Design, implementation, and assessment of a usable multi-biometric continuous authentication system

Design, Implementation, and Assessment of a Usable Multi-biometric Continuous Authentication System

Abstract: Authentication mechanisms typically verify the user identity only at login, or with tedious explicit authentication requests that improve security at the expense of usability. However, especially for critical systems, workstations have to be tightly and continuously secured in order to prevent unauthorized interventions. Recent researches envisage multi-biometric systems for continuous authentication, where biometric traits are acquired transparently to the user and authentication is provided without requiring explicit actions. In this work we propose a multi-biometric authentication system that continuously and transparently verifies the user identity through face, fingerprint and keystroke recognition. This paper presents the design, prototype implementation and assessment of our system. We evaluate the system usability and its trade-off with security in an experiment involving 60 users. Our findings show that security enhancements are provided and users i) perform the actions without additional effort, ii) largely accept the authentication system, which only requires minimal training.

Keywords: continuous authentication; biometrics; security, usability; identity verification; multi-biometric; transparent; face; fingerprint; keystroke; prototype; assessment; design; implementation;

1 Introduction

In many critical systems and applications, it is fundamental that only authorized users are allowed to interact with a machine [1]. User authentication, which is in the process of verifying the identity claimed by or for a human entity [2], is the security service designed for this purpose. Traditional authentication approaches are knowledge-based and take advantage of passwords or PINs [2]; alternative solutions have been proposed, either possession-based (e.g., security token) or methods that make use of biometric traits [3]. In all cases, when the identity verification is performed as a single occurrence during the login phase, and no identity checks are performed during sessions, unauthorized people may take physical control of the computer or device. For instance, in an office environment if a worker leaves the device unattended without logging out e.g., for a short break or to reach the printer, *insiders* may intervene accessing, modifying or deleting sensitive information, or even introducing vulnerabilities [4]. This is just an example, but areas where continuous authentication is desirable are multiple. An important category is safety critical command and control rooms, where operators are responsible for public safety, transportation, air traffic management, or in all the areas where the computer system has to be accessible only to certified and authorized users, otherwise intrusions may be the cause of severe or even catastrophic consequences [51], [52]. Let us think about a crisis management system where human operators working in a control room are in charge of analyzing and interpreting situations that describe the current status of an emergence [53]. Using the information available, the operator from his computer system has to command intervention teams on field and to dispatch instructions to civilians in the target area. Workstations have to be protected from intruders and insiders that may want to acquire privacy-sensitive data, disrupt the crisis management operations, disseminate false information, or simply commit errors, which will be ascribed to the operator in charge of the computer system [1]. In order to address this issue, it would be desirable to verify user identity continuously, for the whole duration of a session. However, repeatedly asking for passwords and secrets over time requires user active participation: it would disturb operations and reduce system usability. It is well-known, in fact, that usability and security are often seen as competing goals [5]. Improving usability is sometimes considered as improving vulnerabilities, to the extent that it has also been perceived as helping the attacker [5]. As an example, reducing the frequency of password changes improves usability but it also implies that a compromised password may be (mis)used longer.

On top of procedural solutions as training employees to logout every time they leave the workstation, solutions based on biometric authentication have been proposed in literature [6]-[31]; some have the potential to continuously verify the identity of the user and without reducing usability. Thanks to this approach, known as biometric continuous authentication, user identity verification is no longer a single occurrence, but a continuous process. Furthermore, it is commonly agreed that the use of multiple biometric traits properly combined can improve the performance of the identity verification process [32]. In addition, appropriate sensors, together with specific design choices, permit to acquire biometric traits transparently i.e., without the active involvement of the user. Consequently, transparent multi-modal biometric continuous authentication solutions are identified and compared in Table VII.

However, there is a real lack of studies with emphasis and focus on end-users. In fact, the evaluation of a proposed system or framework is often conducted as a simulation, rarely with human involvement and almost never through a proper usability assessment. In our opinion, instead, when introducing a new approach, it is fundamental to address not only technical issues, but to consider also the effects of this innovation on humans, their thoughts, and perceptions.

In this paper, we target a multimodal biometric transparent continuous authentication system that is both usable and incurs in little system overhead. We design a solution which integrates face, fingerprint and keystroke recognitions, and removes the necessity of conscious human-computer interactions. Data is transparently acquired by the workstation and transmitted to an authentication server, which performs the identity verification. In case of successful verification, the authentication server permits the establishment of a user session. Then it calculates and updates a trust level that decreases as time passes; the session expires when such level becomes lower than a predefined threshold [1].

We evaluate the system through a usability testing campaign [33], [34], involving a population of 60 users selected amongst academia. Experiments were devised where participants were asked to perform different office processing tasks. Users' feedbacks were collected regarding system usability and their satisfaction. During the tests we also measured the acceptance and rejection rates of the face, fingerprint and keystroke subsystems and of the integrated system, to evaluate the effectiveness of the authentication solution. Furthermore, we considered system efficiency measuring the time interval during which a legitimate user remains authenticated, and the window of time needed by the system to reject an impostor. Finally, the trade-off between usability and security is quantified. Results show that even taking into account usability as a primary goal, security of users' workstations is increased.

We publish a repository of the log files recorded by the system, which, as far as we know, is the first public dataset on logs of a continuous authentication service available to researchers. The supplementary data for this article consists also in detailed questionnaire results regarding user satisfaction, and the documents used for tasks execution [49]. All data in the dataset is anonymized.

The remaining of the paper is organized as follows. Section 2 introduces the background. Section 3 illustrates the proposed authentication solution. Section 4 describes the experiments plan and execution. Section 5 reports the analysis of the collected data. A detailed analysis of the relevant works from state of the art, with the related positioning of our contribution is in Section 6, whereas concluding remarks are in Section 7.

2. BACKGROUND

2.1. Foundations on Biometric Authentication Systems

Authentication can be defined as the process that provides assurance in the claimed identity of an entity [35]. Traditionally, this process is composed of two consecutive steps: *registration* and *verification* [3], [36]. Registration consists in storing an authentication factor associated with the user identity, and which will be verified in the subsequent step (on verification). In literature, many types of authentication factors have been proposed, and they are typically divided in three categories [3], [36]:

- Knowledge factors: something the user knows e.g., passwords, PINs.
- Possession factors: something the user has e.g., passports, private keys.
- Inherence factors: something the user is or does physiological or behavioral biometrics.

Thus, a *biometric authentication system* is a system in which the identity verification of individuals leverages on inherence factors: their biometric characteristics (also called traits). If the biometric system exploits only one type of characteristics, it is referred as *unimodal* [3], [36]. Otherwise, a *multimodal* biometric authentication system (also known as multi-biometric), which uses multiple sources of biometric information, is obtained integrating two or more unimodal *subsystems*, fusing them at one of the different levels of the verification process [3], [36].

The user presents one or multiple of his/her biometric traits during the registration step, (also called *enrollment*, [36], [3]). The system generates a template, i.e., a digital representation of the traits, which is stored in a database.

Then, during the verification step, the features extracted from the new traits are compared with the stored templates belonging to the user. The process generates a comparison score, which has direct impact on the decision about user's identity: accept as legitimate and authenticate, or reject. However, the system's decision sometimes is wrong, and the error can belong to two categories: false accept or false reject. Thus, the two main types of errors metrics are [36]: *False Acceptance Rate (FAR)*, that is the proportion of verification attempts with wrongful claims of identity that are incorrectly confirmed, and *False Rejection Rate (FRR)*, that is the proportion of verification attempts with truthful claims of identity that are incorrectly denied. FAR and FRR are generally the basic measures of the accuracy of a biometric system. The *confusion matrix* is then completed with the *True Rejection Rate (TRR)*, and the *True Acceptance Rate (TAR)*, which intuitive definitions are, respectively: the ratio of impostor authentication attempts that were correctly rejected, and the ratio of legitimate user authentication attempts that were correctly accepted [36], [3].

We refer to *performance* of a biometric system as the achievable recognition accuracy and speed, the resources required to achieve them, and environmental factors that affect them [3]. By "increasing the security of the system" we mean reducing the likelihood that it is physically operated by not legitimate users [7].

We define the trust level *trust(t)* as the likelihood that the user is legitimate at time instant *t*, considering his/her interaction with the system [7]. This score is a value that lies in the interval [0, 1]; it is computed considering the time interval from the last acquisition of biometric traits, and the combination of the individual decisions of the unimodal subsystems. We also define a lower bound, $trust_{min}$, corresponding to the minimum threshold of *trust(t)* requested by the system to authenticate the user [7].

For example, considering a multimodal system composed of three subsystems S_1, S_2, S_3, we define $m(S_1)$, $m(S_2)$, $m(S_3)$ as the trust in the respective subsystems. The $m(S_1)$, $m(S_2)$, $m(S_3)$ are static values in the interval [0, 1] assigned by the system administrator based on the performance of each individual subsystem [7].

2.2. Foundations on Usability Analysis

The formal definition of usability by the ISO (International Organization for Standardization) is *the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use* [42].

There are important attributes characterizing usability. These are:

Effectiveness, which answers the question: "*can users complete tasks with the system*?" In other words, a system is effective if it behaves as expected and can be used easily. This is usually measured quantitatively with error rate, which in our context means FAR, FRR, as well as their complementary values: TRR and TAR [33], [34].

Efficiency, which answers the question: *"how much effort is required from the users to do this?"* [33], [34]. In other words, a system is efficient if users can accomplish goals quickly, accurately, completely and with limited resources consumption. It is usually a measure of time.

Satisfaction, which answers the question: *"what do users think about the easiness of the products' use?"* [33], [34]. It refers to the user's perceptions, feelings, and opinions about the product, their comfort and feedback about the system usage. Usually satisfaction is captured through interviews or questionnaires.

3. OUR APPROACH TO CONTINUOUS AUTHENTICATION

Our Biometric Continuous Authentication System (from now BCAS) architecture is composed of i) a desktop workstation including sensors for the biometric data acquisition, ii) an authentication server, and iii) a database of templates. The different biometric data are acquired continuously by the workstation, and the identity of the user is verified. The foundations of the devised system, that has been initially presented in [50] in a preliminary version, has been re-applied or considered as inspiration in recent works, especially [51] and [52].

The choice of the biometric traits is based on the characteristics of a generic workstation: the user interface typically consists at least of a screen, a keyboard, and a mouse. In our opinion, the best choice to achieve system acceptability is to avoid the introduction of any additional device with which the user actively interacts. In this way, there is no loss of time spent in learning how to use the device and, consequently, no loss of proficiency and efficiency in the working activity. The system only requires the usage of a particular kind of mouse that incorporates a fingerprint scanner where users normally place their thumb [38]. This measure may be unpleasant but is necessary; otherwise the biometrics acquisition would not be possible in a transparent way. Then, the other sensors are a keyboard, and a camera which nowadays is very common to be integrated in a laptop or on top of the screen of a workstation, e.g., for usage in video conferences. Noteworthy, related works as also reviewed in the state of the art e.g., [27], [29], also apply a similar approach to the identification of the platform, based on usage requirements.

