Continuous and Transparent User Identity Verification for Secure Internet Services

Continuous and Transparent User Identity Verification for Secure Internet Services

Andrea Ceccarelli, Leonardo Montecchi, Francesco Brancati, Paolo Lollini, Angelo Marguglio, Andrea Bondavalli, *Member, IEEE*

Abstract— Session management in distributed Internet services is traditionally based on username and password, explicit logouts and mechanisms of user session expiration using classic timeouts. Emerging biometric solutions allow substituting username and password with biometric data during session establishment, but in such an approach still a single verification is deemed sufficient, and the identity of a user is considered immutable during the entire session. Additionally, the length of the session timeout may impact on the usability of the service and consequent client satisfaction. This paper explores promising alternatives offered by applying biometrics in the management of sessions. A secure protocol is defined for perpetual authentication through continuous user verification. The protocol determines adaptive timeouts based on the quality, frequency and type of biometric data transparently acquired from the user. The functional behavior of the protocol is illustrated through Matlab simulations, while model-based quantitative analysis is carried out to assess the ability of the protocol to contrast security attacks exercised by different kinds of attackers. Finally, the current prototype for PCs and Android smartphones is discussed.

Index Terms—Security, Web Servers, Mobile Environments, Authentication

1 Introduction

SECURE user authentication is fundamental in most of modern ICT systems. User authentication systems are traditionally based on pairs of username and password and verify the identity of the user only at login phase. No checks are performed during working sessions, which are terminated by an explicit logout or expire after an idle activity period of the user.

Security of web-based applications is a serious concern, due to the recent increase in the frequency and complexity of cyber-attacks; biometric techniques [10] offer emerging solution for secure and trusted authentication, where username and password are replaced by biometric data. However, parallel to the spreading usage of biometric systems, the incentive in their misuse is also growing, especially considering their possible application in the financial and banking sectors [20], [11].

Such observations lead to arguing that a single authentication point and a single biometric data cannot guarantee a sufficient degree of security [5], [7]. In fact, similarly to traditional authentication processes which rely on username and password, biometric user authentication is typically formulated as a "single shot" [8], providing user verification only during login phase when one or more biometric traits may be required. Once the user's identity has been verified, the system resources are available for a fixed period of time or until explicit logout from the user. This approach assumes that a single verification (at the beginning of the session) is sufficient, and that the identity of the user is constant during the whole session. For instance, we consider this simple scenario: a user has already logged into a security-critical service, and then the user leaves the PC unattended in the work area for a while. This problem is even trickier in the context of mobile devices, often used in public and crowded environments, where the device itself can be lost or forcibly stolen while the user session is active, allowing impostors to impersonate the user and access strictly personal data. In these scenarios, the services where the users are authenticated can be misused easily [8], [5]. A basic solution is to use very short session timeouts and periodically request the user to input his/her credentials over and over, but this is not a definitive solution and heavily penalizes the service usability and ultimately the satisfaction of users.

To timely detect misuses of computer resources and prevent that an unauthorized user maliciously replaces an authorized one, solutions based on multi-modal biometric *continuous authentication* [5] are proposed, turning user verification into a continuous process rather than a one-time occurrence [8]. To avoid that a single biometric trait is forged, biometrics authentication can rely on multiple biometrics traits. Finally, the use of biometric authentication allows credentials to be acquired *transparently*, i.e. without explicitly notifying the user or requiring his/her interaction, which is essential to guarantee better service usability. We present some examples of transparent acquisition of biometric data. Face can be acquired while

- *A. Ceccarelli, L. Montecchi, P. Lollini and A. Bondavalli are with the Department of Mathematics and Informatics, University of Firenze, Viale Morgagni 65, 50134 Firenze, Italy. E-mail: andrea.ceccarelli@unifi.it, leonardo.montecchi@unifi.it, paolo.lollini@unifi.it, bondavalli@unifi.it.*
- *F. Brancati is with Resiltech S.R.L., Piazza Iotti 25, 56025 Pontedera (Pisa), Italy. E-mail: francesco.brancati@resiltech.com.*
- *A. Marguglio is with Engineering Ingegneria Informatica S.p.A., Viale Regione Siciliana 7275, 90146 Palermo, Italy. E-mail: angelo.marguglio@eng.it.*

Manuscript received December 2013.

the user is located in front of the camera, but not purposely for the acquisition of the biometric data; e.g., the user may be reading a textual SMS or watching a movie on the mobile phone. Voice can be acquired when the user speaks on the phone, or with other people nearby if the microphone always captures background. Keystroke data can be acquired whenever the user types on the keyboard, for example when writing an SMS, chatting, or browsing on the Internet. This approach differentiates from traditional authentication processes, where username/password are requested only once at login time or explicitly required at confirmation steps; such traditional authentication approaches impair usability for enhanced security, and offer no solutions against forgery or stealing of passwords.

This paper presents a new approach for user verification and session management that is applied in the CASHMA (Context Aware Security by Hierarchical Multilevel Architectures [1]) system for secure biometric authentication on the Internet. CASHMA is able to operate securely with any kind of web service, including services with high security demands as online banking services, and it is intended to be used from different client devices e.g., smartphones, Desktop PCs or even biometric kiosks placed at the entrance of secure areas. Depending on the preferences and requirements of the owner of the web service, the CASHMA authentication service can complement a traditional authentication service, or can replace it.

