Mind How You Answer Me!

(Transparently Authenticating the User of a Smartphone when Answering or Placing a Call)

Mauro Conti
CS Department
Vrije Universiteit Amsterdam
De Boelelaan 1081a
1081 HV Amsterdam, NL
mconti@few.vu.nl

Irina Zachia-Zlatea
CS Department
Vrije Universiteit Amsterdam
De Boelelaan 1081a
1081 HV Amsterdam, NL
iza600@few.vu.nl

Bruno Crispo
CS Department
University of Trento
Via Sommarive 14
38050 Povo, Trento, IT
crispo@disi.unitn.it

ABSTRACT

In this paper we propose a new biometric measure to authenticate the user of a smartphone: the movement the user performs when answering (or placing) a phone call. The biometric measure leverages features that are becoming commodities in new smartphones, i.e. accelerometer and orientation sensors. We argue that this new biometric measure has a unique feature. That is, it allows a transparent authentication (not requiring an additional specific interaction for this) to check that the user that is answering (or placing) a phone call is the one authorized to do that. At the same time, this biometric measure can also be used as a non transparent authentication method, e.g. the user may need to move the phone as if answering a call, in order to unlock the phone to get access to SMSs or emails. As a consequence of being a biometric measure, an adversary that spies on the movement (e.g. captures it with a camera) and tries to replicate it, will not be granted access to the phone.

We prototyped our solution and conducted several experiments to assess its feasibility. Results show that the method is effective, and the performance is comparable to that of other transparent authentication methods, like face or voice recognition.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: Authentication

General Terms

Security

Keywords

Smartphone Security, Biometric Authentication

1. INTRODUCTION

Mobile phones have become everyday personal devices. People use them for both managing personal data and handling private communications. Hence, many authentication methods have been provided to restrict access to unauthorized users. Non transparent methods (e.g. PIN) are the most commonly used, while requiring an aware interaction with the user—for this reason, some users tend to avoid this type of authentication. Furthermore, even when in place, these methods often do not block a malicious user to get some access to the phone, e.g. answering to an incoming call. On the other hand, current transparent methods (e.g. keystroke analysis) take a significant amount of time to authenticate the user, and they cannot guarantee that an unauthorized user is blocked before she gets access to the desired data or service. In general, most of the current systems leave the possibility to answer a phone call even if the phone is locked (e.g. with a PIN).

Previous works have already investigated the possibility to use the accelerometer as biometric authentication. However, they either only considered secret movements the user needs to remember [15] (i.e. non transparent authentication), or they require the user to do some gesture that is not naturally connected to the phone usage (e.g. walking [16]).

In this paper, we propose a new authentication method that aims to solve the problems mentioned before. The proposed method offers transparency by identifying if the user that is answering (or placing) a call is the authorized one. In particular, we investigate if a user can be authenticated just by using the movement she performs, from the moment she presses "start" (to initiate a call), until she takes the phone to the ear. We will refer to this movement as a pattern. We treat this movement as a biometric feature, and we demonstrate that there are sufficient differences between different users, such that the movement can effectively be used for identification. In this way, as soon as a call is answered (or placed), the phone can promptly evaluate if the user is authorized to perform this action, and block the system in case of non authorized users.

Moreover, the mechanism might be used even in case of phones shared among multiple users. Each user could have its own profile and authentication would be performed by comparing new movements to the existing profiles. If it is determined that the user accessing the device can be matched to one of the existing profiles, then access is granted, otherwise it is denied. Another possible use of our system is to perform forensics analysis, e.g. to investigate who used the phone at a particular point in time.

Finally, we note that the proposed mechanism might be even used just in substitution of current (non secret) unlocking mechanisms, without user identification purposes: these mechanisms are the ones currently in place to avoid accidentally answering (or placing) a call while the phone is, for example, within the user's bag. Checking for an answering movement pattern after the "start" but-

ASIACCS '11, March 22–24, 2011, Hong Kong, China.

ton is pressed (that can be accidental), the phone avoids that unwanted calls are placed or answered.

Contribution. The main contribution of this work is to assess whether the call answering movement is a unique biometric feature—no previous work has investigated on this issue. We consider the movement performed from the instant an user presses "start", to initiate a mobile conversation, until she places the phone to the ear. Moreover, we propose to use also the orientation sensor for biometric measurements. To our knowledge, so far researchers used only the accelerometer sensor when dealing with movement patterns recognitions. We observe that both orientation sensor and accelerometer sensor are common in smartphones today on the market (e.g. the Google Dev phones).

We propose four basic authentication methods, each one being a specific combination of considered sensor and recognition algorithm. Each of these methods needs a training phase. After this, the method compares a newly observed pattern with the reference ones. Depending on the similarity measure and the threshold set, the method outputs a binary result: 0 (rejected user), 1 (accepted user). We also propose a way of combining these basic methods to improve their performances. In particular, we leverage the fact that, combining methods, the similarity measures can give us more information than the binary one gives as output. We implemented our proposals for the Android system, and tested it with the HTC Dev 1 smartphone considering ten different users. The results are very promising and show that: (i) the proposed method is effective; (ii) the performances are comparable to the ones of other transparent authentication methods, like face or voice recognition—in terms of both False Alarm Rate (FAR) and Impostor Pass Rate (IPR); (iii) the proposed way of combining methods is further able to improve the (already good) performances of the basic methods.

Roadmap. In Section 2 we present the main authentication methods that have been proposed over the years, and we discuss why there is a need for improvements. In Section 3 we present the main technologies used by our solution, the recognition algorithms applied, and the other preliminaries for this paper. Section 4 presents the solution we are proposing. Section 5 reports on the experiments that were conducted and discuss the results obtained. Finally, Section 6 concludes this paper and discuss further possible improvements.

2. RELATED WORK

Smartphones nowadays are very popular. They offer support for an increasing number of applications like web browsers, e-mail clients, applications for editing documents, taking pictures, and so on. This increase of capabilities and usage creates also the need to improve the security of these devices. However, authentication methods already available for smartphones do not offer sufficient transparency and security.

Classical (non transparent, non biometric) authentication solutions, like PIN based methods or graphical passwords, have been proposed long time back. However, being non transparent, these methods ask for the aware participation from the user. This leads often to annoy the user, e.g. continuously prompting her with some challenges. As a result, many users tend to completely remove such authentication methods. Moreover, classical methods based on PINs or passwords are easy to break. This is the case because people choose predictable passwords that have a meaning (e.g. important dates or pet names), making them easy to remember but also easy to break [19]. Similarly, graphical passwords use secret drawings [12], instead of secret strings of characters. Even in this case, users tend to choose symmetric figures making the password space small, and again easy to break. Finally, we have to mention possible attacks where the adversary steals the secret by spying (e.g. with a camera) the user while she inputs the secret (password or drawing). For example, a recent work [7] has shown the feasibility of the "smudge attack" to the Android password pattern. That is, using a camera that takes pictures of the screen of the smartphone, it is possible to reconstruct the pattern drawn by the user on the touch screen, to unlock the phone. This is done by leveraging the light reflex of the smudge left on the smartphone. Interestingly, this seems to be feasible even taking pictures long time after the user drawn her unlocking pattern, or the screen has been (apparently) clean.

Some of the mentioned problems of classical authentication methods can be solved by biometric authentication methods. In fact, these methods increase the security since their secrets can not be easily spied and reproduced—since they identify the user based on her natural features. Biometric measures are classified into two main categories: physiological biometrics and behavioral biometrics [20].

Physiological biometrics identify the user based on her physiological features. They include: face recognition [4], fingerprint recognition [20], external ear shape recognition [18], internal acoustic ear shape recognition [6] (i.e. measuring the shape of the ear channel using acoustic data). However, we found the current physiological biometric solutions to be affected by one or more of the following problems: (i) non transparent usage; (ii) performances are heavily influenced by external factors such as illumination, position or occlusion [20] ; (iii) lack of required hardware on current smartphones.

For example, a good recognition rate could be obtained when using external ear shape recognition [18, 9] (recognition rate of some 90%) or internal acoustic ear recognition [6] (Equal Error Rate, EER, of some 5.5%). However, this methods are heavily influenced by external factors. For example, it is hard to transparently get a useful picture of the ear, or get a useful acoustic feedback that characterizes the internal shape [6], when the ear might be obstructed by hair, or because user is wearing things like hats or veils. Similarly, the camera should be at a distance appropriate to get the correct focus on the target. In fact, we note that in the experiments described in [9], the cited recognition rates were achieved under these conditions: (i) a specific setup for capturing the image; (ii) an active participation of the user (e.g. uncovering the ears from the hair). These constraints result in a completely non transparent authentication of the user.