The BCAS is therefore based on three unimodal biometric subsystems, for fingerprint recognition, face recognition, and keystroke recognition. Each subsystem is composed of hw/sw elements necessary for the acquisition of the trait and for the verification process, including sensors and recognition algorithms, such that each one is able to decide independently if the user is genuine or not. The fusion is performed at decision level.

The three biometric traits above have different levels of performance and measurability [3], [37] and complement each other. High measurability of facial images will help covering temporal gaps that could exist between two fingerprint acquisitions, when the operator is not using the mouse.

Keystroke supports the other two traits, despite its lower performance, and it can result useful especially when fingerprint acquisitions are missing.

3.1. The Protocol

The proposed continuous authentication protocol is shown in the sequence diagram in Figure 1. It is divided in two phases: the initial phase and the maintenance phase. Before the initial phase, we assume that the enrollment already took place as a preliminary step.

Initial phase. It is composed of the following steps:

1. The user logs in and a biometric verification is executed by all the three subsystems in a short time interval. At this time instant, indicated with $t_{0,}$ the trust is set to $trust(t_0) = 1$.
2. Biometric data is acquired by the BCAS workstation and transmitted to the authentication server.
3. The authentication server matches the user’s templates with the traits stored in the database and verifies his/her identity.
4. In case of a successful verification, the BCAS application establishes a session and allows access to restricted functions.

Maintenance phase. The biometric continuous authentication protocol works as follows:

5. The user’s biometric data are periodically acquired by the biometric subsystems operating on the workstation and are transmitted to the authentication server.
6. The authentication server waits for fresh biometric data, from any of the three subsystems. When new biometric data is available, it verifies the identity claimed by the user and, depending on the comparison results of each subsystem, it computes and updates $trust(t)$.

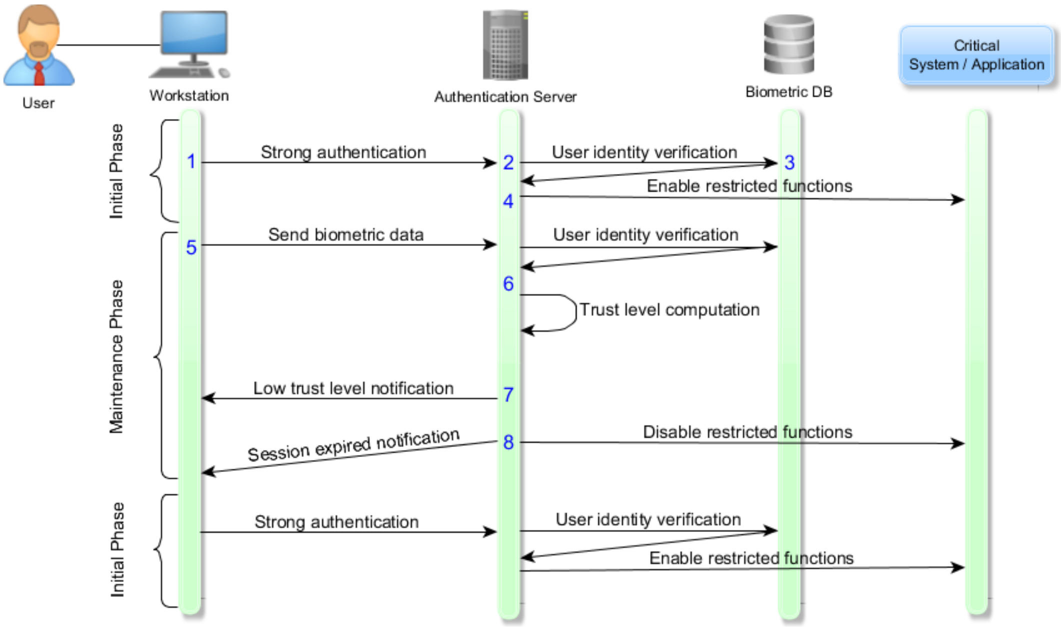

User
Workstation
Authentication Server
Biometric DB
Critical System / Application
Initial Phase
Maintenance Phase
Initial Phase
1
Strong authentication
2
User identity verification
3
Enable restricted functions
4
5
Send biometric data
User identity verification
6
Trust level computation
Low trust level notification
7
Session expired notification
8
Disable restricted functions
Strong authentication
User identity verification
Enable restricted functions

Figure 1 Sequence diagram of the protocol.

7. The session expires when *trust*(t_i) becomes lower than $trust_{min}$.

8. When the trust level is below the threshold, *trust*(t) < $trust_{\mathrm{min}}$, the session expires and the restricted functions are disabled. The user receives a notification of this event, and, if necessary, restarts again from the initial phase.

3.2. Internals: the Trust Level Computation

We describe the algorithm executed by the authentication server to compute the trust level. Our system integrates three unimodal biometric subsystems {S_1=fingerprint recognition, S_2=face recognition, S_3=keystroke recognition} such that each one is able to decide independently if the user is genuine or not.

The algorithm which computes the trust level is executed periodically on the authentication server as follows. During the maintenance phase, the authentication server verifies the user identity thanks to all biometric data provided in a specific time interval. In our implementation this interval is 20s, during which multiple attempts of fingerprint, and face acquisition are sequentially performed, while a keystroke listener runs in parallel for almost the whole interval. In general, let us consider the time interval [t_{i-1}; t_i], where t_i is the current time instant and t_{i-1} is the time instant in which the previous iteration of the protocol has been concluded. Regarding the status of the system at time instant t_i, we have three following alternatives: *three recognitions*, *two recognitions*, *one or no recognitions*.

Three recognitions: for any time interval in which all the three biometric subsystems led to successful verifications, the authentication server sets $trust(t_i) = 1$.

Two recognitions: two-out-of-three biometric subsystems led to successful verification. The trust level is updated to a static value, which can be set a priori based on the estimated accuracy of the subsystems that decided the user legitimacy. The trust level is computed following (1):

$$\mathrm{trust}(\mathrm{t_i}) = \mathrm{m}(\mathrm{S_{k1}}) + \left(\mathrm{r} \cdot \mathrm{m}(\mathrm{S_{k2}})\right) \qquad (1)$$

where:

- S_{k1} and S_{k2} are the subsystems which correctly verified the identity of the user, and S_{k2} is the one with the lower performance;
- r is a parameter to weight $m(S_{k2})$ in order to have $trust(t_0)$ between 0 and 1.

In our implementation, setting $r = 0.1$, $m(S_{k1})= 0.9$, $m(S_{k2})= 0.8$, $m(S_{k3})= 0.7$, we have the combinations of Table I. The selection of these values has been conducted comparing the biometric traits, analyzing how their performance is evaluated in literature [3], [37], and it is related to the number of false accepts produced by each subsystem. We found that these values can represent properly the accuracy of each subsystem, but other different values can be easily adopted, if necessary, following a similar approach.

One or no recognitions: if instead, at most one biometric verification is successful at time instant t_i, $trust(t_i)$ decreases nonlinearly through time. Given $trust(t_{i-1})$, that is the trust level computed at the previous iteration of the algorithm, we have that $trust(t_i)$ will be smaller than $trust(t_{i-1})$. Its value is given by (2) from [7]:

$$trust(t_i) = \frac{\left(-\arctan\left((\Delta \mathrm{t_i} - 5)\cdot k\right) + \frac{\pi}{2}\right)\cdot trust(\mathrm{t_{i-1}})}{-\arctan(-5\cdot k) + \frac{\pi}{2}}. \qquad (2)$$

$\Delta \mathrm{t_i} = \mathrm{t_i}$-$\mathrm{t_{i-1}}$, and k are introduced to tune the decreasing function: k affects the inclination towards the falling inflection point. With regard to [7], in (2) we set the value of s, the parameter which allows delaying the decay.

Table I Trust computation with two out of three successful recognitions

Pair of biometric subsystems	$\boldsymbol{trust(t_i)}$
Fingerprint, Face	$m(S_{k1}) + (r \cdot m(S_{k2})) = 0.98$
Fingerprint, Keystroke	$m(S_{k1}) + (r \cdot m(S_{k2})) = 0.97$
Face, Keystroke	$m(S_{k2}) + (r \cdot m(S_{k2})) = 0.87$

Through an experimental evaluation, we found that 5 is the most appropriate value to manage the delay in our setup.

The selection of *k*, in particular, affects the speed of the decrease of the trust level. We adopt three different values of *k* according to which and how many verifications are successful. A *fast* decrease is set when no verifications are successful or no biometric data is transmitted. The decrease is said *average* if only one verification is successful, for any biometric subsystem. Finally, we have a *slow* decrease if face is correctly verified and the usage of keyboard is detected, although data is not sufficient to perform keystroke recognition or keystroke recognition fails.

The latter is the situation in which: i) the user is actually busy in the usage of the keyboard, ii) the user is not able to send any fingerprint data, iii) the amount of keys pressed is too low or too sparse to permit keystroke recognition. Thus, a small penalization is assigned to the trust in the user, smoothly decreasing the trust level. The triples of *k* values selected for the experiments are discussed in the following of the paper.

It is important to specify that in our prototype the trust computation is only influenced by the number of successful verifications and not by unsuccessful verifications. In fact, as an implementation choice we do not distinguish between a missing biometric characteristic, and a trait which is verified as not legitimate. This is to favor usability, considering also the high number of false rejects that may happen under different operating conditions, for instance when the fingers are sweat, or the room is darker than usual. However, an alternative solution may address this difference: if a subsystem considers the trait belonging to an impostor, this may cause a faster trust level decreasing w.r.t. a missing acquisition of the same trait.

3.3. The Prototype

The hardware is entirely COTS (Commercial of-the Shelf). For fingerprint acquisition, our choice is the SecuGen OptiMouse Plus mouse [38], which incorporates an optical fingerprint scanner at the place where a user normally places the thumb. Such fingerprint scanner does not require active participation by the user, and therefore does not require that the user periodically performs biometric-related tasks that are not part of their normal activities. For acquisition of the images for face recognition, we use the built-in camera of a laptop that can continuously capture images. Finally, we collected keyboard data using the standard PS/2 keyboard integrated in the laptop.

We relied on OTS software as much as possible. For fingerprint, the SecuGen's FDx Software Developer Kit [38] provides low-level APIs for device initialization, fingerprint capture and comparison functions. For face recognition, we customized the software library available in [39]: this is able to (i) analyze the frames captured via a camera, (ii) locate a face in the frames, and then (iii) verify user's identity. This customization is necessary to i) structure the implementation available in [39] in a client and a server side, where the first is in charge of capturing images and deciding if a face is present, and the second performs verification, and ii) make the acquisition of the biometric data transparent and automatic, removing the graphical interface and interactions of the user with the software.