The approach we introduced in CASHMA for usable and highly secure user sessions is a *continuous sequential* (a single biometric modality at once is presented to the system [22]) *multi-modal* biometric authentication protocol, which adaptively computes and refreshes session timeouts on the basis of the trust put in the client. Such *global trust* is evaluated as a numeric value, computed by continuously evaluating the trust both in the *user* and the (biometric) *subsystems* used for acquiring biometric data. In the CASHMA context, each subsystem comprises all the hardware/software elements necessary to acquire and verify the authenticity of one biometric trait, including sensors, comparison algorithms and all the facilities for data transmission and management. Trust in the user is determined on the basis of frequency of updates of fresh biometric samples, while trust in each subsystem is computed on the basis of the quality and variety of sensors used for the acquisition of biometric samples, and on the risk of the subsystem to be intruded.

Exemplary runs carried out using Matlab are reported, and a quantitative model-based security analysis of the protocol is performed combining the SAN (Stochastic Activity Networks [16]) and ADVISE (ADversary VIew Security Evaluation [12]) formalisms.

The driving principles behind our protocol were briefly discussed in the short paper [18], together with minor qualitative evaluations. This paper extends [18] both in the design and the evaluation parts, by providing an in-depth description of the protocol and presenting extensive qualitative and quantitative analysis.

The rest of the paper is organized as follows. Section 2 introduces the preliminaries to our work. Section 3 illustrates the architecture of the CASHMA system, while Sections 4 describes our continuous authentication protocol. Exemplary simulations of the protocol using Matlab are shown in Section 5, while Section 6 presents a quantitative model-based analysis of the security properties of the protocol. Section 7 present the running prototype, while concluding remarks are in Section 8.

2 Preliminaries

2.1 Continuous Authentication

A significant problem that continuous authentication aims to tackle is the possibility that the user device (smartphone, table, laptop, etc.) is used, stolen or forcibly taken after the user has already logged into a security-critical service, or that the communication channels or the biometric sensors are hacked.

In [7] a multi-modal biometric verification system is designed and developed to detect the physical presence of the user logged in a computer. The proposed approach assumes that first the user logs in using a strong authentication procedure, then a continuous verification process is started based on multi-modal biometric. Verification failure together with a conservative estimate of the time required to subvert the computer can automatically lock it up. Similarly, in [5] a multi-modal biometric verification system is presented, which continuously verifies the presence of a user working with a computer. If the verification fails, the system reacts by locking the computer and by delaying or freezing the user's processes.

The work in [8] proposes a multi-modal biometric continuous authentication solution for local access to high-security systems as ATMs, where the raw data acquired are weighted in the user verification process, based on i) type of the biometric traits and ii) time, since different sensors are able to provide raw data with different timings. Point ii) introduces the need of a temporal integration method which depends on the availability of past observations: based on the assumption that as time passes, the confidence in the acquired (aging) values decreases. the paper applies a degeneracy function that measures the uncertainty of the score computed by the verification function. In [22], despite the focus is not on continuous authentication, an automatic tuning of decision parameters (thresholds) for sequential multi-biometric score fusion is presented: the principle to achive multimodality is to consider *monomodal* biometric subsystems *sequentially*.

In [3] a wearable authentication device (a wristband) is presented for a continuous user authentication and transparent login procedure in applications where users are nomadic. By wearing the authentication device, the user can login transparently through a wireless channel, and can transmit the authentication data to computers simply approaching them.

2.2 Quantitative Security Evaluation

Security assessment relied for several years on qualitative analyses only. Leaving aside experimental evaluation and data analysis [26], [25], model-based *quantitative* security

assessment is still far from being an established technique despite being an active research area.

Specific formalisms for security evaluation have been introduced in literature, enabling to some extent the quantification of security. Attack trees are closely related to fault trees: they consider a security breach as a system failure, and describe sets of events that can lead to system failure in a combinatorial way [14]; they however do not consider the notion of time. Attack graphs [13] extend attack trees by introducing the notion of state, thus allowing more complex relations between attacks to be described. Mission Oriented Risk and Design Analysis (MORDA) assesses system risk by calculating attack scores for a set of system attacks. The scores are based on adversary attack preferences and the impact of the attack on the system [23]. The recently introduced ADVISE (Adversary VIew Security Evaluation) formalism [12] extends the attack graph concept with quantitative information and supports the definition of different attackers profiles.

In CASHMA assessment, the choice of ADVISE was mainly due to: i) its ability to model detailed adversary profiles, ii) the possibility to combine it with other stochastic formalisms as the Möbius multi-formalism [15], and iii) the ability to define ad-hoc metrics for the system we were targeting. This aspect is explored in Section 6.

2.3 Novelty of Our Approach

Our continuous authentication approach is grounded on transparent acquisition of biometric data and on adaptive timeout management on the basis of the trust posed in the user and in the different subsystems used for authentication. The user session is open and secure despite possible idle activity of the user, while potential misuses are detected by continuously confirming the presence of the proper user.

Our continuous authentication protocol significantly differs from the work we surveyed in the biometric field as it operates in a very different context. In fact, it is integrated in a distributed architecture to realize a secure and usable authentication service, and it supports security-critical web services accessible over the Internet. We remark that although some very recent initiatives for multimodal biometric authentication over the Internet exist (e.g., the BioID BaaS - Biometric Authentication as a Service is presented in 2011 as the first multi-biometric authentication service based on the Single Sign-On [4]), to the authors' knowledge none of such approaches supports continuous authentication.

Another major difference with works [5] and [7] is that our approach does not require that the reaction to a user verification mismatch is executed by the user device (e.g., the logout procedure), but it is transparently handled by the CASHMA authentication service and the web services, which apply their own reaction procedures.

