IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 31, 2019, accepted April 27, 2019, date of publication May 23, 2019, date of current version June 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2918647

Draw It As Shown: Behavioral Pattern Lock for Mobile User Authentication

YEEUN KU, LEO HYUN PARK, SOOYEON SHIN, (Member, IEEE), AND TAEKYOUNG KWON, (Member, IEEE)
Graduate School of Information, Yonsei University, Seoul 03722, South Korea

Corresponding author: Taekyoung Kwon (taekyoung@yonsei.ac.kr)

This work was supported in part by the Institute for Information and Communications Technology Promotion (IITP) Grant funded by the Korea Government (MSIT), Development of Next Generation User Authentication, under Grant 2017-0-00380.

ABSTRACT Android pattern lock is still popularly used for mobile user authentication. Unfortunately, however, many concerns have been raised regarding its security and usability. User-created patterns tend to be simply structured or reduced to a small set. Complex patterns are hard to memorize. Input patterns are susceptible to various attacks, such as guessing attacks, smudge attacks, and shoulder surfing attacks. This paper presents a novel mechanism based on the pattern lock, in which behavioral biometrics are employed to address these problems. Our basic idea starts from turning the lock pattern into public knowledge rather than a secret and leveraging touch dynamics. Users do not need to create their own lock patterns or memorize them. Instead, our system shows a public pattern along with guidance on how to draw it. All the user needs to do for authentication is to draw the pattern as shown. For adversaries, the above-mentioned attacks are rendered useless by this new mechanism. Specifically, we study how to generate the public patterns and how to perform authentication. We considered segments, angles, directions, and turns as units for constructing the lock patterns, and established the public pattern criteria. The results are utilized to generate four public patterns in our experiment. For authentication, we achieved equal error rates (EERs) as low as 2.66% (sitting), 3.53% (walking), and 5.83% (combined). Furthermore, the results of our additional experiments demonstrated that our system preserved performance over time (F1-score = 89.88%, $SD = 4.60\%$), and was sufficiently secure against camera-based recording attacks (FAR = 3.25%).

INDEX TERMS Behavioral authentication, android pattern lock, smartphone, machine learning.

I. INTRODUCTION

Smartphones have now become a part of our daily lives, and their functionality has significantly increased; hence, mobile user authentication has now turned into an essential mechanism for the security and privacy of users. Currently, various authentication methods such as PIN, passwords, biometrics, and pattern lock, are used among smartphone users, and each scheme has advantages and disadvantages [10], [26].

Android pattern lock, which is still widely used for mobile user authentication, dates back to the earlier recall-based systems such as Draw-A-Secret (DAS) [22] and Pass-Go [40]. Users are asked to create and memorize a graphical pattern on a 3×3 grid. For authentication, they should remember the pattern, and then draw it with a finger on the grid. According to recent studies, a considerable number of Android users prefer to use patterns rather than PINs or textual passwords for unlocking their devices [3], [50]. Although biometric authentication is being increasingly used and has been rapidly replacing traditional authentication mechanisms for over the last five years, the pattern lock is still a popular, frequently used unlocking mechanism. Particularly, it is crucial for secondary authentication. It is reported that approximately 40% of Android users adopt pattern lock into their devices [45]. Pattern lock is also adopted as a part of the login authentication method for many critical applications, such as Alipay, which has 450 million users in China [51].

Unfortunately, however, many researchers have raised concerns about the security and usability of the Android pattern lock system. Users tend to create simple patterns or select biased patterns from a small set [13], [44]. The reason for this is probably related to usability: as with other password

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaofan He.

mechanisms, complex patterns are difficult to memorize, take longer to type in, and are prone to input errors. In contrast, such weak passwords with low entropy are susceptible to password guessing attacks. The Android pattern lock can be vulnerable to guessing attacks unlike those that are expected from theoretical considerations [6], [38]. In addition, researchers have demonstrated various attack methods to threaten the pattern lock in this context: smudge, thermal, shoulder-surfing, and video-based attacks [1], [8], [27], [44], [47], [50], [53].

Behavioral authentication (or behavior-based authentication) is a prospective approach to user authentication. In behavioral authentication, the behavioral biometrics (i.e., touch dynamics and gait) of the user are measured by various sensors and leveraged for authentication. However, the performance of behavioral authentication in terms of accuracy heavily depends on the *shape of gestures* because not all gestures perform fairly for authentication [38]; that is, the high performance of behavioral authentication cannot be expected without a precisely defined shape of gestures. From the perspective of behavioral authentication, our concern is that the lock pattern of the pattern lock system can be used for the gesture shape.

In the vein of multi-factor authentication, the security of user authentication can be greatly increased if behavioral authentication is combined with the pattern lock system. However, problems persist. First, an adversary can still peek into a secret pattern drawn by a user (e.g., through shoulder-surfing attacks, smudge attacks, or occasionally guessing the poorly chosen pattern in a few attempts). Second, an adversary can mimic the user's drawing gestures by recording gestures using videos and practicing based on the recorded gestures. Our study is strongly motivated by these problems, particularly for the combination of behavioral authentication with a pattern lock system for user authentication.

In this paper, we raise the aforementioned problems and present a novel behavioral pattern lock referred to as *draw it as shown* (DIAS). The basic idea of DIAS is to let the lock pattern into public knowledge, instead of being a secret, and leverage touch dynamics for authentication. DIAS displays a public pattern as a shape of gestures along with guidance on how to draw it. As such, users will no longer need to create or memorize lock patterns. Therefore, DIAS makes various attacks on secret lock patterns useless. We study how to effectively build this breakthrough system and evaluate its performance and security through user studies.

The contributions of this study are as follows:

- We design and implement a new pattern lock mechanism, called DIAS (Section III-A and III-B). To the best of our knowledge, this is the first attempt of turning the lock pattern into a public value/knowledge by virtue of touch dynamics. We also define the criteria of public patterns to assure a shape of gestures required for good accuracy in authentication (Section III-C).
- Based on the data set assembled (7200 samples collected from 30 participants), we explore the best-performing classifiers and optimal feature sets for DIAS (Sections IV-A and IV-B). We then evaluate the authentication performance from the perspective of both short- and long-term periods (referred to as permanence) under the optimal settings (Sections IV-C and IV-D). With regard to permanence, we present a novel model training strategy, called the *sliding window approach*, which considers user behavior changes and outperforms the existing approaches.
- We evaluate the security of DIAS against recording attacks (Section V-A) and compare DIAS to a pattern lock system in terms of running time and error rate (Section V-B).

Moreover, we discuss the implications and limitations of our study (Section VI). We then review the related works (Section VII) and conclude the paper (Section VIII).

II. BACKGROUND

A. MOBILE USER AUTHENTICATION

As mobile devices increasingly contain the private information of users, the need for user authentication techniques has steadily grown. There are several mechanisms for smartphone user authentication, including PIN, passwords, biometrics, and pattern lock. It is widely known that each method has advantages and disadvantages.

PINs and passwords comprising a sequence of digits or characters are the most prevalent authentication mechanisms, and can also be used for secondary authentication mechanisms when biometric fails [46]. However, PINs and passwords are known to be easily guessed and vulnerable to smudge and shoulder surfing attacks.

Biometrics, such as fingerprint and facial recognition, have also been widely adopted for authentication. Authenticating with biometrics is very simple and convenient; hence, it is very highly preferred by users. However, biometrics is susceptible to various attacks that decrease security. For example, a user's fingerprint can be exposed and exploited by a smudge attack [29]. A previous research indicated that usability is the most important factor to consider when deciding whether to use biometrics [16]. Despite the common thought that physical biometrics provide convenience, users often feel uncomfortable when using it [43]. Fingerprints are sensitive to finger conditions, such as moisture and dust. Likewise, facial recognition requires moving the camera to an angle where the face is visible. The camera can be blinded in the presence of strong sunlight, direct sunlight, or darkness [21], [23], [31]. In addition, users are often required to register with secondary authentication because authentication errors frequently occur when using biometrics in the real world [7], [24], [32].

B. ANDROID PATTERN LOCK

The Android pattern lock is one of the graphical password mechanisms used to protect sensitive data. Users have to create and memorize a secret pattern, then draw it with a finger for authentication on a 3×3 grid. The pattern must

be connected to at least four dots, and a complex pattern must be selected to provide greater security. The Android pattern lock is still widely accepted as a main authentication mechanism, and is also used as a secondary authentication mechanism when biometric authentication fails. A considerable number of Android users prefer pattern locks instead of PINs or textual passwords [3], [50]. Reportedly, approximately 40% to 48% of Android users choose patterns to protect their devices [49]. Unfortunately, many researchers have raised concerns regarding the security and usability of the Android pattern lock.

Theoretically, the number of possible patterns is 389,112, which is a larger password space than PINs or passwords [8], [14]. However, pattern lock users are generally known to choose weak patterns [13], [38], [44] because complex patterns are difficult to memorize, take longer to type in, and are prone to typing errors, thereby reducing usability.

In aspects of security, the pattern lock is threatened by various attack methods: guessing attacks [38], [44], shoulder-surfing attacks [47], smudge attacks [5], [8], thermal attacks [1], and even side-channel attacks using Wi-Fi signals [53]. According to [50] and [54], even complex patterns can be vulnerable to video-based shoulder-surfing attack and acoustic signal-based attack. In summary, user-created patterns are vulnerable to various types of pattern-inferring attacks regardless of their simplicity or complexity. Therefore, we propose a novel behavioral authentication mechanism based on the pattern lock interface that many users prefer, thereby addressing the abovementioned usability and security issues of the pattern lock.

C. BEHAVIORAL AUTHENTICATION

Behavioral authentication is an authentication method based on human behavior characteristics, such as signature, voice, gait, keystroke, and touch dynamics, which capture the user's behavior from the touchscreen and the sensors when touching the screen. Behavioral authentication is ideal for mobile devices with a touchscreen and various sensors. This is also a prospective approach that improves security because it can be leveraged for front-end and continuous authentication. However, behavior authentication faces a problem in that its authentication performance depends on the *shape of gestures* [36]. We are concerned that the problem applies to patterns as well. We intend to resolve this challenge by designing the criteria of public patterns for performance improvement and applying it to DIAS.

One might ask why it is beneficial to combine the user's behavior characteristics with a pattern lock interface. We answer that such a combination could be advantageous for the improvement of the authentication performance. First, the regular 3×3 grid provided in the pattern lock allows the user to draw patterns easily, whereas gestures do not provide a guidance background. Second, we can extract formal features from a limited space, called *segment*. This concept is introduced in Section III-B) in the grid, and does not need to perform additional feature processing. It is more efficient than gestures in terms of resource efficiency.

D. THREAT MODEL AND ASSUMPTIONS

We consider the following threat model commonly adopted in related studies. An attacker is in close vicinity and may be a stranger or even a malicious insider, such as a close acquaintance, a colleague, or a friend. Considering widely deployed mobile cameras and surveillance cameras, we assume that the attacker may be capable of recording and taking a glance at the user's authentication steps, including input patterns and behavioral gestures. Furthermore, the attacker may acquire physical access to the smartphone in the user's absence, then unlock the smartphone. The biggest problem is that the Android pattern lock eventually just checks whether the shape of the input pattern is correct. Users are supposed to keep the created patterns secret, but attackers can easily know those patterns using various attacks, such as guessing attacks, shoulder-surfing attacks, smudge attacks, and video-based attacks, acquiring all authorities on the target devices. Our insight is to publicly present the shape of the pattern rather than making it secret, and to use the pattern shape as a guide for acquiring the behavior information of users. As a consequence, we can eliminate the value of those attacks which solely tried to acquire the shape of patterns, not the specific user behavior. However, threats utilizing video or sensor attacks may still remain. These attacks may capture not only the shape of patterns, but also a user's input behavior. The great concern here is that the attacker might mimic the user1s behavior by training with the recorded information.

