tuning the hyper-parameters of an estimator*. https://scikit-learn.org/stable/modules/grid_search.html (accessed 27 April 2021)

Siek, K. A., Rogers, Y., & Connelly, K. H. (2005). Fat finger worries: How older and younger users physically interact with pdas. In M. F. Costabile & F. Paternò (Eds.), *Human-computer interaction - INTERACT 2005* (pp. 267–280). Springer.

Teh, P. S., Zhang, N., Teoh, A. B. J., & Chen, K. (2016). A survey on touch dynamics authentication in mobile devices. *Computers & Security*, *59*, 210–235. https://doi.org/10.1016/j.cose.2016.03.003

Unar, J., Seng, W. C., & Abbasi, A. (2014). A review of biometric technology along with trends and prospects. *Pattern Recognition*, *47*(8), 2673–2688. https://doi.org/10.1016/j.patcog.2014.01.016

Zhao, Y., Qiu, Z., Yang, Y., Li, W., & Fan, M. (2017). An empirical study of touch-based authentication methods on smartwatches. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers* (pp. 122–125). Association for Computing Machinery. https://doi.org/10.1145/3123021.3123049

About the Authors

Youngeun Song is a Ph.D. candidate at the Ulsan National Institute of Science and Technology, Republic of Korea. Her research is in human-computer interaction and specifically, the trade-offs between usability and security on wearables.

Ian Oakley received his Ph.D. in Computer Science from the University of Glasgow, UK and is now a full professor at the Department of Design at Ulsan National Institute of Science and Technology. His research focuses on the design, development and evaluation of multi-modal interfaces and social technologies.