The length of the session timeout in CASHMA is calculated according to the trust in the users and the biometric subsystems, and tailored on the security requirements of the service. This provides a trade-off between usability and security. Although there are similarities with the overall objectives of the decay function in [8] and the approach for sequential multi-modal system in [22], the reference systems are significantly different. Consequently, different requirements in terms of data availability, frequency, quality, and security threats lead to different solutions.

2.3 Basic Definitions

In this section we introduce the basic definitions that are adopted in this paper. Given n unimodal biometric subsystems S_k, with $k= 1, 2, \ldots, n$ that are able to decide independently on the authenticity of a user, the False Non-Match Rate, $FNMR_k$ is the proportion of genuine comparisons that result in false non-matches. False non-match is the decision of non-match when comparing biometric samples that are from same biometric source (i.e., genuine comparison) [10]. It is the probability that the unimodal system S_k wrongly rejects a legitimate user. Conversely, the False Match Rate, FMR_k is the probability that the unimodal subsystem S_k makes a *false match* error [10] i.e., it wrongly decides that a non legitimate user is instead a legitimate one (assuming a fault-free and attack-free operation). Obviously, a false match error in a unimodal system would lead to authenticate a non legitimate user. To simplify the discussion but without loosing the general applicability of the approach, hereafter we consider that each sensor allows acquiring only one biometric trait; e.g., having n sensors means that at most n biometric traits are used in our sequential multimodal biometric system.

The *subsystem trust level* $m(S_k, t)$ is the probability that the unimodal subsystem S_k at time t does not authenticate an impostor (a non-legitimate user) considering both the quality of the sensor (i.e., FMR_k) and the risk that the subsystem is intruded.

The *user trust level* $g(u, t)$ indicates the trust placed by the CASHMA authentication service in the user u at time t, i.e., the probability that the user u is a legitimate user just considering his behavior in terms of device utilization (e.g., time since last keystroke or other action) and the time since last acquisition of biometric data.

The *global trust level trust*(u, t) describes the belief that at time t the user u in the system is actually a legitimate user, considering the combination of all subsystems trust levels $m(S_{k=1,\ldots,n}, t)$ and of the user trust level $g(u, t)$.

The *trust threshold* g_{min} is a lower threshold on the global trust level required by a specific web service; if the resulting global trust level at time t is smaller than g_{min} (i.e., $g(u,t) < g_{min}$), the user u is not allowed to access to the service. Otherwise if $g(u,t) \geq g_{min}$ the user u is authenticated and is granted access to the service.

3 The CASHMA Architecture

3.1 Overall View of the System

The overall system is composed of the CASHMA authentication service, the clients and the web services (Fig. 1), connected through communication channels. Each communication channel in Fig. 1 implements specific security measures which are not discussed here for brevity.

The CASHMA authentication service includes: i) an *authentication server*, which interacts with the clients, ii) a set

of high-performing *computational servers* that perform comparisons of biometric data for verification of the enrolled users, and iii) *databases of templates* that contain the biometric templates of the enrolled users (these are required for user authentication/verification). The *web services* are the various services that use the CASHMA authentication service and demand the authentication of enrolled users to the CASHMA authentication server. These services are potentially any kind of Internet service or application with requirements on user authenticity. They have to be registered to the CASHMA authentication service, expressing also their trust threshold. If the web services adopt the continuous authentication protocol, during the registration process they shall agree with the CASHMA registration office on values for parameters h, k and s used in Section 4.2.

Finally, by *clients* we mean the users' devices (laptop and desktop PCs, smartphones, tablet, etc.) that acquire the *biometric data* (the *raw data*) corresponding to the various biometric traits from the users, and transmit those data to the CASHMA authentication server as part of the authentication procedure towards the target web service. A client contains i) sensors to acquire the raw data, and ii) the CASHMA application which transmits the biometric data to the authentication server. The CASHMA authentication server exploits such data to apply user authentication and successive verification procedures that compare the raw data with the stored biometric templates.

Transmitting raw data has been a design decision applied to the CASHMA system, to reduce to a minimum the dimension, intrusiveness and complexity of the application installed on the client device, although we are aware that the transmission of raw data may be restricted for example due to National legislations.

CASHMA includes countermeasures to protect the biometric data and to guarantee users' privacy, including policies and procedures for proper registration; protection of the acquired data during its transmission to the authentication and computational servers and its storage; robustness improvement of the algorithm for biometric verification [24]. Privacy issues still exist due to the acquisition of data from the surrounding environment as for example voices of people nearby the CASHMA user, but are considered out of scope for this paper.

The continuous authentication protocol explored in this paper is independent from the selected architectural choices and can work with no differences if templates and feature sets are used instead of transmitting raw data, or independently from the set of adopted countermeasures.

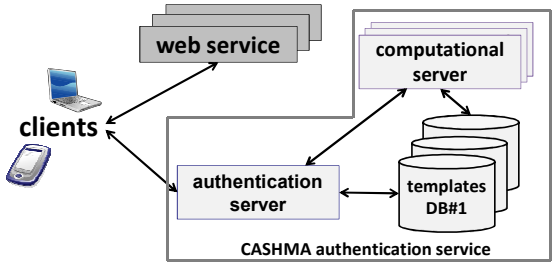

Fig. 1. Overall view of the CASHMA architecture.

3.2 Sample Application Scenario

CASHMA can authenticate to web services, ranging from services with strict security requirements as online banking services to services with reduced security requirements as forums or social networks. Additionally, it can grant access to physical secure areas as a restricted zone in an airport, or a military zone (in such cases the authentication system can be supported by biometric kiosk placed at the entrance of the secure area). We explain the usage of the CASHMA authentication service by discussing the sample application scenario in Fig. 2 where a user u wants to log into an Online Banking service using a smartphone.

