PushID: A Pressure Control Interaction-Based Behavioral Biometric Authentication System for Smartwatches

Youngeun Song(✉) and Ian Oakley

Ulsan National Institute of Science and Technology, Ulsan, Republic of Korea
soyo61@unist.ac.kr

Abstract. Smartwatches support a wide range of functionality, including mediating access to sensitive data and services. However, securing access to them is difficult due to their small size—it is difficult, for example, to enter alphanumeric passwords accurately due to the limited space available to present a keyboard. Consequently, 4-digit PIN is commonly used to secure smartwatches, a technique widely acknowledged to be highly vulnerable to simple guessing attacks. To address these usability and security issues, we propose PushID, a new behavioral biometric technique for a smartwatch that combines four on-screen targets with five pressure levels to enable input of any one of 20 unique symbols from a single screen touch. In addition to this relatively large input space, PushID captures behavioral features during pressure input (e.g., finger touch profile, wrist motions) and uses this as a behavioral biometric. We report on a preliminary study of PushID and its security against random guessing attack: it achieves good usability for a single input (approximately 2 s) and high resistance to guessing (false-positive rates of 1.05%). We argue that pressure-based input can improve the security and maintain the usability of smartwatch lock systems.

Keywords: Smartwatch · Behavioral biometrics · Usability

1 Introduction

The Smartwatch market is steadily growing and is anticipated to achieve sales of 230.30 million units by 2026 [17]. The majority of smartwatches are currently paired with a smartphone to support all functionality. However, as devices become more advanced standalone functionality is starting to be introduced to support users during tasks such as exercise, situations in which users may prefer not to be encumbered by a smartphone. Building on this trend, in the future commercial smartwatches could provide a wide range of services in standalone settings such as collecting health data, storing personal messages, and processing payments. Due to the sensitive nature of these applications, we argue there is a need to secure access to standalone smartwatches.

However, the small size of these devices makes traditional knowledge-based authentication schemes, such as alphanumeric passwords, slow and awkward

A. Moallem (Ed.): HCII 2022, LNCS 13333, pp. 255–267, 2022.
https://doi.org/10.1007/978-3-031-05563-8_17

to enter. Simpler schemes such as Personal Identification Number (PIN) are more feasible but are widely acknowledged to be vulnerable to random guessing attacks [14]. Physiological biometrics-based authentication could be an alternative solution that can improve usability and security over such knowledge-based schemes—such systems have proven popular on smartphones. However, the sensors for established physiological biometrics, such as fingerprints, are hard to integrate into the small case of a smartwatch. As an alternative, researchers have proposed behavioral biometrics based on data from built-in smartwatch sensors such as the touch screen [14] or inertial motion unit [3] with promising results. Pressure-based input is another form of touch input that may lead to rich variations in behavioral characteristics. In addition, it can help support authentication by increasing the scope of different inputs that are available on small input areas—multiple pressure levels can be entered on each on-screen button. Reflecting these benefits, pressure has been proposed as a behavioral biometric feature during tap-based smartwatch authentication [14]. Additionally, explicitly controlling touch force has been used to enable pressure-based explicit authentication schemes on a smartphone [10,16].

In this paper, we extend these approaches by proposing PushID, a smartwatch authentication system based on a novel pressure-based behavioral biometric. This paper presents the design of the scheme and results of a user study exploring its usability and security against random guessing attacks. PushID is based on features extracted from screen touches and wrist motion data in which users intentionally modulate the force they exert. Specifically, we extracted a total of 165 features from raw touch and wrist motion data while the participants generated, sustained, and released one of five discrete pressure levels on a wrist-mounted touch screen. We performed a simple empirical study (N = 30) to collect user behavior while operating PushID. The participants entered 20 randomly assigned PushID entries according to instructions provided during the study. We used this data to train recognizers for each participant and compared user verification performance in a simulated random guessing attack scenario in terms of False-Positive Rate (FPR), False-Negative Rate (FNR), and Equal Error Rate (EER). We also measured the completion time to input a single PushID entry to evaluate the usability of PushID.