Within physiological biometric measures, methods that do not suffer much by obstruction problems are fingerprint recognition and internal ear recognition. The area that needs to be captured for fingerprint is small, and usually there is no occlusion that may intervene between the user's finger and the scanner [20] (unless the user wears gloves). However, this method would suffer by the other highlighted problems. That is, it is not transparent to the user and, most importantly, it cannot be leveraged by the technologies already available in smartphones. Similarly, internal ear recognition [6] needs extra devices: a special device that is placed in the ear to emit acoustic signals and a special microphone needs to be attached to the smartphone.

The other classification of biometric measures is the one of behavioral biometrics, where user is identified based on her behavioral features: e.g. keystroke dynamics [10], voice pattern [17], or gesture (e.g. the user's walking pattern [16]). However, for these currently implemented methods the recognition process takes a long period of time. For example, in order to recognize the user from her walking pattern [16], the user would have to walk first for the device to figure out whether she is the correct user or not. For keystroke dynamics the user has to type a phrase, e.g. up to over

100 characters before recognition can be performed [13, 14, 10]. Similarly, for voice recognition the user would have to output some predefined phrases, or sounds, before being authenticated [17].

Recently, researchers proposed other interesting authentication methods, that are non biometric but use accelerometer devices—commonly present in smartphones. These mechanisms aim at identifying the user based on a secret movement pattern [15] (e.g. moving the phone as if to draw an "8" in the air, where "8" is the secret). The movement pattern is measured using data from the accelerometer sensor. The security obtained is high. However, similar to classical PIN or password methods, an adversary might spy the movement, replay it, and get access to the phone and its data.

In this paper, we present a method that solve the problems of the cited methods. The proposed solution is to measure the movement pattern performed by a user while answering (or placing) a phone call. In particular, the considered pattern initiates from the moment the user presses "start"—to answer an incoming call or to initiate a new one—up to the moment she brings the phone to the ear. Hence, as soon as the phone reaches the ear, the measuring ends and the recognition process starts. We observe that, differently from the solution in [15], in our proposal the secret is not the movement itself (e.g. the "8" drawn in the air) but is the biometric measures of a specific user's movement—i.e. even if an adversary spies how the user answers the phone, she is not able to replay the movement in a way such that it can replicate the biometric features of the correct user.

Furthermore, we find particularly interesting to compare our method with the ones of (external or internal) ear recognition. In fact, similar to our method, also these methods could be used for authenticating the user answering a phone call, without requiring a specific interaction for the authentication. However, these methods have the drawbacks discussed before: they suffer from influence of external factors (e.g. hair, hats, and veils); they are not so transparent (the camera must be at an appropriate distance to get the focus); they require devices not commonly present in smartphones (e.g. a camera close to the ear position for external ear recognition, or a microphone close to the ear position for the internal acoustic ear recognition). In addition, even if we assume these problems can be solved (e.g. equipping the new smartphones with new—and costly—devices, and requiring the user not to wear hats or veils), we observe the following. In the ear recognition the registration of the measure of interest starts when the phone is at the ear—that is the moment at which the measure of our pattern ends. Hence, this would further prolong the recognition process, so further delaying the beginning of the actual phone communication.

We propose to measure movement patterns using both the accelerometer sensor and the orientation sensor. To the best of our knowledge the second one has never been used before for authentication purposes. Also, leveraging the idea of combining different authentication methods [21], we propose a specific way of combining our different proposed authentication methods (based on different sensors and different recognition algorithms). In this way, we manage to further improve the performances of the basic methods involved in the combination: reducing at the same time both FAR (False Alarm Rate) and IPR (Impostor Pass Rate).

3. PRELIMINARIES AND NOTATION

In this section, we introduce some building block concepts for our proposal. In particular, our proposal leverages technologies widely available on smartphones: accelerometer sensor and orientation sensor. Hence, in Section 3.1 we introduce the working of this sensors. Furthermore, we introduce the building block algorithms that we use to measure similarity between patterns: the Dynamic Time Wrapping (DTW) algorithm (Section 3.2). In particular, we implement two different versions of this algorithm: Dynamic Time Warping Distance (DTW-D, Section 3.2.1), and Dynamic Time Warping Similarity (DTW-S, Section 3.2.2). Finally, in Section 3.3 we introduce definitions and notation used in the rest of the paper.

3.1 Considered Sensors

Current smartphones come equipped with a wide range of sensors, e.g. to measure acceleration, light, magnetic field, orientation, and temperature. Some of these sensors have already been used for authentication purposes. In particular, the accelerometer sensor has been used both for capturing secret movements [15], and for measuring biometric features like the walking pattern [11]. On the contrary, orientation sensor has never been used before our proposal.

Our proposal leverages both accelerometer sensor and orientation sensor. In particular, we implemented a proof of concept of our proposal for the Android system. Android implements the OpenGL ES coordinate system [3]. The coordinate system on the Android platform is defined with relation to the screen of the phone, when the phone is in its default position (the default position—either portrait or landscape—depends on the specific smartphone model, e.g. it is portrait for the Dev 1 phone used in our testing). The origin of the coordinates is given by the lower left corner of the screen. The x-axis are horizontal and point right, the y-axis are vertical and point up, and the z-axis point outside the front face of the screen. This coordinate system applies both to the accelerometer and the orientation sensor. Also, these coordinates do not change when the orientation of the phone is changed.

3.1.1 Accelerometer Sensor (S_a)

The accelerometer sensor measures the forces applied to the phone (minus the force of gravity) on the three axis: x, y and z. This means that when the phone is pushed toward the sky with an acceleration a (expressed in meter per second squared $\frac{m}{s^2}$), the acceleration measured by the sensor will be a + 9.81 $\frac{m}{s^2}$. This value represents the acceleration of the device: a minus the force of gravity (9.81 $\frac{m}{s^2}$).

Let us denote the values of the acceleration of the device on the axis x, y and z, as a_x, a_y, and a_z, respectively. Similarly, let us denote the values of the force of gravity on the axis x, y and z, as g_x, g_y, and g_z, respectively. The values provided by the accelerometer sensor are the following:

- force applied by the device on the x-axis (a_x - g_x);
- force applied by the device on the y-axis (a_y - g_y);
- force applied by the device on the z-axis (a_z - g_z).

3.1.2 Orientation Sensor (S_o)

The orientation sensor measures values of the angles in degrees of arc, representing the orientation of the phone on the three axis. For example, let us assume a user is standing in a point holding the phone in one hand. If the user rotate her body, this will mainly imply a modification of the value referring to the rotation around z-axis. Similarly, a rotation on z is also observed rotating the device from portrait to landscape. The values provided by the accelerometer sensor are the following:

- rotation around z-axis, yaw ($0^\circ \leq yaw \leq 360^\circ$, 0°= North, 90°= East, 180°= South, 270°= West);
- rotation around x-axis, $pitch$ ($-180^\circ \leq pitch \leq 180^\circ$, with positive values when z-axis moves toward y-axis);

- rotation around y-axis, $roll$ (-90°$\leq roll \leq$ 90°, with positive values when z-axis moves toward x-axis).

3.2 Considered Similarity algorithms

As for the similarity algorithm used, we focused on the Dynamic Time Warping (DTW), being it widely used in the literature for authentication purposes [22, 23]. The DTW is an algorithm for measuring similarity between two sequences which may vary in time or speed. It was first used in speech recognition in the 70s but it is currently used in many areas: handwriting recognition, signature recognition, sign language recognition, and gesture recognition.

This algorithm gained its popularity in this field due to its capability of minimizing the effects of shifting and distortion in time, for time series data [22, 23]. The continuity of the input patterns is less important in the DTW than in other pattern matching algorithms (e.g. Support Vector Machines, Bayesian Networks, and Decision Tree Learning) as it is particularly suited for matching sequences with missing information.

In the following, we describe the two specific algorithms that we use in our proposal, both derived from DTW: DTW-D (Section 3.2.1), and DTW-S (Section 3.2.2). Using these two algorithms, and two different sensors S_a and S_o, we propose four different methods, that are: 1) DTW-D with data from S_a; 2) DTW-S with data from S_a; 3) DTW-D with data from S_o; 4) DTW-S with data from S_o.

3.2.1 Dynamic Time Warping Distance (DTW-D)

The Dynamic Time Warping Distance (DTW-D) uses as comparison algorithm the classical DTW algorithm. The data is represented in the form of time series. By a time series we mean a sequence of pairs: each pair represents a 3D point (values x, y, and z) and the corresponding time. In our scenario, the time is normalized such that each sequence starts at zero and all the other values represent the time interval that passed from the starting point.