It is required that the user and the web service are enrolled to the CASHMA authentication service. We assume that the user is using a smartphone where a CASHMA application is installed.

The smartphone contacts the Online Banking service, which replies requesting the client to contact the CASHMA authentication server and get an authentication *certificate*. Using the CASHMA application, the smartphone sends its unique identifier and biometric data to the authentication server for verification. The authentication server verifies the user identity, and grants the access if: i) it is enrolled in the CASHMA authentication service, ii) it has rights to access the Online Banking service and, iii) the acquired biometric data match those stored in the templates database associated to the provided indentifier. In case of successful user verification, the CASHMA authentication server releases an authentication certificate to the client, proving its identity to third parties, and includes a timeout that sets the maximum duration of the user session. The client presents this certificate to the web service, which verifies it and grants access to the client.

The CASHMA application operates to continuously maintain the session open: it transparently acquires biometric data from the user, and sends them to the CASHMA authentication server to get a new certificate. Such certificate, which includes a new timeout, is forwarded to the web service to further extend the user session.

3.3 The CASHMA certificate

In the following we present the information contained in the body of the CASHMA certificate transmitted to the client by the CASHMA authentication server, necessary to understand details of the protocol.

Timestamp and *sequence number* univocally identify each certificate, and protect from replay attacks.

ID is the user ID e.g., a number.

Decision represents the outcome of the verification pro-

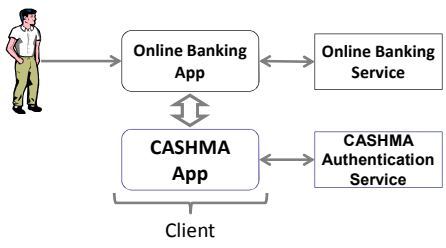

Fig. 2. Example scenario: accessing an online banking service using a smartphone.

cedure carried out on the server side. It includes the *expiration time* of the session, dynamically assigned by the CASHMA authentication server. In fact, the global trust level and the session timeout are always computed considering the time instant in which the CASHMA application acquires the biometric data, to avoid potential problems related to unknown delays in communication and computation. Since such delays are not predicable, simply delivering a relative timeout value to the client is not feasible: the CASHMA server therefore provides the absolute instant of time at which the session should expire.

4 The Continuous Authentication Protocol

The continuous authentication protocol allows providing adaptive session timeouts to a web service to set up and maintain a secure session with a client. The timeout is adapted on the basis of the trust that the CASHMA authentication system puts in the biometric subsystems and in the user. Details on the mechanisms to compute the adaptive session timeout are presented in Section 4.2.

4.1 Description of the Protocol

The proposed protocol requires a sequential multi-modal biometric system composed of n unimodal biometric subsystems that are able to decide independently on the authenticity of a user. For example, these subsystems can be one subsystem for keystroke recognition and one for face recognition.

The idea behind the execution of the protocol is that the client continuously and transparently acquires and transmits evidence of the user identity to maintain access to a web service. The main task of the proposed protocol is to create and then maintain the user session adjusting the session timeout on the basis of the confidence that the identity of the user in the system is genuine.

The execution of the protocol is composed of two consecutive phases: the initial phase and the maintenance phase. The *initial phase* aims to *authenticate* the user into the system and establish the session with the web service. During the *maintenance phase*, the session timeout is adaptively updated when *user identity verification* is performed using fresh raw data provided by the client to the CASHMA authentication server. These two phases are detailed hereafter with the help of Fig. 3 and Fig. 4.

Initial phase. This phase is structured as follows:

- The user (the client) contacts the web service for a service request; the web service replies that a valid certificate from the CASHMA authentication service is required for authentication.
- Using the CASHMA application, the client contacts the CASHMA authentication server. The first step consists in acquiring and sending at time t_0 the data for the different biometric traits, specifically selected to perform a strong authentication procedure (step 1). The application explicitly indicates to the user the biometric traits to be provided and possible retries.
- The CASHMA authentication server analyzes the biometric data received and performs an authentication procedure. Two different possibilities arise here. If the user identity is not verified (the global trust level is below the trust threshold g_{min}), new or additional biometric data are requested (back to step 1) until the minimum trust threshold g_{min} is reached. Instead if the user identity is successfully verified, the CASHMA authentication server authenticates the user, computes an initial timeout of length T_0 for the user session, set the expiration time at $T_0 + t_0$, creates the CASHMA certificate and sends it to the client (step 2).
- The client forwards the CASHMA certificate to the web service (step 3) coupling it with its request.
- The web service reads the certificate and authorizes the client to use the requested service (step 4) until time $t_0 + T_0$.

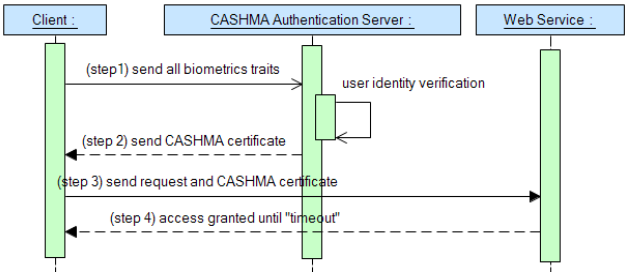

Fig. 3. Initial phase in case of successful user authentication.