Keystroke data acquisition relies on the library JnativeHook, which provides keyboard listeners for Java language. In particular, this library allows detecting keys press and release events and captures, in correspondence to those events, the time instant of the events. JnativeHook also permits to detect the keyboard usage (and the keys pressed), both if the user is typing in a specific text area or not: the cursor position is not relevant. This is consistent with our needs as we can capture keystroke data without being invasive for the activity of the user.

We implemented the keystroke recognition algorithm described in [40]. Such algorithm continuously collects the keystroke dynamics (the typed key and related pressing and release time) and applies a penalty/reward function on the dataset to measure the confidence that the user has not changed in the selected time interval. In our implementation of [40], the system listens for keystroke dynamics for a defined time interval (approximately 20s), and then transmits all the values to the authentication server. The selection of the time interval is critical because if the number of values collected is too low, the verification will most likely fail: a short time interval would probably result ineffective for keystroke verification [40]. Conversely, a long time interval would imply to postpone verification, thus risking that the session expires meanwhile. We experimentally evaluated that acquiring the keystroke for 10s of continuous typing is sufficient to allow successful verification.

All the software we developed is implemented in Java. The communication between workstation and authentication server is based on RESTful web services, and developed using the Jersey framework.

3.4. Parameters Configuration

The proposed solution offers a wide set of parameters that can be tuned according to system requirements, in order to manage the trade-off between security and usability. The three subsystems possess their own parameters that can be managed. For example, if we consider the keystroke subsystem in our prototype, possible configurations are the penalty/reward function, and the time interval for the keystroke listening.

More in general, a company's IT administrator responsible of workstation security can act on the parameters which affect the trust level computation:

- The weight *r* and the trust in the individual subsystems $m(S_k)$ in equation (1).
- The decreasing function parameters, *k*, and *s*, in equation (2).
- The time interval $\Delta t_i = t_i - t_{i-1}$ between two consecutive verification attempts.

Finally, the minimum threshold $trust_{min}$ required by the system to maintain the user authenticated can be set. In our prototype, all these parameters are easily set via configuration file.

In Section 4 and Section 5, our solution will be exercised with different parameters values, to show their impact on system behavior.

3.5. Exemplary Run

For clarity, we show in Fig. 2 an exemplary run. As discussed in Section 3.1, the user initially performs a strong authentication, which sets $trust(t_0) = 1$, indicated by square markers in the figure. The BCAS acquires biometric traits at time intervals $\Delta t_i = t_i - t_{i-1}$ of about 20s. In this run, we have three traits contemporarily verified during the intervals ending at seconds 347, 1749, and 2002; this means that the trust level is raised to *1.0* three times. Instead, when exactly two traits are recognized, the $trust(t_i)$ is set to the corresponding value of Table I. In Fig. 2 round marker signals that the two recognized traits are face and fingerprint, and $trust(t_i)$ is *0.98*; instead, with a diamond marker we indicate that face and keystroke are recognized, and $trust(t_i)$ is *0.87*. In the run of Fig. 2, the situation of having fingerprint and keystroke recognized in the same interval has never arisen, thus $trust(t_i)$ has never been set to *0.97*.

When the face trait is recognized and keyboard usage is detected (but it is not possible to complete keystroke recognition), the $trust(t_i)$ starts decaying slowly. This is indicated Fig. 2 by a triangle. The decaying becomes a bit faster (average speed, star marker in Fig. 2) if exactly one

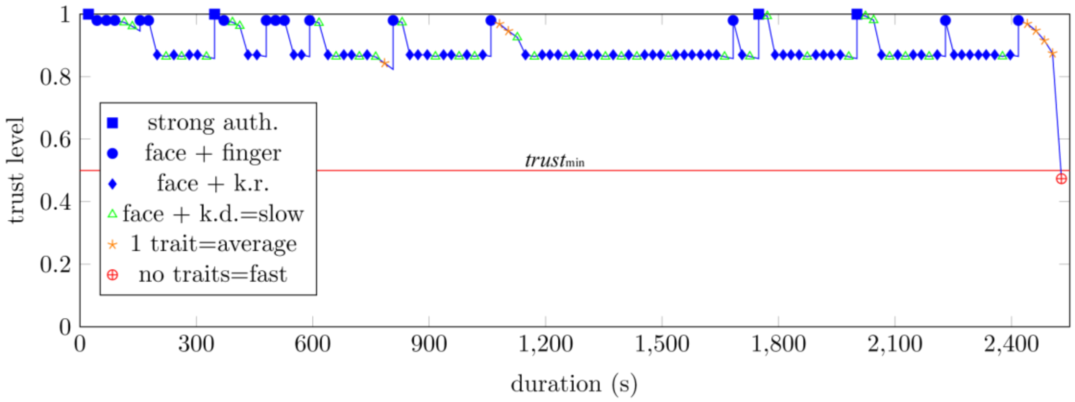


Figure 2 An exemplary run with the user remaining authenticated for about 42 minutes until he leaves.

trait is recognized, and even faster if no traits are recognized (fast speed, target marker). In the run of Fig. 2, we can see that the user remains authenticated for about 42 minutes, then at second 2441 he stops using both mouse and keyboard: only the face is recognized and the *trust*(t_i) decreases with an average speed for four intervals. Finally, after second 2506, the user leaves the workstation and no data is recognized: the trust quickly decreases to *0.47*, below the $trust_{min}$ threshold of *0.5* selected for this run.

4. EXPERIMENTS PLAN AND EXECUTION

4.1. Overview and Goals of the Study

The best way to investigate the usability of our system is conducting a study involving real users, because it provides direct information about how they perceive the system and interact with it. Following the definition of usability by the ISO, we want to study our BCAS in terms of effectiveness, efficiency and satisfaction, taking into account also the trade-off with security. We will compare our results mainly with [27] and [29], which among the works in literature, as it will be discussed in Section 6, are the closest approaches. We will also compare with [3] the acceptability of biometric traits.

Effectiveness

Can users of our system complete their tasks while the continuous authentication is running? How often are they disturbed or even rejected as impostors? Are the impostors rejected if they try to intrude the system?

Effectiveness itself is in a certain way a measure of the trade-off between usability and security. We want to measure the effectiveness of our solution calculating the FAR and the FRR for the individual biometric subsystems (fingerprint, face, and keystroke biometric traits), and for the BCAS. These two metrics, in fact, are indicators of system effectiveness: lower are the error rates, more effective is the continuous authentication.

Efficiency and Resources Utilization

How long the legitimate user remains authenticated during a task execution? How fast an intrusion is detected? Are the system and the user activity slowed down by continuous authentication?

In our tests, the efficiency of user-system interaction is represented by the time that the legitimate user remains authenticated before session termination. Our goal is to assess the efficiency of the system measuring the time interval between the initial strong authentication and the unexpected session termination. We call this measure *Authentication Time* (AT). Similarly, we are interested in the *Time to Impostor Rejection* (TIR), namely the time necessary for the authentication system to reject an impostor that gains possession of a workstation left unattended.

We also want to clarify if the overhead introduced by the continuous authentication system slows down the workstation and consequently increases the effort required to the users. For this purpose, we asked participants to complete four tasks on a workstation provided with our BCAS application running in background. One of the tasks is performed with a placebo application, that resembles the real one but actually does not perform continuous authentication. Consequently, it has an insignificant overhead. We want to measure the *Completion Time* (CT) for each task, and compare the CT difference between the same tasks completed with the real and with the placebo applications. The tasks resemble real-life work using Microsoft Office, and the participants are requested to reproduce documents using Word, Excel and PowerPoint. This choice will also permit a comparison with [27].

In addition, we want to calculate the overhead of the BCAS in terms of percentage of CPU usage.

Satisfaction

What do users think about working at a workstation with continuous authentication running in background?

The satisfaction of the participants has been measured with a Likert scaled post questionnaire [43], [44], [45], designed to gather users opinions and comments about their acceptance to provide the biometrics,

Table II.A) Parameter k configurations

k 1)	fast=8x10^{-3}
	average=8x10^{-4}
	slow=5x10^{-4}
k 2)	fast=1x10^{-2}
	average=1x10^{-3}
	slow=8x10^{-4}

Table II.B) Trust minimum threshold configurations

$\textit{trust}_{\min}$ a)	0.3	highly usable system
$\textit{trust}_{\min}$ b)	0.5	trade-off usability/ security
$\textit{trust}_{\min}$ c)	0.7	highly secure system

and their interaction with the BCAS for both the enrollment and the continuous authentication phases.

Trade-off Security/Usability

How do changes in parameters configuration affect security and usability?

Another goal is to perform the specified measurements with different parameters configuration, such as varying the $\textit{trust}_{\text{min}}$ threshold or the triple of *k* parameter values, which modifies the speed of the trust decaying function, as discussed in Section 3.2. We tested two configurations of *k* (Table II.A), where its value is proportional to the decaying speed, combined with three configurations of $\textit{trust}_{\text{min}}$ (Table II.B). For instance, with the first triple of k values in Table II.A, the trust level decreases slower than with the second triple.

We also want to compute the *Probability of Time to Impostor Rejection (PTIR)*, which is the probability that the TIR is lower than a *Window of vulnerability (W)*.

4.2. Design of the Single Experiment

During the briefing, the observer welcomes the test participant and gives a brief explanation of the purpose of the participation: testing a biometric continuous authentication system. Participants are asked to complete a set of four extremely simple tasks, designed to represent realistic tasks of an office worker. For each participant, the entire session lasts from 1 to 2 hours, depending on the participant’s speed to perform the required tasks. Figure 3 shows the workflow of the experiment.

After a brief introduction, the enrollment phase is performed: exploiting the GUI of the BCAS, and with the help of the observer, the users register 10 facial images, their right thumb fingerprint and their keystrokes. This phase lasts approximately 17 minutes. Acquiring the trait and training the related algorithm requires approximately 1 minute for the face subsystem,

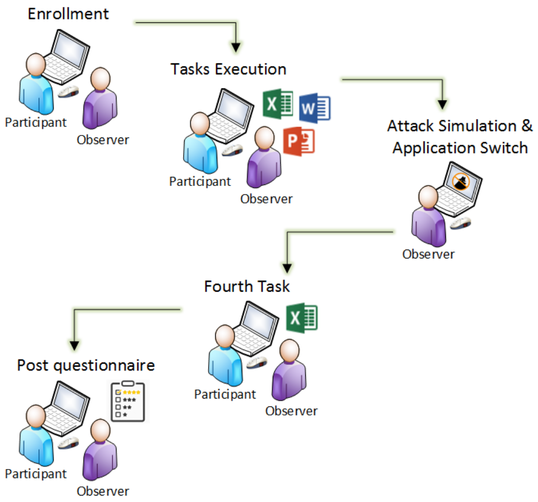


Figure 3 Workflow of the experiment.

and less than 1 minute for the fingerprint subsystem respectively. The keystroke acquisition phase needs approximately 15 minutes of keyboard typing in order to acquire sufficient statistical data about key pressure and release. We decided to have an enrollment stage of this duration because, even if it may appear long and boring to the user, having a long text to type usually increases the recognition accuracy [46].