The results indicate that the best verification performance of the PushID recognizer was as follows: mean FPR was 1.05%, mean FNR was 42.76%, and mean EER was 8.34%. In the case of FPR, PushID shows improved values compared to other behavioral biometric authentication schemes for smartwatches, such as the 21.65% reported for AirSign [3], and the 7.2% in Beat-PIN [8]. However FNRs, and correspondingly, EERs are higher compared to these closely related schemes [3,7,8,14]. In terms of usability, participants took a mean of 2.09 s for each input, a good level of performance compared to both popular authentication schemes such as PIN (2.195 s) [14] or other behavioral biometrics-based authentication proposed for smartwatches [7,8,14].

Based on these results we argue that authentication via pressure-based behavioral biometrics (based on data captured when users are asked to input specific pressure levels) is a promising approach to smartwatch security that can enhance resistance to random guessing attacks while maintaining good authentication time.

2 Related Work

User authentication systems can be divided into two different modes [20]: identification and verification. In the case of identification, there is an assumption that multiple users share a device so the authentication task is to clarify the identity of the current user among the stored set of genuine users. On the other hand in verification, there is a single genuine user (a device or account owner) and the task is to verify whether or not the submitted data represents that captured from the genuine user or any other individual (e.g., another user or an imposter). In this paper, we consider only verification scenarios.

2.1 Behavioral Biometrics in Smartwatches

Biometric authentication, which identifies or verifies a user according to their sensed characteristics (either physiological or behavioral) is an important and popular authentication method that can achieve both strong security and good usability in platforms as diverse as smartphones and door locks. This method is known as a viable solution to memorability issues and has high resistance against guessing attacks compared to knowledge-based authentication, which authenticates an individual based on information that they know (e.g., a password or PIN). There are two different authentication methods—first, physiological biometrics, which is based upon the unique body features of an individual (i.e., fingerprint) and, second, behavioral biometrics, which authenticate an individual based on their unique activity patterns (e.g., typing patterns) [19]. Though physiological biometrics are well established in many devices, such as the fingerprint or face recognition systems that appear on smartphones, they are hard to implement on smartwatches [14] because they typically require specialized sensors (e.g., fingerprint readers, high-resolution cameras) they are difficult to integrate into small watch form factors. On the other hand, behavioral biometrics may be a more appropriate approach, as many smartwatches are already designed to accurately track the detailed activities of their user to support applications such as exercise or physiological monitoring.

We summarize previously reported verification performance for behavioral biometrics on smartwatches in Table 1. We express performance data in terms of *False Positive Rate* (FPR, the ratio of the number of accepted attempts by non-legitimate users to the total number of attempts by non-legitimate users), *False Negative Rate* (FNR, the ratio of the number of rejected attempts by legitimate users to the total number of attempts by legitimate users), and *Equal Error rate* (EER, the trade-off point where a recognizer is tuned to match FNR and FPR as equal). Various user behaviors, which can be captured by popular smartwatch sensors such as motion sensors or the touchscreen, have been studied

Table 1. Verification performance in random guessing attack for existing smartwatch authentication systems using behavioral biometrics. FPR (false positive rate), FNR (false negative rate), and EER (equal error rate) expressed in %.

Work	FPR	FNR	EER
Li and Xie [12]	0	22	NA
AirSign [3]	21.65	19.48	NA
VeriNet [13]	10.24	20.77	7.17
TapMeIn [14]	0.98	5.3	1.3

for biometrics. Specific modalities include arm gestures [12], mid-air gestures [3], wrist motion data during PIN input [13], and screen tapping rhythm [14]. Their performance is complex and reporting of metrics is not completely consistent. Nonetheless, it is obvious that attaining high performance, in terms of low EERs (or the combination of low FPRs and FNRs) is demanding: a majority of this work achieves scores of 20% or higher on at least one of these metrics. However, bucking this trend, TapMeIn [14] achieved 0.98% EER in response to random guessing attacks. To do this, TapMeIn extracted different touch features from a customized explicit authentication code—a passcode in the form of a rhythmical tapping pattern. This highlights the possible advantages of extracting behavioral features from novel touch actions beyond standard taps. We argue that the more expressive performance inherent in this type of input may help to increase the uniqueness of the behavioral biometrics that can be derived from a user‘s input.