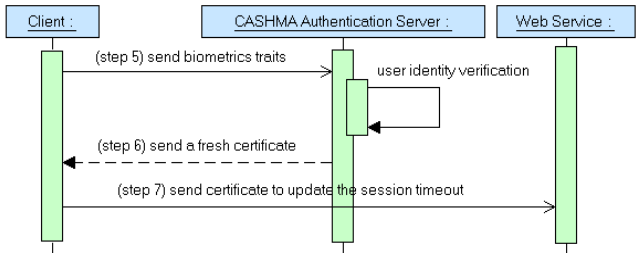

Fig. 4. Maintenance phase in case of successful user verification.

For clarity, step 1 to step 4 are represented in Fig. 3 for the case of successful user verification only.

Maintenance phase. It is composed of three steps repeated iteratively:

- When at time t_i the client application acquires fresh (new) raw data (corresponding to *one* biometric trait), it communicates them to the CASHMA authentication server (step 5). The biometric data can be acquired transparently to the user; the user may however decide to provide biometric data which are unlikely acquired in a transparent way (e.g., fingerprint). Finally when the session timeout is going to expire, the client may explicitly notify to the user that fresh biometric data are needed.
- The CASHMA authentication server receives the biometric data from the client and verifies the identity of the user. If verification is not successful, the user is marked as not legitimate, and consequently the CASHMA authentication server does not operate to refresh the session timeout. This does not imply that the user is cut-off from the current session: if other

biometric data are provided before the timeout expires, it is still possible to get a new certificate and refresh the timeout. If verification is successful, the CASHMA authentication server applies the algorithm detailed in Section 4.2 to adaptively compute a new timeout of length T_i, the expiration time of the session at time $T_i + t_i$ and then it creates and sends a new certificate to the client (step 6).

- The client receives the certificate and forwards it to the web service; the web service reads the certificate and sets the session timeout to expire at time $t_i + T_i$ (step 7).

The steps of the maintenance phase are represented in Fig. 4 for the case of successful user verification (step 6b).

4.2 Trust Levels and Timeout Computation

The algorithm to evaluate the expiration time of the session executes iteratively on the CASHMA authentication server. It computes a new timeout and consequently the expiration time each time the CASHMA authentication server receives fresh biometric data from a user. Let us assume that the initial phase occurs at time t_0 when biometric data is acquired and transmitted by the CASHMA application of the user u, and that during the maintenance phase at time $t_i > t_0$ for any i=1, … m new biometric data is acquired by the CASHMA application of the user u (we assume these data are transmitted to the CASHMA authentication server and lead to successful verification i.e., we are in the conditions of Fig. 4). The steps of the algorithm described hereafter are executed.

To ease the readability of the notation, in the following the user u is often omitted; for example $g(t_i) = g(u, t_i)$.

4.2.1 Computation of Trust in the Subsystems

The algorithm starts computing the trust in the subsystems. Intuitively, the subsystem trust level could be simply set to the static value $m(S_k, t)$=1-$FMR(S_k)$ for each unimodal subsystem S_k and any time t (we assume that information on the subsystems used, including their FMRs, is contained in a repository accessible by the CASHMA Authentication Server). Instead we apply a penalty function to *calibrate* the trust in the subsystems on the basis of its usage. Basically, in our approach the more the subsystem is used, the less it is trusted: to avoid that a malicious user is required to manipulate only one biometric trait (e.g., through sensor spoofing [10]) to keep authenticated to the online service, we decrease the trust in those subsystems which are repeatedly used to acquire the biometric data.

In the initial phase $m(S_k, t_0)$ is set to $1\text{-}FMR(S_k)$ for each subsystem S_k used. During the maintenance phase, a penalty function is associated to consecutive authentications performed using the same subsystem as follows:

$$penalty\ (x, h) = e^{x \cdot h}$$

where x is the number of consecutive authentication attempts using the same subsystem and $h>0$ is a parameter used to tune the penalty function. This function increases exponentially; this means that using the same subsystem for several authentications heavily increases the penalty.

The computation of the penalty is the first step for the computation of the subsystem trust level. If the same subsystem is used in consecutive authentications, the subsystem trust level is a multiplication of i) the subsystem trust level $m(S_k, t_{i-1})$ computed in the previous execution of the algorithm, and ii) the inverse of the penalty function (the higher is the penalty, the lower is the subsystem trust level):

$$m(S_k, t_i) = m(S_k, t_{i-1}) \cdot (penalty\ (x, h))^{-1}.$$

Otherwise if the subsystem is used for the first time or in non-consecutive user identity verification, $m(S_k, t_i)$ is set to $1\text{-}FMR(S_k)$. This computation of the penalty is intuitive but fails if more than one subsystem are compromised (e.g., two fake biometric data can be provided in an alternate way). Other formulations that include the history of subsystems usage can be identified but are outside the scope of this paper.

4.2.2 Computation of Trust in the User

As time passes from the most recent user identity verification, the probability that an attacker substituted to the legitimate user increases i.e., the level of trust in the user decreases. This leads us to model the user trust level through time using a function which is asymptotically decreasing towards zero. Among the possible models we selected the function in (1), which: i) asymptotically decreases towards zero; ii) yields $trust(t_{i-1})$ for Δt_i=0; and iii) can be tuned with two parameters which control the delay (s) and the slope (k) with which the trust level decreases over time (Fig. 5 and Fig. 6). Different functions may be preferred under specific conditions or users requirements; in this paper we focus on introducing the protocol, which can be realized also with other functions.