Then, when the enrollment is completed, the observer starts the BCAS, which can either be the real or the placebo version. As in [27], the users are not informed of the presence of the placebo version of the BCAS, which does not perform any authentication and, consequently, does not introduce a significant overhead, but which has the same interface and appearance of the real BCAS.

The identified tasks represent some of the ordinary operations, as realistic as possible, that users may perform in a working environment:

- *Task Word:* writing a given text document with Microsoft Word;
- *Task Excel:* producing a spreadsheet file with Microsoft Excel;
- *Task PowerPoint:* creating a presentation with Microsoft PowerPoint.

Tasks order is selected randomly before assignment. After the participants complete the third task, they are asked to leave the workstation for a short break.

For the participants who have been using the actual BCAS, they are asked to exit the room without logging out from the BCAS. In that time interval, and with the BCAS application running, the observer sits in front of the computer and is able to verify if the BCAS rejects him as an

Table III Configurations of the groups of participants

TABLE II.B **TABLE II.A**	***trust*$_{\text{min}}$ a)** ***0.3***	***trust*$_{\text{min}}$ b)** ***0.5***	***trust*$_{\text{min}}$ c)** ***0.7***
k 1)	group I	group III	group V
k 2)	group II	group IV	group VI

impostor -as it is supposed to do- and the *Time to Impostor Rejection.* The impostor looks at the screen and uses the mouse until being rejected by the BCAS. This attack scenario may look artificial, because an impostor would probably avoid contact with the fingerprint sensor if he/she needed to use the mouse. However, in our implementation –as explained in Section 3.2- in terms of trust level and TIR, the consequence of no fingerprint recognition is the same as presenting a non-legitimate fingerprint to the sensor. The scenario also allowed to test if the face and fingerprint recognition subsystems recognize the intrusion, or if and how many times they erroneously accept the intruder. After having performed the attack and just before the end of the short break, the observer switches the BCAS to the placebo version.

Instead, for the participants who have been using the placebo version of the BCAS, they are simply asked to exit the room for a short break. In that time interval, the observer switches the placebo to the real BCAS. The attack scenario is then executed exactly in the same way as for the other group, but at the end of whole experiment.

After the short break, all the participants are asked to complete a fourth task, which is the replication of the first task, and it is supposed to take approximately the same *CT*. We introduced changes to the documents in order to reduce the learning effects [27]. The changes are on the format of the documents and on their appearance, but not on their length.

4.3. Participants and Experiments Plan

Participants were students and researchers of the University of Campinas (UNICAMP), in Brazil, and the tests took place at the Institute of Computing of the same University. We spent some weeks looking for participants, sending them an invitation through mailing lists and contacting them in laboratories and classrooms. Among the 60 respondents, 65% were male (39) and 35% (21) female. The mean age of the sample was 27.72, ranging from 19 to 41 years, with a standard deviation of 4.54. Their educational level varied from undergraduate to postdoc, and their field of study was mainly computer science or engineering. Even if the participants are computer experts, the task completion did not require any particular skill except from being capable of writing documents using mouse and keyboard, and the basic knowledge of the Microsoft Office suite.

In preparation of the experiment, we divided the 60 participants in 6 groups, having 10 participants per group. Each group had assigned a combination of $trust_{min}$ and *k* parameters. The assignment of participants to groups followed the order of appearance: for example, group I had participants number 1, 7, 13, 19,…, 55. Participants were not aware of groups' existence, neither of differences in parameters configuration.

We ordered the six groups (see Table III) based on our expectations about system security: group I conducted the test with the most usable parameter configuration, and group VI with the most secure one. Each user had the possibility to test both the real and the placebo version of the BCAS before or after the break. In detail, 80% of the participants (48 users, 8 per group) performed the three main tasks with the real application running, and the fourth task with the placebo version. Instead, the other 20% (12 users, 2 per group), had the placebo version running during the execution of the three tasks, and the real BCAS for the fourth repeated task.

4.4. Data Collection Techniques

The Completion Time of the repeated task was logged by the observer, using a chronometer. We used the BCAS application to track all the other data. An extract from the log file is the following:

```
2017/05/02 12:24:33,1,1,0,0,nodecay,acq,acq,not,0.98
```

The data contained in the log is respectively:

(i) a timestamp of each continuous authentication iteration,

(ii) a Boolean value representing the result of face, fingerprint, and keystroke recognition, and keyboard usage detection (1 for legitimate user, 0 when the trait is not acquired or the user is not legitimate),

(iii) the decaying speed of trust (fast, average, slow or no decay),

(iv) data about the biometric traits acquisition by the server -in other words, if face, fingerprint and keystroke were acquired (*acq*) in that time window or *not*-,

(v) the trust level *trust*(t_i).

As an example, a 40 minutes session correspond approximately to a log file with a length of 120 rows, where each row is generated by an iteration

Table IV **True and False Rejection and Acceptance Rates of the System and of its Subsystems**

System FRR	System FAR	Face FRR	Fingerprint FRR	Keystroke FRR	Face FAR	Fingerprint FAR	Keystroke FAR
0,61%	3,33%	4,61%	25,20%	19,78%	3,43%	0,00%	-
System TAR	**System TRR**	**Face TAR**	**Fingerprint TAR**	**Keystroke TAR**	**Face TRR**	**Fingerprint TRR**	**Keystroke TRR**
99,39%	96,67%	95,39%	74,80%	80,22%	96,57%	100%	-

5. ANALYSIS OF THE COLLECTED DATA AND DISCUSSION

5.1. BCAS Effectiveness

We are now able to discuss the effectiveness of BCAS in terms of error rates. Analyzing the results of Table IV, we can see that during the tests the face recognition subsystem had an FRR of 4,61%, the fingerprint an FRR of 25,20% and the keystroke recognition subsystem showed an FRR of 19,78%. If considered individually, the FRR of our three subsystems are slightly higher than the error rates declared by other approaches in literature as [46], [47], and [48]. In some of our tests, a high FRR is found, probably because the users were busy completing the tasks and did not focus on how well the biometric trait was presented; this may have caused imperfect traits acquisition and consequent errors in the recognition process.

Another important measurement taken for each test, in addition to logging the three subsystems' FRR, is *system false rejection*, which corresponds to any unexpected session termination. In other words, we have a system false rejection whenever the user trust level is below $trust_{min}$. As a consequence, the system FRR is obtained dividing the number of false rejections by the total number of identity verification attempts. As shown in Table IV, the system FRR is 0,61%. This means that, despite the high FRR of the individual subsystems, our algorithm for trust level calculation properly integrates the three subsystems decisions in order to: i) reduce the rejection errors and ii) let the legitimate user remaining authenticated. Furthermore, if we consider the users that performed three tasks with the real BCAS (*real system group*) separately from the 12 users that completed only one with it (*placebo group*), we obtain a FRR of 0,40% and 1,44% respectively. As expected, for the placebo group the FRR is higher than for the real system group, because performing three tasks with the placebo version of the application means that they used the real BCAS for a short period. However, the results are good if compared with [27], in which FRR was 0,86% for the real system group and 3,13% for the placebo group.

Performing the attacks, we are also able to measure the individual FAR of face and fingerprint subsystems. It is the number of times that each of the traits, belonging to the impostor sitting in front of the computer, was erroneously recognized as legitimate. In order to recreate the same conditions of trust level decreasing for all participants, the substitution of the legitimate user with the attacker happens when *trust*(t_i) is *0.98*. This is realized asking the legitimate user to look at the webcam and use the mouse right after the third task was completed, and before leaving the workstation unattended for a short break. As explained previously, during the attack scenario the impostor sits in front of the workstation looking at the screen and using the mouse until session expiration, and this has been rigorously repeated for all the 60 tests in order not to alter conditions. As a consequence, we were not able to calculate the keystroke FAR during the experiments. However, we know from [40] that with the selected algorithm, the average number of keystrokes needed to lockout an intruder varies between 79 and 348, that means about 30 words. We considered a *system false acceptance* any iteration in which the user was erroneously recognized by at least one of the subsystems, so when the trust decaying was *average* or *slow*. Results regarding acceptance rates of the system are shown in Table IV: the face subsystem has a FAR of 3,43%, and the fingerprint subsystem has an interesting FAR of 0,00%. Therefore, the system FAR is the ratio of the number of false acceptances divided by the number of identification attempts; for the BCAS it is 3,33%. We cannot compare our result with [27] because the authors did not report the FAR of their system.

In order to provide a more complete confusion matrix we also provide in Table IV the TAR and TRR of the system and of its subsystems.

5.2. BCAS Efficiency and Resources Utilization

Regarding efficiency, we first comment the results in Table V. Analyzing the log files of each experiment, we calculate the Time to Impostor Rejection (TIR) as the time needed by the BCAS to determine the instant of session expiration from the substitution of the legitimate user. In other words, it is a measure of the time necessary to decrease the trust level under the threshold.

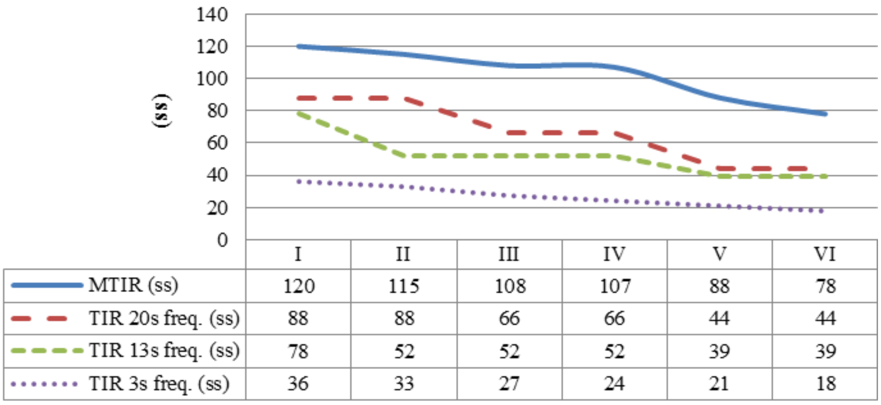
Table V System Efficiency Measures

	Mean Time to Impostor Rejection (MTIR) (mm.ss)	Mean Authentication Time (MAT)
I	02.00±0.32	100,00%
II	01.55±0.25	99,36%
II	01.48±0.49	97,96%
IV	01.47±0.47	96,09%
V	01.28±0.28	99,85%
VI	01.18±0.37	98,09%

As expected, the MTIR is proportional to the threshold (see Table V). For instance, the group I of participants, which has the lowest threshold of 0.3 and the most usable *k* parameter configuration, shows a MTIR of 2 minutes. Conversely, the MTIR of group VI, is 1 minute and 18 seconds, because of the most secure configuration of *k* parameter, and the highest $trust_{min}$ of 0.7 designed for this group of users.