The result obtained when comparing two time series is a real value ($\in \Re_+$), and it represents a distance measure. The minimum distance that can be obtained is zero. Smaller the result, smaller the distance between the two pattern, higher the similarity. When two identical time series are compared the outputted result is zero.

During the training phase, T patterns (i.e. their corresponding measures) are added to the database. Let us denote these patterns as $t_1, \ldots, t_T$. Once the patterns are added, each two series are compared using the DTW algorithm. The maximum distance ($maxDist$) value obtained in this comparison is stored, and used during the recognition phase. That is:

$$maxDist = max_{i,j=0}^{T}\{DTW(t_i, t_j)\}, \quad (1)$$

where DTW(t_i,t_j) is the similarity measure compute by DTW-D between the patterns t_i and t_j. The $maxDist$ value has the purpose to make the authentication mechanism being dependent on the specific user's behaviour: instead of choosing a general maximum distance allowed between two patterns, we will consider $maxDist$, which is dependent on the training set.

During the recognition phase, a new test pattern, given to the system for recognition, is compared to each pattern in the training set, this resulting in T similarity measure d_i, $i = 1 \ldots T$. If for more than half of these similarity measures are smaller than the maximum distance plus a given threshold (τ_D) the user is considered to be the correct one, and the access is allowed. Formally, a user is accepted if the following holds:

$$|\{d_i | d_i < maxDist + \tau_D, i = 1 \ldots T\}| > \frac{T}{2} \quad (2)$$

If Equation 2 does not hold, the user is considered an impostor—hence, the access is not granted.

3.2.2 Dynamic Time Warping Similarity (DTW-S)

The Dynamic Time Warping Similarity (DTW-S) [5] uses for comparison an adaptation of the classical DTW algorithm. That is, instead of giving as a result a distance measure, this method gives as output a percentage of similarity between the two series. in DTW-S the three axis are considered independently. In particular, three instances are created, one for each axis (x, y and z). When two patterns are compared, the instances corresponding to the same axis are compared. These three results are averaged and the outcome is returned as the final result—that is a percentage value. Two instances that are identical will give as result 100%. Hence, differently from DTW-D, for DTW-S the higher the result, the higher the similarity between two patterns.

During the training phase a number of T patterns is added to the database. Since the results that can be obtained by this method are bounded both on the lower side (0%) and on the upper side (100%), no processing is performed for the training set. That is, instead of using a maximum allowed distance that is dependent on the training set (as done for DTW-D), for DTW-S we only use a maximum accepted threshold τ_S.

During the recognition phase, each new pattern is compared with each training pattern and the results are averaged. If the average obtained is bigger than a given threshold, the user is considered to be the correct one—and the access is guaranteed. On the contrary, if the average is smaller, the user is consider an impostor—and the access is denied.

3.3 Definitions and Notation

We remind that we investigate on the feasibility of using the call answering movement (that the user performs when answering or placing a phone call) as a biometric authentication measure. We also assume that only one user is authorized to answer or place calls. Coherently with previous work on biometric authentication [8, 10, 13, 14], we use the following two definitions to evaluate the performances of our proposal.

Definition 1. **False Alarm Rate (FAR)**. The FAR is the percentage of accesses attempted by the authorized user of the system, erroneously rejected.

Definition 2. **Impostor Pass Rate (IPR)**. The IPR is the percentage of successful accesses to the system by impostors, pretending to be the authorized users.

Table 1 summarizes the notation and abbreviations used in this paper.

4. A NOVEL BIOMETRIC AUTHENTICATION

In the literature (cfr. Section 2) there is no efficient and effective solution to transparently authenticate a user while she answers (or places) a phone call. However, we argue that it is possible to use the values obtained from the accelerometer and orientation sensors (while the user answers, or places, a call) as a biometric measure in order to authenticate the user.

In this section, we first report on a preliminary assessment of our intuition (Section 4.1). Based on the preliminary confirmation of our intuition, we propose four different basic methods (Section 4.2), that are the result of all the possible combinations of considered sensors (accelerometer sensor S_a, and orientation sensor S_o),

Table 1: Notation and Abbreviations.

Symbol	Meaning
FAR	False Alarm Rate
IPR	Impostor Pass Rate
DTW-D	Dynamic Time Warping algorithm with distance feature
DTW-S	Dynamic Time Warping algorithm with similarity feature
S_A	Accelerometer sensor
S_O	Orientation sensor
T	Size of training set
t_i	i-th training pattern (i=1, ..., T)
τ	A generic threshold value
τ_D	Threshold value for algorithm DTW-D
τ_S	Threshold value for algorithm DTW-S
$\hat{\tau}$	Threshold value for combined methods

and considered similarity algorithms (DTW-D and DTW-S). Furthermore, we propose to combine together the basic methods. First, we do this with simple boolean operations (Section 4.3). After, we propose a novel way of combining the basic methods. In particular, we leverage the following fact.

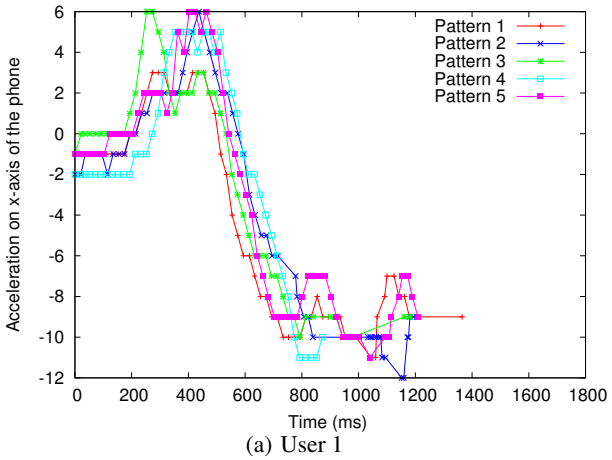
While the final answer of a basic method can only be 1 (accept the user) or 0 (reject the user), we argue that the similarity values computed by these methods actually express more information on the likelihood that the current user is the correct one. Our proposed non-boolean combination (Section 4.4) aims at combining methods in a way such that this additional information can be leveraged to improve, at the same time, both FAR and IPR. The evaluation of all the proposed methods (basic methods and combined ones) is shown in Section 5.

4.1 Intuition Assessment

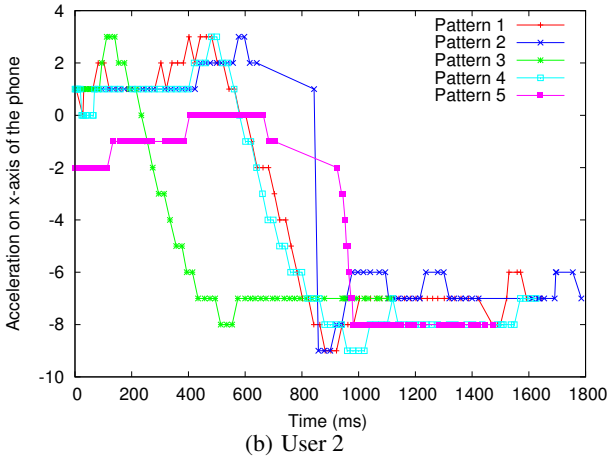
Our proposal is based on the intuition that the movement a user performs while answering (or placing) a call might be used as a biometric measure. In the following, we will refer to this movement (as well as the measures associated with it) as a pattern. In particular, we assume that when the phone rings first the user handles the phone in front of her—to see who is calling—then she presses the "start" button to initiate the call. Similarly, when the user places a call, we assume she handles the phone in front of her—to compose the number or search for a name in the contact list—, then she presses the "start" button to initiate the call. We specifically consider the movement that begins from the instant the user presses "start", until she handles to phone close to the ear.

We run some preliminary experiments to have a confirmation of our intuition. That is, we were looking for a preliminary assessment of the following question: can the measurements associated with the described movement be used as a biometric measure to discern between different users? In practice, the aim of these experiments were just to observe how close are the values measured for the same user, and how far are the patterns observed for different users.

We wrote an Android application that logs the values sensed by the accelerometer sensor (acceleration on x, y, and z-axis) and the orientation sensors ($pitch$, $roll$, and yaw), while the user moves the phone accordingly to the described pattern. We collected this data by asking 10 users to use the test application to trace data of several movement patterns. For space limitation, we report here only part of the results we obtained. In particular, we only report here (Figure 1) the values of the acceleration on x-axis (of the accelerometer sensor) obtained with two users (User 1 and User 2), each one performing the movement five times (Pattern 1, Pattern 2, Pattern 3, Pattern 4, and Patterns 5). Figure 1(a) shows the results of the five patterns of User 1. Figure 1(b) shows the results of the five patterns of User 2. In particular, the x-axis (of the graph) represents the time (starting from the moment the call is initiated), and the y-axis (of the figure) represents the corresponding measured acceleration (on the x-axis of the phone).