During the initial phase, the user trust level is simply set to $g(t_0)$ = 1. During the maintenance phase, the user trust level is computed for each received fresh biometric data. The user trust level at time t_i is given by:

$$g(t_i) = \frac{(-\arctan((\Delta t_i - s) \cdot k) + \frac{\pi}{2}) \cdot trust(t_{i-1})}{-\arctan(-s \cdot k) + \frac{\pi}{2}}. \quad (1)$$

Value $\Delta t_i = t_i - t_{i-1}$ is the time interval between two data transmissions; $trust(t_{i-1})$ instead is the global trust level computed in the previous iteration of the algorithm. Parameters k and s are introduced to tune the decreasing function: k impacts on the inclination towards the falling inflection point, while s translates the inflection point horizontally i.e., allows anticipating or delaying the decay.

Fig. 5 and Fig. 6 show the user trust level for different values of s and k. Note that s and k allow adapting the algorithm to different services: for example, services with strict security requirements as banking services may adopt a high k value and a small s value to have a faster decrease of the user trust level. Also we clarify that in Fig. 5, Fig. 6 and in the following of the paper, we intentionally avoid using measurements units for time quantities (e.g., seconds), since they depend upon the involved application and do not add significant value to the discus-

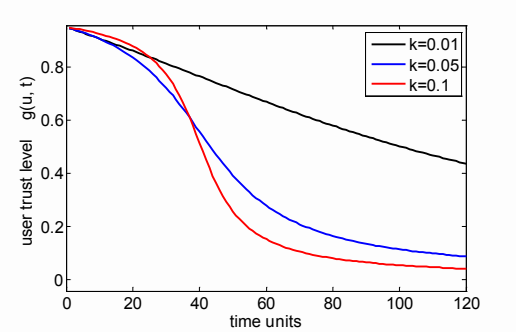

Fig. 5. Evolution of the user trust level when k=[0.01, 0.05,0.1] and s=40.

sion.

4.2.3 Merging User Trust and Subsystems Trust: the Global Trust Level

The global trust level is finally computed combining the user trust level with the subsystem trust level.

In the initial phase, multiple subsystems may be used to perform an initial strong authentication. Let n be the number of different subsystems, the global trust level is firstly computed during the initial phase as follows:

Equation (2) includes the subsystem trust level of all subsystems used in the initial phase. We remind that for

$$trust(t_0) = 1 - \Pi_{k=1,\ldots,n}(1 - m(S_k, t_0)). \quad (2)$$

the first authentication $m(S_k, t_0)$ is set to 1-$FMR(S_k)$. The different subsystems trust levels are combined adopting the *OR*-rule from [2], considering only the false acceptance rate: each subsystem proposes a score, and the combined score is more accurate than the score of each individual subsystem. The first authentication does not consider trust in the user behavior, and only weights the trust in the subsystems. The FNMR is not considered in this computation because it only impact on the reliability of the session, while the user trust level is intended only for security.

Instead, the global trust level in the maintenance phase is a linear combination of the *user trust level* and the *subsystem trust level*. Given the user trust level $g(t_i)$ and the subsystem trust level $m(S_k, t_i)$, the global trust level is computed again adopting the *OR*-rule from [2], this time with only two input values. Result is as follows:

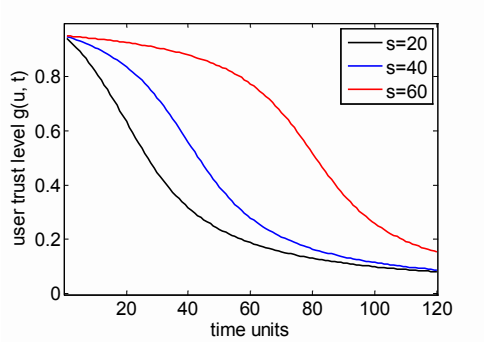

Fig. 6. Evolution of the user trust level when k=0.05 and s=[20, 40, 60].

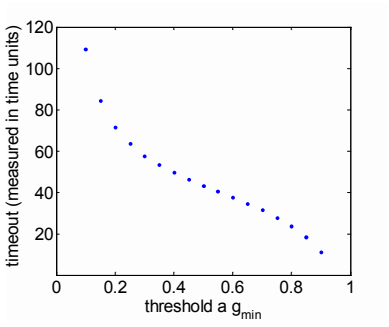

Fig. 7. Timeout values for $g_{min} \in$ [0.1, 0.9], k = 0.05 and s = 40.

$$\begin{aligned} trust\ (t_i) &= 1 - (1 - g(t_i))\ (1 - m(S_k, t_i)) = \\ &= g(t_i) + m(S_k, t_i) - g(t_i)\ m(S_k, t_i) = \\ &= g(t_i) + (1 - g(t_i))\ m(S_k, t_i). \end{aligned} \quad (3)$$

4.2.4 Computation of the Session Timeout

The last step is the computation of the length T_i of the session timeout. This value represents the time required by the global trust level to decrease until the trust threshold g_{min} (if no more biometric data are received). Such value can be determined by inverting the user trust level function (1) and solving it for Δt_i.

Starting from a given instant of time t_i, we consider t_{i+1} as the instant of time at which the global trust level reaches the minimum threshold g_{min}, i.e., $g(t_{i+1})=g_{min}$. The timeout is then given by $T_i=\Delta t_i=t_{i+1}-t_i$. To obtain a closed formula for such value we first instantiated (1) for i+1 i.e., we substituted $trust(t_{i-1})$ with $trust(t_i)$, $\Delta t_i = T_i$ and $g(t_i) = g_{min}$.

By solving for T_i, we finally obtain Equation (4), which allows the CASHMA service to dynamically compute the session timeout based on the current global trust level. The initial phase and the maintenance phase are computed in the same way: the length T_i of the timeout at time t_i for the user u is:

$$T_i = \begin{cases} \tan\left(\dfrac{g_{\min} \cdot (\arctan(-s \cdot k) - \frac{\pi}{2})}{trust(t_i)} + \dfrac{\pi}{2}\right) \cdot \dfrac{1}{k} + s & \text{if } T_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

It is then trivial to set the *expiration time* of the certificate at $T_i + t_i$.