However, the MTIR proportionality, and more in general all the analysis, applies as long as our system is not modified. With other implementation choices, as for instance a trust computation which distinguish between a missing trait and a not legitimate one, we may have different results in terms of usability and security measures.

Generally speaking, the TIR is influenced by the false acceptances of the recognition subsystems: between the tests that did not show false acceptances, the lowest TIR is 44s. This measure corresponds to the time needed by the BCAS to close the session if the workstation was left unattended with the configuration of group VI.



	I	II	III	IV	V	VI
MTIR (ss)	120	115	108	107	88	78
TIR 20s freq. (ss)	88	88	66	66	44	44
TIR 13s freq. (ss)	78	52	52	52	39	39
TIR 3s freq. (ss)	36	33	27	24	21	18

Figure 4 MTIR and expected TIR of the six groups.

Regarding the number of unexpected session terminations, we have that 75% of the participants were able to complete the tasks without any session termination, so the BCAS execution was actually transparent to them. We can considered it satisfying, especially if compared with the result of [27], where the percentage of completion without logout was 48%. It is also interesting to observe that the most usable parameters configuration let 100% of group I users to complete the tasks without interruptions.

In Figure 5 we show how many tests would have been completed without session expiration with the corresponding *trust*$_{\text{min}}$ value varying between 0.9 and 0.3. The Authentication Time (AT) is calculated as a fraction of the total time that the user remains authenticated [29].

For instance, suppose the total time needed for a user to complete the tasks with the real BCAS running is *T* seconds and, during this time, the system rejects the user once or more times, preventing him/her accessing the protected resources for *a* seconds. Then, the AT is calculated as *(T-a)/T*. The Mean Authentication Time (MAT) of group I was 100% because they did not have any unexpected expiration. We are not able to formally compare our results in term of MAT with [29], because of the differences in terms of tasks, length of the experiment and number of participants. However, our MAT seems to be very similar to the one obtained in [29].

In order to determine whether our system had any significant effect on Completion Time (CT), we executed a paired t-test on the difference between the tasks CT with the BCAS, versus the CT of the placebo

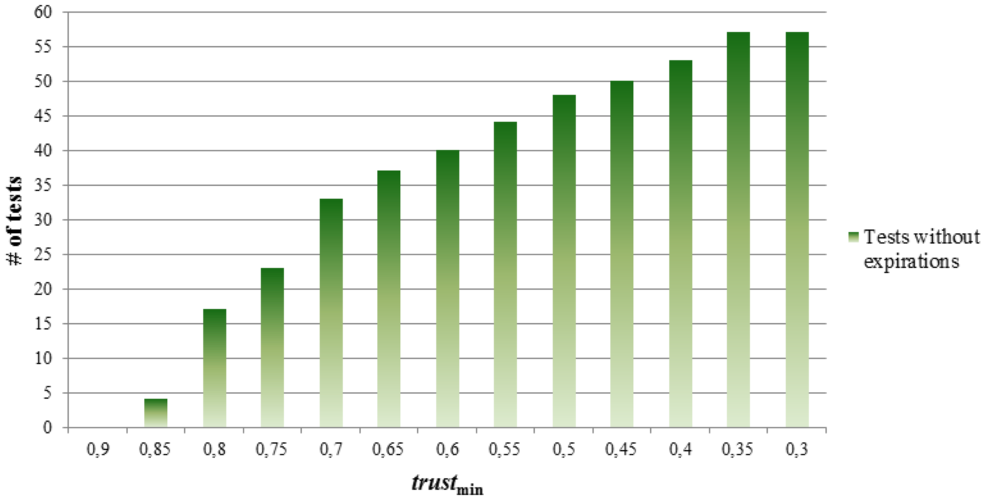


Figure 5 Expected number of tests without expirations.

version. It is generally used to compare two population means to test the null hypothesis that the true mean difference is zero.

The combined paired t-test results are in Table VI. N is the number of tests executed with the BCAS and then repeated with the placebo version, and it is 20 for each task.

The *Mean Time* is the mean difference between tests with BCAS and tests with the placebo version, *Std.Dev* is the standard deviation of the differences and *Std.Error.Mean* is the standard error of the mean difference. Table VI also shows the 95% confidence interval of their difference. Under the null hypothesis, the t-statistic follows a t-distribution with *df=n-1=19* degrees of freedom.

Comparing the values obtained for t with the t_{19} distribution, we obtain the p-value for the test. The result is that at the $p< .01$ level:

- There was *no significant difference* between time taken to complete the *Word* task with the BCAS and the placebo version (p=.1040) [t_{19}=1.7076, p>.05]).
- There was *no significant difference* between time taken to complete the *PowerPoint* task with the BCAS and the placebo version (p=.8314) [t_{19}= .2159, p>.05]).
- There was *no significant difference* between time taken to complete the *Excel* task with the BCAS and the placebo version (p=.1605) [t_{19}=1.4605, p>.05]).

We can conclude that there is no significant difference in completion time for all tasks. This gives evidence that there isn’t any significant impact on task performance, which is the same result obtained by the authors of [27].

As explained previously, in order to avoid introducing any additional overhead that could affect the Completion Time, the system overhead has been computed as a separate test and without the involvement of participants. The observer executed the PowerPoint task twice, with the real BCAS running and then repeating it with the placebo version,

Table VI Paired t-test results for task completion time.

Task	**N**	**Paired Differences: BCAS – Placebo version**					***t***	***df***	***Sig-2 tailed***
		Mean Time	***Std.Dev***	***Std.Error Mean***	***95% Confidence Interval of the Difference***				
					Lower	***Higher***			
Word	20	32.80	5.34	1.20	-7.40	73.00	1.7076	19	0.1040
PowerPoint	20	6.95	61.43	13.74	-60.42	74.32	0.2159	19	0.8314
Excel	20	42.75	-37.62	-8.41	-104.1	18.51	1.4605	19	0.1605

obtaining a comparative analysis of CPU usage with Windows Performance Analyzer ® (WPA). The resulting overhead for our machine, in terms of CPU usage, is 2,06%. This result is promising if compared with [29] and [27], who declared an overhead of 25% and 42% respectively, and is also an indication that nowadays, thanks to the technological progress of the last decade, biometric continuous authentication be actually integrated without slowing down the operating system.

5.3. User Satisfaction

Analyzing Fig. 6, we can discuss to which extent the participants are willing to provide each of the biometric traits in order to perform the enrollment. It is interesting to compare our results with [3], in which the authors perceived keystroke, fingerprint and face characteristics having respectively medium, medium and high acceptability. We have the highest acceptability for the keystroke trait; in fact, 80% of users said that they did not felt uncomfortable in providing it.

We find a high acceptability also for fingerprint (76,67%), and 70% of

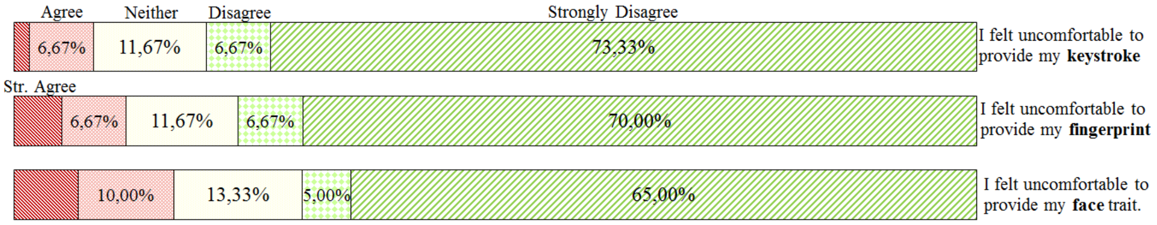


Figure 6 Questionnaire results about acceptability.

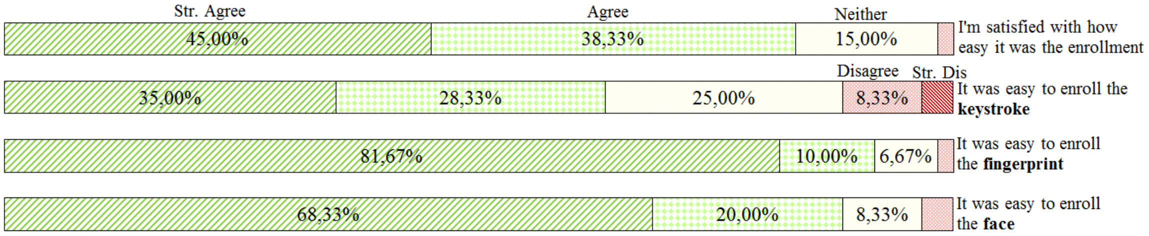


Figure 7 Questionnaire results about enrollment.

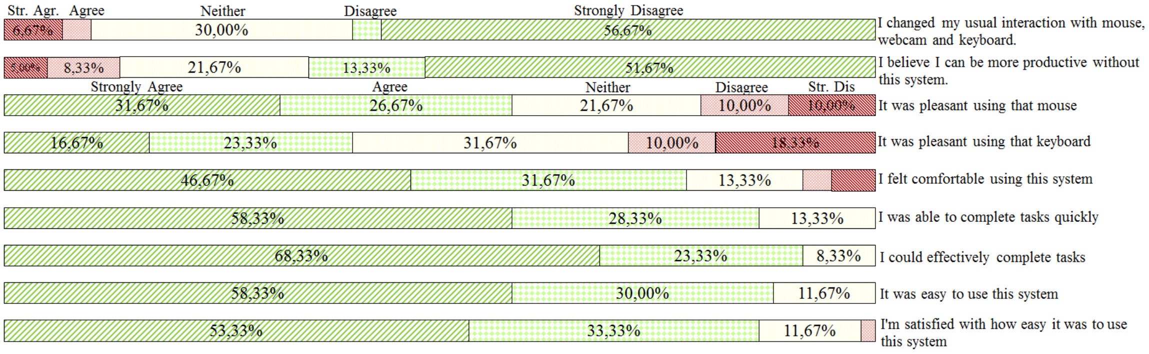


Figure 8 Questionnaire results about usability of the system.