III. DIAS SYSTEM

This section describes the DIAS system in terms of acquiring touch dynamics from the touch screen and sensors, extracting useful features to classify users from the acquired touch dynamics, and which types of classifiers should be considered. We also define the criteria of public patterns by considering the user preferences, ease of drawing patterns, and accuracy of patterns drawn.

A. DESIGN

DIAS is a novel behavioral pattern lock system that does not require users to create or memorize lock patterns; instead, it turns a lock pattern into public knowledge and leverages touch dynamics for authentication. DIAS shows a public pattern along with guidance on how to draw it and authenticates a user through the behavior differences that occur when drawing the public pattern. Figure 1 illustrates the user interface of DIAS and an authentication example with a virtual finger to aid explanation. A public pattern is shown on the top grid and a user draws it on the bottom grid, viewing the given pattern. Visual feedback is given as the user draws. DIAS grants access to the smartphone if the behavior data captured when drawing the pattern are determined to belong to that user.

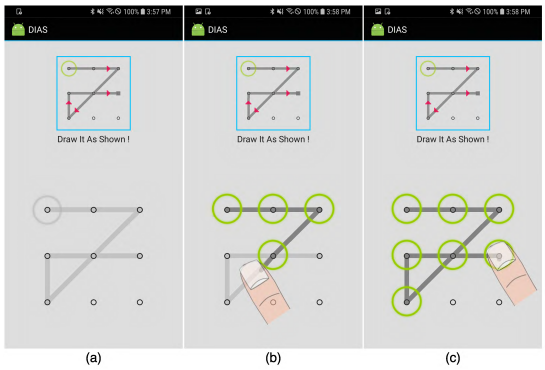

FIGURE 1. DIAS user interface and an authentication example. (a) The upper grid shows a public pattern with a circle and directional arrows. The lower grid also shows a sketch of the pattern. (b) The user draws the pattern on the lower grid as shown. (c) Visual feedback is given while drawing.

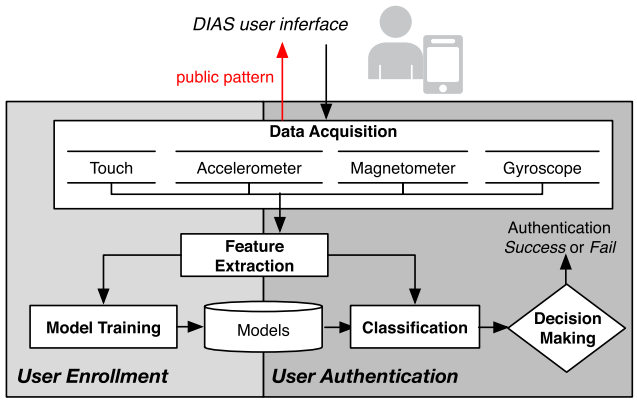

FIGURE 2. Overview of DIAS consisting of user enrolment and user authentication phases.

Figure 2 gives an overview of the DIAS system comprising five modules: data acquisition, feature extraction, model training, classification, and decision making. Utilizing those modules, the DIAS system takes two phases: *user enrollment* and *user authentication*. In the enrolment phase, a user must repeatedly draw the given m (≥ 1) patterns r times. The data acquisition module acquires the touch dynamics when the user draws a pattern. The feature extraction module extracts multiple distinctive features from the obtained touch dynamics and selects the optimal subset of all features for optimizing the authentication performance. Through a machine learning technique, the model training module uses the selected optimal feature set as input and trains a model (referred to as a classifier) that uniquely represents the user.

In the authentication phase, DIAS displays one of the patterns registered in the enrolment phase, and the user is required to draw this pattern for authentication. When the user draws the pattern, the data acquisition module acquires the touch dynamics, and the feature extraction module extracts the features contained in the optimal feature subset. The classification module employs the extracted features as the test sample for the model trained in the model training module. The decision-making module determines whether the test sample belongs to that user; if so, DIAS grants access to the smartphone; otherwise, it rejects the user.

B. IMPLEMENTATION

Various built-in sensors are installed in current smartphones, from which we select a sensor that can accurately represent the user's behavioral characteristics while the public pattern is drawn. In DIAS, we consider the touchscreen and the following three sensors that are commonly embedded in current smartphones: accelerometer, gyroscope, and magnetometer. The touchscreen and the three sensors represent different degrees of information regarding the user's behavior. The touchscreen logs the physical interaction, such as how the user draws; the accelerometer records the user's large motion patterns, such as how the user walks; the gyroscope reflects the user's fine-grained motion, such as how the user holds the smartphone; and the magnetometer measures the ambient magnetic field. They collect the touch dynamics herein.

1) FEATURES

The touch dynamics obtained are used for feature extraction on a *segment* basis to obtain the granular and unique characteristics of each user. A segment implies a single line that connects point-to-point (i.e., dot-to-dot) on a grid [38]. For instance, as shown in Figure 3, the number of segments of the S1 pattern and the S2 pattern is one and two, respectively.

In DIAS, a user must touch the touchscreen with his/her fingertip and continue sliding to draw a pattern in a single movement. At the same time, the touch events capturing the changes in the state of contacting the touchscreen surface are acquired. The touch events are intimately related to the physical interactions between the touchscreen and the fingertip movements of each user [42]. In other words, the touch events can differ for users, although the pattern shapes are identical; hence, they need to be considered first in DIAS. We extract 16 features from touch events, which are listed in Table 1, including statistical information (i.e., *avg*. or *std*.) to avoid noise [49].

TABLE 1. Features from the touch events (a total of 16 types).

Touch events	Description
numTE	Number of touch events of each segment
avgTP	Average of touch pressure of each segment
stdTP	Standard deviation of touch pressure of each segment
maxTP	Maximum of touch pressure of each segment
minTP	Minimum of touch pressure of each segment
avgTS	Average of touch size of each segment
stdTS	Standard deviation of touch size of each segment
maxTS	Maximum of touch size of each segment
minTS	Minimum of touch size of each segment
avgSS	Average of sliding speed of each segment
stdSS	Standard deviation of sliding speed of each segment
maxSS	Maximum of slide speed of each segment
minSS	Minimum of slide speed of each segment
SS	Slide speed when moving from one point to the next
SA	Slide angle when moving from one point to the next
TD	Time duration of each segment

The user's touch actions that occur while the user is drawing the pattern typically cause subtle tilts, micro-movements, and orientation changes of the smartphone. These changes

can be profiled by leveraging the accelerometer, gyroscope, and magnetometer built into current smartphones. In particular, for tilt correction, we obtain the orientation data reflecting the orientation changes of the smartphone. The orientation data are calculated with the magnetometer and the accelerometer from the rotation matrix. The `getRotationMatrix` method lets us translate the magnetometer readings into a fixed global coordinate system. Accordingly, the traces can be easily compared, regardless of the attitude changes that occur on the compared smartphones. We also consider the magnitude of each sensor datum. The *magnitude* is very effective for smartphone user authentication [12]. We derive the magnitude $S_M = \sqrt{s_x^2 + s_y^2 + s_z^2}$, where s_x, s_y, and s_z are the readings obtained from each sensor along the x, y, and z-axis, respectively. Table 2 lists 90 features of the three sensors.

TABLE 2. Features from sensors (a total of 90 types).

Accelerometer	Magnetometer	Gyroscope	Description
avgAX	*avgOX*	*avgGX*	Average x-axis value of each sensor
avgAY	*avgOY*	*avgGY*	Average y-axis value of each sensor
avgAZ	*avgOZ*	*avgGZ*	Average z-axis value of each sensor
avgAM	*avgOM*	*avgGM*	Average magnitude value of each sensor
numPAX	*numPOX*	*numPGX*	Number of positive x-axis values of each sensor
numPAY	*numPOY*	*numPGY*	Number of positive y-axis values of each sensor
numPAZ	*numPOZ*	*numPGZ*	Number of positive z-axis values of each sensor
numNAX	*numNOX*	*numNGX*	Number of negative x-axis values of each sensor
numNAY	*numNOY*	*numNGY*	Number of negative y-axis values of each sensor
numNAZ	*numNOZ*	*numNGZ*	Number of negative z-axis values of each sensor
stdAX	*stdOX*	*stdGX*	Standard deviation of x-axis value of each sensor
stdAY	*stdOY*	*stdGY*	Standard deviation of y-axis value of each sensor
stdAZ	*stdOZ*	*stdGZ*	Standard deviation of z-axis value of each sensor
stdAM	*stdOM*	*stdGM*	Standard deviation of magnitude value of each sensor
maxAX	*maxOX*	*maxGX*	Maximum x-axis value of each sensor
maxAY	*maxOY*	*maxGY*	Maximum y-axis value of each sensor
maxAZ	*maxOZ*	*maxGZ*	Maximum z-axis value of each sensor
maxAM	*maxOM*	*maxGM*	Maximum magnitude value of each sensor
minAX	*minOX*	*minGX*	Minimum x-axis value of each sensor
minAY	*minOY*	*minGY*	Minimum y-axis value of each sensor
minAZ	*minOZ*	*minGZ*	Minimum z-axis value of each sensor
minAM	*minOM*	*minGM*	Minimum magnitude value of each sensor
skewAX	*skewOX*	*skewGX*	Skewness of x-axis of each sensor
skewAY	*skewOY*	*skewGY*	Skewness of y-axis of each sensor
skewAZ	*skewOZ*	*skewGZ*	Skewness of z-axis of each sensor
skewAM	*skewOM*	*skewGM*	Skewness of magnitude of each sensor
kurAX	*kurOX*	*kurGX*	Kurtosis of x-axis of each sensor
kurAY	*kurOY*	*kurGY*	Kurtosis of y-axis of each sensor
kurAZ	*kurOZ*	*kurGZ*	Kurtosis of z-axis of each sensor
kurAM	*kurOM*	*kurGM*	Kurtosis of magnitude of each sensor

2) CLASSIFIERS

Machine learning is categorized as supervised learning and unsupervised learning. Supervised learning needs a training phase, and the output of the test data is labeled. In contrast, unsupervised learning can group test data without the training phase, but the label of output cannot be identified. We adopt supervised learning, especially classification, because we need to identify the ownership of the input data. We collect the training data from the enrolment phase in Figure 2.

We adopt *multi-class* classification herein. *One-class* is more realistic, but does not guarantee high authentication performance [11]. We consider two models for authentication in the multi-class classification: *discrimination* and *authentication* [49]. The discrimination model identifies more than two users; thus, more than two classes are defined. The authentication model verifies a valid user and an invalid user; thus, only two classes are defined. One problem of the authentication model is that the system collects the data of other invalid users for training [49]. We mitigate this problem by storing and using the data of all users using the same smartphone to train the invalid users' data.

We used the following six classifiers that are known to be effective: Decision Tree (DT), Support Vector Machine (SVM), k-Nearest Neighbor (kNN), Gaussian Naive Bayes (GNB), Random Forest (RF), and Logistic Regression (LR) [35], [42].

