AirSign: A Gesture-based Smartwatch User Authentication

Attaullah Buriro*‡, Rutger Van Acker§, Bruno Crispo*† and Athar Mahboob‡
*Department of Information Engineering and Computer Science (DISI),
University of Trento, Via Sommarive, 38123, Italy,
Email*:{attaullah.buriro, bruno.crispo}@unitn.it
†DistriNET, KULeuven, Belgium
‡Department of Information Security, Khwaja Fareed University of Engineering & Information Technology
Rahim Yar Khan, Pakistan
Email*:{attaullah.buriro, athar.mahboob}@kfueit.edu.pk
§Swift via Exellys, Hamont-Achel Hamont, Flanders, Belgium
Email‡: rutgervanacker@outlook.com

***Abstract*—Smartwatches have been seen as the next generation personal devices, after smartphones, due to their use for wide range of applications, e.g., communication, financial applications, social networking, and fitness tracking, etc. These applications generate and store a lot of personal and private data which needs to be protected from unauthorized access. Several authentication schemes based on PIN/password, graphical password have been proposed, however, it is quite difficult to enter or sketch them because of small smartwatch screen. In addition, these schemes still suffer from unsolved security issues. In this paper, we present a novel movement-based behavioral biometric authentication solution for smartwatch, namely, AIRSIGN. AIRSIGN requires smartwatch worn user to write her name in the air. AIRSIGN leverages the arm-movements while the user swings her finger in the air during the process of writing her name. More specifically, AIRSIGN generates arm-movement fingerprint to profile and identify the user. We implemented and evaluated our scheme on Motorola Moto 3G smartwatch. We applied Dynamic Time Warping (DTW), in two settings, and One-class Multilayer Perceptron (OMLP) as classifiers. AIRSIGN has the potential to replace the traditional authentication schemes and be accepted by users.**

***Index Terms*—Behavioral Biometrics, User authentication, Smartwatch, Sensors**

I. INTRODUCTION

Smartwatches are the most personal devices these days because users wear them all the time and use them for a variety of applications, such as call making/listening, fitness recording/tracking, performing online transactions[1], etc. Since these devices are meant to generate and store users personal data, it becomes extremely important to protect that data from unauthorized access.

Knowledge-based authentication schemes, i.e., PIN/passwords, require users to remember them and enter them every time they would want to interact with the device. These schemes put an additional cognitive load on the users and thus users choose simpler passwords/sketches, which makes easier for the impostor to attack [1]. Additionally, users find them cumbersome to enter or sketch them on the smartphones because of their small screen size.

Biometric-based authentication solutions leverage human biological modalities, i.e., face, fingerprint, retina, etc., for identity verification. Despite having security [2] and usability [3] limitations, these solutions have already been deployed on smartphones, e.g., Apple Face ID [4] and fingerprint in iPhones [5], however, their addition to smartwatches is still awaited.

This paper presents AIRSIGN - a behavioral biometric-based novel and user-friendly authentication solution, for smartwatch unlocking, which establishes the user identity with the motion-based signature, collected during the process of user's signing-in-the-air action. More specifically, AIRSIGN collects 3d sensory readings from smartwatch's built-in accelerometer and gyroscope sensors at a sample rate of 50Hz, while the user writes her name in the air, to perform user profiling. For the testing, it repeats the process and performs the identity confirmation. AIRSIGN using state-of-the-art machine learning classifiers, decides if the smartwatch is worn by the owner or by an impostor. Access to the smartwatch is granted to the owner upon identity confirmation otherwise it is denied. AIRSIGN offers three advantages: (i) it does not require any additional dedicated hardware, (ii) it is secure, as it is based on the person-specific gesture and mimicking this behavior would be extremely difficult, and (iii) it is user-friendly - because it does not require any token or password from the user, making it usable. We validated the efficacy of AIRSIGN on a real Android device (Motorola Moto 3G smartwatch) and obtained acceptable results.

The main contributions of this work are listed below:

- The proposal of AIRSIGN - an arm-motion-based user-friendly user authentication scheme for Android smartwatch. The scheme authenticates the users based on the differences in the arm-movements generated while the smartwatch worn user performs signature-in-the-air gesture with his finger.
- Proof-of-the-concept prototype implementation of the scheme on a smartwatch.