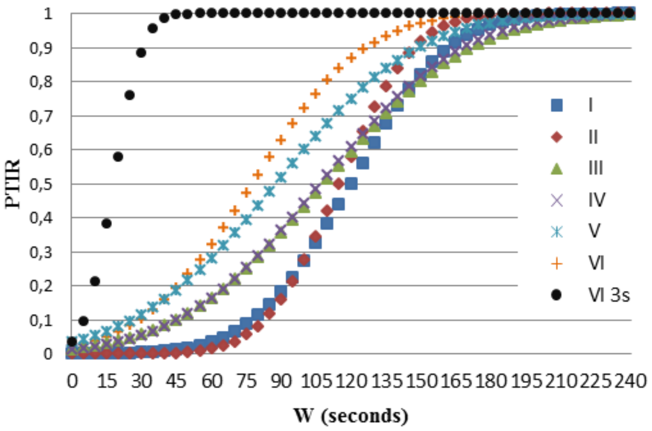


Figure 9 Plot of PTIR versus W for the six groups.

the participants felt comfortable in providing their face.

We report in Fig. 7 the users' opinion about the enrollment. Generally speaking, 83,33% of them were satisfied with its easiness. Between the three traits, as expected they felt more uneasy with the keystroke acquisition (11,67%), probably because of the 15 minutes length of the process. The results about users' satisfaction regarding system usability are shown in Fig. 8. A consistent amount of participants (28,33%) found the keyboard unpleasant: the notebook used for the tests had a column of special keys on the left that the users often pressed unintentionally. Also the mouse, with the optical sensor for the right thumb fingerprint acquisition, was not pleasant to use for 20% of the users.

These two elements probably influenced the perception on the system's usability; still it was comfortable for 78,34% of them. 13,33% of them believed they could be more productive without this system: probably because of the higher comfort and the familiarity they have with their own system, and also because they were forced to change their usual interaction with mouse and keyboard (10%). Nevertheless, the participants found the system easy to use (88,33%) and were satisfied with it (86,66%). They also said to be able to complete tasks effectively (91,66%) and quickly (86,33%), and this was one of our main objectives thinking about our system's usability. A proper comparison with [27] is impossible, because the authors did not report the complete results of their questionnaire nor all the questions; we only know that their users were satisfied with system's responsiveness, with the overall system, and with the comfort they felt.

5.4. The Trade-off between Security and Usability

In order to analyze the trade-off between usability and security, we follow the approach of [29]. As discussed in Section 4.1, we call Probability of Time to Impostor Rejection (PTIR) the probability that the TIR is lower than a vulnerability window (W). Vulnerability windows can be seen as the minimum time frames needed by an impostor to damage the critical system. In the ideal situation, the PTIR is 1, meaning that the impostor is certainly rejected.

In Fig. 9 we report the PTIR of the six groups of users, for different values of vulnerability window. The higher the PTIR, the higher is the security provided. As we can see, configuration VI is the most secure for almost all the windows of vulnerability, and this confirms our expectations. As discussed, we designed our BCAS to execute authentication iterations every 20s (plus a delay of 2-3 seconds basically for acquiring the fingerprint). For this reason, comparison with [29] is not straightforward: their system acquires 10 frames per second and requires less than 3s to reject an impostor. With our configuration, after 150s we can see that for all the curves the reached PTIR is between 0.82 and 0.97.

However, as discussed in Section 5.1 and already shown in Figure 4, increasing the frequency of user verification can easily reduce the TIR. When the interval $[t_{i-1,} t_i]$ is set to 3s, the TIR decreases to 18s. The resulting PTIR for configuration VI is shown in Fig. 9 (called “VI 3s” in the legend). It is obtained synthetically and just for comparison, but clearly demonstrates that with this configuration an impostor is rejected after 25s with a probability of 76%, and the curve tends to 1 after about 30s.

Another interesting evaluation can be done analyzing the trade-off between usability and security with the same approach of [29]. We can discuss the PTIR and the MAT, being respectively a measure of system’s security and usability. A PTIR of 1 is obtained when the impostor is always rejected, and this, for a low vulnerability window, means a MAT close to 0%. If instead we have a high MAT, it should be more unlikely to have the impostor rejected, especially with a low W.

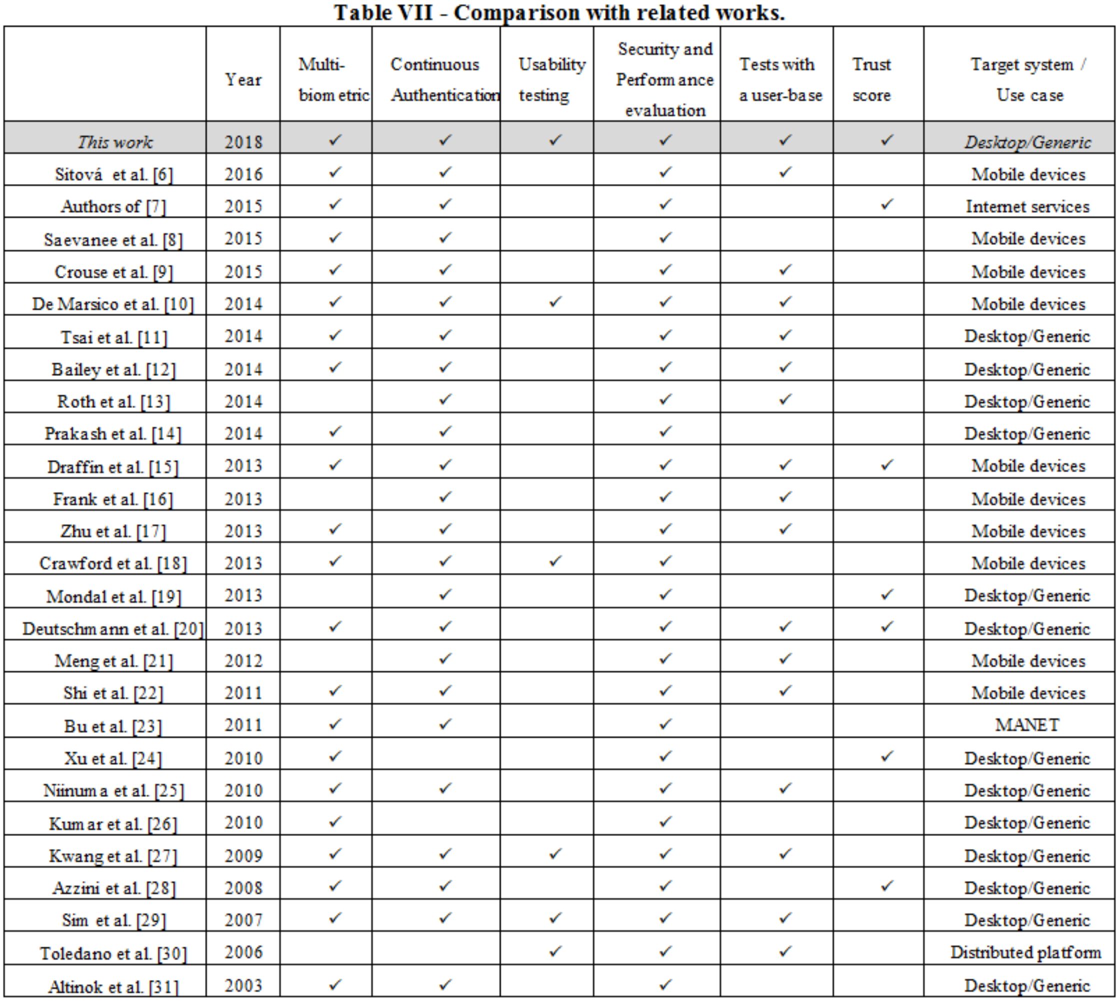
The results show that the decreasing of usability was always very low if compared with [29]. In fact, the MAT of BCAS varies from 96% to 100%. However, even if group I had the best results in terms of MAT, for the six configurations of BCAS we tested, a higher security didn’t correspond linearly to lower usability. This anomaly is probably due to the influence of FRR and FAR.

As explained in Section 3.2, in our prototype the trust computation is only influenced by the number of successful verifications. A different implementation choice, which may distinguish between a missing trait and a not legitimate trait, would have direct impact on TIR, AT and on the tradeoff between usability and security in general. In fact, we can easily imagine that it would probably reduce the TIR. On the other side, it would be interesting to study if and to which extent the modification causes any side effect, for instance in terms of FRR, AT and user satisfaction.

6. RELATED WORKS

Several studies describe frameworks, systems and novel characteristics for biometric authentication of humans. The following literature review includes the most relevant and recent papers available so far on this topic and is especially concentrated on approaches based on continuous authentication which conducted usability tests involving real users. We are not claiming to be the first authors discussing the relevance of usability.

Table VII - Comparison with related works.



	Year	Multi-biometric	Continuous Authentication	Usability testing	Security and Performance evaluation	Tests with a user-base	Trust score	Target system / Use case
This work	2018	✓	✓	✓	✓	✓	✓	*Desktop/Generic*
Sitová et al. [6]	2016	✓	✓		✓	✓		Mobile devices
Authors of [7]	2015	✓	✓		✓		✓	Internet services
Saevanee et al. [8]	2015	✓	✓		✓			Mobile devices
Crouse et al. [9]	2015	✓	✓		✓	✓		Mobile devices
De Marsico et al. [10]	2014	✓	✓	✓	✓	✓		Mobile devices
Tsai et al. [11]	2014	✓	✓		✓	✓		Desktop/Generic
Bailey et al. [12]	2014	✓	✓		✓	✓		Desktop/Generic
Roth et al. [13]	2014		✓		✓	✓		Desktop/Generic
Prakash et al. [14]	2014	✓	✓		✓			Desktop/Generic
Draffin et al. [15]	2013	✓	✓		✓	✓	✓	Mobile devices
Frank et al. [16]	2013		✓		✓	✓		Mobile devices
Zhu et al. [17]	2013	✓	✓		✓	✓		Mobile devices
Crawford et al. [18]	2013	✓	✓	✓	✓			Mobile devices
Mondal et al. [19]	2013		✓		✓		✓	Desktop/Generic
Deutschmann et al. [20]	2013	✓	✓		✓	✓	✓	Desktop/Generic
Meng et al. [21]	2012		✓		✓	✓		Mobile devices
Shi et al. [22]	2011	✓	✓		✓	✓		Mobile devices
Bu et al. [23]	2011	✓	✓		✓			MANET
Xu et al. [24]	2010	✓			✓		✓	Desktop/Generic
Niinuma et al. [25]	2010	✓	✓		✓	✓		Desktop/Generic
Kumar et al. [26]	2010	✓			✓			Desktop/Generic
Kwang et al. [27]	2009	✓	✓	✓	✓	✓		Desktop/Generic
Azzini et al. [28]	2008	✓	✓		✓		✓	Desktop/Generic
Sim et al. [29]	2007	✓	✓	✓	✓	✓		Desktop/Generic
Toledano et al. [30]	2006			✓	✓	✓		Distributed platform
Altinok et al. [31]	2003	✓	✓		✓			Desktop/Generic

Instead, we argue that an investigation on continuous authentication needs to take into explicit account the usability perspective. Our state of the art allows understanding the general awareness on the subject, and the specific initiatives and solutions for Desktop applications and especially control rooms.