(a) User 1



(b) User 2

Figure 1: Examples of patterns. Accelerometer Sensor: acceleration observed on x-axis of the phone (plotted on the y-axis on the graphs).

From Figure 1(a) we observe that the different patterns of the User 1 are very close to each other. On the other hand, the patterns of User 1 are far from the ones of User 2. However, we also observe as the patterns of User 2 are not close to each other as they are the ones of User 1. A similar behavior has been observed also for other movement patterns, for other users, and for other measured values—not reported in Figure 1—i.e. for the other sensor values (y and z-axis for the accelerometer sensor; $pitch$, $roll$, and yaw for the orientation sensor). While the intuition seems to be confirmed by these preliminary results, these data are not sufficient to assess the effectiveness of a possible approach.

4.2 Basic Methods

We propose four different basic methods. These are the result of all the possible combinations of the considered sensors (accelerometer sensor S_a, and orientation sensor S_o), and considered similarity algorithms (DTW-D and DTW-S). Hence, the resulting methods are:

- DTW-D-S_a: this method applies DTW-D algorithm to the data collected by the accelerometer sensor (S_a);
- DTW-D-S_o: this method applies DTW-D algorithm to the data collected by the orientation sensor (S_o);
- DTW-S-S_a: this method applies DTW-S algorithm to the data collected by the accelerometer sensor (S_a);
- DTW-S-S_o: this method applies DTW-S algorithm to the data collected by the orientation sensor (S_o).

We remind that both DTW-D and DTW-S need a training phase, that is, they need to store a set of T training patterns. Furthermore, the results of these algorithms are influenced by the value that is considered as threshold, τ, in order to either accept or reject a new test pattern (not in the training set). In particular, we denote the threshold used for DTW-D and DTW-S, with τ_D and τ_S, respectively. We expect to get more users accepted while having a less strict threshold. We remind that, because of the specific working of the algorithms, for DTW-D this happens when increasing τ_D, while for DTW-S this happens when decreasing τ_S. We investigate the influence of these parameters (T, τ_D, and τ_S) in Section 5.1.

We finally observe that varying a threshold will always influence FAR and IPR in an opposite way. For example, increasing τ_D in the DTW-D algorithm would decrease FAR. However, IPR will be increased. Similarly for other variation of parameter: whenever a variation brings a positive influence on one of the performance metrics (FAR or IPR), the same variation brings a negative influence on the other performance metric.

4.3 Boolean Combinations

In this section, we describe some simple boolean ways of combining the basic methods presented in Section 4.2. The boolean combinations take as input the binary results of the basic methods: 1, for accepted patterns, and 0, for rejected patterns. We build these methods to discuss the behaviour of FAR and IPR, and also as building blocks for the methods proposed in Section 4.4.

First, we propose to combine two methods with a boolean operator. This mechanism of combination is independent from the specific basic methods considered as building blocks. So, we refer in general terms to the building block methods as Method A and Method B. Similarly, we refer to FAR and IPR of Method A as FAR_A and IPR_A (FAR_B and IPR_B, for Method B). When combining two basic methods with AND (cfr. Table 2), a test pattern will be accepted only if both methods accept it (fourth line of Table 2).

Line n.	method A	method B	AND combination
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

Table 2: Output of combination with AND.

Let us now discuss the case of the third line of Table 2: Method A accepts the user, while Method B rejects her. It is clear that in this case, one of the methods has failed; either (i) the user is an impostor and Method A is wrong, or (ii) the user is the correct one and Method B is wrong. In case (i), Method A makes a mistake that would lead to an higher IPR. Hence, taking the AND combination of the two results, the Method B might help to reduce the mistakes of Method A, hence reducing the IPR. However, it could also be the other way around: case (ii). That is, Method B is wrongly rejecting a pattern from the correct user, this leading to an increase of the FAR. Unfortunately, in this scenario, the AND combination does not allow Method A to help reduce these type of mistakes of Method B, hence not allowing a reduction of FAR (note that an OR combination would allow this to happen). As a general result, considering the cases laying in the other lines of Table 2, the AND combination:

- can only reduce the IPR. Resulting IPR will enjoy $\text{IPR} \leq \text{IPR}_A$, $\text{IPR} \leq \text{IPR}_B$. That is:
$$IPR \leq min\{IPR_A, IPR_B\}. \quad (3)$$
- can only increase the FAR. Resulting FAR will enjoy $\text{FAR} \geq \text{FAR}_A$, $\text{FAR} \geq \text{FAR}_B$. That is:
$$FAR \geq max\{FAR_A, FAR_B\}. \quad (4)$$

In terms of sets we can describe the AND combination as follows. Let us consider the following two sets: the set IP_A, of patterns that result to be impostor pass cases for Method A, and the set IP_B, of patterns that results to be impostor pass cases for Method B. The improvement that an AND combination can lead to the resulting IPR depends on the size of $\text{IP}_A \cap \text{IP}_A$. In fact, if $\text{IP}_A \cap \text{IP}_A = \emptyset$, the IPR of the combinations will be 0. On the other side, if $\text{IP}_A \cap \text{IP}_A = \text{IP}_A = \text{IP}_B$ then IPR will be $\text{IPR}=\text{IPR}_A=\text{IPR}_B$.

In general, using AND the patterns will be accepted only when both methods accept it. In this way, the number of accepted patterns decreases, potentially decreasing IPR (i.e. the actual decreasing depends on the size of $\text{IP}_A \cap \text{IP}_A$), and potentially increasing FAR.

When using OR, the patterns will be accepted even if only one of the methods accepts it. In this way, the number of patterns that get accepted increases, potentially reducing FAR and potentially increasing IPR. Hence, the IPR resulting from an OR combination, will enjoy:

$$IPR \leq min\{IPR_A, IPR_B\}. \quad (5)$$

While for the FAR of an OR combination, the following equation holds:

$$FAR \geq max\{FAR_A, FAR_B\}. \quad (6)$$

We might also combine the four building blocks methods all together at the same time, instead of just two by two as described. In this way, the combination mechanism accepts the user based on how many basic methods accept the user (i.e. if n = 1, 2, 3, or 4 out of 4 methods accept her). We expect to get less patterns accepted as we increase n, hence a decreasing IPR, and an increasing FAR.

We can conclude this section by observing that, combining basic methods with boolean operations can be helpful to select a different combination of FAR and IPR. That is, as it is possible by varying T and τ, also with boolean combinations we might look for a different value of one of the performance metrics (FAR and IPR). However, we observe that also with boolean combinations is not possible to improve both FAR and IPR at the same time.

The best possible scenario expected for boolean combinations is that, compared to one of the basic method, one of the metric improve, while the other remain the same. The analysis cannot tell us the chance to have such cases—the actual behaviour depends on the specific users' patterns, that is it depends on the size of the set being intersections of cases, as discussed before for IP_A and IP_B. Hence, we have to conduct experiments (results shown in Section 5.2) to better understand the behaviour of incorrect recognitions (FAR and IPR).

4.4 Leveraging non boolean output—can we do it better, for free?

By combining the basic methods in a boolean way, as presented in Section 4.3, we considered only the binary output (1, accepted, or 0, rejected) of each single basic method involved in the combination. Instead, we argue that each single basic method has potentially more information, rather than just the binary output (accept/reject). In fact, each of the two algorithms (DTW-D and DTW-S) considered in the basic methods gives its output based on a similarity measure. The similarity measure is a value that expresses how close (or how far) is the new test pattern compared to the T patterns in the training set. While a single method can only output a binary value with respect to a threshold, combining methods together, as described below, can convey more information. In particular, the intuition is the following. Assume for example that Method A suggests that the new pattern is very close to the one of the correct user, while Method B suggests that the pattern does not belong to the correct user just because the similarity goes beyond the threshold for a small value. We expect that in this case, the likelihood that Method B is making a mistake is significantly higher than the probability that Method A is making a mistake. We describe combinations that can be applied to both DTW-D (Section 3.2.1) and DTW-S (Section 3.2.2) to leverage additional information from the similarity measure, rather than only binary outputs. We expect that this combination is able to significantly reduce, at the same time, both FAR and IPR (not just one of them as we expect for the boolean combination).