Based on these arguments, we propose PushID to explore the value of the behavioral features extracted from both the touch screen (including touch force data) and 3-dimensional motion sensors (accelerometer and gyroscope to track wrist motion) while users explicitly perform a complex and dynamic input action; controlling specific forces during an entry of a single pressure-based input.

2.2 Pressure Input-Based Authentication

There have long been studies claiming that pressure input is expressive and precisely controllable [2]. Highly accurate pressure sensors are currently integrated into many commercial laptops and smartphones, while binary pressure sensors appear on smartwatches. We argue that pressure input is particularly useful in small form factor devices, which frequently suffer from fat-finger problems, because it can provide additional input options even when screen real estate is highly limited [16].

The potential of pressure-based input has been studied in various studies and showed diverse results. For instance, ForcePIN [10] is a PIN system that features two pressure levels, doubling the number of possible input symbols available. ForcePIN achieved a reasonable authentication time of 3.66 s to complete input of 4-digit passcodes. Pressure has also been studied in the area of touch-based behavioral biometrics. One noteworthy focus for this work is to improve the security of standard lock inputs. For example, De Luca et al. [4] collected touch-related features including location, size, speed, duration, and pressure during pattern lock input on a smartphone, and achieved 77% accuracy while performing authentication tasks. Salem and Obaidat [18] report the peak verification performance in this area—0.9% EER—in a study using 10-keystroke dynamics-related features, including pressure during the task of entering eight alphanumeric passwords. We assert these prior results indicate that pressure input tasks can generate data that is appropriate for verifying an individual using a behavioral biometric approach. The system we present in this paper borrows methods from much of the work reviewed in this section; it leverages the idea that pressure input can increase the expressivity of touch input on small screen wearables to create a behavioral biometric authentication system that provides a large number of attainable passcodes on a small input surface. The goal of this work is to retain authentication usability while improving resistance to random guessing attacks. We do this by investigating the efficiency (time to enter) and FPR, FNR, and EER of our proposed PushID system that is based on extracted touch and wrist motion behavioral features that occur during pressure input.

3 PushID System Design

3.1 Threat Model

PushID was designed to enhance resistance to random guessing attacks, a simple and common attack strategy. In this study, we set the attack scenario as that of an attacker who has gained a victim's device via methods such as theft and tries to unlock it without any preliminary knowledge related to the user or passcode [12]. We assume the attacker was not able to previously observe genuine unlock attempts.

3.2 PushID Interface

Existing commercial smartwatches, such as the Apple Watch First Generation and above, provide pressure input on their touch screen. However, they only support binary levels of touch force so app developers can not use multi-level pressure input at this time. Since this study required a platform that could measure detailed pressure, we used an iPhone X smartphone (iOS 12.1.4) that supports analog pressure measurements in place of a smartwatch—this device has also been widely used for research about touch force-based interaction [6,10]. In order to use the smartphone as a watch, we placed the phone on an armband and made a prototype to receive touch input from only a 24 by 30 mm area in the center of the screen. Emulating a watch with a smartphone in this fashion has been previously adopted for investigating the user experience of next-generation smartwatch interfaces [5]. Although the smartphone (174 g) and a smartwatch (e.g., Apple Watch 6's 30.5 g) are substantially different in weight, we believe these changes likely had a limited impact on the user behavior data we collected