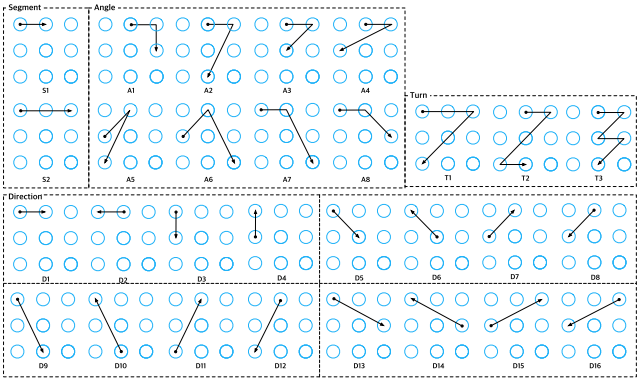

FIGURE 3. Unit patterns used to design the public pattern criteria in terms of different properties (i.e., segment, angle, turn, and direction).

C. PUBLIC PATTERN CRITERIA

The public pattern is a key element for collecting touch dynamics, which can have a significant impact on the classification/authentication performance. We need to define the criteria for designing public patterns by considering the accuracy and the user's preference. For this purpose, we conducted an experiment using a total of 29 unit patterns (Figure 3) to measure their accuracy with different attributes and see how the attributes in each property affect the authentication performance. The lock patterns generally have four properties: number of segments on a straight line, angle, direction, and number of turns. The direction unit patterns are particularly grouped into subgroups by angle. The possible number of unique attribute values in four properties is 29. The unit patterns in the same property have different values, but the values of the other properties that they do not belong to are fixed. We compare the difference in the classification performance among different attribute values for each property.

For the study that includes further experiments, we installed DIAS on a Samsung Galaxy S8 running Android OS 8.0. We used scikit-learn [33], a Python-based machine learning library, to implement the six classifiers.

We recruited 20 participants (two females and 18 males) by promotion via social networking services (SNSs). The average age of the participants was 27.7 years ($SD = 5.7$). We asked them to draw 29 unit patterns at least 25 times using DIAS on a given device while sitting. We then asked several questions to check their preferences about the properties and their attributes.

We applied six classifiers to the collected data using each of the 106 individual features. We used the discrimination model

TABLE 3. Results of the statistical tests for the unit experiment, including p-values. An inequality sign between the patterns indicates the difference in their accuracy. "sig." means significant differences at the 0.05 level. " n.s." means that the differences are not significant. "B-C Wilcoxon signed rank" stands for Bonferroni-corrected Wilcoxon signed rank.

Property	Test	Pattern	DT (decision tree)		SVM (support vector machine)		kNN (k-nearest neighbor)		GNB (gaussian naive bayes)		RF (random forest)		LR (logistic regression)	
			p	Results	*p*	Results	*p*	Results	*p*	Results	*p*	Results	*p*	Results
Segment	B-C Wilcoxon signed rank	S1,S2	0.000	S2>S1	0.000	S2>S1	0.000	S2>S1	0.000	S2>S1	0.000	S2>S1	0.000	S2>S1
Angle	Friedman	A1–A8	0.000	sig.	0.000	sig.	0.000	sig.	0.000	sig.	0.000	sig.	0.000	sig.
	B-C Wilcoxon signed rank	A1,A2	0.417	n.s.	0.321	n.s.	0.421	n.s.	0.138	n.s.	0.186	n.s.	0.416	n.s.
		A1,A3	0.362	n.s.	0.279	n.s.	0.467	n.s.	0.061	n.s.	0.437	n.s.	0.490	n.s.
		A1,A4	0.034	n.s.	0.001	A1>A4	0.024	n.s.	0.000	A1>A4	0.017	n.s.	0.045	n.s.
		A1,A5	0.000	A1>A5	0.001	A1>A5	0.001	A1>A5	0.000	A1>A5	0.000	A1>A5	0.000	A1>A5
		A1,A6	0.046	n.s.	0.032	n.s.	0.010	n.s.	0.217	n.s.	0.018	n.s.	0.026	n.s.
		A1,A7	0.011	n.s.	0.000	A7>A1	0.014	n.s.	0.015	n.s.	0.003	n.s.	0.006	n.s.
		A1,A8	0.020	n.s.	0.004	n.s.	0.006	n.s.	0.005	n.s.	0.016	n.s.	0.014	n.s.
		A2,A3	0.448	n.s.	0.185	n.s.	0.273	n.s.	0.256	n.s.	0.399	n.s.	0.419	n.s.
		A2,A4	0.094	n.s.	0.003	n.s.	0.091	n.s.	0.000	A2>A4	0.010	n.s.	0.071	n.s.
		A2,A5	0.000	A2>A5	0.003	n.s.	0.001	A2>A5	0.000	A2>A5	0.000	A2>A5	0.000	A2>A5
		A2,A6	0.011	n.s.	0.013	n.s.	0.001	A6>A2	0.018	n.s.	0.019	n.s.	0.010	n.s.
		A2,A7	0.014	n.s.	0.000	A7>A2	0.018	n.s.	0.002	n.s.	0.029	n.s.	0.031	n.s.
		A2,A8	0.047	n.s.	0.001	A8>A2	0.010	n.s.	0.000	A8>A2	0.039	n.s.	0.027	n.s.
		A3,A4	0.062	n.s.	0.004	n.s.	0.027	n.s.	0.000	A3>A4	0.022	n.s.	0.028	n.s.
		A3,A5	0.000	A3>A5	0.001	A3>A5	0.000	A3>A5	0.000	A3>A5	0.000	A3>A5	0.000	A3>A5
		A3,A6	0.006	n.s.	0.003	n.s.	0.013	n.s.	0.022	n.s.	0.010	n.s.	0.008	n.s.
		A3,A7	0.002	n.s.	0.000	A7>A3	0.011	n.s.	0.000	A7>A3	0.006	n.s.	0.008	n.s.
		A3,A8	0.013	n.s.	0.000	A8>A3	0.019	n.s.	0.000	A8>A3	0.027	n.s.	0.013	n.s.
		A4,A5	0.004	n.s.	0.056	n.s.	0.009	n.s.	0.031	n.s.	0.006	n.s.	0.003	n.s.
		A4,A6	0.000	A6>A4	0.000	A6>A4	0.000	A6>A4	0.000	A6>A4	0.000	A6>A4	0.000	A6>A4
		A4,A7	0.000	A7>A4	0.000	A7>A4	0.001	A7>A4	0.000	A7>A4	0.000	A7>A4	0.001	A7>A4
		A4,A8	0.004	n.s.	0.000	A8>A4	0.001	A8>A4	0.000	A8>A4	0.000	A8>A4	0.001	A8>A4
		A5,A6	0.000	A6>A5	0.000	A6>A5	0.000	A6>A5	0.000	A6>A5	0.000	A6>A5	0.000	A6>A5
		A5,A7	0.000	A7>A5	0.000	A7>A5	0.000	A7>A5	0.000	A7>A5	0.000	A7>A5	0.000	A7>A5
		A5,A8	0.000	A8>A5	0.000	A8>A5	0.000	A8>A5	0.000	A8>A5	0.000	A8>A5	0.000	A8>A5
		A6,A7	0.344	n.s.	0.033	n.s.	0.434	n.s.	0.103	n.s.	0.495	n.s.	0.470	n.s.
		A6,A8	0.475	n.s.	0.117	n.s.	0.370	n.s.	0.055	n.s.	0.457	n.s.	0.459	n.s.
		A7,A8	0.184	n.s.	0.168	n.s.	0.425	n.s.	0.244	n.s.	0.465	n.s.	0.451	n.s.
Direction	Friedman	D1–D4	0.185	n.s.	0.487	n.s.	0.514	n.s.	0.359	n.s.	0.414	n.s.	0.391	n.s.
	Friedman	D5–D8	0.822	n.s.	0.161	n.s.	0.253	n.s.	0.553	n.s.	0.893	n.s.	0.221	n.s.
	Friedman	D9–D12	0.000	sig.	0.000	sig.	0.000	sig.	0.000	sig.	0.000	sig.	0.000	sig.
	B-C Wilcoxon signed rank	D9,D10	0.000	D9>D10	0.000	D9>D10	0.000	D9>D10	0.000	D9>D10	0.000	D9>D10	0.000	D9>D10
		D9,D11	0.487	n.s.	0.118	n.s.	0.098	n.s.	0.131	n.s.	0.388	n.s.	0.484	n.s.
		D9,D12	0.000	D9>D12	0.000	D9>D12	0.000	D9>D12	0.000	D9>D12	0.000	D9>D12	0.000	D9>D12
		D10,D11	0.000	D11>D10	0.001	D11>D10	0.000	D11>D10	0.000	D11>D10	0.000	D11>D10	0.000	D11>D10
		D10,D12	0.212	n.s.	0.007	D10>D12	0.174	n.s.	0.040	n.s.	0.130	n.s.	0.161	n.s.
		D11,D12	0.000	D11>D12	0.000	D11>D12	0.000	D11>D12	0.000	D11>D12	0.000	D11>D12	0.000	D11>D12
	Friedman	D13–D16	0.168	n.s.	0.403	n.s.	0.470	n.s.	0.358	n.s.	0.037	sig.	0.195	n.s.
	B-C Wilcoxon signed rank	D13,D14	-	-	-	-	-	-	-	-	0.260	n.s.	-	-
		D13,D15									0.014	n.s.		
		D13,D16									0.037	n.s.		
		D14,D15									0.010	n.s.		
		D14,D16									0.068	n.s.		
		D15,D16									0.029	n.s.		
Turn	Friedman	T1–T3	0.237	n.s.	0.000	sig.	0.311	n.s.	0.005	sig.	0.082	n.s.	0.529	n.s.
	B-C Wilcoxon signed rank	T1,T2	-	-	0.108	n.s.	-	-	0.082	n.s.	-	-	-	-
		T1,T3			0.001	T3>T1			0.001	T3>T1				
		T2,T3			0.001	T3>T2			0.008	T3>T2				

for six classifiers to measure the classification accuracy of the unit patterns with a five-fold cross-validation. With the measurements of the six classifiers, we performed statistical tests to ascertain whether a difference existed in the accuracy of the unit patterns with different attributes. Statistical tests were conducted to compare the average accuracies of individual features of unit patterns with different attributes. The significance level α was set at 5%. The null hypothesis was that no significant differences existed in accuracy to discriminate a participant among the patterns with different attribute values. The Friedman test was conducted to determine the overall significance of the differences between the patterns with two attribute values. The Friedman test results showed that if a difference existed, the Bonferroni-corrected Wilcoxon signed-rank test must again be performed as a post-hoc analysis by grouping two attribute values among all attribute values used in the Friedman test. Table 3 presents the statistical results, including the p-values of the Friedman and Bonferroni-corrected Wilcoxon signed-rank tests.

The pattern with two segments had a higher accuracy, and no significant difference was found in user preference for the patterns with different segments. Slight differences in the accuracy of angles for each classifier were found, but overall, the patterns with angles 75°, 120°, and 135° had the highest accuracy, and the patterns with angles 15° and 30° had the lowest. For the angle property, a large difference was observed between the accuracy comparison result and the user preference; thus, we set its criterion by considering both accuracy and preference. Even if the three angle values with the highest accuracy yield a better performance, we chose the other three angle values that had moderate accuracy and better user preference. No significant difference was found in the accuracy of directions for each classifier, except in the 60° subgroup. In the 60° subgroup, for all classifiers, the patterns

with upper left to lower right direction and lower left to upper right direction had a higher accuracy than the other two patterns. In the case of user preference, regardless of the subgroup, most participants said they preferred the direction from left to right. No significant difference was found in the preference between the downward and upward directions. For the property of the number of turns, no significant difference was found, although the participants preferred patterns with fewer turns. However, the experimental results demonstrated that the pattern with three turns had the highest accuracy.