[1]http://www.passengertransport.co.uk/2015/01/your-transport-ticket-on-your-watch/

II. Related Work

Behavioral biometric-based user authentication for user authentication on smartwatches, is comparatively a less-explored area. We survey, in this section, the most relevant studies proposed for smartwatch user authentication.

Draw-a-pin [6] exploits drawing style and correctness of the drawn PIN, to authenticate the user. Authors used the Samsung Gear Live smartwatch to collect data from 30 participants in sitting and walking activities and achieved 4.48% (for PIN attacks) and 14.11% (for Imitation attacks) Equal Error Rate (EER) using Dynamic Time Warping (DTW), as the classifier. In another study, Lewis et al. [7] proposed a free-form arm movement as a behavioral modality for user authentication on smartwatches. They also used DTW as a classifier, on their collected dataset of 5 users, and achieved an accuracy of 84.6%. "VeriNET" [8] also treats motion signals as password and proposes deep recurrent neural network as a classification model for user authentication on smartwatches. "VeriNET" achieved an EER of 7.17% on PINs and 6.09% on Android lock patterns, on the collected $\approx 60k$ passcode entries from 310 users.

We consider the works of Kumar et al. [9] and Buriro et al. [10] very relevant to our work. The study [9] proposes arm movement based continuous smartwatch user authentication. Their scheme leverages built-in smartwatch motion sensors, i.e., accelerometer and gyroscope. The scheme was evaluated in 3 different environments, i.e., intra-session (40 testers), inter-session (40 testers) and inter-phase (12 testers) using K-Nearest Neighbor (KNN), Logistic Regression (LR), Multilayer Perceptron (MLP) and Random Forest (RF) as classifiers. Authors achieved 0% mean Dynamic False Accept Rate (DFAR) and dynamic False Reject Rate (DFRR), for all of the twelve authentication mechanisms in the intra-session analysis on raw sensory readings. Similarly, in the inter-session environment, KNN outperformed all other chosen classifiers and yielded 2.2% and DFRR of 4.2% on fused features. However, DFAR and DFRR increased up to 15.03% and 14.62%, respectively, on the same features using KNN classifier.

The study by Buriro et al. [10] investigates finger-snapping gesture as a behavioral-biometric modality for smartwatch unlocking. Their proposed scheme SnapAuth collects arm-movements during the process of finger-snapping gesture. SnapAuth also exploits the accelerometer and gyroscope sensor for user profiling and based on the differences in the generated movements verifies the users' identity. Using one-class MLP as the classifier, they achieved 82.34% True Acceptance Rate (TAR) at 34.12% False Accept Rate (FAR), on very limited training samples, i.e., just 15.

AirSign is different from the state-of-the-art solutions in at least two ways: firstly, it exploits a novel human behavior (signing-in-the-air) gesture which is easy to perform, and secondly, the data collection is fully unobtrusive, making it attractive for designing the frictionless user authentication solutions.

III. Our Solution - AirSign

This work uses the Motorola Moto 360 smartwatch for the implementation of our proposed authentication scheme. AirSign leverages very common human behavior, i.e., signing - the way a smartwatch worn user writes her name, in the air. AirSign collects movement data from smartwatch sensors, i.e., accelerometer and gyroscope, process this data to extract relevant features and applied machine learning to confirm the users identity. In this work, we explore two data fusion levels, namely, at sensor and feature level. At sensor level, we explore two existing DTW approaches for decision making. Then, we extract features and apply feature fusion. We test AirSign in two settings: (i) testing the attempts of the legitimate user, and (ii) testing the attempts of would-be adversaries. Every successful and unsuccessful legit attempt is added to TAR and FRR table. Similarly, every successful and unsuccessful adversarial attempt is added in FAR and True Reject Rate (TRR).

IV. Methodology

In this section, we discuss the steps taken to design AirSign. This work uses the Motorola Moto 360 smartwatch for designing AirSign. We exploited built-in accelerometer and gyroscope sensors for the generation of raw sensory data, for both data collection and validation phases. Moto 360 delivers a 50Hz sampling rate for both the accelerometer and gyroscope meaning that both sensors are able to ideally generate 50 samples per second with an error margin around +/- 3 samples per second. Figure 1 shows the main screen of our developed application, and Figure 2 illustrates the recording process while the user performed signing-in-the-air action.