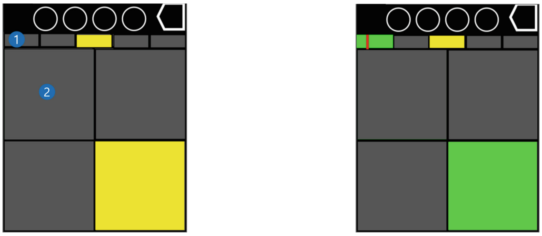

Fig. 1. GUI of PushID (left) showing touch force gauge (1) and buttons (2). Highlight items are in yellow. Example screen during input (right) showing red pressure cursor over first pressure level.

in this study. According to a study about physical loading on the wrist [9], significantly increased stress and fatigue appeared when participants wore a wrist-mounted wearable computer weighing more than 0.54 kg, and lifted their arm for more than 10 s. In the case of our study, the apparatus was about one-third of this weight limit and, generally, the authentication completion time was considerably shorter than 10 s. Accordingly, we do not believe the use of a phone in place of a watch invalidates the work we report. For the rest of this paper, we refer to our wrist-mounted prototype as a "watch".

Single PushID entries were achieved by pressing an on-screen button with a specific level of touch force. Figure 1 shows the current graphical user interface (GUI)—four large (12 by 12 mm) square buttons are provided and one of five pressure levels can be entered per button, so a total of twenty different input behaviors are available. We chose four targets because we want to provide users with large and easy-to-select targets and this design is used for prior research on smartwatch authentication with similar goals [15]. We picked five pressure levels based on prior work [6]—participants in this study indicated that the use of five discrete pressure levels led to high precision and accuracy.

We used the full touch pressure range that could be measured on the watch, but not all five pressure levels were equally divided; the interval of each pressure level was adjusted to improve selection performance following prior designs of five pressure level input systems [6]. Specifically, we reduced the intervals of highest and lowest pressure levels, as these levels have been reported to be easy to select [1]. There was no official way to convert the pressure values measured by iPhoneX to International System Units. Therefore, we first calculated the five pressure levels based on the values measured by the iPhone, and after obtaining a conversion formula through a calibration procedure using an electronic scale and a set of weights. This enabled converting the iPhone's sensor data into gram-force units. After this process the pressure levels were defined as (in gram-force): 0 to 54.85; 54.85 to 167.50; 167.50 to 280.16; 280.16 to 333.17 and; 333.17 to 392.81.

The graphic interface of PushID prompted users to select a yellow highlighted target button and pressure level. It showed the pressure levels as segments on a horizontal gauge (Fig. 1). When a user touched the screen, the touched button was displayed in green, and the current touch force was displayed as a red line that moved across along the gauge (and its segments) in real-time. The five different pressure levels were marked on the gauge. A pressure level was selected when the currently exerted pressure remained in the same pressure level for 300 milliseconds, a technique borrowed from prior work [6]. Progress during this pressure-dwell was marked by feedback in the form of the currently selected level filling up with a blue highlight; after 300ms it was full. If the correct pressure level was selected this highlight turned green, whereas it turned purple if the wrong pressure level was selected. The use of this pressure dwell enabled accurate selection of pressure levels during finger release as this process was rapid (less than 50 ms) and did not result in a selection of any new pressure levels. As this system has the potential to be combined with a knowledge-based authentication scheme in the future, the interface also contained graphic feedback for entering four entries as a set; the entry history was displayed using four circles, and it also supported the ability to modify entered entries with a delete button. These functions were not used in the current study.

3.3 System Overview

Like other behavioral biometric authentication technologies, PushID requires two separate processes: *Enrollment* and *Verification*. A summary of the overall system is shown in Fig. 2 and these two stages are briefly described below.