We describe our findings with the support of Table VII. The surveyed works are ordered by the year of publication starting from the most recent. Then, the table reports on various aspects: the integration of multiple traits, the continuity of authentication, the presence of proper usability testing, the presence of security and performance evaluation, the involvement of real users (instead of simulations), the application of a trust for authentication, and the target system or use case.

In [6], the authors introduced a set of behavioral biometric features for continuous authentication of smartphone users: hand movement, orientation, and grasp (HMOG). They evaluated them from three perspectives—continuous authentication, biometric key generation performance, and energy consumption. The evaluation was performed on multi-session data collected from 100 subjects under two motion conditions (i.e., sitting and walking). The results demonstrate that HMOG is well suited for continuous authentication of smartphone users.

In [7], a sequential multi-modal biometric authentication system is composed of an authentication service, web services and clients. Clients are users' devices (e.g., laptop and desktop PCs, smartphones, tablets) that acquire the biometric data, and transmit those data to an authentication server for a single-sign on procedure.

The authors of [8] propose a novel text-based multimodal biometric approach utilizing linguistic analysis, keystroke dynamics and behavioural profiling. They present a framework that is able to provide robust, continuous and transparent authentication. Due to the lack of public datasets, the effectiveness of the proposed framework of providing security and user convenience was evaluated via a simulation approach (using the MATLAB environment). The simulation process involved implementing a virtual user.

The result showed that it is able to provide a 91% reduction in the number of intrusive authentication requests required for high security applications.

The paper [9] presents a work on a face-based continuous authentication system that operates in an unobtrusive manner. The authors present a methodology for fusing mobile device (unconstrained) face capture with gyroscope, accelerometer, and magnetometer data to correct for camera orientation and, by extension, the orientation of the face image. Experiments demonstrate (i) improvement of face recognition accuracy from face orientation correction, and (ii) efficacy of the prototype continuous authentication system.

In [10], a biometric application is proposed based on a multimodal recognition of face and iris, which is designed to be embedded in mobile devices. The system is inspiring for our purpose even if specific for a different target system.

A framework complementing face recognition and clothing colors for continuous authentication is proposed in [25]. Similarly, the work in [11] builds a passive continuous authentication system based on both hard and soft biometric features (e.g., clothes color). These approaches could be integrated in ours to further improve the detection capability; however, at present stage a proper usability study is missing and the improvements in security, especially tolerance to counterfeit, are not detailed.

The article in [12] presents a behavioral biometric system that fuses user data from keyboard, mouse, and Graphical User Interface (GUI) interactions. As a multimodal system, authentication decision is based on a broader view of the user's computer activity while requiring less user interaction to train the system than previous work. The system performs authentication every two minutes. Testing over 31 users shows that fusion techniques significantly improve behavioral biometric authentication accuracy over single modalities on their own.

In [13], the authors propose a novel biometric modality named typing behavior (TB) for continuous user authentication. Given a webcam pointing toward a keyboard, they develop real-time computer vision algorithms to automatically extract hand movement patterns from the video stream. Unlike the typical continuous biometrics, such as keystroke dynamics (KD), TB provides a reliable authentication with a short delay, while avoiding explicit key-logging. They collected in a database videos of 63 unique subjects, with type static text and free text for multiple sessions. The experimental results demonstrate a superior performance of TB when compared with KD.

The work in [14] introduces a new continuous user authentication scheme which is designed to authenticate the user irrespective of their posture in front of the system. The system continuously monitors the user by using *soft* biometrics (color of user's clothing and facial skin) along with *hard* biometrics. It automatically registers soft biometric traits every time the user logs in and fuses soft biometric matching with the conventional face biometric authentication.

In [15] a novel passive authentication method for mobile devices users is proposed. The authors show that how rapidly a user types with the device soft keyboard, and a variety of soft-keyboard specific micro-behavior features, reflect their unique physical and behavioral characteristics. Using this data, plus a variety of statistical tools, they generate a certainty score of whether the user's phone is in a stranger's hands. Without any contextual information, they can passively identify that a mobile device is being used by a non-authorized user.

The work in [16] investigates whether touchscreen gestures are a viable biometric trait for continuous authentication of smartphone users. Experiments to assess security and performance involving users are presented. Based on the results, the authors identify this method as suitable for multimodal biometric authentication system. However, the solution is specific for smartphones, and the authors acknowledge that it cannot securely serve as an exclusive authentication mechanism of a device.

The authors of [17] investigate the usage of passive sensory data collected from accelerometers, gyroscopes and magnetometers to ensure the security of applications and data on mobile devices. They build the gesture model of how a user uses the device and propose a framework which calculates the sureness that the mobile device is being used by its owner. Based on the sureness score, mobile devices can dynamically request the user to provide active authentication, or disable certain features of the mobile devices to protect user's privacy and information security.

The work in [18] addresses mobile device authentication, as provided by a password or sketch. It proposes an extensible Transparent Authentication Framework that integrates multiple behavioral biometrics with conventional authentication to implement a continuous authentication mechanism. The security and usability evaluation of the proposed framework showed that a legitimate device owner can perform tasks while being asked to authenticate explicitly 67% less often than without a transparent authentication method. Furthermore, the evaluation showed that attackers are soon denied access to on-device tasks as their behavioral biometrics is collected.

Both [19], and [20] are based on a trust model and influenced by [40]. Mondal et al. [19] propose to perform continuous authentication using Mouse Dynamics as the behavioral biometric modality. They used a publicly available mouse dynamics with data of 49 users and evaluated the system performance with 6 machine learning algorithms. Their continuous authentication scheme is based on a trust model which uses both global thresholds and person specific thresholds.

Deutschmann et al. [20], investigate the possibility of authenticating users continuously on desktop computers. They tested a continuous *behaviometric* authentication system on 99 users over 10 weeks, focusing on keystroke dynamics, mouse movements, application usage, and the system footprint. They continuously monitored users' activity during an entire working session. Such a continuous-authentication system uses the set of behavioral traits to calculate a similarity ratio (score) between the user's current and expected behavior.

In [21], the authors propose a novel user authentication scheme based on touch dynamics that uses a set of behavioral features related to touch dynamics for accurate user authentication. They select 21 features, collect and analyze touch gesture data of 20 Android phone users, comparing several known machine learning classifiers.

In [22], the authors describe SenGuard, a user identification framework that enables continuous and implicit user identification service for smartphone. It leverages availability of multiple sensors on smartphones and passively uses sensor inputs as sources of user authentication. SenGuard invokes active user authentication when there is mounting evidence that the phone user has changed. A prototype of SenGuard was created using voice, location, multitouch, and locomotion. Preliminary empirical studies with a set of users indicate that those four modalities are suited as data sources for implicit mobile user identification.

The authors of [23] focus on user-to-device authentication in high security mobile ad hoc networks (MANETs). The paper studies distributed combined authentication and intrusion detection with data fusion in such MANETs. Multimodal biometrics is deployed to work with intrusion detection systems (IDSs). The system decides whether user authentication (or IDS input) is required and which biosensors (or IDSs) should be chosen, depending on the security posture.

In [24] the authors propose a feature level fusion method called matrix-based complex PCA (MCPCA), for bimodal biometrics that uses a complex matrix to denote two biometric traits from one subject. The method respectively takes the two images from two biometric traits of a subject as the real part and imaginary part of a complex matrix. In order to obtain features with a small number of data items, they have devised a two-step feature extraction scheme and shown through experiments that it can achieve higher classification accuracy than other techniques as 2DPCA and PCA. The authors used different existing unimodal databases (of ear, palm print and face images) to simulate bimodal databases and create virtual subjects for the experiments.

Authors of [26] present a new approach for adaptive combination of multiple biometrics, employed to determine the optimal fusion strategy and the corresponding fusion parameters. The score-level fusion rules are adapted to ensure the desired system performance. The experimental results presented in the paper illustrate that the proposed score-level approach can achieve significantly better and stable performance over the decision-level approach. Their experiments leverage on publicly available biometric databases, which were combined one another to obtain multimodality.

In [27], a usability study is presented for a bi-modality continuous biometrics authentication system that combines fingerprint and facial biometrics to authenticate users. The system suffers from a computational overhead of up to 42% to the computer system.

The goal of paper [28] is to investigate if a multimodal biometric system can be used as input of a fuzzy controller for preventing user substitution. The chosen modalities are face and fingerprint. The fuzzy controller requests the fingerprint data only if the face recognition matching produces a trust level that is below a threshold. Experiments have not been performed with specific biometric systems, but simulating them in different conditions. In our opinion, the explicit request of fingerprint does not seem to be a proper transparent acquisition of biometric traits, which we think is a fundamental requirement to meet usability of continuous authentication.

The work in [29] presents a multimodal biometric verification system that continuously verifies the presence of a logged-in user based on face and fingerprint modalities. The imposter attacks were detected within 3 s, but at the cost of an overhead of 25% of time completion for CPU-intensive tasks.

The work in [30] promotes user-centered design and usability and security evaluation of biometric technologies, including fingerprint, voice and signature verification. However, the biometric modalities are studied in isolation, so they are not combined for a single authentication decision and there is no tailoring to a specific algorithm or context.

Authors of [31] proposed a multi-modal biometric continuous authentication system which integrates information temporally as well as between modalities. Simulations show that temporal integration improves authentication accuracy.

Therefore, the literature review highlights that, while all the studies focus on security or performance evaluation, there is a real lack of usability testing in the field of continuous authentication. Furthermore, if we do not consider works specifically tailored for mobile devices, MANETs or distributed platforms, which in our opinion are solutions not applicable in the field of control rooms or office environments, the list becomes even shorter. To our knowledge, this paper is the first to present the design, implementation and evaluation of a multi-biometric continuous authentication system from a user-centered perspective.

7. CONCLUDING REMARKS

In many critical systems and applications, it is fundamental that only authorized users are allowed to interact with the system. In some working environments, in fact, users are in charge of analyzing potentially sensitive data, taking decisions for which they are directly responsible and which may have serious implications on company's assets or even citizen's safety. Their workstations should be properly protected in order to prevent undesired consequences.

In this paper, we presented our design, implementation and experimental evaluation of a multimodal biometric continuous authentication system conducted taking into account user needs and behavior and having end users in mind in all phases of the work. Towards this end, we designed a solution which integrates face, fingerprint and keystroke recognitions and removes the necessity of explicit interactions to prove the user identity.

