# FINGER-DRAWN PIN AUTHENTICATION ON TOUCH DEVICES

*Toan Van Nguyen* *Napa Sae-Bae* *Nasir Memon*

New York University Polytechnic School of Engineering, New York, USA
toan.v.nguyen@nyu.edu nsb261@nyu.edu memon@nyu.edu

## ABSTRACT

PIN authentication is widely used thanks to its simplicity and usability, but it is known to be susceptible to shoulder surfing. In this paper, we propose a novel online finger-drawn PIN authentication technique that lets a user draw a PIN on a touch interface with her finger. The system provides some resilience to shoulder surfing without increasing authentication delay and complexity by using both the PIN as well as a behavioral biometric in user verification. Our approach adopts the Dynamic Time Warping (DTW) algorithm to compute dissimilarity scores between PIN samples. We evaluate our system in two shoulder surfing scenarios: 1) PIN attack where the attacker only knows the victim's PIN but has no information about it's drawing characteristic and 2) Imitation attack where an attacker has access to a dynamic drawing sequence of a victim's finger-drawn PIN in the form of multiple observations. Experimental results with a data set of 40 users and 2400 imitating samples from two attacks yield an Equal Error Rate (EER) of 6.7% and 9.9% respectively, indicating the need for further study on this promising authentication mechanism.

***Index Terms—*** Finger-drawn PIN, behavioral biometric, online signature, shoulder surfing, mobile authentication

## 1. INTRODUCTION

A PIN is a secret sequence of digits widely used to authenticate a user while unlocking a phone as well as in many financial and mobile applications. Typically a user enters her PIN into a system by pressing or tapping buttons corresponding to the digits in a sequential order. A user is then authenticated only if the sequence of digits entered matches the one stored in the system during enrollment. That is, a traditional PIN authentication system only verifies knowledge of the PIN and utilizes no other user characteristic. As a result, the system is not capable of distinguishing between a legitimate user and an adversary with the correct PIN, and, therefore, is vulnerable to shoulder surfing attack. In addition, studies have shown that PINs are not hard to guess. They often include repeated digits, sequential digits, memorable patterns, or birthdays of users [1]. Lastly, entering PINs on virtual keyboards of devices equipped with touch sensitive display may leave a smudge that could be used as a cue to recover the device PIN [2]. However, recent touch sensitive display technology has enabled a user to interact with the system in new ways. In this paper, we exploit this new interaction mechanism by proposing a finger-drawn PIN authentication system.

The proposed system works as follows. A user is asked to enter her PIN by drawing it on a touch sensitive display instead of typing it. Intuitively, the way people draw PINs differs from person to person. Hence, the system authenticates a user based on what they know (PIN) as well what they are (the way they draw the PIN). As a result, by adding this behavioral biometric information derived from drawing sequences, we create a security-enhanced finger-drawn PIN that is reasonably difficult to imitate by an adversary with limited resources. Another advantage of this approach is that it can be built on top of existing PIN authentication systems thereby allowing service providers to adopt the scheme by just adding an overlay to the existing infrastructure. In addition, this system does not require users to learn a new scheme or new type of secrets thereby maintaining the high adoption rate associated with PINs. The proposed mechanism could have a potential impact on many legacy systems that use PINs for authentication.

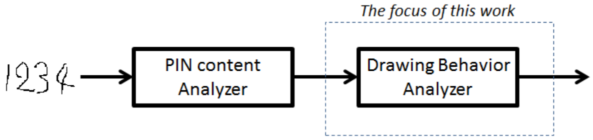

**Fig. 1**. The proposed finger-drawn PIN authentication system

As shown in Figure 1, the complete system comprises of two cascaded modules: a PIN content analyzer and a PIN drawing behavior analyzer. A finger-drawn PIN sample is first processed by the PIN drawing behavior analysis module only if the recognized PIN number matches the one stored in the system. While multiple approaches to recognize digits from handwritten trace have been proposed [3, 4, 5, 6], its application to user authentication by leveraging inherent biometric information has not been studied before.

**Contribution** In this work, we propose and evaluate the verification performance of a finger-drawn 4-digit PIN on mobile devices under two attack scenarios, namely PIN attack and Imitation attack, using a dataset collected in a mobile environment over time and an imposter dataset collected for the two scenarios. To this end, we first collected genuine samples from 40 users where each user contributed one sample each over six separate sessions with intervals between sessions ranging from 12 to 96 hours. Then, we collected imposter samples according to two attack scenarios. The first scenario is where an attacker has a knowledge of the PIN of a victim but has no information about how the PIN is drawn. The second scenario is where an attacker is provided access to an animation of the victim's finger-drawn 4-digit PIN in the form of multiple observations. In each scenario, we collected 30 imposter samples for each user, resulting in a dataset of 2400 samples in total. All these samples were drawn by a fingertip on touch screens of iOS devices in an uncontrolled environment.