- *Enrollment*: If a genuine user successfully entered the target pressure level within the target button shown on the screen, a feature vector was calculated based on the behavioral data collected from the beginning of the touch to finger release of the screen. Then the system also generated feature vectors of attackers, that should be distinguished from the genuine user based on a random guessing attack scenario. Finally, the system finished pre-processing the genuine user data and attacker data and then trained a per-user recognizer for verification using machine learning techniques.
- *Verification*: If the genuine user or attacker succeeded in entering the target pressure level within the target button according to the instructions shown on the screen, a feature vector was calculated. Then the user-specific recognizer generated during *Enrollment* was used to judge whether the data represented the genuine user or an imposter.

During enrollment, the PushID system recorded touch screen data (X and Y coordinates of touchpoint, touch radius, and touch force) 100 Hz, and captured wrist-motion data from the watch inertial measurement unit (IMU) 250 Hz while genuine users entered a PushID item. Specifically, we logged acceleration and rotational velocity (gyroscope data) in X, Y, and Z axes.

In the feature extraction stage, a feature vector consisting of summary statistics of 12 variables derived from four distinct behavioral traits captured during input of a single PushID entry was created. These are touch (force, x-position, y-position, radius), acceleration (X, Y, and Z axes, and magnitude), and rotational

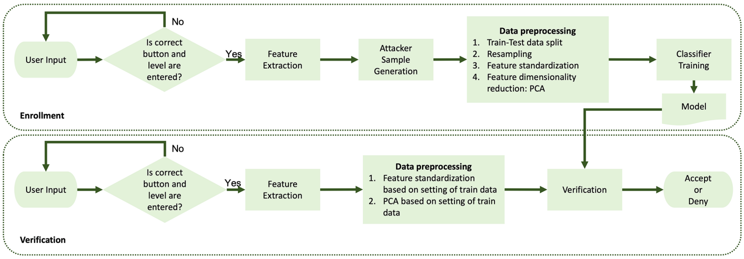
Enrollment
User Input
Is correct button and level are entered?
No
Yes
Feature Extraction
Attacker Sample Generation
Data preprocessing
1. Train-Test data split
2. Resampling
3. Feature standardization
4. Feature dimensionality reduction: PCA
Classifier Training
Model
Verification
User Input
Is correct button and level are entered?
No
Yes
Feature Extraction
Data preprocessing
1. Feature standardization based on setting of train data
2. PCA based on setting of train data
Verification
Accept or Deny

Fig. 2. Overview of PushID system implemented for this study

velocity (X, Y, and Z axes, and magnitude). The calculated summary statistics were: minimum, maximum, range, mean, and standard deviation. Additionally, frequency domain analysis was done via zero-padded Fast Fourier Transforms (FFT). We used the top four amplitudes and the top frequencies at which they occur as features. The highest amplitude frequency was always zero, so it was dropped as a feature. Moreover, we included skewness and kurtosis for all variables except positions of touch and radius since these variables exhibited very narrow deviations. Lastly, we included three features related to input time—touch duration in milliseconds and the number of samples of collected touch and wrist motion data. Each calculated feature value was also converted to a z-score according to the combination of button and pressure level selected; the mean and standard deviation of each feature per combination of button and pressure level were calculated from the full set of user data. In the end, a total of 165 features were created to form the feature vector from each input trial.

To train and test binary authentication classifiers to support user verification, it is also necessary to prepare the feature vectors of attackers. We applied two different methods for this. Firstly, we used the traditional method of extracting feature vectors from randomly sampled pre-collected data from other users. We created attacker feature vectors in the same way as in the case of genuine user data. Secondly, we synthesized feature vectors based on the distributions of individual feature values in the pre-collected user data set. This approach has been applied in prior work [14] and offers the advantage that it does not require storing any genuine data from other users. We generated 720 feature vectors using both of these methods and based on a set of 28 users data collected in an empirical study.