We set the design criteria of the DIAS public pattern as follows based on the results of the experiments and the user preference:

- Lines with two segments are first considered.
- The angles of 45°, 60°, and 90° are mainly used instead of any other angles.
- The left-to-right direction is first used at the beginning of the pattern. All directions are available in other cases.
- The number of turns is preferred by having over three turns.

We derived four public patterns reflecting the abovementioned design criteria (Figure 4). These patterns will be used for the experimental study and evaluation.

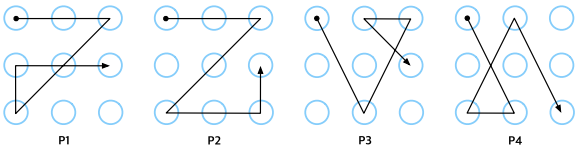

FIGURE 4. Four public patterns satisfying the criteria for the public patterns. We split these patterns into unit patterns and used them in the experimental study.

IV. PERFORMANCE EVALUATION

A. DATA COLLECTION

For the experimental study, we recruited 30 participants (11 females, 19 males) through word of mouth, SNSs, and fliers on bulletin boards in a university. They were students, general office workers, and university staff aged between 24 and 44 years ($M = 28$ y, $SD = 3.97$). All participants were smartphone users (13 Android users and 17 iPhone users), and most of them had experience using the pattern lock system.

We explained the purpose and process of all the experiments to the participants before the experiment. Each participant signed an explicit agreement to participate in the experimental study. We also informed them that their touch dynamics would be collected for the experiments. We gave a Samsung Galaxy S8 device to each participant. They were asked to draw a public pattern 30 times in each of the two postures (i.e., sitting and walking). We required them to use the device as usual, except restricting their postures to make realistic settings. We collected a total of 7200 samples for the experiments in Sections IV-B and IV-C. A few volunteers conducted the attack experiment or the comparison experiment the day after the authentication experiment. All participants basically received a gift of $15 for participation. Each participant completed the experiments within five days. All experiments were completed after approximately three months.

We recruited 10 volunteers from among 30 participants for the long-term experiment in Section IV-D. We asked them to draw four public patterns with the same settings daily for 20 days. We asked each volunteer to draw each pattern at least 25 times a day. We collected a total of 40,000 samples from the volunteers. Those volunteers received an additional $10 reward. In addition, we deployed other 10 participants for the security experiment (Section V-A) and the remaining 10 participants for the comparison experiment (Section V-B).

B. BEST-PERFORMING CLASSIFIERS AND OPTIMAL FEATURE SETS

1) EXPERIMENT DESIGN

The authentication performance of behavioral authentication may depend on which classifier is used and which combination of features is applied to the classifier. In this respect, we need to check which classifier and feature sets are effective in terms of user discrimination. For this purpose, we applied the *discrimination model* and evaluated 106 individual features using only one feature at a time. We measured the F1-score as the discrimination performance of an individual feature. Each execution was based on a 10-fold cross validation.

We applied the recursive feature elimination (RFE) to select the optimal feature set for each classifier and avoid overfitting issues. In RFE, we used the F1-score of individual features as their weights. We constructed a feature set in descending order by their weights. We recursively evaluated the performance of the remaining feature set, pruning the individual feature with the smallest weight. This process was performed until the remaining feature sets became empty. This feature selection was performed for each of the six classifiers and the three postures.

TABLE 4. Average accuracy and F1-score of each classifier with an optimal feature set in each posture (%). Classifiers are denoted with abbreviations as Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Gaussian Naive Bayes (GNB), Random Forest (RF), and Logistic Regression (LR).

Classifiers	Sitting		Walking		Combined	
	Acc.	F-sco.	Acc.	F-sco.	Acc.	F-sco.
DT	92.28	92.42	73.89	72.90	82.65	82.34
SVM	95.69	95.54	85.83	85.11	91.61	91.45
KNN	95.28	95.50	68.61	66.30	85.40	85.39
GNB	**97.19**	**96.90**	**95.25**	**94.59**	**95.97**	**95.26**
RF	97.53	97.49	91.58	90.49	94.53	93.77
LR	96.47	96.02	89.17	88.00	92.11	91.03

2) RESULT

Table 4 shows the highest accuracy and F1-score of each posture and each classifier from RFE. Table 5 presents the

TABLE 5. F1-score of individual features from the touchscreen and each sensor (a total of 106 types of features ranked in a descending order by F1-score).

Rank 1 (top) ∼ Rank 53 (bottom)						Rank 54 (top) ∼ Rank 106 (bottom)					
Sitting (GNB)	F1-score	Walking (GNB)	F1-score	Combine (GNB)	F1-score	Sitting (GNB)	F1-score	Walking (GNB)	F1-score	Combine (GNB)	F1-score
minOX	69.89	avgTS	54.88	avgTS	54.55	stdGM	35.56	numPGY	21.58	stdOM	29.80
maxOX	69.21	maxTS	50.63	maxTS	49.28	TD	35.34	stdAX	21.30	maxGZ	29.31
avgOX	68.14	minTS	45.86	avgOX	47.06	maxOZ	35.30	maxAM	20.81	stdAY	28.79
avgOM	62.89	SS	38.98	minOX	46.84	numTE	35.16	minGZ	20.48	stdGM	28.70
minOM	61.69	numTE	38.67	avgOM	46.01	numPAZ	34.97	numPOX	19.23	stdOX	28.29
maxOM	61.38	avgSS	38.42	maxOX	45.85	stdGX	34.13	stdAZ	19.06	stdGX	28.16
avgGZ	55.19	TD	38.22	minTS	45.42	stdTS	33.98	numNOX	18.93	numNGZ	27.54
avgGX	54.31	numPAZ	37.82	minOM	43.73	stdAZ	32.56	stdAM	18.74	numPGX	26.67
avgTS	54.23	numNOY	37.55	avgGX	42.24	maxAM	32.04	minAM	18.52	maxAM	26.43
avgGY	52.96	numPAY	37.55	maxOM	42.02	minAM	31.42	avgAM	18.43	stdAZ	25.81
minGZ	49.52	numNAX	33.35	avgGY	41.55	numPGZ	31.21	stdOY	18.42	minSS	25.54
avgAZ	49.49	numPOZ	33.33	avgAZ	38.79	numPGX	31.13	numNOZ	17.61	numPGY	25.17
avgAM	48.24	minOZ	33.29	SS	38.39	numPGY	28.76	numPAX	17.61	minAM	24.97
maxTS	47.93	avgAX	32.62	minOY	38.30	minGM	27.96	numNGZ	17.37	numPGZ	23.63
maxAY	47.32	avgOZ	32.14	minOZ	37.34	stdAM	27.82	avgGZ	16.70	stdAM	23.28
avgAY	47.20	maxAX	31.83	avgSS	37.15	numNOZ	24.14	maxGZ	16.41	SA	22.82
avgGM	47.19	minAX	31.30	numNOY	37.07	numPAX	24.14	stdOM	16.28	stdSS	21.41
avgOY	47.03	stdTS	30.21	numPAY	37.07	minSS	23.79	numPGZ	16.06	numNOZ	20.87
minOY	46.51	avgGX	30.18	numTE	36.91	SA	22.94	stdOX	15.76	numPAX	20.87
minTS	44.98	avgGY	30.13	TD	36.78	stdSS	20.65	stdAY	15.30	minGM	19.63
numPOX	44.94	minOY	30.09	avgOY	36.54	skewGZ	17.88	maxSS	13.96	numNOX	16.07
maxGY	44.85	avgOM	29.14	avgGM	36.47	skewGY	15.81	minGM	11.29	maxSS	13.59
minGY	44.45	maxOZ	28.89	numPAZ	36.39	skewGX	15.60	skewGX	9.03	skewGX	12.32
minAZ	44.11	numNGX	28.22	maxAY	36.33	skewOX	14.57	skewGY	8.74	skewGY	12.27
maxGM	43.85	avgAZ	28.09	avgAY	36.27	skewGM	13.79	kurAX	8.36	skewGZ	12.19
numNGX	43.37	maxGY	27.38	maxGY	36.12	skewOM	13.60	kurOZ	7.94	skewOX	10.58
stdOM	43.32	minSS	27.29	maxAX	36.07	skewOZ	13.28	kurGY	7.77	skewOM	10.32
minAY	42.80	stdGY	26.71	avgOZ	36.01	maxSS	13.22	kurGZ	7.52	kurGY	10.25
maxOY	42.64	minGX	26.69	avgGZ	35.95	numNOX	13.20	kurGM	7.41	kurGZ	10.06
stdAY	42.29	avgOY	26.06	avgAX	35.84	skewAY	12.96	skewOM	7.05	skewGM	10.05
stdGY	42.28	minGY	26.02	numNGX	35.80	kurAY	12.87	skewOX	6.58	skewOZ	9.78
maxGZ	42.22	avgOX	25.98	minGY	35.24	kurGY	12.74	skewGZ	6.50	kurAY	9.40
minGX	42.14	minAY	25.90	numNAX	35.10	kurGZ	12.60	skewGM	6.32	kurGM	9.27
minOZ	41.40	avgGM	25.76	numPOZ	35.10	skewOY	12.39	skewOZ	6.28	skewAY	9.23
stdOY	41.32	minOM	25.76	minGZ	35.00	skewAZ	12.17	skewAX	6.16	skewAZ	9.12
stdOX	40.81	maxAZ	25.43	minAX	34.51	kurOY	11.83	skewAZ	6.07	skewAX	8.99
maxAZ	40.80	avgAY	25.34	stdGY	34.50	skewAX	11.82	kurAY	5.93	kurAX	8.91
maxAX	40.31	maxAY	25.34	minGX	34.42	kurAZ	11.67	kurAM	5.89	kurOZ	8.69
stdOZ	40.06	maxGM	24.38	minAY	34.35	kurGM	11.13	skewAM	5.63	kurAZ	8.63
avgOZ	39.89	minOX	23.79	maxGM	34.11	kurOM	11.00	kurAZ	5.59	skewOY	8.50
numNGY	39.71	stdGZ	23.78	avgAM	33.34	kurGX	10.69	skewAY	5.51	kurOY	8.50
maxGX	39.63	stdOZ	23.10	maxAZ	33.11	skewAM	9.92	kurGX	5.49	kurGX	8.09
stdAX	39.53	numNGY	23.01	minAZ	33.04	kurAX	9.46	numPOY	5.38	kurOM	8.03
avgAX	39.06	SA	22.71	maxOY	32.44	kurOZ	9.44	numNAY	5.38	skewAM	7.78
stdGZ	38.92	maxOM	22.66	maxOZ	32.10	kurAM	9.29	kurOY	5.17	kurAM	7.59
SS	37.80	maxOX	22.50	stdTS	32.10	kurOX	8.96	kurOM	5.07	kurOX	6.44
minAX	37.72	maxOY	22.24	numPOX	32.08	numPOY	3.44	skewOY	4.60	numPOY	4.41
numNGZ	37.72	numPGX	22.22	stdOZ	31.58	numNAY	3.44	kurOX	3.91	numNAY	4.41
numPOZ	36.86	stdGX	22.20	numNGY	31.36	numNAZ	0.65	numNAZ	0.65	numNAZ	0.65
numNAX	36.86	maxGX	22.19	stdGZ	31.35	minTP	0.65	minTP	0.65	minTP	0.65
numNOY	36.59	stdSS	22.17	maxGX	30.91	maxTP	0.65	maxTP	0.65	maxTP	0.65
numPAY	36.59	minAZ	21.97	stdAX	30.42	stdTP	0.65	stdTP	0.65	stdTP	0.65
avgSS	35.87	stdGM	21.83	stdOY	29.87	avgTP	0.65	avgTP	0.65	avgTP	0.65