To evaluate the verification performance of the system, we used a Dynamic Time Warping (DTW) based algorithm to compute dissimilarities between pairs of finger-drawn PIN samples. Evaluation results reveal that the system achieves verification performance of 6.7% equal error rate under PIN attack and 9.9% equal error rate under Imitation attack. These results demonstrate the potential of finger-drawn 4-digit PINs as another authentication factor in order to provide stronger security against common attacks as compared to the traditional PIN authentication system.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 presents the design of our system and process to verify a finger-drawn PIN sample. Section 4 describes the experiment design we used for collecting a new dataset as well as experimental results and analysis. Section 5 concludes and discusses future work.

## 2. RELATED WORK

Numerous approaches have been proposed to cope with the problem of shoulder surfing in PIN-based authentication systems. One approach is to increasing the noise in what is observed by the shoulder surfer [7, 8]. Nevertheless, this approach increases PIN entry delay and complexity of the system as well as its usability. Another approach that is immune to shoulder surfing attack is to eliminate the need to use fingers for PIN entry by using eye-gaze [9, 10]. The general idea is instead of interacting with keypad or touch screen, user can utilize an eye tracker to enter the PIN by looking at the proper symbols on the screen in appropriate order.

Biometric-based authentication that utilizes physiological or behavioral characteristics to distinguish individuals are believed to be a better solution for user authentication. Such systems make use of physiological characteristics [11] of a user and are resistant to shoulder surfing. However their adoption has been limited perhaps partly due to privacy concerns. Systems using behavioral characteristics are considered less intrusive and are becoming a potential replacement and are attracting more attention from researchers. Numerous methods which exploit diverse behavioral traits have been proposed, such as keystroke dynamics [12, 13], mouse movements [14, 15], online signature [16, 17, 18, 19, 20], multi-touch gestures [21, 22, 23, 24], etc. Our proposed system verifies users based on their PINs (what they know) as well as the way they draw the PINs (what they are), thus combining advantages of both knowledge-based and behavioral biometric-based authentications.

One popular instance of behavioral biometric that alleviates the threat of shoulder surfing and at the same time is socially and legally accepted for authenticating an individual is a handwritten signature. There are two types of signature-based authentication systems: offline and online systems. In an offline system, the signature is just an image of the user's signature without additional attributes, whereas, in an online system, the signature is represented by a sequence of x-y coordinates along with associated attributes such as time-stamp and pressure. With the increasing number of touch interface devices, online signatures have gradually replaced offline signatures for user authentication. The general method for verifying an online signature first extracts features from a signature and then comparing against a stored template. For this purpose, two well-known algorithms including Dynamic Time Warping (DTW) and Hidden Markov Models (HMM) have been widely used [16, 17, 25, 26, 27]. Finger-drawn PIN shares similarity with an online handwritten signature since the format of the data captured in both cases is the same, thus, we can adopt a DTW and HMM approach for our system. However, an HMM usually requires a large amount of data to train an accurate model, which is not always readily available. In our system, for usability, we do not require a user to provide a large number of training samples in enrollment phase. Therefore, we choose DTW as our verification algorithm where a small number of samples (3 to 5) taken during enrollment are adequate for use as templates during verification.

## 3. FINGER-DRAWN PIN VERIFICATION

Similar to any other biometric authentication systems, finger-drawn PIN authentication comprises of two phases: enrollment and verification. In the enrollment phase, the system captures a few finger-drawn samples from a user and stores them as a training set for that user. In the verification phase, a user requesting access to the system enters her PIN. The system then verifies the entered PIN as follows. First, it computes the average of pairwise distances between this sample and the samples in training set. Second, it computes the average of pairwise distances between samples in the stored training set in order to estimate within-user variation. These two average distances are then used to compute a dissimilarity score in order to decide whether to grant a requested access or not.