After preparing feature vectors for both a genuine user and attackers, the following pre-processing steps were applied:

1. Split data into train data and test data. We divided the feature vectors of the genuine user and the attackers into data for training a recognizer and data for evaluating verification performance. In the case of genuine user data, we use initial data for training to reflect a realistic unlock scenario during enrollment. We varied the set size of training data (nTrain) between 3 to 14

genuine user entries and sought to identify the optimal nTrain size, in terms of the verification performance of recognizers, via a grid-search procedure.

2. Re-sampling. We used the random re-sampling technique to match the amount of genuine user samples and attacker samples in the training data to reduce the influence of imbalanced classes on verification performance. The target re-sampling number of each class (nResampling) was varied from 20 to 720 in hops of 50 according to a grid-search procedure.
3. Feature standardization. The values of each feature were converted to a Z-score according to the statistical distributions in the training data.
4. Feature dimensionality reduction. As a large number of features can overestimate classification performance during training, we incorporated dimensionality reduction methods [20]. For this study, we used principal component analysis (PCA) and adjusted the retention percentage of variance explained by all of the selected components (PCATh) from 0.98 to 0.6 via grid-search.

After completing data pre-processing, two different types of classifiers were generated: one-class classifiers and binary classifiers. For both, we used a support vector machine (SVM) with a Radial Basis Function kernel, as this approach has been frequently used in behavioral biometrics research [20]. We applied a 10 fold cross-validation grid-search for tuning classifier hyperparameters [11].

The verification process determined whether a new PushID input was entered by a genuine user. Firstly the system checked whether the user correctly entered input according to guidance on screen. If this was correct. the feature vector was calculated for the input in the same way as during the enrollment process. Finally, this feature vector was submitted to the appropriate user classifier.

4 Data Collection Study

We performed an empirical study to explore the usability and security of PushID. We collected 551 valid PushID entries, and evaluated the verification performance of PushID in a simulated random guessing attack scenario. This study was approved by the local institutional review board (IRB).

4.1 Participants

A total of 30 participants (mean age = 23.07, σ = 2.66) were recruited through a post on a social media site for members of a local university. Among the participants, 18 were male and 12 were female. Left-handed people were excluded to increase the homogeneity of collected data. We recorded familiarity with smartphones, smartwatches, and pressure interaction on these devices via a questionnaire with 5-point Likert scales. Results indicated a high familiarity with smartphones (μ = 4.47, σ = 1.31), a low familiarity with smartwatches (μ = 1.33, σ = 0.76), moderate experience with pressure interaction with smartphones (μ = 2.27, σ = 1.51) and low experience with pressure on smartwatches (μ = 1.27, σ = 0.69). 5 USD in local currency was given as compensation for study participation. One participant showed low compliance with study instructions (in terms of accurate button and pressure level selection), so their data was excluded from all analyses. We report on data from the remaining 29 participants in this paper.

4.2 Procedure

The study was conducted in a silent laboratory environment while the participants sat in a chair without armrests. Each participant completed the below steps:

Instructions: The study started with a participant reading the study guide and then filling out the consent form. The participant could ask questions about the study at any time. The participant read paper guidelines about how to enter PushID items. The participant was guided to perform all input tasks as quickly and accurately as possible. Also, the participant had to keep their arm suspended in space while performing each input trial. There was no restriction on posture between trials; this minimized fatigue. In addition, we further reduced accumulated fatigue by mandating a break of at least 5 s after every 8 trials.

Input task: After receiving all the experimental instructions, participants put on the watch to perform the study input tasks. They achieved this by correctly selecting the yellow target button and the target pressure level displayed on the screen. The combination of target buttons and pressure levels were randomized for each trial, and each participant completed 19 trials.

4.3 Measures

To evaluate the usability of the PushID input task, *Input time*, the period between screen contact and release during each correct PushID entry was recorded. Based on collected behavioral data of the participants, verification FPR, FNR, and EER were explored for a wide set of classifier parameters using the grid-search procedures discussed in Sect. 3.3. This revealed how to generate a recognizer with the best possible EER.