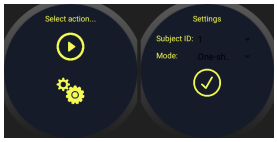
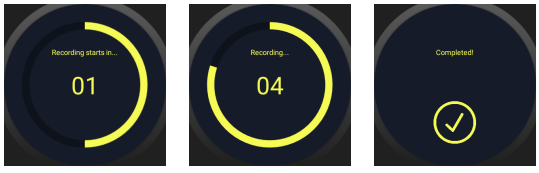


Fig. 1: Main screen and settings

We could attract 11 volunteers (8 males) to our 3-days & 3-session (one session per day) long experiment. All the participants had somehow a computer science background, but not necessarily the security. The testers wore the smartwatch in their natural hand (in the same wrist they normally wear their watch in).



(a) (b) (c)

Fig. 2: (a) start of data recording & (b) data recording (c) recording complete [10].

A. Feature Extraction

We process data in two settings: firstly, we compare the raw data generated from the three dimensions of accelerometer and gyroscope. We also added fourth dimension, computed by taking average of all the three dimensions and taking the square root of the output [11], [12]. Mathematically, it can be computed as below:

$$S_m = \sqrt{s_x^2 + s_y^2 + s_z^2} \tag{1}$$

Where S_m represents the magnitude of sensor S and s_x, s_y and s_z represent the values of the X, Y and Z streams, respectively, from sensor S.

Every sign-in-the-air gesture contains n-records of sensory readings where *n* is the sample rate of sensor *s*. A sensor vector is then formally defined as:

$$X = (s_x^1, \ldots, s_x^n); Y = (s_y^1, \ldots, s_y^n); Z = (s_z^1, \ldots, s_z^n) \tag{1}$$

$$M = (s_{magn}^1, \ldots, s_{magn}^n) \tag{2}$$

$$FV_s = (X, Y, Z, M) \tag{3}$$

where sensor *s* represents either the accelerometer or gyroscope.

In order to form a final combined vector, the four dimensions for both the accelerometer and gyroscope are fused producing an eight-dimensional sensor vector. Formally a combined sensor vector is defined as:

$$FV_{combined} = (X_a, X_g, Y_a, Y_g, Z_a, Z_g, M_a, M_g) \tag{4}$$

Where X_s, Y_s, Z_s and M_s are equally defined as in (1) for both the accelerometer and gyroscope represented by *a* and *g* respectively.

In the 2nd setting, we compute the features from the collected raw streams and concatenate together to form a feature vector. We extracted four statistical features, namely, mean(μ), standard deviation(σ), skewness(γ) and kurtosis(γ'), from every acquired raw stream. A feature set for stream *d* of sensor *s* is formally defined as:

$$FS_{s_d} = (\mu_{s_d}, \sigma_{s_d}, \gamma_{s_d}, \gamma'_{s_d}) \tag{5}$$

With these four available features, i.e., one per dimension, an individual feature vector for sensor *s* is constructed similar to sensor vectors at sensor level.

B. Classification

In the experimental analysis, the two levels of feature fusion are explored. At sensor level, two existing DTW approaches are tested [13], [14]. Next, we perform feature level analysis. The performance is tested in two different settings, i.e. *attempts of the legitimate user* and *attempts of adversaries*. Where both scenarios measure the amount of times legitimate and adversary users are accepted. This evaluation is expressed in terms of a TAR and a FAR. When reporting results, all individual results are averaged over all the participants to compare the accuracy.

1) Dynamic Time Warping: In the first stage of the experimental analysis, the sensor level is explored using DTW. To compare similarity between two time series, which may vary in length, DTW is an efficient and practical algorithm to use. To sketch how DTW works, the algorithm takes two time series $T_1 = (t_1^1, t_2^1, \ldots, t_n^1)$ and $T_2 = (t_1^2, t_2^2, \ldots, t_n^2)$ with length n and m, respectively, and a distances function δ as input. A $n \times m$ matrix *D*, called a distance matrix, is then calculated where each cell in D is computed by applying the following formula:

$$D[i,j] = \delta(t_i^1, t_j^2) + min(D[i-1, j-1], D[i, j-1], D[i-1, j])$$

With this distance matrix, DTW tries to find the optimal similarity between T_1 and T_2 by constructing a so called warping path *W*. This path is the optimal alignment between the two time series and has the minimal cumulative distance among all possible paths in *D*.