We defined a protocol that improves security based on the trust in the user, which is continuously computed by an authentication server. The security provided by the proposed solution can be managed through a wide set of configuration parameters.

A significant number of experiments with human participants has been performed to prove usability and security of the solution. The tests clearly stated that our system is usable and incurs in litte system overhead. Evaluations showed that the system is satisfyingly effective and efficient. The number of users who completed the test without unexpected expirations (75%) is very interesting if compared with the previous studies. However, it could be further improved reducing the FAR and FRR of the three subsystems.

Participants declared to be satisfied with the solution, and 91,66% of them said to be able to complete tasks effectively. As expected, with the change in the parameters we were able to obtain a highly usable configuration, or a more secure one, without markedly decreasing usability.

As a further contribution, we propose in this paper a repository of the multi-biometric continuous authentication logfiles, which, as far as we know, is the first of this kind publicly available [49]. In addition, as supplementary data we publish the detailed questionnaire results regarding user satisfaction, and the documents used for tasks execution [49].

We also observe that this solution has been integrated in the prototype of the *Name removed* crisis management system [41], in which users have to command intervention teams during emergencies. It is required to protect the workstations from unauthorized people that may want to acquire privacy-sensitive data, disrupt the crisis management operations, disseminate false information, or simply commit errors which will be ascribed to the person in charge of the workstation. Usability is one of the main requirements, so that users are not requested to explicitly interact to prove their identity, in order to not interfere with operations. This allowed us to describe our solution to enterprises that are actively working on the field of crisis management systems and control room operation.

As a future work, together with continuous authentication we are focusing on non-repudiation. The latter is a security service which provides evidence of users' actions, protects against their denial, and may help solving disputes between parties. We are studying if evidences can be produced for the entirety of a continuous information flow and not only at specific points. Our idea is to explore if traditional solutions as digital signature, if necessary integrated with other mechanisms, can provide *continuous non-repudiation* without reducing usability and interfering with user activities.

REFERENCES

[1] Reference removed for blind revision.

[2] Stallings, W., & Brown, L. (2008). Computer security. *Principles and Practice*.

[3] Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, *14*(1), 4-20.

[4] Hunker, J., & Probst, C. W. (2011). Insiders and Insider Threats-An Overview of Definitions and Mitigation Techniques. *JoWUA*, *2*(1), 4-27.

[5] Sasse, M. A., & Flechais, I. "Usable security: Why do we need it? How do we get it?" in *Security and Usability: Designing secure systems that people can use,* Sebastopol, US, O'Reilly, 2005, pp- 13-30.

[6] Sitová, Z., Šeděnka, J., Yang, Q., Peng, G., Zhou, G., Gasti, P., & Balagani, K. S. (2016). HMOG: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security*, *11*(5), 877-892.

[7] A. Ceccarelli, L. Montecchi, F. Brancati, P. Lollini, A. Marguglio, A. Bondavalli, "Continuous and transparent user identity verification for secure internet services," IEEE Transactions on Dependable and Secure Computing, vol. 12, n.3, pp. 270-283, 2015.

[8] Saevanee, H., Clarke, N., Furnell, S., & Biscione, V. (2015). Continuous user authentication using multi-modal biometrics. *Computers & Security*, *53*, 234-246.

[9] Crouse, D., Han, H., Chandra, D., Barbello, B., & Jain, A. K. (2015, May). Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data. In *Biometrics (ICB), 2015 International Conference on* (pp. 135-142). IEEE.

[10] De Marsico, M., Galdi, C., Nappi, M., & Riccio, D. (2014). Firme: face and iris recognition for mobile engagement. *Image and Vision Computing*, *32*(12), 1161-1172.

[11] Tsai, P. W., Khan, M. K., Pan, J. S., & Liao, B. Y. (2014). Interactive artificial bee colony supported passive continuous authentication system. *IEEE Systems Journal*, *8*(2), 395-405.

[12] Bailey, K. O., Okolica, J. S., & Peterson, G. L. (2014). User identification and authentication using multi-modal behavioral biometrics. *Computers & Security*, *43*, 77-89.

[13] Roth, J., Liu, X., & Metaxas, D. (2014). On continuous user authentication via typing behavior. *IEEE Transactions on Image Processing*, *23*(10), 4611-4624.

[14] Prakash, A., & Mukesh, R. (2014). A Biometric Approach for Continuous User Authentication by Fusing Hard and Soft Traits. *IJ Network Security*, *16*(1), 65-70.

[15] Draffin, B., Zhu, J., & Zhang, J. Y. (2013, November). KeySens: Passive User Authentication through Micro-behavior Modeling of Soft Keyboard Interaction. In *MobiCASE* (pp. 184-201).

[16] Frank, M., Biedert, R., Ma, E., Martinovic, I., & Song, D. (2013). "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication." *IEEE transactions on information forensics and security*, *8*(1), 136-148.

[17] Zhu, J., Wu, P., Wang, X., & Zhang, J. (2013, January). Sensec: Mobile security through passive sensing. In *Computing, Networking and Communications (ICNC), 2013 International Conference on* (pp. 1128-1133). IEEE.

[18] Crawford, H., Renaud, K., & Storer, T. (2013). A framework for continuous, transparent mobile device authentication. *Computers & Security*, *39*, 127-136.

[19] Mondal, S., & Bours, P. (2013, September). Continuous authentication using mouse dynamics. In *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the* (pp. 1-12). IEEE.

[20] Deutschmann, I., Nordström, P., & Nilsson, L. (2013). Continuous authentication using behavioral biometrics. *IT Professional*, *15*(4), 12-15.

[21] Meng, Y., Wong, D. S., & Schlegel, R. (2012, November). Touch gestures based biometric authentication scheme for touchscreen mobile phones. In *International Conference on Information Security and Cryptology* (pp. 331-350). Springer, Berlin, Heidelberg.

[22] Shi, W., Yang, J., Jiang, Y., Yang, F., & Xiong, Y. (2011, October). Senguard: Passive user identification on smartphones using multiple sensors. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on* (pp. 141-148). IEEE.

[23] Bu, S., Yu, F. R., Liu, X. P., Mason, P., & Tang, H. (2011). Distributed combined authentication and intrusion detection with data fusion in high-security mobile ad hoc networks. *IEEE transactions on vehicular technology*, *60*(3), 1025-1036.

[24] Xu, Y., Zhang, D., & Yang, J. Y. (2010). A feature extraction method for use with bimodal biometrics. *Pattern recognition*, *43*(3), 1106-1115.

[25] Niinuma, K., Park, U., & Jain, A. K. (2010). Soft biometric traits for continuous user authentication. *IEEE Transactions on information forensics and security*, *5*(4), 771-780.

[26] Kumar, A., Kanhangad, V., & Zhang, D. (2010). A new framework for adaptive multimodal biometrics management. *IEEE transactions on Information Forensics and Security*, *5*(1), 92-102.

[27] Kwang, G., Yap, R. H., Sim, T., & Ramnath, R. (2009, June). An usability study of continuous biometrics authentication. In *International Conference on Biometrics* (pp. 828-837). Springer, Berlin, Heidelberg.

[28] Azzini, A., Marrara, S., Sassi, R., & Scotti, F. (2008). A fuzzy approach to multimodal biometric continuous authentication. *Fuzzy Optimization and Decision Making*, *7*(3), 243-256.

[29] Sim, T., Zhang, S., Janakiraman, R., & Kumar, S. (2007). Continuous verification using multimodal biometrics. *IEEE transactions on pattern analysis and machine intelligence*, *29*(4), 687-700.

[30] Toledano, D. T., Pozo, R. F., Trapote, Á. H., & Gómez, L. H. (2006). Usability evaluation of multi-modal biometric verification systems. *Interacting with Computers*, *18*(5), 1101-1122.

[31] Altinok, A., & Turk, M. (2003, December). Temporal integration for continuous multimodal biometrics. In *Proceedings of the Workshop on Multimodal User Authentication*.

[32] Hong, L., Jain, A. K., & Pankanti, S. (1999, October). Can multibiometrics improve performance?. In *Proceedings AutoID* (Vol. 99, pp. 59-64). Citeseer.

[33] Rubin, J., and Chisnell D. “Handbook of usability testing: how to plan, design and conduct effective tests”. John Wiley & Sons, 2008.

[34] J. Nielsen. “Usability Engineering”. Boston: AP Professional, 1993.

[35] ISO/IEC 18014-2:2009 – “Information technology – Security techniques – Time-stamping services – Part 2: Mechanisms producing independent tokens”.

[36] Li, S. Z. (2009). *Encyclopedia of Biometrics: I-Z* (Vol. 1). Springer Science & Business Media.

[37] Tripathi, K. P. (2011). A comparative study of biometric technologies with reference to human interface. *International Journal of Computer Applications*, *14*(5), 10-15.

[38] SecuGen OptiMouse Plus, http://www.secugen.com/products/po.htm

[39] Davison, A.: Killer Game Programming in Java. O’Reilly Media Inc. (2005)

[40] Bours, P., Barghouthi, H.: Continuous Authentication using Biometric Keystroke Dynamics. In: The Norwegian Information Security Conference (NISK), (2009).

[41] Reference removed for blind revision.

[42] ISO 9241-100:2010- “Ergonomics of human-system interaction – Part 100: Introduction to standards related to software ergonomics”.

[43] R. Likert, “A technique for the measurement of attitudes.” *Archives of psychology* (1932).

[44] J. R. Lewis, “IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use.” International Journal of Human-Computer Interaction, 7(1), pp. 57-78, 1995.

[45] J. Prümper, “ISONORM 9241/10 – Beurteilung von Software aufGrundlage der Internationalen Ergonomie-Norm ISO 9241/10.” John Wiley & Sons, Inc, 1993.

[46] A. Alsultan, and K. Warwick, “Keystroke dynamics authentication: a survey of free-text methods.” *International Journal of Computer Science Issues*, *10*(4), 1-10, 2013.

[47] A. K. Jain, B. Klare, and U. Park, “Face recognition: Some challenges in forensics.” In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on* (pp. 726-733). IEEE, 2011.

[48] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, “*Handbook of fingerprint recognition*.” Springer Science & Business Media, 2009.

[49] *Reference removed for blind revision: dataset*

[50] *Reference removed for blind revision: the paper first introducing the algorithm here explored and assessed*

[51] E. Schiavone, A. Ceccarelli, A. Bondavalli, "*Continuous Biometric Verification for Non-Repudiation of Remote Services*", ARES 2017: 4:1-4:10.

[52] K. Dragerengen, "*Access Control in Critical Infrastructure Control Rooms using Continuous Authentication and Face Recognition*", MSc Thesis, Norwegian University of Science and Technology, 2018.

[53] T. Zoppi et al., "*Labelling relevant events to support the crisis management operator*", Journal of Software: Evolution and Process 30(3), 2018.