5 Results and Discussion

In terms of usability measurements, the participants took a mean of 2.09 s ($\sigma = 1.68$) for each input. This represents an on par level of performance compared to both popular authentication schemes such as PIN (2.195 s) [14] and other behavioral biometrics-based authentication proposed for smartwatches [7,8,14]. In terms of resistance to random guessing attack, the best PushID recognizer used an nTrain of 14, nResampling of 520, and a PCATh of 0.98 with the binary classifier built using synthesized imposters—mean FPR was 1.05% ($\sigma = 0.76$), mean FNR was 42.76% ($\sigma = 22.50$), and mean EER was 8.34% ($\sigma = 8.06$). In the case of the real user imposter set, peak performance was achieved with nTrain of 14, nResampling of 720, and a PCATh of 0.98 with the binary classifier —mean FPR was 2.04% ($\sigma = 1.98$), mean FNR was 42.07% ($\sigma = 23.51$), and mean EER was 11.67% ($\sigma = 7.54$). Interpreting these results, we note that in the case of FPR, PushID shows improved values compared to other behavioral biometric authentication schemes for smartwatches, such as the 21.65% reported for Air-Sign [3], and the 7.2% in Beat-PIN [8]. However FNRs, and correspondingly, EERs are higher compared to these closely related schemes [3,8,14,21]. This result also showed a discussion point about the method to prepare imposter data to train and test the security of recognizers against random guessing attacks—the lower FPR when using synthesized imposter than real human data may mean that either the former method was effective at training the recognizer, or it led to weaker attacks than the latter one. Further work with collecting more user data is needed to systemically explore this point.

We can draw some wider conclusions from these results. The high FNR may occur because within-subject variability may be elevated by the fact that participants selected various random combinations of buttons and pressure levels. More consistent selections may lead to improved performance. Additionally, the high FNR values may be due to the limited size of train/test data in our current study; collecting an extended data set is a clear next step for this work. In addition, performance may be improved by considering multiple touch events, each featuring production of a different pressure level. Furthermore, PushID could also be combined with a knowledge-based authentication scheme involving entering a series of symbols each associated with a different button/pressure level combination. We see value in exploring these ideas in future work.

6 Conclusion

This paper proposes PushID, a behavioral biometric authentication system based on touch and motion traits extracted from five-level touch force input that seeks to achieve good usability and security as a lock system for a smartwatch. Our study indicates PushID ably resisted simulated random guessing attacks. Furthermore, it also achieves an input time that is on-par with other authentication techniques. We believe these results are promising and support the future development of the PushID pressure-based behavioral biometric concept as a viable smartwatch lock system.

Acknowledgement. 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).

References

1. Accot, J., Zhai, S.: Refining fitts' law models for bivariate pointing. In: Proceeding of The SIGCHI Conference on Human Factors in Computing Systems (2003). https://doi.org/10.1145/642611.642646
2. Brewster, S.A., Hughes, M.: Pressure-based text entry for mobile devices. In: Proceeding of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (2009). https://doi.org/10.1145/1613858.1613870