4.4.1 DTW-D Normalized

In DTW-D, a test pattern is compared with the threshold $maxDist + \tau_D$ (cfr. Section 3.2.2). We modify DTW-D to have an output that is normalized, in the range of possible distances from $maxDist + \tau_D$. In presenting the proposed normalization, we refer to Figure 2, where: the line represents an axis where the possible output similarity values lie (lower bounded by 0—for identical patterns); above the line we indicate the range of values that are accepted, [0, $maxDist + \tau_D$), and the range of values that are rejected, [$maxDist + \tau_D$, ∞); a_v and r_v indicate the values assigned to an example of accepted and an example of a rejected pattern, respectively (their computation are explained below); the dotted interval indicates the distance between the value a_v and $maxDist + \tau_D$; similarly, the dashed interval indicates the distance between the example of a rejected value r_v and $maxDist + \tau_D$.

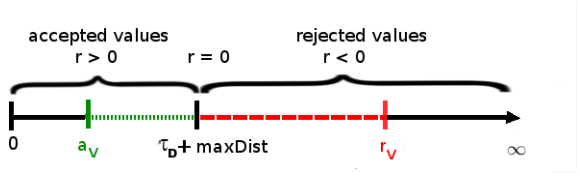


Figure 2: Normalized output for DTW-D.

We remind that DTW-D, in its intermediate steps (cfr. Section 3.2.2), computes for the test pattern a distance d_i from each pattern in the training set ($i = 1 \dots T$). Then, these d_i are evaluated to decide whether to accept or reject the test pattern (cfr. Section 3.2.2). Here, we propose a different usage of these d_i. First, we compute the average of this values ($\sum_{i=1}^{T} d_i/T$). In case this average is $\leq maxDist + \tau_D$ we denote this result as a_v; otherwise, we denote the result as r_v (examples of a_v and r_v are reported in Figure 2). However, we are now looking for a normalized value. That is, we want to make the obtained value normalized to a reference interval. We consider as reference interval the one from 0 to $maxDist + \tau_D$, that is the interval of all possible accepted values. This normalization will apply to both a_v and r_v. The result r outputted by the normalized DTW-D will be:

- $r > 0$, if $(\sum_{i=1}^{T} d_i/T) < (maxDist + \tau_D)$;
- $r < 0$, if $(\sum_{i=1}^{T} d_i/T) > (maxDist + \tau_D)$;
- $r = 0$ if $(\sum_{i=1}^{T} d_i/T) = (maxDist + \tau_D)$.

In general, r is described by the following equation:

$$r = \frac{(maxDist + \tau_D) - \sum_{i=1}^{T} d_i/T}{(maxDist + \tau_D)}. \quad (7)$$

4.4.2 DTW-S Normalized

We give a normalized version of DTW-S in a way that is similar to the one used for DTW-D (Section 4.4.1). The main difference is the way users get accepted. In this case, a test pattern is considered to correspond to the authorized user if the result is greater (not smaller, as for the DTW-D) than a given threshold. We refer to Figure 3 to describe the normalized version of DTW-S. The notation used in the figure is consistent with the one described in Section 4.4.1 for Figure 2. However, we underline that the accepted values are now on the right of the threshold (that here is τ_S), and the rejected values are on the left of the same threshold.

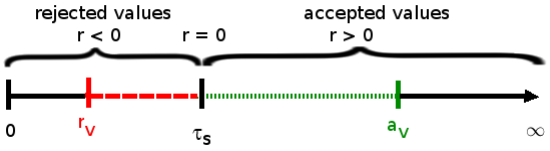


Figure 3: Normalized output for DTW-S.

The result r, for the normalized version of DTW-S, is computed according to the following equation (where s_i is the similarity value that the non normalized version of DTW-S outputs):

$$r = \frac{\sum_{i=1}^{T} s_i/T - \tau_S}{\tau_S} \quad (8)$$

4.4.3 Combining Normalized Results

Now that we have the normalized version of both DTW-D and DTW-S, we need to design the combination mechanism. First, we observe that the result r of the normalized algorithms is no more just binary (1, accept, or 0, reject). The mechanism to combine the results is simple as just computing the sum of the normalized results for each method, and compare it to a new threshold, $\hat{\tau}$.

Let us first consider the case of two methods, say Method A and Method B, that use the normalized versions DTW-D and DTW-S. Let us also denote their results are m_A (for Method A) and m_B (for Method B). In the combined method, the user is accepted if the following equation holds (where α and β are parameters of the combination mechanism):

$$(\alpha m_A + \beta m_C) \geq \hat{\tau}, \quad (9)$$

Parameters α and β are used to regulate the influence of the two building block methods on the overall result. If Equation 9 does not hold, the user is rejected.

More generally, combining all four methods, we propose a similar procedure, except that we compute the sum of all the four methods (m_A, m_B, m_C and m_D), and compare them again with the threshold $\hat{\tau}$. Finally, the user is accepted if the following equation holds:

$$(\alpha m_A + \beta m_B + \gamma m_C + \delta m_D) \geq \hat{\tau}, \qquad (10)$$

where α, β, γ, and δ are used to control the importance given to each method. If Equation 10 does not hold, the user is rejected.

5. EVALUATION

To evaluate our proposal, we performed a wide range of experiments. In particular, we investigated the performances of all the presented basic methods (cfr. Section 4.2), and the possible combinations (cfr. Section 4.4 and Section 4.4). We wrote an Android application, named FANTASY-app to get movement patterns, that is the corresponding values over time of the accelerometer and orientation sensors. We installed FANTASY-app on the Android Dev Phone 1 [1], equipped with Android platform version 1.6. More information on our proposal, the FANTASY-app, and its source code can be found on the project website [2]. We involved in our experiments 10 test users (User 1, . . . , User 10), each of them providing us with 50 movement patterns (for answering or placing a call). In particular, we asked the users to answer the phone in the way we depicted before, put the phone in front of them, press the "start" button to imitate the call, then bring the phone to the ear. We did not consider different ways of starting a call, like using hand-free devices or using voice-recognition to find a name in the contact list and automatically initiate the call.

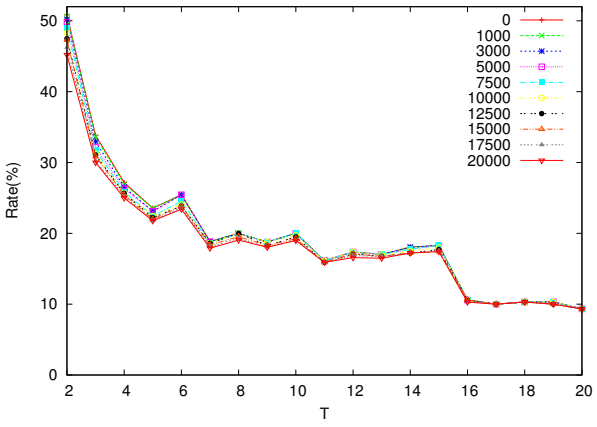
As for performance metrics, we used the ones commonly used for evaluation of biometric authentication systems [8, 10, 9], that are: the percentage of times the correct user is not granted access—FAR (False Alarm Rate)—, and the percentage of times an impostor is granted access—IPR (Impostor Pass Rate). For computing FAR, for each user we trained the system with her first T out of all (50) patterns. Then, we gave as input to the authentication method the remaining $T - 50$ patterns, hence considered test patterns. We counted the percentage of times the system were not accepting this patterns—hence not granting access to the correct user. Similarly, we computed IPR by using the first T patterns of User 1 as training patterns, and the patterns of the the other users as test patterns.

Given the described setting, in the following we show and discuss the results of our experiments. In particular, Section 5.1 discusses the experimental results for the basic methods, Section 5.2 the ones for the boolean combinations, while Section 5.3 present the ones for the combinations considering the normalized version of both DTW-D and DTW-S algorithms. When combining the basic methods, we considered, for each of them, the choice of parameters T and τ as summarized in Table 3.