In this analysis, two existing approaches are explored on the collected data set. First, Conti et al. [13] proposed to use the classical DTW algorithm combined with the calculation of the maximum distance between all time series used for training. This maximum distance is obtained by calculating the pairwise distance between every training sample and then selecting the largest distance from these calculated distances. Once this maximum distance is obtained, and thus the classifier is considered to be trained, the authentication phase is activated. In this phase, a time series T_{cred}, representing a login attempt, is compared with DTW to every sample in the training set. The sample is considered similar and thus a positive match to a training sample once the following formula is satisfied:

$$d_i < maxDist + \tau \tag{6}$$

In this formula, d_i represents the distance computed by DTW between T_{cred} and training sample T_i, *maxDist* represents the maximum distance calculated during training and τ represents a predefined threshold. Authentication is based on the constraint that T_{cred} gives a positive match with at least more than half the available training samples, otherwise access is denied. By slightly altering this approach at decision level, this part of the analysis explores the impact on the acceptance rates by altering the decision variable. Instead of requiring that only >50% of the training samples should match the attempt, thresholds of 60, 70 and 80% are also explored. Since the addition of τ did not made any sense, calculation of a positive match is done by omitting τ in equ. 6 and accepting users when formally the following holds:

$$|d_i : d_i < maxDist, i = 1 \ldots T| > |T| * \tau \tag{7}$$

Were τ represents a percentage rather than a value as done in the referenced paper.

Next, a more recent use of DTW by Nguyen et al. [14] is considered. In contrast to calculating the maximum pairwise distance in the training set, here the average pairwise distance is calculated per template from the training set. The pairwise

distance of a template is obtained by calculating the distance of T_i with all the other training samples T_j where $i \neq j$ and dividing it by the amount of different training samples available. The template with the lowest average, called T_{ref}, is then selected as the most viable template for authentication. In the authentication phase, a matching or dissimilarity score is calculated for a login sample $T_{attempt}$ with the following formula:

$$Score(T_{attempt}, T) = \frac{DTW(T_{attempt}, T_{ref})}{median(DTW(T_{ref}, T_i))} \quad (8)$$

Note that in the original paper, pairwise distances are calculated differently because a template of a drawn PIN contains n-digits and thus n time series. The pairwise distance is then represented by the sum of all distances from the one-to-one mapping of digits between two templates. However, with the used data in this analysis, there is only one time series per template, thus resulting in easier calculation of pairwise distances with no changes in the fundamental idea of the authentication scheme.

2) One-class Multilayer Perceptron Classifier: We also analyzed our data using popular one-class Multilayer Perceptron (OMLP) as the classifier. OMLP classifier is implemented by using Weka's metaclass; the *OneClassClassifier*[2]. The authentication scheme (Fig.3) is drawn using the *KnowlegdeFlow* module of Weka and performs both training and verification of the *OneClassClassifer*.

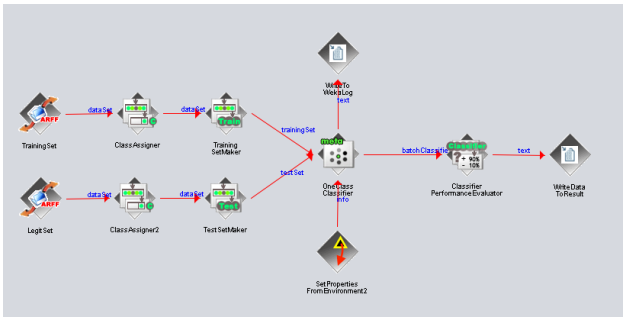


Fig. 3: Authentication scheme using OMLP

C. Analysis

We collected 10 observations per activity per user (in total 50 observations in sitting, standing, walking in the corridor, walking upstairs, and walking downstairs). In 1st setting, we chose the first two observations from each activity (10 in total) to train the classifier and used the remaining 40 observations for testing. Similarly, in the 2nd setting, we chose the first three observations from each activity (15, in total) to train the classifier and used the remaining 35 observations for testing. We are reluctant to use more data, e.g., 50% observations, for training, because in real-world settings, users do not feel comfortable providing more samples.

[2]http://weka.sourceforge.net/doc.packages/oneClassClassifier/weka/classifiers/meta/OneClassClassifier.html

User authentication on smart devices is essentially a one-class classification problem where the data from one user - "the owner" is used to train the classifier and later that classifier is tested on the remaining samples of that one user (to obtain TAR and FRR) and on the data of all the would-be adversaries (to obtain FAR and TRR [15]. We followed the above-mentioned scheme and trained all the chosen classifiers on the data of one user and tested in two settings (as mentioned above). The process is repeated for all the 11 users and the obtained average results are reported.