### 3.1. Pairwise distance between finger-drawn PIN samples

A finger-drawn PIN sample consists of a series of x-y coordinates and time-stamps. This series is divided into strokes where each stroke is defined as a sequence of consecutive points beginning from a touch-down event to the next touch-up event. A preprocessing step for each finger-drawn PIN sample normalizes its sampling rate. To this end, linear interpolation is used to derive a resampled signal with a uniform rate. Then the system performs stroke concatenation by translating the origin of the latter stroke to the end of the former stroke in order to derive a stroke translation invariant signal. More details of such preprocessing techniques can be seen in [23].

Once the preprocessing step is performed, the PIN is represented by a sequence of uniform sampled points. Then a pairwise distance between finger-drawn PIN samples is computed using Dynamic Time Warping algorithm as follows. Let $P$ and $Q$ be the representations of two processed finger-drawn PINs with lengths $N_1$ and $N_2$,

$$P = \{\bar{s}_1^P, \bar{s}_2^P, ..., \bar{s}_{N_1}^P\},$$

and

$$Q = \{\bar{s}_1^Q, \bar{s}_2^Q, ..., \bar{s}_{N_2}^Q\},$$

where each is a time series of touchpoints $\bar{s}_i^P$ and $\bar{s}_j^Q$ and each touchpoint is represented by $x$ and $y$ coordinates: $\bar{s}_i^* = (x_i^*, y_i^*)$. To compute a DTW distance between these two time series, the DTW algorithm constructs a $N_1 \times N_2$ distance matrix $DTW$ where each element $dtw(i, j)$ is computed as,

$$dtw(i,j) = d(\bar{s}_i^P, \bar{s}_j^Q) + \min \begin{cases} dtw(i-1, j) \\ dtw(i, j-1) \\ dtw(i-1, j-1) \end{cases}$$

where $d(\bar{s}_i^P, \bar{s}_j^Q)$ is defined as Euclidean distance between two points on the plane.

$$d(\bar{s}_i^P, \bar{s}_j^Q) = \sqrt{(x_j^Q - x_i^P)^2 + (y_j^Q - y_i^P)^2}$$

The warping distance $W(P, Q)$ between two time series is defined as $dtw(N_1, N_2)$ which is the minimum cumulative distance derived from the warping path that best match these two sequences. This warping distance is then normalized with the minimum length between these two sequences. Finally, the pairwise distance between the two samples is defined as,

$$D(P, Q) = \frac{W(P, Q)}{\min(N_1, N_2)}$$

### 3.2. Dissimilarity score between a set of training samples and a finger-drawn PIN sample

Let $\{T_1, T_2, ..., T_n\}$ be the stored templates of a user. To verify a test sample $S$, we compute all pairwise distances between the test sample $S$ and the training samples $T_i$, in order to quantify the difference between the test sample and the training samples. In addition, we compute all pairwise distances between the training samples $T_i$ and use their sum as denominator to compensate for within-user variation difference. That is, the dissimilarity score between a set of training samples $T_i$ and a finger-drawn PIN sample $S$ is defined as,

$$\bar{D}(S|T_i) = \frac{\sum_{i=1}^{n} D(T_i, S)}{\sum_{i=1}^{n} \sum_{j=1}^{n} D(T_i, T_j)}$$

If this score is smaller than a threshold, verification succeeds, otherwise, it fails.

### 3.3. Evaluation metric

In this work, as per normal practice, verification performance of the system is evaluated using Equal Error Rate (EER), or the rate at which False Acceptance Rate (FAR) and False Rejection Rate (FRR) are equal. In addition, the ROC curve, which is a trade off curve between True Positive Rate (TPR) and False Positive Rate (FPR) is used as another evaluation metric to demonstrate the performance of the system under different attack scenarios at different operating points.

## 4. EXPERIMENTS

In this section, we present the experiments that were performed to evaluate the performance and effectiveness of the proposed system on a dataset comprising of finger-drawn 4-digit PINs collected from users with iOS devices in an uncontrolled environment.

### 4.1. Data acquisition

Genuine samples of finger-drawn PIN were collected during the online signature research project described in [23]. The participants in this project were recruited from a departmental mailing list where each volunteer was offered a $10 gift card to participate in six sessions over the course of approximately seven days. The minimum time interval between each session varied from 12 to 96 hours. The interval between the first and the second session, between the second and the third session, and between the third and the forth session were set at 12 hours. The interval between the fourth and the fifth session and between the fifth and the sixth session were set at 96 hours and 24 hours, respectively. These intervals were chosen to capture variations in times of the days when user performed the experiment. In each session, a user draws a PIN (assigned when she created account) once at the end of the session. At the end of the experiment, each user provided 6 finger-drawn 4-digit PIN samples.