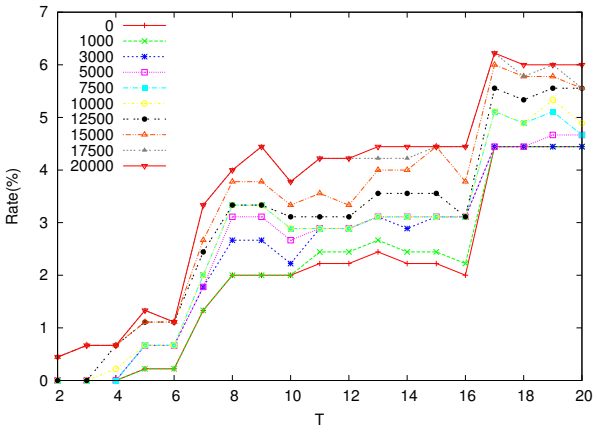
5.1 The basic methods

In this section we show the results obtained for the four basic methods: DTW-D-S_a, DTW-S-S_a, DTW-D-S_o, and DTW-S-S_o. For each of this methods, we varied the number of training patterns, T, from 2 to 20, and we tested 10 different values for the threshold (τ). In particular, since the two considered algorithm DTW-D and DTW-S work in different ways, we also considered for the two of them different set of threshold values. That is, we considered the following values for the threshold τ_D: 0, 1000, 3000, 5000, 7500, 10000, 12500, 15000, 17500, and 20000. Similarly, for τ_S (threshold of DTW-S), we considered the following values: 7%, 8%, 9%, 10%, 11%, 12%, 13%, 14%, 15%, and 16%. We run experiments with all the combination of these parameters and for each combination, we computed FAR and IPR.

Figures 4 and 5 show how the variation of T and τ_D influences FAR and IPR, in the DTW-D-S_o method. Figure 4 reports the results for different values of τ_D, when varying T on the x-axis. In particular, Figure 4(a) shows (on y-axis) the corresponding FAR, and Figure 4(b) the corresponding IPR. Similarly, Figure 5 gives a different view on the same data. It reports the results for the variation of τ_S (on the x-axis) for different values of T. In particular, Figure 5(a) shows the FAR, while Figure 5(b) shows the IPR.



(a) FAR

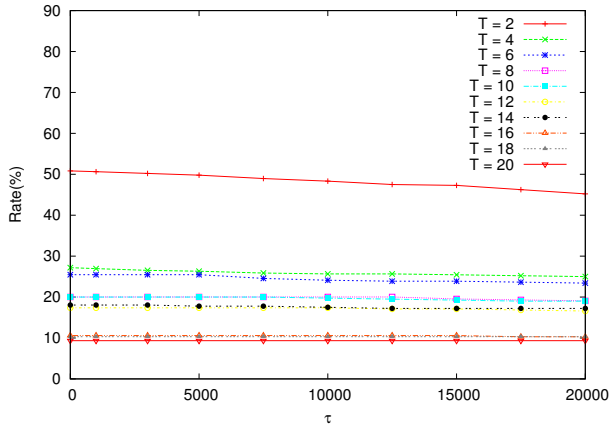


(b) IPR

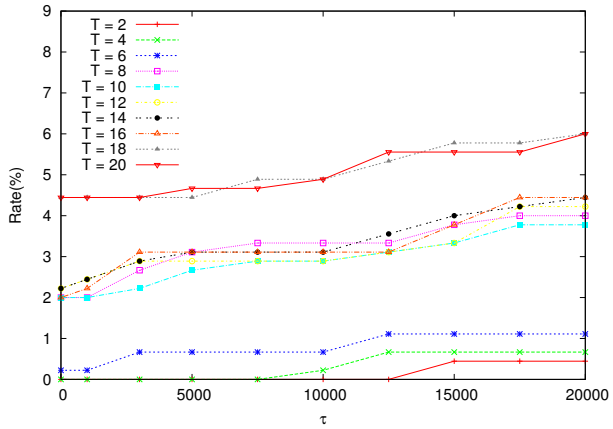
Figure 4: DTW-D-S_o, Varying T.

We observe from Figure 4 that an increase in the number of training patterns decreases the FAR by 30% (from 45%, for $T = 0$, to 15% for T = 20), and increases the number of IPR by only 10% (from 0%, for T = 0, to less then 10% for T = 20). Also, we observe that the results for the several considered τ_S are close to each other. Hence, within the considered range, the variation of the threshold does not significantly influence the results. This observation can also be drawn from the other view we have on the same data (Figure 5)—the curves for different threshold values are almost parallel to the x-axis.

Due to space limitation, we do not report here the same detailed results for the other methods: DTW-D-S_a, DTW-S-S_a, and DTW-



(a) FAR



(b) IPR

Figure 5: DTW-D-S_o, Varying τ_D.

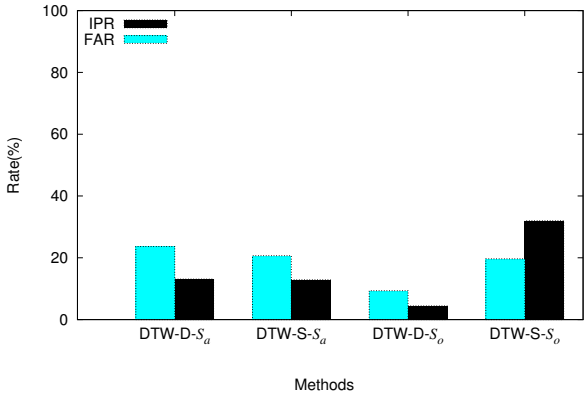


Figure 6: Selection of basic methods with best performances.

Method	τ	T	IPR	FAR
DTW-D-S_a	0	6	13.1111	23.6666
DTW-S-S_a	58	20	12.8888	20.6666
DTW-D-S_o	0	20	4.4444	9.3333
DTW-S-S_o	14	20	32.0000	19.6666

Table 3: Parameters for methods comparison.

S-S_o, for which we observed behaviour similar to the one shown for DTW-D-S_o. However, we report some results that summarize the overall performances of all the basic methods. We observe that any variation of the parameters T and τ might improve one of the two metrics (FAR and IPR), while decreasing the other one. Since there is no variation that at the same can improve both metrics, it is up to the user of the system to prefer to have a smaller FAR (at a cost of an higher IPR), or to have a smaller IPR (at a cost of an higher FAR). To summarize the result we used to following method. For each combination of the considered T and τ, we measured the corresponding FAR and IPR. Then, we computed the average between this two values. For each method, we looked for the parameters setting (T and τ) giving the smallest average of the two metrics. In Figure 6 we report, for each method, the IPR and FAR obtained in this way. Also, for each method, we report on Table 3 the values of T and τ for which we obtained the lowest average between FAR and IPR.

We conclude this section by observing that the performances obtained for the basic methods are comparable to the ones of other transparent authentication methods (Section 2). In fact, we obtained for a single (not combined) method, DTW-D-S_o: IPR $\sim$ 4.4% and FAR $\sim$ 9.3%. As an example of performances of concurrent transparent authentication system, we remind that gait recognition (i.e. walking pattern) enjoys Equal Error Rate (EER) close to 7% [16]. That is, the performances of walking patterns recognition are similar to the ones of our system. However, while being transparent, walking pattern recognition takes a long time before the system can detect that the person using the phone is not the correct one. Keystroke dynamic, while obtaining an EER close to zero when performed on computers, it gave for mobile devices an EER = 12.8% [10, 8]. We argue that not only this result is high (making this method more insecure), but also the system requires long time before it collects enough data to take a decision; again, this giving the intruder enough time to access sensitive data. Finally, acoustic (internal) ear recognition has the best EER between the cited methods. It enjoys an EER of some 5.5% [6]. In this case, the result is slightly better than the one obtained by our method. However, we remind that this method actually requires devices that are not available on current smartphones. Also, we underline that the result of EER $\sim$ 5.5% were obtained in a specific experimental setting, not influenced by external factors like hair, or users wearing things like hats or veils. Hence, we can claim that the performances of our basic solutions are comparable (even better in most of the cases) to the ones of other authentication methods. Furthermore, we remind that our solution can solve a very specific problem that the other solution can not solve: transparently authenticate the user of a smartphone when she answers (or places) a phone call, using devices already available on current smartphones, and without the system being influenced by external factors like ears being covered.

5.2 Boolean Combinations

In this section, we report on the experiments we run in order to evaluate the boolean combination approach discussed in Section 4.3.

Figure 7(a) (in Appendix) summarizes the results of all the possible AND combinations of the basic methods. As expected, in all the possible combinations of two basic methods (say Method A and Method B), resulting IPR and FAR satisfy Equation 3 and Equation 4, respectively. For example, let us see the case of the AND

combination of DTW-D-S_o (Method A) and DTW-D-S_a (Method B). IPR$_A$ is some 4.4; IPR$_B$ is some 13.3; the resulting IPR of the AND combination is some 1.5 (that is, smaller than the smallest between IPR$_A$ and IPR$_B$). On the other side: FAR$_A$ is some 9.3; FAR$_B$ is some 23.6; the resulting FAR of the AND combination is some 24.6 (that is, bigger than the biggest between FAR$_A$ and FAR$_B$). For the OR combination, results are shown in Figure 7(b) (in Appendix), we observed, as expected, the opposite effect on the combined results. That is, resulting IPR and FAR behave accordingly to Equation 5 and Equation 6, respectively.