3. Buriro, A., Van Acker, R., Crispo, B., Mahboob, A.: AirSign: a gesture-based smartwatch user authentication. In: 2018 International Carnahan Conference on Security Technology (2018). https://doi.org/10.1109/CCST.2018.8585571
4. De Luca, A., Hang, A., Brudy, F., Lindner, C., Hussmann, H.: Touch me once and i know it's you! implicit authentication based on touch screen patterns. In: Proceeding of the SIGCHI Conference on Human Factors in Computing Systems (2012). https://doi.org/10.1145/2207676.2208544
5. Gil, H., Lee, D., Im, S., Oakley, I.: TriTap: identifying finger touches on smartwatches. In: Proceeding of the 2017 CHI Conference on Human Factors in Computing Systems (2017). https://doi.org/10.1145/3025453.3025561
6. Goguey, A., Malacria, S., Gutwin, C.: Improving discoverability and expert performance in force-sensitive text selection for touch devices with mode gauges. In: Proceeding of the 2018 CHI Conference on Human Factors in Computing Systems (2018). https://doi.org/10.1145/3173574.3174051
7. Guerar, M., Migliardi, M., Palmieri, F., Verderame, L., Merlo, A.: Securing pin-based authentication in smartwatches with just two gestures. Concurr. Comput. Pract. Exp. **32**(18), e5549 (2020). https://doi.org/10.1002/cpe.5549
8. Hutchins, B., Reddy, A., Jin, W., Zhou, M., Li, M., Yang, L.: Beat-Pin: a user authentication mechanism for wearable devices through secret beats. In: Proceeding of the 2018 on Asia Conference on Computer and Communications Security (2018). https://doi.org/10.1145/3196494.3196543
9. Knight, J.F., Baber, C.: Assessing the physical loading of wearable computers. Appl. Ergon. **38**(2), 237–247 (2007). https://doi.org/10.1016/j.apergo.2005.12.008
10. Krombholz, K., Hupperich, T., Holz, T.: Use the force: evaluating force-sensitive authentication for mobile devices. In: Twelfth Symposium on Usable Privacy and Security (2016). https://www.usenix.org/conference/soups2016/technical-sessions/presentation/krombholz
11. scikit learn: 3.2. tuning the hyper-parameters of an estimator (2021). https://scikit-learn.org/stable/modules/grid_search.html. Accessed 27 Apr 2021
12. Li, Y., Xie, M.: Understanding secure and usable gestures for realtime motion based authentication. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (2018). https://doi.org/10.1109/INFCOMW.2018.8406912
13. Lu, C.X., Du, B., Kan, X., Wen, H., Markham, A., Trigoni, N.: VeriNet: user verification on smartwatches via behavior biometrics. In: Proceeding of the First ACM Workshop on Mobile Crowdsensing Systems and Applications (2017). https://doi.org/10.1145/3139243.3139251
14. Nguyen, T., Memon, N.: Tap-based user authentication for smartwatches. Comput. Secur. **78**, 174–186 (2018). https://doi.org/10.1016/j.cose.2018.07.001
15. Oakley, I., Huh, J.H., Cho, J., Cho, G., Islam, R., Kim, H.: The personal identification chord: a four buttonauthentication system for smartwatches. In: Proceeding of the 2018 on Asia Conference on Computer and Communications Security (2018). https://doi.org/10.1145/3196494.3196555
16. Ranak, M.N., Azad, S., Nor, N.N.H.B.M., Zamli, K.Z.: Press touch code: a finger press based screen size independent authentication scheme for smart devices. PLOS One, **12**(10), e0186940 (2017). https://doi.org/10.1371/journal.pone.0186940
17. Reportlinker: global smartwatch market-growth, trends, covid-19 impact, and forecasts (2021–2026). yahoo!finance (2021). https://finance.yahoo.com/news/global-smartwatch-market-growth-trends-113500113.html
18. Salem, A., Obaidat, M.S.: A novel security scheme for behavioral authentication systems based on keystroke dynamics. Secur. Priv. **2**(2), e64 (2019). https://doi.org/10.1002/spy2.64

19. Shah, S.W., Kanhere, S.S.: Recent trends in user authentication-a survey. IEEE Access **7**, 112505–112519 (2019). https://doi.org/10.1109/ACCESS.2019.2932400
20. Teh, P.S., Zhang, N., Teoh, A.B.J., Chen, K.: A survey on touch dynamics authentication in mobile devices. Comput. Secur. **59**, 210–235 (2016). https://doi.org/10.1016/j.cose.2016.03.003
21. Zhang, H., Xiao, X., Ni, S., Dou, C., Zhou, W., Xia, S.: Smartwatch user authentication by sensing tapping rhythms and using one-class DBSCAN. Sensors **21**(7), 2456 (2021). https://doi.org/10.3390/s21072456