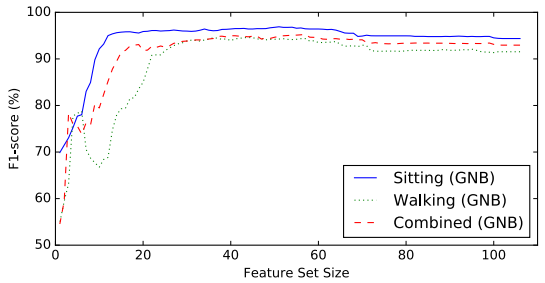

FIGURE 5. Variation in the F1-score caused by the number of used features for each posture and selected classifier.

rank of 106 individual features with the selected classifiers for each posture. Figure 5 depicts the F1-score variation caused by the feature length with the selected classifiers for each posture. In all postures, the F1-score was stable for the feature set sizes of 20 to 106. Meanwhile, the performance significantly decreased when the feature set size was below 20. The results of the best-performing classifier for each posture are highlighted in bold in Table 4. The best-performing classifiers and the size of the optimal feature sets for sitting, walking, and combined were (51, GNB), (45, GNB), and (57, GNB). RF showed a higher F1-score than the other algorithms for the sitting posture. However, GNB also showed an F1-score similar to that of RF, and its optimal feature length was shorter than with RF. In addition, it is more comfortable to find the equal error rate (EER) using GNB than using RF. For this reason, we chose GNB as the optimal algorithm for the sitting posture. In terms of postures, sitting provided a better performance than walking. These optimal settings for each posture were applied to the remaining experiments.

C. AUTHENTICATION PERFORMANCE

1) EXPERIMENT DESIGN

For DIAS to be practically used as a user authentication system, the authentication model using a multi-class classifier with two classes should be applied to DIAS. In other words, DIAS should be able to verify a valid user as a valid user and an invalid user as an invalid user. The number of valid users in the authentication model is one. The number of invalid users in training may affect the authentication performance. Having more invalid users increases the false rejection rate (FRR), while having fewer invalid users increases the false acceptance rate (FAR). Therefore, we need to derive the optimal number of invalid users to achieve balanced FAR and FRR.

Consequently, we evaluated the authentication performance while varying the number of invalid users in training. We measured the average values of FAR and FRR while changing the invalid user size to 4, 9, 14, 19, 24, and 29. We trained each of the 30 participants as a valid user. For each invalid user count, we trained randomly selected other participants, except for the valid user. We measured the average values of FAR and FRR for the 30 valid user cases (i.e., all participants). We applied a 10-fold cross validation to all authentication executions.

2) RESULT

Figure 6 illustrates the ratio of the real FAR and FRR values to the initial FAR and FRR values. As expected, the FAR values decreased, and FRR increased as the invalid user count grew in all postures. However, the real FAR and FRR values for each posture did not cross in our setting. We could not find the EER because we only considered the invalid user size rather than compromise the algorithm parameters. To deal with this problem, we measured the ratios of the real values to the initial values. The initial FAR was set to the FAR for the count of four, whereas the initial FRR was set to that for the count of 29. In this manner, we could find the cross-points of FAR and FRR that indicate the optimal invalid user count for each posture: 13 for sitting, 23 for walking, and 8 for combined.

We found EER by compromising the algorithm parameters based on the derived invalid user size. Table 6 shows the authentication performance of DIAS in terms of EER and AUC (area under the curve). The sitting posture showed the lowest error rate. The authentication model demonstrated the worst performance when combining the sitting and

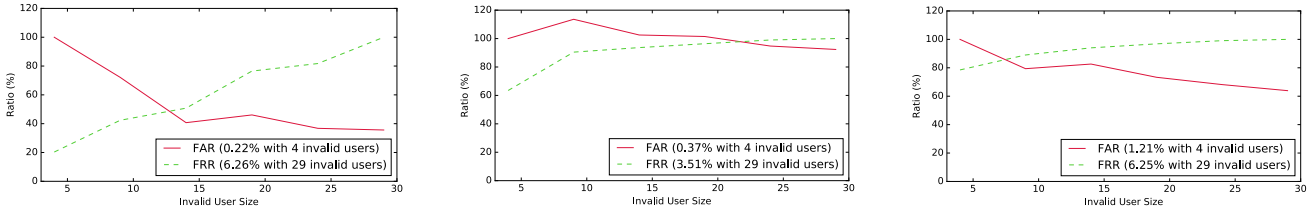

FIGURE 6. Variation in the ratio of the real false acceptance rate (FAR) and false rejection rate (FRR) caused by the number of invalid training users to the initial FAR and FRR. We measured those values because the real FAR and FRR values for each posture did not cross in our setting. The initial FAR is the FAR for the count of four, while the initial FRR is set to that for the count of 29. The cross points of the FAR and FRR are approximately 13 for sitting, 23 for walking, and 8 for combined. (a) Sitting condition. (b) Walking condition. (c) Combined condition.

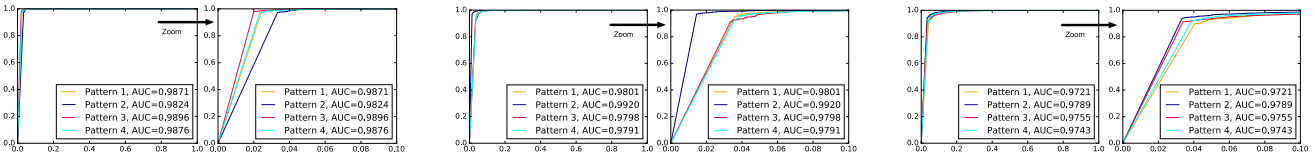

FIGURE 7. ROC curve for each pattern and each posture in user authentication. (a) Sitting condition. (b) Walking condition. (c) Combined condition.

TABLE 6. Authentication model performance for each combination of posture and invalid user count used for training (GNB is used as a classifier for all postures).

Patterns	Sitting, 13		Walking, 23		Combined, 8	
	EER(%)	AUC	EER(%)	AUC	EER(%)	AUC
P1	2.75	0.9871	3.83	0.9801	6.63	0.9721
P2	3.47	0.9824	1.88	0.9920	4.18	0.9789
P3	2.07	0.9896	4.32	0.9798	6.52	0.9755
P4	2.34	0.9876	4.07	0.9791	5.99	0.9743
Average	2.66	0.9867	3.53	0.9828	5.83	0.9752

walking postures. The average EER of each posture was 2.66%, 3.53%, and 5.83%, respectively. Figure 7 shows the ROC curve for each pattern and posture. The AUC values appeared with aspects similar to EER: the best AUC from sitting and the worst AUC from the combined condition. The average AUC for each posture was 0.9867, 0.9828, and 0.9752, respectively.

D. LONG-TERM EXPERIMENT

As a user frequently and continuously unlocks his/her smartphone, DIAS should be able to properly authenticate the user each time. However, according to Xu *et al.* [49], the accuracy of a model trained only with samples taken on a single day is unstable and unreliable because the touch biometrics of the users is not stable over time. In other words, the authentication accuracy of DIAS would also be unstable and the error rates may increase over time if DIAS employs a model trained with only the samples collected in a short period of time. From this perspective, we need to consider model training approaches appropriate for long-term authentication. Thus, we examine the existing training approaches [49] and propose a novel training approach, called the sliding window approach. In the sliding window approach, a model is refreshed with recent samples in a sliding window to provide a more flexible and stable authentication even though time elapses. Considering three approaches, we conducted an experiment to evaluate the authentication performance from the perspective of a long-term period (referred to as permanence).

1) EXPERIMENT DESIGN

In this experiment, DIAS utilized the authentication model with optimal settings obtained from the experimental results of the previous two sections. We applied the existing two approaches and our sliding window approach as a long-term training method and compared their permanence in terms of FAR and FRR. We used the long-term data set of 10 volunteers, 40,000 samples, for training a valid user of the model. The short-term data set of 30 participants, 7200 samples, was used for training the invalid users of the model.

For model training and testing, we first considered each of the 10 volunteers individually. We selected one of them, then used his/her long-term data set according to each approach to model the valid user. In all approaches, we randomly selected N users according to the optimal invalid user count of each posture from the remaining 29 participants. We then modeled the invalid user using his/her short-term data set. The samples of the valid user for the corresponding day and the samples of the remaining nine volunteers for the first day were used for prediction. Note that the training and testing samples were not associated. The three approaches we applied for training of the long-term data set are as follows:

Approach 1 (Training With a Fixed-Sized Data Set): In this approach, the samples of the first day are used for training

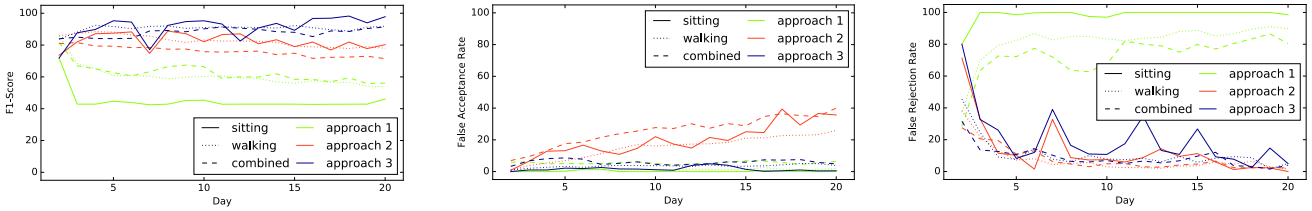

FIGURE 8. Variation of F1-score, FAR, and FRR of the three approaches over time. Approach 1 was trained with a fixed-sized dataset. Approach 2 was trained with an expanded-sized data set (i.e., adaptive approach). Approach 3 was trained with a sliding window. Approaches 1 and 2 are existing approaches. Approach 3 is our proposed approach. (a) F1-score. (b) False acceptance rate. (c) False rejection rate.

the model, and the samples of the days thereafter are used for testing. In the same manner, we used the samples of the first day for model training, then used the remaining samples from the second to the 20th day for testing authentication.

Approach 2 (Training With an Expanded-Sized Data Set (Adaptive Approach): We applied Xu et al.'s adaptive approach as the second approach [49]. In this approach, the samples from the first day to the $(N-1)$th day were used for the model training. The remaining samples of the Nth day were used for testing authentication. We used all the samples from the first day to the 19th day for training and authenticated the user utilizing samples from the 20th day.

Approach 3 (Training With a Sliding Window): The training with a fixed-sized data set degrades the permanence performance of user identification according to Xu *et al.* [49]. In the adaptive approach, the outdated touch biometrics for the user might be applied because the training data were accumulated from the first day. Maintaining the freshness of the training data is essential considering that the touch biometrics of a user is unstable and changes over time. Therefore, we proposed a new and novel approach training with a sliding window. Given a sliding window of size K, the samples from the $(N-K)$th day to the $(N-1)$th day were used for the model training. The samples of the Nth day were used for testing user authentication.