Finally, we considered the combination where the user is accepted when at least n out of the four methods accept her (n = 1, 2, 3, and 4). The results are shown in Figure 7(c) (in Appendix). From this figure we observe that, as expected, when we increase n, fewer users get accepted—hence decreasing IPR, while increasing FAR. For example, IPR goes from some 41.1 to some 0.6 moving from $n = 1$ to $n = 4$. During the same variation of n, FAR goes from some 1 to some 40.9.

5.3 Normalized Combinations

In this section, we report the results we obtained for the evaluation of the combination of basic methods proposed in Section 4.4, where we consider the normalized version of DTW-D and DTW-S (cfr. Section 4.4.1 and Section 4.4.1, respectively). We run experiments considering both of the following: (i) combining methods two by two; (ii) combining all the four methods together. In case (i), we considered $\alpha = \beta = 1$ in Equation 9 (i.e. giving the two methods equal importance). In case (ii), we considered $\alpha = \beta = \gamma = \delta = 1$ in Equation 10. That is, each building block method influences in the same way the overall decision whether the user's pattern should be accepted. For the threshold $\hat{\tau}$ we used these values: -0.5, 0.0, 0.5, and 1.0.

The results obtained combining the methods two by two are shown in figures 8(a), 8(b), 8(c), 8(d), 8(e), and 8(f) (in Appendix), for all the possible combinations of the basic methods. The results for the combination of all the four methods at the same time are shown in Figure 8(g) (in Appendix).

As we can note from Figure 8, increasing $\hat{\tau}$ leads to an increased FAR and a decreased IPR—this observation is consistent for all the considered combinations. This is because increasing $\hat{\tau}$ means requiring the combined basic methods to output a normalized value that is closer to the one of the training set. After the influence of $\hat{\tau}$, the interesting point is that the proposed normalization of DTW-D and DTW-S, and the proposed combination of basic methods using these algorithms, is able to significantly improve the results. In fact, using this combination, we were able to improve both FAR and IPR, i.e. from IPR $\sim$ 4.4%, FAR $\sim$ 9.3% (obtained for DTW-D-S_o with parameters in Table 3) to IPR $\sim$ 2.5%, FAR $\sim$ 8% (for combination of DTW-S-S_a and DTW-D-S_o—cfr. Figure 8(f)—, with $\hat{\tau} = 0$, using normalized version of DTW-D and DTW-D, and again considering for them the parameters in Table 3). For the described combination and parameter setting we observed EER $\sim$ 7%. We remind that varying T or τ in the basic methods, or combining methods with boolean operator, is not possible to reduce at the same time both FAR and IPR. Instead, with the proposed algorithms normalization and combination, we are able to reduce the IPR of some 50% from the best result we observed considering the single methods, and at the same time also significantly reduce FAR.

6. CONCLUSION

In this paper we propose a new biometric measure for users of smartphones. That is, we focus on the movement that a user performs when answering (or placing) a phone call, and we investigate whether this movement can be used as a biometric authentication measure. We propose four basic methods (leveraging different sensors available on the phone and several similarity algorithms). In this way, we manage to obtained for a single method (i.e. DTW-D-S_o): IPR (Impostor Pass Rate) of some 4.5% , and FAR (False Alarm Rate) of some 9.5%. Also, we propose a novel way of combining the basic methods together. The results show that the proposed combination can improve, at the same time, both FAR and IPR. In fact, for a specific combination, we observed IPR $\sim$ 2.5%, and FAR $\sim$ 8%, thus reducing again IPR by $\sim$ 2% and FAR by $\sim$ 1.5%. The proposed biometric measure is not only effective, as proven by the results; it also enjoys a unique feature. That is, it can be transparently used to authenticate a user that is answering (or placing) a phone call, without this being affected by external factors (like light exposure or users wearing hats or veils).

As future work, we mainly aim at more thorough experiments (e.g. more users, different devices, different similarity algorithms). We would also like to investigate the effectiveness in using this mechanism in constrained environments (e.g. a crowded bus), in case of shared devices (more authorized users on the same device), and combining our mechanism with other authentication methods (e.g. acoustic ear recognition). Finally, we aim at possible optimization like the one on the similarity algorithm [23].

7. ACKNOWLEDGMENTS

This work is partly funded by the European Network of Excellence NESSoS contract no. FP7-256980.

8. REFERENCES

[1] Android dev phone 1. `http://www.htc.com/www/product/dream/overview.html`, 2010.

[2] FANTASy project (url anonymized for submission). `http://ANONYMIZED`, 2010.

[3] Public class sensorevent. `http://developer.android.com/reference/android/hardware/SensorEvent.html`, 2010.

[4] A. F. Abatea, M. Nappi, D. Riccio, and G. Sabatino. 2D and 3D face recognition: A survey. *Pattern Recognition Letters*, 28(14):1885 – 1906, 2007.

[5] T. Abeel, Y. Van de Peer, and Y. Saeys. Java-ml: A machine learning library. *J. Mach. Learn. Res.*, 10:931–934, 2009.

[6] A. H. M. Akkermans, T. A. M. Kevenaar, and D. W. E. ant Schobben. Acoustic ear recognition for person identification. In *AUTOID '05*, pages 219–223, 2005.

[7] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. In *USENIX WOOT '10*, 2010.

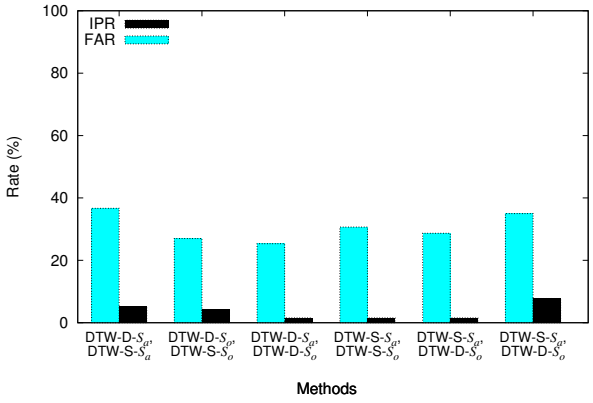
[8] F. Bergadano, D. Gunetti, and C. Picardi. User authentication through keystroke dynamics. *ACM TISSEC*, 5(4):367–397, 2002.

[9] N. B. Boodoo and R. K. Subramanian. Robust multi biometric recognition using face and ear images. *International Journal of Computer Science and Information Security*, 6(2), 2009.

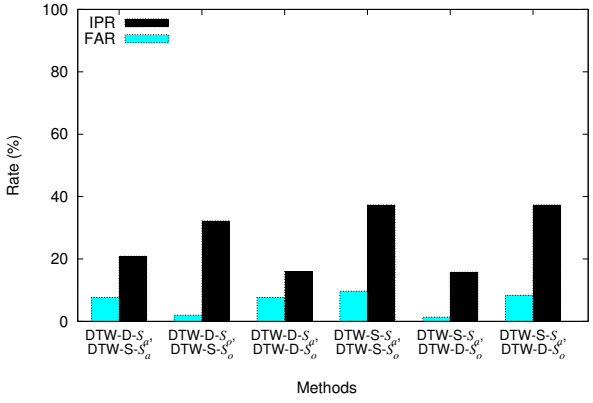
[10] D. Gunetti and C. Picardi. Keystroke analysis of free text. *ACM TISSEC*, 8(3):312–347, 2005.

[11] M. Jani, L. Mikko, V. Elena, M. Satumarja, and A. Heikki. Identifying users of portable devices from gait pattern with accelerometers. In *ICASSP '05*, pages 973 – 976, 2005.

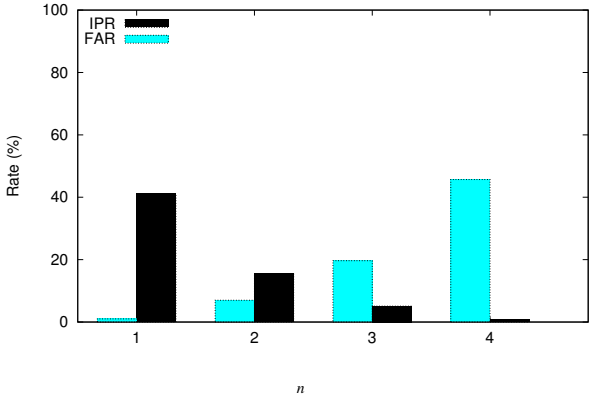
[12] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D.



(a) AND



(b) OR



(c) n out of 4

Figure 7: Boolean Combinations.

Rubin. The design and analysis of graphical passwords. In *USENIX SSYM'99*, pages 1–1, 1999.