A user entered a PIN by visiting the project website written in PHP and on a HTML5 platform. Thus a user could draw the PIN anytime (after waiting interval had passed) and anywhere without supervision. As a result, we were able to capture intra-user variations caused by numerous contextual factors, for example, device orientation and environmental conditions. When the user was drawing on the screen, she received visual feedback to see what she drawn. She could clear out and redraw until she satisfied with her drawing. Once she clicked to save her PIN, the data consisting of a series of x-y coordinates and times-stamps were sent to our server. In total, we have 40 users and 240 genuine samples.

We considered and collected imposter samples from two attack scenarios.

1. **PIN Attack**: In this scenario, an attacker knows a user's PIN probably by observations or from insiders, but does not know how a user draws the PIN. Attackers in this experiment were presented the PINs before drawing them in their own way. Each attacker provided 6 samples for each PIN and, for each genuine user, we collected 30 PIN attack samples from 5 attackers. In total, we collected 1200 PIN attack samples.
2. **Imitation Attack**: This scenario reflects the situation where an attacker observes how a target victim user draws the PIN several times and then tries to imitate this user. In this attack mode, an attacker was presented with an animation of an exact reproduction on how the PIN was drawn by a victim. The attacker then had 15-20 seconds to learn from this animation. Each attacker then provided 6 samples imitating the observed finger-drawn PIN of a particular user. Similar to the PIN attack, each user had 30 attack samples from 5 attackers, and in total, we have 1200 imitating samples.

### 4.2. Evaluation results

Verification performance of the proposed system was evaluated under the two attack scenarios. First, we examined the impact of training size on verification performance. To this end, we use the first $n$ samples from enrollment phase as a training set for each genuine user. The rest of the samples from the same user are used as positive test samples. The negative samples of those users were collected under two attack scenarios as described in the previous subsection. Note that the training sample in this experiment was chosen in a timely manner in order to preserve causality in real-world applications. The performance in terms of EER for each attack scenario with training set size ranging from 3 to 5 is reported in Table 1.

**Table 1**. The performance (EER) when the training samples are drawn from first $n$ enrolled samples

| Attack | EER(%) at | | |
|---|---|---|---|
| | $n = 3$ | $n = 4$ | $n = 5$ |
| PIN Attack | 24.96 | 17.17 | 12.5 |
| Imitation Attack | 30.79 | 21.25 | 12.83 |

The results show that the verification performance of the system improves when a larger number of samples are used for training. This is not surprising since, with a larger training size, the system could potentially learn more about variation boundary of users finger-drawn PINs thereby resulting in higher performance. In addition, the results also reveal that the verification performance of the system under imitation attack is worse than that of the system under PIN attack. This result indicates that if an attacker gains more knowledge about a user's finger-drawn PIN, she could potentially use that knowledge to increase her chance of success.

Next, we performed a cross-validation test when any five samples are used as a training set in order to arrive at a better estimation

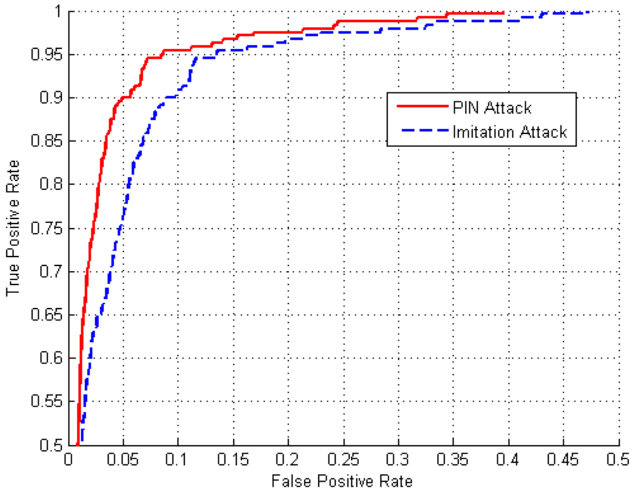

**Fig. 2**. ROC curves of 2 attacks in cross-validation test

**Table 2**. The verification performance (EER) when using 5 training samples in cross-validation test

|  | PIN Attack | Imitation Attack |
|---|---|---|
| EER (%) | 6.70 | 9.89 |

of the verification performance. That is, performance evaluated on too few number of samples could be inaccurate. In particular, when only 6 genuine samples are available for each user, the evaluation result of the system that uses the first 5 samples for training is evaluated from only 40 positive samples. In other words, each positive sample would account for 2.5% FRR if it is rejected. Consequently, rejected genuine samples could make a huge impact on reported verification performance.