V. Results

We report the accuracy of our AirSign in terms of TAR and FAR, to avoid redundancy, only as $FRR = 1 - TAR$ and $TRR = 1 - FAR$.

Recall from the previous section, that the 1st iteration of the analysis uses two training samples per activity per user and the 2nd iteration uses three samples per activity. We explain below the results of both iterations which explore both fusions at the sensor and feature level.

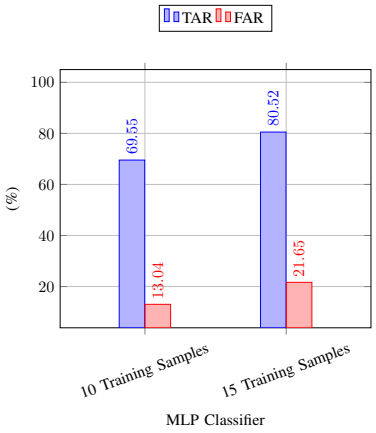


Fig. 4: Authentication results of OMLP classifier

First of all, the two implementations of DTW are checked at sensor level with two different distance functions. The first approach verifies an altered version of the methods applied in *Mind How You Answer Me!* [13]. DTW using the *Euclidean distance squared distance function* does not provide the satisfactory results (see Table I and Table II). We achieved higher TAR, but at higher FAR also, using both distances in both iterations. We obtained best results, i.e., TAR of 85.71% at 43.27% FAR, using 15 training samples,

OMLP classifier performed better on extracted features fused at the feature level (see Fig. 4). We obtained $\approx 70\%$ TAR at 13.04% FAR using this classifier, in its default settings, on just 10 training samples. Similarly, on 15 training samples, OMLP achieved 80.52% TAR at 21.65% FAR. This increase in FAR, in conjunction to TAR, is because of the less number of training samples, i.e., the results could be improved if the classifier is trained on more samples and on the samples of one-activity.

TABLE I: Authentication Results using DTW as the classifier on 10 training samples. The results are averaged over 11 users.

	Euclidean Distance								***Squared Distance***							
	$T = 0.5$		$T = 0.6$		$T = 0.7$		$T = 0.8$		$T = 0.5$		$T = 0.6$		$T = 0.7$		$T = 0.8$	
Reference papers	TAR	FAR	TAR	FAR	TAR	FAR	TAR	FAR	TAR	FAR	TAR	FAR	TAR	FAR	TAR	FAR
[13]	99.55	80.53	97.95	73.80	92.50	68.31	86.14	55.45	99.32	77.80	98.18	74.07	96.36	69.38	91.82	62.87
[14]	55.45	14.87	81.14	33.89	83.64	40.76	84.55	42.69	32.95	11.45	60.00	21.20	67.05	24.40	67.95	25.62

TABLE II: Authentication Results using DTW as the classifier on 15 training samples. The results are averaged over 11 users.

	Euclidean Distance								***Squared Distance***							
	$T = 0.5$		$T = 0.6$		$T = 0.7$		$T = 0.8$		$T = 0.5$		$T = 0.6$		$T = 0.7$		$T = 0.8$	
Reference papers	TAR	FAR	TAR	FAR	TAR	FAR	TAR	FAR	TAR	FAR	TAR	FAR	TAR	FAR	TAR	FAR
[13]	99.48	85.36	99.48	83.98	99.48	79.91	99.22	78.47	99.74	85.18	99.74	83.98	99.22	83.89	98.96	79.76
[14]	65.97	13.04	82.60	33.05	85.45	40.49	85.71	43.27	63.38	13.73	77.14	24.40	77.92	27.56	78.70	29.07

AIRSIGN is a bit less accurate because it is in the early stages and we applied it in the wild. Extensive experimentation with more samples and more testers along with classifier tuning would certainly improve the accuracy of AIRSIGN.