[13] R. Joyce and G. Gupta. Identity authentication based on keystroke latencies. *Commun. ACM*, 33(2):168–176, 1990.

[14] J. Leggett and G. Williams. Verifying identity via keystroke characteristics. *Int. J. Man-Mach. Stud.*, 28(1):67–76, 1988.

[15] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. User evaluation of lightweight user authentication with a single tri-axis accelerometer. In *MobileHCI '09*, pages 1–10, 2009.

[16] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S.-M. Makela, and H. Ailisto. Identifying users of portable devices from gait pattern with accelerometers. 5779(7), 2005.

[17] J. A. Markowitz. Voice biometrics. *Commun. ACM*, 43(9):66–73, 2000.

[18] L. Nanni and A. Lumini. A multi-matcher for ear authentication. *Pattern Recognition Letters*, 28(16):2219–2226, 2007.

[19] P. C. v. Oorschot and J. Thorpe. On predictive models and user-drawn graphical passwords. *ACM TISSEC*, 10(4):1–33, 2008.

[20] P. J. Phillips, A. Martin, C. l. Wilson, and M. Przybocki. An introduction to evaluating biometric systems. *Computer*, 33(2):56–63, 2000.

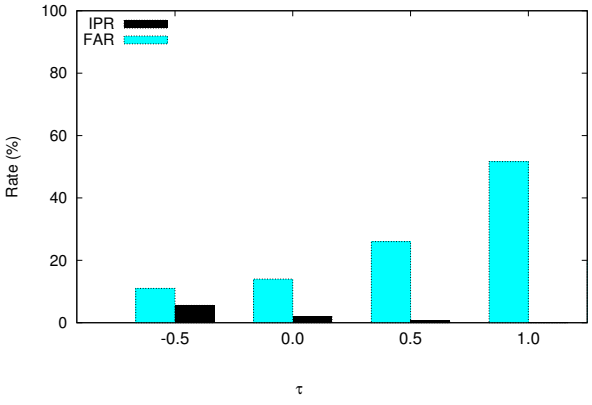
[21] A. Ross and A. Jain. Multimodal biometrics: an overview. In *EUSPICO '05*, pages 1221–1224, 2005.

[22] P. Senin. *Dynamic Time Warping Algorithm Review*. Technical Report - University of Hawaii at Manoa, 2008.

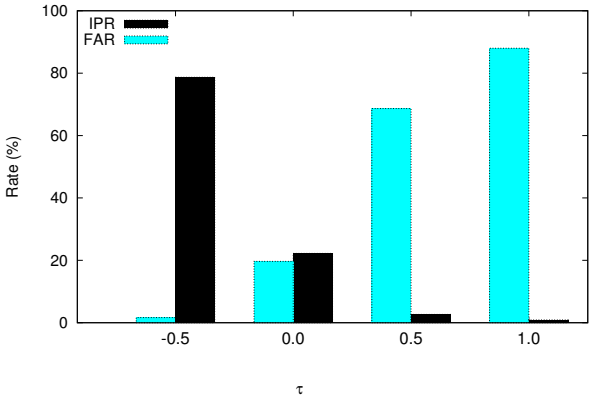
[23] Y. Shou, N. Mamoulis, and D. W. Cheung. Fast and exact warping of time series using adaptive segmental approximations. *Mach. Learn.*, 58(2-3):231–267, 2005.

Appendix

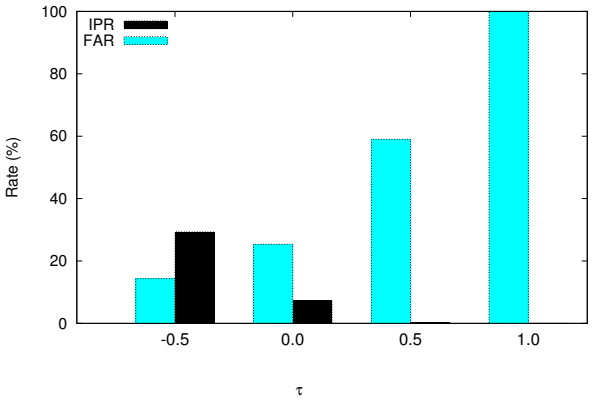
This section gives further details on the proposal evaluation. In particular, we report FAR and IPR for the several proposed ways of combining methods. First we report the results for boolean combinations (Section 4.3). Figure 7(a) shows the results for any possible combination of two basic methods with AND. Figure 7(b) shows the results for any possible combination of two basic methods with OR. Finally, Figure 7(c) shows the results for the combination where a user is accepted when n out of the four methods accept it (n = 1, 2, 3, and 4). Then, we report the results for the combination that is based on the normalized versions of DTW-D and DTW-S (Section 4.4). Figures 8(a), 8(b), 8(c), 8(d), 8(e), and 8(f) report the results for all the possible combinations of two basic methods. Figure 8(g) reports the results for the combination of all the four methods together.



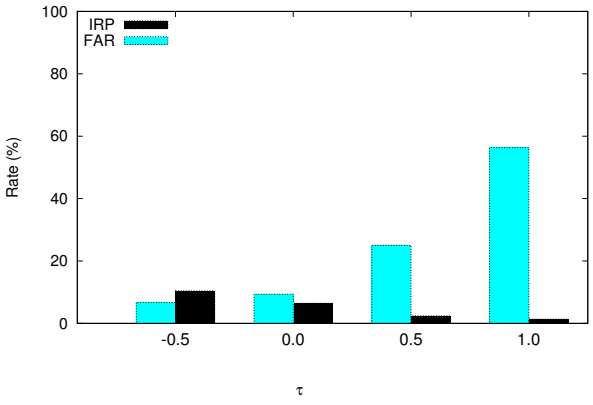
(a) DTW-D-S_a, DTW-D-S_o



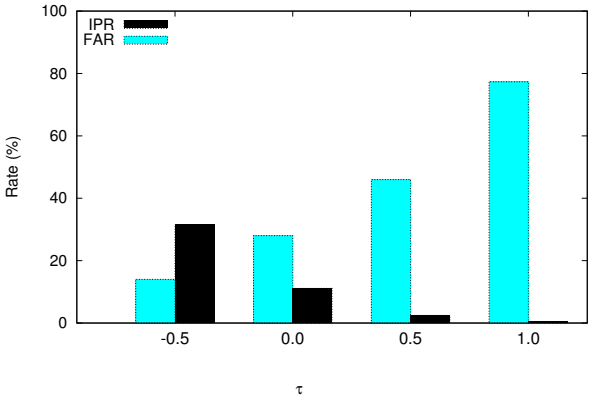
(b) DTW-S-S_a, DTW-S-S_o



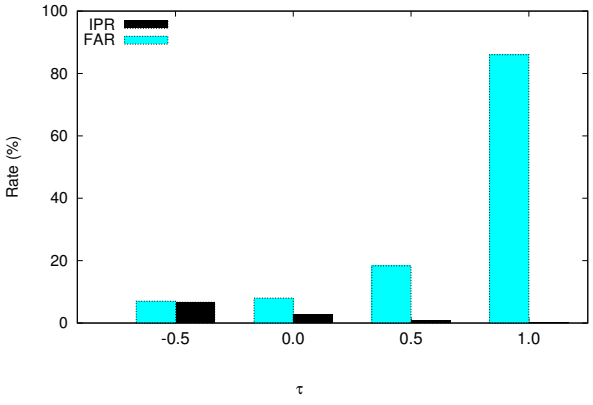
(c) DTW-D-S_a, DTW-S-S_a



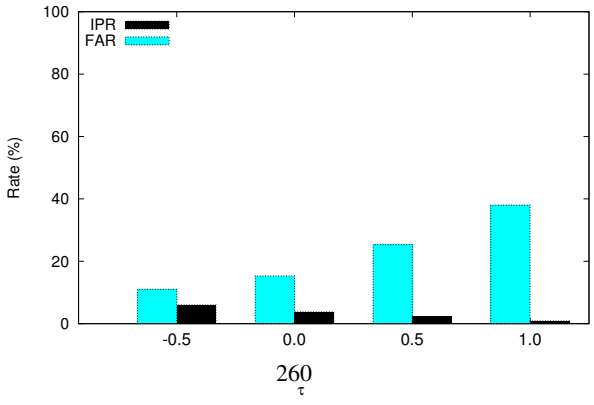
(d) DTW-D-S_o, DTW-D-S_a



(e) DTW-D-S_a, DTW-S-S_o



(f) DTW-D-S_o, DTW-S-S_a



(g) All four methods

Figure 8: Non-Boolean Combinations.