2) RESULT

Figure 8 shows the variation of the F1-score, FAR, and FRR for the three approaches over time. For Approach 1, FAR was kept stably low ($M = 2.34\%$, $SD = 2.33\%$), but FRR stayed high ($M = 84.50\%$, $SD = 14.61\%$). In contrast, for Approach 2, FRR decreased over time ($M = 9.36\%$, $SD = 11.71\%$), but FAR grew linearly over time ($M = 20.27\%$, $SD = 9.75\%$). We set the sliding window size to four because the day with the highest F1-score was day 5 in Approach 2 (82.45%). For Approach 3, FAR stayed stably low ($M = 3.69\%$, $SD = 2.19\%$), and FRR decreased over time ($M = 13.20\%$, $SD = 12.93\%$, $min = 1.58\%$). For most cases, the F1-scores of the proposed approach ($M = 89.88\%$, $SD = 4.60\%$) were higher than those of the other two approaches ($M = 55.98\%$, $SD = 10.11\%$ and $M = 80.41\%$, $SD = 4.80\%$, respectively). Therefore, for all postures, the proposed approach outperformed the existing approaches in terms of permanence.

V. SECURITY EVALUATION AND COMPARISON

A. SECURITY AGAINST RECORDING ATTACKS

Although DIAS removes most existing threats to the pattern lock, which uses the pattern as a secret, a new threat, in which an attacker can mimic the user's behavior, can appear. We conducted a shoulder–surfing attack scenario, specifically a recording attack, which is the most powerful attack on the pattern lock. We assumed that an attacker can obtain a video of legitimate user drawing patterns during the initial enrolment phase. We also assumed that the data set stacked from the data collection phase is insufficient. Furthermore, the attacker is assumed to be familiar with the mechanism of DIAS and behavioral factors used for authentication.

1) EXPERIMENT DESIGN

For the attack experiment, we chose a participant as a legitimate user. From the attacker's point of view, we fully recorded the movements of the arms and the hands when the legitimate user was drawing patterns. We held the camera over the user's head at a 45° angle for the sitting posture. We held the camera over the user's shoulder horizontally and tried to follow the user as much as possible for the walking posture. We filmed not only the user's pattern drawing, but also the user's gait.

We deployed 10 participants as attackers among 30 participants. We explained to them the recording attack and how to process it. They observed the user's hand–arm gestures and movements in the video. They could watch the videos for an unlimited number of times and practice mimicking the behavior of the user to perform skilled attacks. They attempted an attack for 10 times for each pattern and posture when they were ready. A total of 800 samples were used as a test set. For encouragement, we awarded them with $1 whenever an attack succeeded. The short-term samples of the legitimate user shown to them and those of other random users pursuant to each posture were used for training.

2) RESULT

The samples from the participant chosen as a legitimate user during the authentication experiment in Section IV-A were used to train the model for each posture. The samples from the attackers were used as a test set. We measured FAR as an

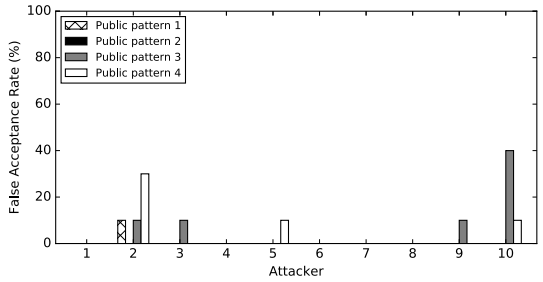

FIGURE 9. Recording attack success rate for each attacker with the sitting posture.

attack success rate. Surprisingly, the attacks in the walking posture never succeeded. Figure 9 shows the FAR of the attackers in the sitting posture. The FAR values of P1, P2, P3, and P4 were 1%, 0%, 7%, and 5%, respectively. A slight difference was observed in the attack success rate depending on the pattern shape. Five attackers never succeeded, and others succeeded in a few attempts. The average attack success rate in the sitting posture was 3.25%. Our results implied that if an attacker draws a pattern while walking, a recording attack against DIAS cannot be successful. Moreover, even if an attacker inputs a pattern in a fixed posture, succeeding in the recording attack is still difficult.

TABLE 7. Results of the statistical tests for the recording attack experiment. *Z* is the test statistic of the Wilcoxon Signed Rank Test. The p-value is calculated by the position of the *Z* value in the normal distribution. There is no p-value less than or equal to a significance level of ($\alpha = 0.05$) for all conditions.

Posture	Walking				Sitting			
Patterns	P1	P2	P3	P4	P1	P2	P3	P4
Z	-1	-1.342	0	0	-1	-1.89	-1.633	0
p	1.000	0.500	1.000	1.000	1.000	0.125	0.250	1.000

We utilized the data collected from the attackers in Section IV-A as a comparison to verify that the recording attack was statistically meaningful. Those data were the input data of the attackers when they were valid users, and not attackers; hence, they represent the attackers' intrinsic behaviors. We used 30 inputs of a valid user for each posture and pattern. We conducted the Wilcoxon-signed rank test with a significance level of 0.05 to compare the FAR of each pattern. The statistical result is described in Table 7. As a result, no significant difference was found in FAR in all patterns and postures. We can conclude that the recording attack was statistically worthless to attempt in DIAS.

B. COMPARISON WITH THE ANDROID PATTERN LOCK

In this section, we compare DIAS to the Android pattern lock system in terms of the authentication time (pattern-recall and pattern-input time) and the error rate.

1) EXPERIMENTAL DESIGN

This experiment was evaluated using repeated measures within the participant design. The independent variable was the authentication mechanism (Android pattern lock, DIAS). The sequence of the independent variable was counterbalanced to minimize the learning effects. We measured the authentication time (pattern-recall and pattern-input time) and the error rate.

We involved the remaining 10 participants who were not involved in neither the long-term nor the security experiment. Before starting the experiment, we explained the goal of the experiment and the DIAS concept in detail. We followed the same procedure for each mechanism: (a) they were asked to define their own secret pattern, or a public pattern was assigned to them; (b) they were allowed unlimited free training to become familiar with the mechanism and interface; and (c) they were asked to authenticate five times with a maximum of five trials, in which five incorrect trials were considered an authentication failure; and (d) they were given a mental rotation task (MRT)[1] for memory distraction before moving on.

After the abovementioned steps were undertaken, we again requested the participants to perform an authentication session for each mechanism after 1 h, 6 h, a day, and a week. We did not control the participants and required them to use the device naturally. The following research questions were developed for the experiment: (1) Does DIAS take less time than the Android pattern lock? (2) Is DIAS more error-resistant than the Android pattern lock? The Wilcoxon signed-rank test was employed to check for significant differences in the authentication time for the mechanisms. The results were based on 90 authentication sessions (9 sessions × 10 participants) per mechanism performed by 10 participants.

2) RESULT

During the experiment, we measured the authentication time for each authentication session from the beginning of the authentication session to the release of the pattern drawing. This decision was made to reflect the time the user thinks about the pattern to draw after the authentication session begins. We counted only the successful authentication sessions for this analysis. Figure 10 depicts the authentication times for the authentication mechanisms. DIAS took a slightly shorter authentication time ($M = 1.89$ *s*, $SD = 0.38$ *s*) than the Android pattern lock ($M = 2.30$ *s*, $SD = 1.29$ *s*). However, no significant difference was found ($Z = -1.274$, $p = .116$). Regarding the time for the authentication performed after a certain period of time, the authentication time in DIAS tends to be shorter as the authentication session interval is longer. For the Android pattern lock, the authentication time tends to be longer as the time gap between the authentication sessions increases. We observed statistically significant differences in the authentication time for each authentication session interval, except for the 1 h session interval: *1 hour* ($Z = -.255$, $p = .423$), *6 hours* ($Z = -1.886$, $p < .032$), *a day* ($Z = -1.886$, $p < .032$), *a week* ($Z = -2.448$, $p < .006$).

[1] https://www.psytoolkit.org/experiment-library/mentalrotation.html

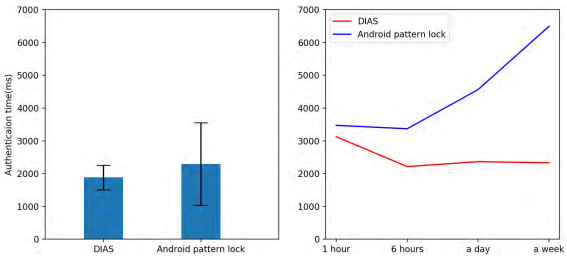

FIGURE 10. Authentication time for DIAS and Android pattern lock.

We also measured whether a participant could correctly authenticate with the mechanism within the five trials. For the Android pattern lock, four users for one or two authentication sessions needed two or three trials to authenticate. One of them failed five authentication sessions, which means that he/she could not remember the secret pattern five times. Regarding DIAS, three users for each authentication session needed two or three trials to authenticate; however, none of the authentication sessions failed.

TABLE 8. Comparison in terms of the equal error rate between DIAS and previous behavior authentication schemes.

Scheme	Mechanism	Equal Error Rate
DIAS*	Pattern Lock	S : 2.66%, W : 3.53%
Shahzad et al. [36]	Gesture-based	4.8% for one gesture
Xu et al. [49]	Touch-based	under 10% in general
Sitova et al. [37]*	Typing-based	S : 10.05%, W : 7.16%
Alpar et al. [2]		4.1%

* S for sitting posture and W for walking posture

C. COMPARISON WITH THE EXISTING BEHAVIORAL AUTHENTICATION

We compared the performance of DIAS in each posture measured in Section IV-C and those of the previous studies in the field of behavioral authentication. Table 8 describes the EER values of each scheme. The suggested schemes were studied based on various mechanisms, such as gesture-, touch-, and typing-based authentication. Although their subjects and models were different, we tried to compare them in an equal condition. Shahzad *et al.* [36] proposed GEAT, in which a user can enter the gesture for several times. We used the EER value of 4.8% when a user entered only one gesture in GEAT because we used the data of a user's one pattern input. Xu *et al.* [49] utilized several operation types and achieved EER values lower than 10% for all operation types. Sitová *et al.* [37] proposed HMOG and achieved the lowest EER of 10.05% for sitting and 7.16% for walking. Alpar *et al.* [2] achieved an EER of 4.1%. Meanwhile, the average EER values of DIAS were 2.66% for sitting and 3.53% for walking. We verified that the performance of DIAS was comparable to those of the other behavior-based schemes for all postures.

VI. DISCUSSION

A. IMPLICATION

Although one might think that there is no benefit in implementing the behavioral authentication mechanism on a 3×3 grid, we argue that DIAS has a beneficial implication. DIAS can present various patterns that guarantee high authentication performance on a strict grid. We designed the criteria of the public pattern by identifying the correlations between shape and accuracy. Therefore, DIAS can create those patterns by virtue of the criteria. In addition, DIAS can extract formal features on a "*segment*" basis in the grid. It does not need to perform additional feature processing, and was observed to be more efficient than gestures. Our statistical feature extraction approach plays an important role in the removal of sensor noise in smartphones. Although this may not be sufficient to deal with old phones that generate significantly more noise, combining our approach with other noise removal techniques [17], [25] will solve the problem.