In Fig. 7 the length T_i of the timeout for different values of g_{min} is shown; the higher is g_{min}, the higher are the security requirements of the web service, and consequently the shorter is the timeout.

5 Exemplary Runs

This section reports Matlab executions of the protocol. Four different biometric traits acquired through four different subsystems are considered for biometric verification: voice, keystroke, fingerprint, and face.

We associate the following *FMRs* to each of them: 0.06 to the voice recognition system (vocal data is acquired through a microphone), 0.03 to the fingerprint recognition system (the involved sensor is a fingerprint reader; the corresponding biometric data are not acquired transparently but are explicitly provided by the user), 0.05 to the facial recognition system (the involved sensor is a cam-

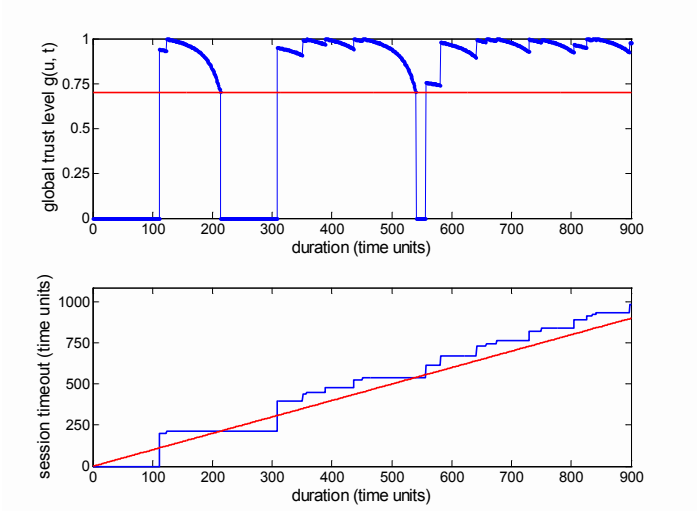

Fig. 8. Global trust level (top) and session timeout (bottom) in a nominal scenario.

era), and 0.08 to keystroke recognition (a keyboard or a touch/tactile-screen can be used for data acquisition). Note that the FMRs must be set on the basis of the sensors and technologies used. We also assume that the initial phase of the protocol needs only one raw data.

The first scenario, depicted in Fig. 8, is a simple but representative execution of the protocol: in 900 time units, the CASHMA authentication server receives 20 fresh biometric data from a user and performs successful verifications. The upper part of Fig. 8 shows the behavior of the user trust level (the continuous line) with the g_{min} threshold (the dashed line) set to $g_{min} = 0.7$. In the lower graph the evolution of the session timeout is shown (it is the continuous line). When the continuous line intersects the dashed line, the timeout expires. The time units are reported on the x-axis. The k and s parameters are set to $k = 0.05$ and $s = 100$. The first authentication is at time unit 112, followed by a second one at time unit 124. The global trust level after these first two authentications is 0.94. The corresponding session timeout is set to expire at time unit 213: if no fresh biometric data are received before time unit 213, the global trust level intersects the threshold g_{min}. Indeed, this actually happens: the session closes, and the global trust level is set to 0. Session remains closed until a new authentication at time unit 309 is performed. The rest of the experiment runs in a similar way.

The next two runs provide two examples of how the threshold g_{min} and the parameters k and s can be selected to meet the security requirements of the web service. We represent the execution of the protocol to authenticate to two web services with very different security requirements: the first with low security requirements, and the second with severe security requirements.

Fig. 9 describes the continuous authentication protocol for the first system. The required trust on the legitimacy of the user is consequently reduced; session availability and transparency to the user are favored. The protocol is tuned to maintain the session open with sparse authentications. Given $g_{min} = 0.6$, and parameters $s = 200$ and $k = 0.005$ set for a slow decrease of user trust level, the plot in Fig. 9 contains 10 authentications in 1000 time units, showing a unique timeout expiration after 190 time units

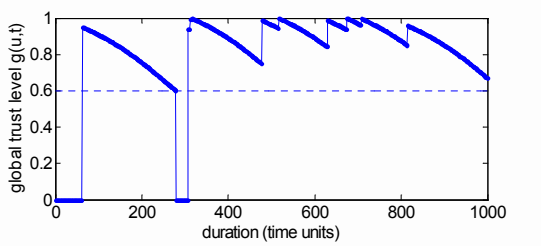

Fig. 9. Global trust level and 10 authentications for a service with low security requirements.

om the first authentication;

Fig. 10 describes the continuous authentication protocol applied to a web service with severe security requirements. In this case, session security is preferred to session availability or transparency to the user: the protocol is tuned to maintain the session open only if biometric data are provided frequently and with sufficient alternation between the available biometric traits. Fig. 10 represents the global trust level of a session in which authentication data are provided 40 times in 1000 time units using $g_{min} = 0.9$, and the parameters $s = 90$ and $k = 0.003$ set for rapid decrease. Maintaining the session open requires very frequent transmissions of biometric data for authentication. This comes at the cost of reduced usability, because a user which does not use the device continuously will most likely incur in timeout expiration.