In cross-validation test, each and every genuine sample is used as a positive test sample exactly once. When a sample is used as a positive test sample, the other five remaining samples are used for training. By this strategy, we now have $40 \times 6 = 240$ positive samples to evaluate the performance of each attack model. Note that the number of negative samples is also multiplied by 6 as each negative samples is verified against six different training sets. The result is reported in Table 2. In addition, ROC curve representing verification performance of the system with 5 training samples using cross-validation is illustrated in Figure 2. The performance is noticeably better in both attack models. In particular, EER decreases from 12.5% to 6.7% and from 12.83% to 9.89% in PIN attack and Imitation attack, respectively. This is encouraging since compare with traditional PIN authentication system where only the knowledge of PIN is used to ensure the identity of a user and once attacker knows the PIN, the attack success rate is 100%. However, this result needs to be verified on a larger dataset in order to confirm that the improvement is from using a larger set of positive test samples for evaluation and not from training bias caused by cross-validation strategy itself.

### 4.3. Verification time

In our dataset, as shown in Figure 3, 90% of PINs were drawn in less than 6 seconds whereas average drawing time for a PIN was 3.7 seconds with a standard deviation of 1.4 seconds. This is similar to the time a user takes to enter a PIN on keypads. That is, our system does not increase the verification time of a PIN-based authentication system.

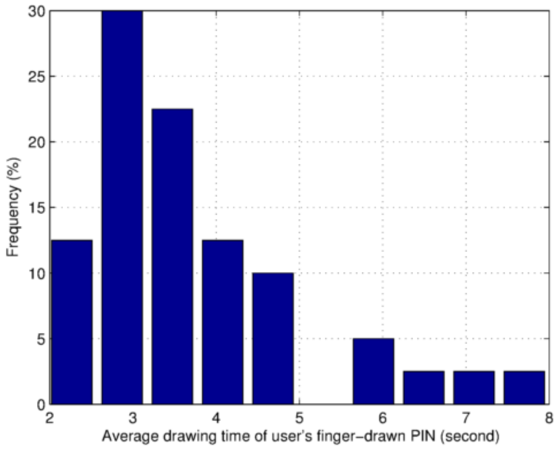

**Fig. 3**. The distribution of drawing time for finger-drawn PINs

## 5. CONCLUSION AND FUTURE WORK

We have proposed an online finger-drawn PIN authentication that can be built on top of existing PIN authentication system. In the proposed approach, a user enters a PIN by drawing it on a touch sensitive display including mobile devices, and tablets. To evaluate the performance, we collected 240 genuine samples from 40 users and 2400 attack samples to model two shoulder surfing attack scenarios. The proposed system achieves verification performance at 6.7% EER under PIN code attack, i.e., the scenario where an attack has knowledge about user's PIN, and achieves a worse performance at 9.9% EER under Imitation attack, i.e., the scenario where an attack has knowledge about user's PIN as well as a dynamic construction of drawing sequences.

However, there are several limitations in this study which shall be addressed in future work. First, the current performance can be improved by developing a problem specific verification algorithm for finger-drawn PINs that works more effectively. Second, the performance has to be verified on a larger dataset with more users and samples in order to increase the accuracy of the evaluation result. In addition, the effect of template aging on verification performance reduction and template updating strategies to counter this issue has to be explored.

Another direction of future work is to evaluate usability and the effectiveness of invisible finger-drawn PIN system which can provide better protection to shoulder surfing attack. Invisible PIN is an approach where users draw PINs on touch sensitive display without live visual feedback on the screen. In addition, the proposed approach could also be extended to text password where users draw their passwords on the display instead of typing them.

## 6. REFERENCES

[1] DataGenetics, "Pin analysis," http://www.datagenetics.com.-blog/blog/september32012, 2012.

[2] Adam J Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the 4th USENIX conference on Offensive technologies*. USENIX Association, 2010, pp. 1–7.

[3] Monji Kherallah, Lobna Haddad, Adel M. Alimi, and Amar Mitiche, "On-line handwritten digit recognition based on trajectory and velocity modeling," *Pattern Recognition Letters*, vol. 29, no. 5, pp. 580 – 594, 2008.

[4] Wen-Li Jiang, Zheng-Xing Sun, Bo Yuan, Wen-Tao Zheng, and Wen-Hui Xu, "User-independent online handwritten digit recognition," in *Machine Learning and Cybernetics, 2006 International Conference on*, Aug 2006, pp. 3359–3364.