Even if the system is secure, but detrimental to user interface and user experience, users may not choose to use it. We implemented DIAS based on the existing Android pattern lock scheme. Therefore, users can obtain two advantages: familiarity from a similar interface with the Android pattern lock and reduced burden of remembering secret lock patterns.

Machine learning-based authentication needs enough input data for the performance and stability of the model, although considerable inputs of training data reduce usability. In this context, DIAS contains an enrolment phase, in which some users may feel uncomfortable. However, we tried to focus on the improvement of long-term usability rather than short-term usability of the training phase. The users' behavior changes over time, and reflecting these changes with the model is an important issue. The sliding window approach that we proposed in Section IV-D can be one solution to offer both security and usability for a long time. In this approach, the user's daily test inputs can be the latest training input data that update the model. Therefore, if users endure the little discomfort brought about by the enrolment phase, they will certainly feel comfortable in their long-term usage.

Turning lock patterns into public values also has several advantages. In Section IV-C, we observed that DIAS could accurately authenticate a valid user based only on the user's behavior while using public patterns for multiple users. Moreover, as a result of the long-term experiment in Section IV-D, we observed that DIAS can preserve the efficient authentication performance over time based on a sliding window approach. We tried to overcome the security problems of the Android pattern lock. We expect that DIAS can eliminate the existing vulnerabilities that can occur in secret patterns. Remaining/new threats, such as recording attacks, can exist, but DIAS is resistant to threats, as shown in the experiment in Section V-A. One might question the unsafe results obtained in some cases. We would like to emphasize that the experimental scenarios were not realistic in terms of mimicking users in the wild. Section V-B showed that DIAS

was competitive in terms of the authentication time compared to the Android pattern lock, and was more error-resistant than the Android pattern lock. The experiment results are in some ways preliminary, but we argue that they are a big step forward toward validating DIAS. We believe those are promising given both the exploratory nature of the study and the small user size.

Some users might avoid using behavior-based authentication due to concerns regarding battery consumption. However, an authentication session in DIAS is intermittent and only takes a moment. Therefore, we expect that the battery consumption of DIAS will be insignificant. In other words, we expect that DIAS will not cause a significant impediment to the battery performance of smartphones because it uses sensors for a very short time.

B. LIMITATIONS AND FUTURE WORK

Although our concept is interesting, it has several limitations:

(1) We need to conduct experiments for a realistic scenario that was not considered herein. First, we need to consider other postures, such as lying down and using two hands. A user in the real world unlocks a smartphone in various postures that are not considered in our study. However, asking the user to train too many postures, even if he/she unlocks the phone in trained postures, decreases the usability of DIAS. Even if the user unlocks the phone in trained postures, the model might still not be able to authenticate the user because of shaky hands or short breath. As a future work, we are planning to convert the user's data of a non-trained form into a trained form to improve the system stability in various environments. Second, we have not yet conducted a long-term experiment either in the field or on a large scale. User behavior can change during the time period between two authentication sessions because fallback authentication is not frequently used, and user behavior is unstable. We need to evaluate DIAS in light of the changes in user behavior in the long term. Thus, we are planning to address these limitations in an upcoming study.

(2) We are still far from producing a practical system for DIAS. First, we need to further improve the FAR and FRR values of DIAS. In fact, a slight difference was found in the authentication performance of DIAS according to the shape of four public patterns. We believe that more public patterns with a better authentication performance than those used in the current study may exist. Therefore, we consider the guided approach as another future work. We are planning to experiment on more public patterns that satisfy the criteria and derive ones with optimal authentication performance to guide the users in the enrolment phase. Second, we stored users' behavioral information on a server, but this may invade privacy. Smartphone users may worry about exposing their information outside the phone. Eventually, one-class classification that is more realistic for a mobile device should be considered because it only takes the owner's data for training.

(3) A user may adopt physical biometrics for main authentication and behavioral biometrics, such as DIAS, as a secondary authentication method. In that case, the user may also disable both methods because of an accidental hand or arm injury. Although this scenario is very specific, it should be prevented. It can be solved by recommending to users a knowledge-based method as a complement of secondary authentication when both methods they use are biometrics-based.

VII. RELATED WORK

A. ANDROID PATTERN LOCK

The Android pattern lock is the most commonly used popular authentication mechanism on mobile devices [19], [26]. The pattern lock is preferred over PIN or textual passwords and known to be more usable/memorable [9]. However, pattern locks are insecure because users tend to use a simple pattern for usability. Uellenbeck *et al.* [44] and Andriotis *et al.* [4] identified the fact that only a few pattern spaces are used to draw actual patterns within the theoretical limits of pattern space. The simple patterns drawn are easily stolen and replicated. Meanwhile, usability decreases as input time and pattern input error rates increase when complex patterns for enhanced security are selected [39].

Moreover, security issues easily occur via a leakage of pattern shapes [34]. Many attack methods may be used against patterns, such as guessing attacks that try to attack by guessing the pattern [6], [13], [38], shoulder surfing attacks that obtain the victim's pattern through an intentionally direct observation [28], [41], smudge attacks leveraging oily residues remaining on a touchscreen [8], and thermal imaging attacks exploiting thermal residues (i.e., heat traces left on a touchscreen, making them visible through thermal imaging) [1]. Ye *et al.* [51] proposed a video-based attack followed by a computer vision algorithm to track the fingertip movement on the video. Zhou *et al.* [54] used the speaker and the microphone in the victim's smartphone to convert acoustic signals into the lock pattern. These two studies showed that complex patterns do not offer stronger security against suggested attacks.

Various efforts based on the Android pattern lock have been made to prevent the leakage of lock patterns from such attacks. Zakaria *et al.* [52] presented three recall-based graphical passwords as a method to prevent shoulder surfing attacks. However, they did not significantly improve resistance to attacks. Von Zezschwitz *et al.* [48] presented three graphic-based authentication schemes, which were more secure than the Android pattern lock against smudge attacks, but the increased input time and reduced usability were the result of these schemes. Furthermore, this technique did not solve the trade-off between usability and security. Cho *et al.* [14] proposed "Syspal," which required the use of randomly selected points when entering patterns. This mechanism requires forcing the pattern selection; hence, usability reduction is inevitable. Furthermore, it still depends on the memory of the user, and no improvement in security against smudge attacks can be observed. Higashikawa *et al.* [20] proposed a pattern lock scheme that ensures high usability

and resistance from shoulder surfing attacks. However, their secure mode can never be turned on when users cannot recognize that they are under a shoulder surfing attack. All of the abovementioned works generally propose a method that targets an increment of resistance against only one specific attack. Their schemes may be vulnerable to other attacks that they have not dealt with because more than two attacks that target the Android pattern lock can coexist in the real world. In contrast, we make the intention of the attackers, who want to crack the shape of the secret lock pattern, worthless by turning patterns into public knowledge.

B. BEHAVIOR-BASED AUTHENTICATION

The behavior-based authentication research in the mobile environment began because of the universalization of mobile devices equipped with touch screens and various sensors. These bodies of research have been actively exploiting data from touch screens and sensors as new authentication factors.

Li *et al.* [30] distinguished the user according to his/her finger movements when he/she uses a smartphone using the touch data. The accuracy was at least 79.74% and as high as 95.78% with SVM. Frank *et al.* [18] classified users by concentrating on their scrolling behavior. The performance was measured separately using kNN and SVM, and 0%–4% EERs were achieved. Xu *et al.* [49] evaluated the performance of various types of touch actions made by the user on the touchscreen. The results showed that all actions generally have an EER of less than 10% for the SVM classifier.

The experimental results demonstrated that the error rate was too high in some scenarios, indicating that the current mechanism cannot be implemented as an independent authentication method. All studies presented so far utilized only touch events for authentication without using other sensors. However, the microscopic movement changes of the mobile devices generated by the touch action of the user can be used as information for identifying the user.

Meanwhile, a few studies reported on user posture recognition. Buriro *et al.* [12] collected accelerometer, gravity, and magnetometer sensor data and touch data to authenticate according to the movement of a smartphone during sign-in. The experimental results demonstrated that the maximum EER with a multi-layer perception algorithm was 2.46%. Sitová *et al.* [37] used accelerometer, gyroscope, and magnetometer sensor data and touch data to distinguish users according to their device usage and touch behavior. The experiments were performed under two conditions of sitting and walking. The maximum EER had a 7.16% accuracy when walking. Crawford and Ahmadzadeh [15] designed an authentication system using user typing patterns with keystrokes. They measured the typing posture using gyroscope data and classified users with different classification models classified by each posture. Decision Tree and Logistic Regression showed a performance of over AUC 90%.

These studies considered user postures when generating touch dynamics. Considering postures, the change in the slope of the device was measured using accelerometer and magnetometer sensors and used as features. Therefore, users can be distinguished using various features instead of using only a touch event. Furthermore, our study demonstrated an authentication performance that is considerably better than those that displayed existing results presented in the literature.

VIII. CONCLUSION

We proposed herein DIAS, a novel extension of the pattern lock mechanism. We believe that DIAS is even beneficial to secondary authentication because the user does not need long-term memory recall. DIAS shows a non-secret lock pattern to users; hence, the concerns of the Android pattern lock system can be addressed. Users draw the public pattern as shown for authentication. We leverage the user's behavior acquired when the user draws the pattern for authentication. Eventually, DIAS can provide high authentication performance by presenting pattern shapes that satisfy the public pattern criteria. Our attack experiment demonstrated that launching a recording attack on DIAS is difficult. Our prototype implementation needs further improvements to be practically used. Although this study is exploratory, we believe that this is a promising approach of combining user behavior with pattern lock systems.

ACKNOWLEDGMENT

Our study was approved by the Institutional Review Board (IRB) of Yonsei University. The authors would like to thank Seungyeon Kim for the helpful comments.

REFERENCES

[1] Y. Abdelrahman, M. Khamis, S. Schneegass, and F. Alt, "Stay cool! understanding thermal attacks on mobile-based user authentication," in *Proc. Conf. Hum. Factors Comput. Syst. (CHI)*, New York, NY, USA, 2017, pp. 3751–3763. [Online]. Available: http://doi.acm.org/10.1145/3025453.3025461

[2] O. Alpar, "Frequency spectrograms for biometric keystroke authentication using neural network based classifier," *Knowl.-Based Syst.*, vol. 116, pp. 163–171, Jan. 2017.

[3] P. Andriotis, G. Oikonomou, A. Mylonas, and T. Tryfonas, "A study on usability and security features of the Android pattern lock screen," *Inf. Comput. Secur.*, vol. 24, no. 1, pp. 53–72, 2016.

[4] P. Andriotis, T. Tryfonas, and G. Oikonomou, "Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method," in *Proc. Int. Conf. Hum. Aspects Inf. Secur. Privacy, Trust (HAS)*. Cham, Switzerland: Springer, 2014, pp. 115–126.

[5] P. Andriotis, T. Tryfonas, G. Oikonomou, and C. Yildiz, "A pilot study on the security of pattern screen-lock methods and soft side channel attacks," in *Proc. Conf. Secur. Privacy Wireless Mobile Netw. (WiSec)*, 2013, pp. 1–6.

[6] A. J. Aviv, D. Budzitowski, and R. Kuber, "Is bigger better? Comparing user-generated passwords on 3×3 vs. 4×4 grid sizes for Android's pattern unlock," in *Proc. Annu. Comput. Secur. Appl. Conf. (ACSAC)*, 2015, pp. 301–310.