VI. CONCLUSIONS

This paper investigates the use of simple and user-friednly *signing-in-the-air* gesture as a behavioral modality for smartwatch user authentication. The proposed scheme, i.e., AIRSIGN, exploits the arm-motion differences in signing-in-the-air gesture, for user authentication. It authenticates users based on the different arm micro-movements, collected using smartwatch sensors, while the user performs the signing-in-the-air gesture. AIRSIGN is simple and easy for the user because they are used to this gestures. Additionally, AIRSIGN is user-friendly, because it does not require any secret to remember and/or to type. Finally, since the scheme leverages the built-in hardware, it avoids any additional costs.

As future work, we are planning to repeat the experiments with more testers and more samples. We will also investigate the performance and usability analysis by performing corresponding experiments. Additionally, the accurate security testing, i.e., how easier is to spoof AIRSIGN, is also a matter of future work.

ACKNOWLEDGMENT

The work was partially supported by the European Training Network for CyberSecurity (NeCS) grant number 675320.

REFERENCES

[1] Lashkari, Arash Habibi, Samaneh Farmand, Dr Zakaria, Omar Bin, and Dr Saleh, "Shoulder Surfing attack in graphical password authentication", arXiv preprint: 0912.0951 (2009).

[2] Hacker fakes German minister's fingerprints using photos of her hands, https://www.theguardian.com/technology/2014/dec/30/hacker-fakes-german-ministers-fingerprints-using-photos-of-her-hands.

[3] De Luca, Alexander, Alina Hang, Emanuel Von Zezschwitz, and Heinrich Hussmann, "I feel like I'm taking selfies all day! towards understanding biometric authentication on smartphones", 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 1411-1414. ACM, (2010)

[4] About Face ID advanced technology, https://support.apple.com/en-us/HT208108.

[5] How the iPhone 5S Fingerprint Scanner WorksAnd What It Means For You, https://gizmodo.com/how-the-iphone-5ss-fingerprint-scanner-works-and-what-1265703794.

[6] Nguyen, Toan and Memon, Nasir.: "Smartwatches Locking Methods: A Comparative Study", Symposium on Usable Privacy and Security (SOUPS) (2017)

[7] Lewis, Antwane and Li, Yanyan and Xie, Mengjun, "Real time motion-based authentication for smartwatch", IEEE Conference on Communications and Network Security (CNS), pp. 380–381, (2016)

[8] Lu, Chris Xiaoxuan and Du, Bowen and Kan, Xuan and Wen, Hongkai and Markham, Andrew and Trigoni, Niki, "VeriNet: User Verification on Smartwatches via Behavior Biometrics", The 1st ACM Workshop on Mobile Crowd sensing Systems and Applications, pp. 68–73, (2017)

[9] Kumar, Rajesh, Vir V. Phoha, and Rahul Raina, "Authenticating users through their arm movement patterns", arXiv preprint arXiv:1603.02211 (2016).

[10] Buriro, Attaullah, Crispo, Bruno, Eskandri. Mojtaba, Gupta, Sandeep, Mahboob, Athar, and Van Acker, Rutger, "SNAPAUTH: A Gesture-based Unobtrusive Smartwatch User Authentication Scheme", 1st International Workshop on Emerging Technologies for Authorization and Authentication, in press.

[11] Buriro, Attaullah and Crispo, Bruno and Del Frari, Filippo and Wrona, Konrad, "Touchstroke: smartphone user authentication based on touch-typing biometrics", International Conference on Image Analysis and Processing, pp. 27–34, Springer (2015)

[12] Buriro, Attaullah and Crispo, Bruno and Del Frari, Filippo and Wrona, Konrad, "Hold and sign: a novel behavioral biometrics for smartphone user authentication", IEEE Security and Privacy Workshops (SPW), pp. 276–285, IEEE (2016)

[13] Conti, Mauro and Zachia-Zlatea, Irina and Crispo, Bruno, "Mind how you answer me!: transparently authenticating the user of a smartphone when answering or placing a call", 6th ACM Symposium on Information, Computer and Communications Security, pp. 249–259, ACM (2011)

[14] Van Nguyen, Toan and Sae-Bae, Napa and Memon, Nasir, "DRAW-A-PIN: Authentication using finger-drawn PIN on touch devices", computers & security, vol. 66, pp. 115–128, Elsevier (2017)

[15] Buriro, Attaullah and Akhtar, Zahid and Crispo, Bruno and Gupta, Sandeep, "Mobile biometrics: Towards a comprehensive evaluation methodology", International Carnahan Conference on Security Technology (ICCST), pp. 1–6, IEEE (2017)