[5] Xiaolin Li, Rejean Plamondon, and M. Parizeau, "Model-based online handwritten digit recognition," in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, Aug 1998, vol. 2, pp. 1134–1136 vol.2.

[6] R. Plamondon and S.N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 1, pp. 63–84, Jan 2000.

[7] Volker Roth, Kai Richter, and Rene Freidinger, "A pin-entry method resilient against shoulder surfing," in *Proceedings of the 11th ACM conference on Computer and communications security*. ACM, 2004, pp. 236–245.

[8] Desney S Tan, Pedram Keyani, and Mary Czerwinski, "Spy-resistant keyboard: more secure password entry on public touch screen displays," in *Proceedings of the 17th Australia conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future*. Computer-Human Interaction Special Interest Group (CHISIG) of Australia, 2005, pp. 1–10.

[9] Manu Kumar, Tal Garfinkel, Dan Boneh, and Terry Winograd, "Reducing shoulder-surfing by using gaze-based password entry," in *Proceedings of the 3rd symposium on Usable privacy and security*. ACM, 2007, pp. 13–19.

[10] Alexander De Luca, Roman Weiss, and Heiko Drewes, "Evaluation of eye-gaze interaction methods for security enhanced pin-entry," in *In: Proceedings of OZCHI*, 2007.

[11] N.K. Ratha, J.H. Connell, and R.M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.

[12] Fabian Monrose, Michael K Reiter, and Susanne Wetzel, "Password hardening based on keystroke dynamics," *International Journal of Information Security*, vol. 1, no. 2, pp. 69–83, 2002.

[13] Kenneth Revett, "A bioinformatics based approach to user authentication via keystroke dynamics," *International Journal of Control, Automation and Systems*, vol. 7, no. 1, pp. 7–15, 2009.

[14] Ahmed Awad E Ahmed and Issa Traore, "A new biometric technology based on mouse dynamics," *Dependable and Secure Computing, IEEE Transactions on*, vol. 4, no. 3, pp. 165–179, 2007.

[15] Nan Zheng, Aaron Paloski, and Haining Wang, "An efficient user verification system via mouse movements," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 139–150.

[16] Marcos Faundez-Zanuy, "On-line signature recognition based on vq-dtw," *Pattern Recognition*, vol. 40, no. 3, pp. 981 – 992, 2007.

[17] Hao Feng and Chan Choong Wah, "Online signature verification using a new extreme points warping technique," *Pattern Recognition Letters*, vol. 24, no. 16, pp. 2943 – 2951, 2003.

[18] Anil K Jain, Friederike D Griess, and Scott D Connell, "On-line signature verification," *Pattern recognition*, vol. 35, no. 12, pp. 2963–2972, 2002.

[19] Jing Tian, Chengzhang Qu, Wenyuan Xu, and Song Wang, "Kinwrite: Handwriting-based authentication using kinect," in *Annual Network & Distributed System Security Symposium (NDSS'13)*. Internet Society, 2013.

[20] Napa Sae-Bae and Nasir Memon, "A simple and effective method for online signature verification," in *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the*. IEEE, 2013, pp. 1–12.

[21] Napa Sae-Bae, Kowsar Ahmed, Katherine Isbister, and Nasir Memon, "Biometric-rich gestures: a novel approach to authentication on multi-touch devices," in *Proceedings of the 2012 ACM annual conference on human factors in computing systems*. ACM, 2012, pp. 977–986.

[22] Napa Sae-Bae, Nasir Memon, and Katherine Isbister, "Investigating multi-touch gestures as a novel biometric modality," in *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*. IEEE, 2012, pp. 156–161.

[23] N. Sae-Bae and N. Memon, "Online signature verification on mobile devices," *Information Forensics and Security, IEEE Transactions on*, 2014.

[24] Napa Sae-Bae and N. Memon, "Online signature verification on mobile devices," *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 6, pp. 933–947, June 2014.

[25] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," *Pattern Recognition*, vol. 36, no. 10, pp. 2271 – 2285, 2003.

[26] A. Kholmatov and B. Yanikoglu, "SUSIG: an on-line signature database, associated protocols and benchmark results," *Pattern Analysis & Applications*, 2008.

[27] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, D. Escudero, and Q.-I. Moro, "Mcyt baseline corpus: a bimodal biometric database," *Vision, Image and Signal Processing, IEE Proceedings -*, vol. 150, no. 6, pp. 395 – 401, dec. 2003.