[7] A. J. Aviv, J. T. Davin, F. Wolf, and R. Kuber, "Towards baselines for shoulder surfing on mobile authentication," in *Proc. Annu. Comput. Secur. Appl. Conf. (ACSAC)*, 2017, pp. 486–498.

[8] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proc. USENIX Conf. Offensive Technol. (WOOT)*, 2010, pp. 1–7.

[9] R. Biddle, S. Chiasson, and P. C. van Oorschot, "Graphical passwords: Learning from the first twelve years," *ACM Comput. Surv.*, vol. 44, no. 4, p. 19, Aug. 2012.

[10] A. Bier, A. Kapczyński, and Z. Sroczyński, "Pattern lock evaluation framework for mobile devices: Human perception of the pattern strength measure," in *Proc. Int. Conf. Man–Mach. Interact.* Cham, Switzerland: Springer, 2017, pp. 33–42.

[11] C. Bo, L. Zhang, X.-Y. Li, Q. Huang, and Y. Wang, "SilentSense: Silent user identification via touch and movement behavioral biometrics," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 187–190.

[12] A. Buriro, B. Crispo, F. Delfrari, and K. Wrona, "Hold and sign: A novel behavioral biometrics for smartphone user authentication," in *Proc. Secur. Privacy Workshops (SPW)*, 2016, pp. 276–285.

[13] S. Cha, S. Kwag, H. Kim, and J. H. Huh, "Boosting the guessing attack performance on Android lock patterns with smudge attacks," in *Proc. Asia Conf. Comput. Commun. Secur. (ASIACCS)*, 2017, pp. 313–326.

[14] G. Cho, J. H. Huh, J. Cho, S. Oh, Y. Song, and H. Kim, "SysPal: System-guided pattern locks for Android," in *Proc. Symp. Secur. Privacy*, 2017, pp. 338–356.

[15] H. Crawford and E. Ahmadzadeh, "Authentication on the Go: Assessing the effect of movement on mobile device keystroke dynamics," in *Proc. Symp. Usable Privacy Secur. (SOUPS)*, 2017, pp. 163–173.

[16] A. De Luca, A. Hang, E. von Zezschwitz, and H. Hussmann, "I feel like i'm taking selfies all day!: Towards understanding biometric authentication on smartphones," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, 2015, pp. 1411–1414.

[17] T. Feng, X. Zhao, and W. Shi, "Investigating mobile device picking-up motion as a novel biometric modality," in *Proc. IEEE 6th Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep./Oct. 2013, pp. 1–6.

[18] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 136–148, Jan. 2013.

[19] M. Harbach, A. De Luca, and S. Egelman, "The anatomy of smartphone unlocking: A field study of Android lock screens," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2016, pp. 4806–4817.

[20] S. Higashikawa, T. Kosugi, S. Kitajima, and M. Mambo, "Shoulder-surfing resistant authentication using pass pattern of pattern lock," *IEICE Trans. Inf. Syst.*, vol. 101, no. 1, pp. 45–52, 2018.

[21] iMore. (2017). *Face ID Limitations*. [Online]. Available: https://www.imore.com/limitations-face-id-what-you-need-know

[22] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, "The design and analysis of graphical passwords," in *Proc. 8th USENIX Secur. Symp.*, 1999, pp. 1–15.

[23] kaspersky. (2013). *Forge You: Can Biometric Authentication be Trusted?* [Online]. Available: https://www.kaspersky.com/blog/biometric-authentication/2520/

[24] H. Khan, U. Hengartner, and D. Vogel, "Evaluating attack and defense strategies for smartphone PIN shoulder surfing," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2018, pp. 1–10, Paper no. 164.

[25] R. Kumar, V. V. Phoha, and A. Serwadda, "Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns," in *Proc. IEEE 8th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Sep. 2016, pp. 1–8.

[26] D. Kunda and M. Chishimba, "A survey of android mobile phone authentication schemes," in *Mobile Networks and Applications (On-Line)*. 2018, pp. 1–9.

[27] T. Kwon and S. Na, "TinyLock: Affordable defense against smudge attacks on smartphone pattern lock systems," *Comput. Secur.*, vol. 42, pp. 137–150, May 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404813001697

[28] A. H. Lashkari, S. Farmand, O. B. Zakaria, and D. Saleh, "Shoulder surfing attack in graphical password authentication," 2009, *arXiv:0912.0951*. [Online]. Available: https://arxiv.org/abs/0912.0951

[29] H. Lee, S. Kim, and T. Kwon, "Here is your fingerprint!: Actual risk versus user perception of latent fingerprints and smudges remaining on smartphones," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf.*, 2017, pp. 512–527.

[30] L. Li, X. Zhao, and G. Xue, "Unobservable re-authentication for smartphones," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2013, pp. 57–59.

[31] Y. M. Lui, D. Bolme, P. J. Phillips, J. R. Beveridge, and B. A. Draper, "Preliminary studies on the good, the bad, and the ugly face recognition challenge problem," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2012, pp. 9–16.

[32] I. Oakley, J. H. Huh, J. Cho, G. Cho, R. Islam, and H. Kim, "The personal identification chord: A four button authentication system for smartwatches," in *Proc. Asia Conf. Comput. Commun. Secur.*, 2018, pp. 75–87.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[34] V. V. Rao and A. S. N. Chakravarthy, "Analysis and bypassing of pattern lock in Android smartphone," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Dec. 2016, pp. 1–3.

[35] A. Serwadda, V. V. Phoha, and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms," in *Proc. IEEE 6th Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep./Oct. 2013, pp. 1–8.

[36] M. Shahzad, A. X. Liu, and A. Samuel, "Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 39–50.

[37] Z. Sitová, J. Šedenka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, "HMOG: New behavioral biometric features for continuous authentication of smartphone users," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 5, pp. 877–892, May 2016.

[38] Y. Song, G. Cho, S. Oh, H. Kim, and J. H. Huh, "On the effectiveness of pattern lock strength meters: Measuring the strength of real world pattern locks," in *Proc. Conf. Hum. Factors Comput. Syst. (CHI)*, New York, NY, USA, 2015, pp. 2343–2352.

[39] C. Sun, Y. Wang, and J. Zheng, "Dissecting pattern unlock: The effect of pattern strength meter on pattern selection," *J. Inf. Secur. Appl.*, vol. 19, nos. 4–5, pp. 308–320, 2014.

[40] H. Tao and C. Adams, "Pass-Go: A proposal to improve the usability of graphical passwords," *Int. J. Netw. Secur.*, vol. 7, no. 2, pp. 273–292, 2008.

[41] F. Tari, A. A. Ozok, and S. H. Holden, "A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords," in *Proc. Symp. Usable Privacy Secur. (SOUPS)*, 2006, pp. 56–66.

[42] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "A survey on touch dynamics authentication in mobile devices," *Comput. Secur.*, vol. 59, pp. 210–235, Jun. 2016.

[43] S. Trewin, C. Swart, L. Koved, J. Martino, K. Singh, and S. Ben-David, "Biometric authentication on a mobile device: A study of user effort, error and task disruption," in *Proc. 28th Annu. Comput. Secur. Appl. Conf.*, 2012, pp. 159–168.

[44] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz, "Quantifying the security of graphical passwords: The case of Android unlock patterns," in *Proc. Conf. Comput. Commun. Secur. (CCS)*, 2013, pp. 161–172.

[45] D. Van Bruggen, "Studying the impact of security awareness efforts on user behavior," Ph.D. dissertation, Graduate Program Comput. Sci. Eng., Univ. Notre Dame, Notre Dame, IN, USA, 2014.

[46] T. Van Nguyen, N. Sae-Bae, and N. Memon, "DRAW-A-PIN: Authentication using finger-drawn PIN on touch devices," *Comput. Secur.*, vol. 66, pp. 115–128, May 2017.

[47] E. Von Zezschwitz, A. De Luca, P. Janssen, and H. Hussmann, "Easy to draw, but hard to trace?: On the observability of grid-based (Un)lock patterns," in *Proc. Conf. Hum. Factors Comput. Syst. (CHI)*, 2015, pp. 2339–2342.

[48] E. von Zezschwitz, A. Koslow, A. De Luca, and H. Hussmann, "Making graphic-based authentication secure against smudge attacks," in *Proc. Int. Conf. Intell. User Interfaces (IUI)*, 2013, pp. 277–286.

[49] H. Xu, Y. Zhou, and M. R. Lyu, "Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones," in *Proc. Symp. Usable Privacy Secur. (SOUPS)*, vol. 14. 2014, pp. 187–198.

[50] G. Ye, Z. Tang, D. Fang, X. Chen, K. I. Kim, B. Taylor, and Z. Wang, "Cracking Android pattern lock in five attempts," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2017, pp. 1–15.

[51] G. Ye *et al.*, "A video-based attack for Android pattern lock," *ACM Trans. Privacy Secur.*, vol. 21, no. 4, p. 19, 2018.

[52] N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan, "Shoulder surfing defence for recall-based graphical passwords," in *Proc. 7th Symp. Usable Privacy Secur. (SOUPS)*, 2011, Art. no. 6.

[53] J. Zhang, X. Zheng, Z. Tang, T. Xing, X. Chen, D. Fang, R. Li, X. Gong, and F. Chen, "Privacy leakage in mobile sensing: Your unlock passwords can be leaked through wireless hotspot Functionality," *Mobile Inf. Syst.*, vol. 2016, Mar. 2016, Art. no. 8793025.

[54] M. Zhou, Q. Wang, J. Yang, Q. Li, F. Xiao, Z. Wang, and X. Chen, "PatternListener: Cracking Android pattern lock using acoustic signals," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 1775–1787.

YEEUN KU received the B.S. degree in information security from Sejong University, Seoul, South Korea, in 2017. She is currently pursuing the M.S. degree with the Information Security Laboratory, Yonsei University, Seoul. Her research interests include information security and privacy, system security, and machine learning.

LEO HYUN PARK received the B.S. degree in computer engineering from Kwangwoon University, Seoul, South Korea, in 2017. He is currently pursuing the Ph.D. degree with the Information Security Laboratory, Yonsei University, Seoul. His research interests include information security and privacy, malware analysis, usable security, AI security, machine learning algorithms, and adversarial machine learning.

SOOYEON SHIN received the B.S., M.S., and Ph.D. degrees in computer science and engineering from Sejong University, Seoul, South Korea, in 2004, 2006, and 2012, respectively, where she was a Postdoctoral Researcher, from 2012 to 2013. In 2013, she joined Yonsei University, Seoul, to continue her postdoctoral research. Her current research interests include cryptographic protocol, network security, usable security, and human–computer interaction.

TAEKYOUNG KWON received the B.S., M.S., and Ph.D. degrees in computer science from Yonsei University, Seoul, South Korea, in 1992, 1995, and 1999, respectively, where he is currently a Professor of information security, where he is the Director of the Information Security Laboratory. From 1999 to 2000, he was a Postdoctoral Research Fellow with the University of California at Berkeley. From 2001 to 2013, he was a Professor of computer engineering with Sejong University, Seoul. He is on the Director Board of the Korea Institute of Information Security and Cryptology (KIISC) and on the Editorial Committee of the Korean Institute of Information Scientists and Engineers (KIISE). His research interests include authentication, cryptographic protocols, network security, software and system security, usable security, AI security, and adversarial machine learning. He is a member of ACM and Usenix.