6 Security Evaluation

A complete analysis of the CASHMA system was carried out during the CASHMA project [1], complementing traditional security analysis techniques with techniques for quantitative security evaluation. Qualitative security analysis, having the objective to identify threats to CASHMA and select countermeasures, was guided by general and accepted schemas of biometric attacks and attack points as [9], [10], [11], [21]. A quantitative security analysis of the whole CASHMA system was also performed [6]. As this paper focuses on the continuous authentication protocol rather than the CASHMA architecture, we briefly summarize the main threats to the system identified within the project (Section 6.1), while the rest of this section (Section 6.2) focuses on the quantitative security assessment of the continuous authentication protocol.

6.1 Threats to the CASHMA System

Security threats to the CASHMA system have been analyzed both for the *enrollment* procedure (i.e., initial registration of an user within the system), and the *authentica-*

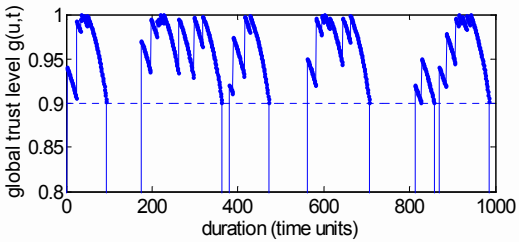

Fig. 10. Global trust level and 40 authentications for a service with high security requirements.

tion procedure itself. We report here only on authentication. The biometric system has been considered as decomposed in functions from [10]. For authentication, we considered collection of biometric traits, transmission of (raw) data, features extraction, matching function, template search and repository management, transmission of the matching score, decision function, communication of the recognition result (accept/reject decision).

Several relevant threats exist for each function identified [9], [10], [11]. For brevity, we do not consider threats generic of ICT systems and not specific for biometrics (e.g., attacks aimed to Deny of Service, eavesdropping, man-in-the-middle, etc.). We thus mention the following.

For the collection of biometric traits, we identified sensor spoofing and untrusted device, reuse of residuals to create fake biometric data, impersonation, mimicry and presentation of poor images (for face recognition). For the transmission of (raw) data, we selected fake digital biometric, where an attacker submits false digital biometric data. For the features extraction, we considered insertion of imposter data, component replacement, override of feature extraction (the attacker is able to interfere with the extraction of the feature set), and exploitation of vulnerabilities of the extraction algorithm. For the matching function, attacks we considered are insertion of imposter data, component replacement, guessing, manipulation of match scores. For template search and repository management, all attacks considered are generic for repositories and not specific to biometric systems. For the transmission of the matching score, we considered manipulation of match score. For the decision function, we considered hill climbing (the attacker has access of the matching score, and iteratively submits modified data in an attempt to raise the resulting matching score), system parameter override/modification (the attacker has the possibility to change key parameters as system tolerances in feature matching), component replacement, decision manipulation. For the communication of recognition result, we considered only attacks typical of Internet communications.

Countermeasures were selected appropriately for each function on the basis of the threats identified.

6.2 Quantitative Security Evaluation

6.2.1 Scenario and measures of interest

For the quantitative security evaluation of the proposed protocol we consider a mobile scenario, where a registered user uses the CASHMA service through a client installed on a mobile device like a laptop, a smartphone or a similar device. The user may therefore lose the device, or equivalently leave it unattended for a time long enough for attackers to compromise it and obtain authentication. Moreover, the user may lose the control of the device (e.g. he/she may be forced to hand over it) while a session has already been established, thus reducing the effort needed by the attacker. In the considered scenario the system works with three biometric traits: voice, face, and fingerprint.

A security analysis on the first authentication performed to acquire the first certificate and open a secure session has been provided in [6]. We assume here that the attacker has already been able to perform the initial authentication (or to access to an already established session), and we aim to evaluate how long he is able to keep the session alive, at varying of the parameters of the continuous authentication algorithm and the characteristics of the attacker. The measures of interest that we evaluate in this paper are the following: i) $P_k(t)$: Probability that the attacker is able to keep the session alive until the instant t, given that the session has been established at the instant t=0; ii) T_k: Mean time for which the attacker is able to keep the session alive.

Since most of the computation is performed server-side, we focus on attacks targeting the mobile device. In order to provide fresh biometric data, the attacker has to compromise one of the three biometric modalities. This can be accomplished in several ways; for example, by spoofing the biometric sensors (e.g., by submitting a recorded audio sample, or a picture of the accounted user), or by exploiting cyber-vulnerabilities of the device (e.g., through a "reuse of residuals" attack [9]). We consider three kind of abilities for attackers: *spoofing,* as the ability to perform sensor spoofing attacks, *hacking* as the ability to perform cyber attacks, and *lawfulness*, as the degree to which the attacker is prepared to break the law.

The actual skills of the attacker influence the chance of a successful attack, and the time required to perform it. For example, having a high *hacking* skill reduces the time required to perform the attack, and also increses the success probability: an attacker having high technological skills may able to compromise the system is such a way that the effort required to spoof sensors is reduced (e.g., by altering the data transmitted by the client device).

6.2.2 The ADVISE [12] formalism

The analysis method supported by ADVISE relies on creating executable security models that can be solved using discrete-event simulation to provide quantitative metrics. One of the most significant features introduced by this formalism is the precise characterization of the attacker (the "adversary") and the influence of its decisions on the final measures of interest.

The specification of an ADVISE model is composed of two parts: an Attack Execution Graph (AEG), describing how the adversary can attack the system, and an adversary profile, describing the characteristics of the attacker. An AEG is a particular kind of attack graph comprising different kinds of nodes: attack steps, access domains, knowledge items, attack skills, and attack goals. Attack steps describe the possible attacks that the adversary may attempt, while the other elements describe items that can be owned by attackers (e.g., intranet access). Each attack step requires a certain combination of such items to be held by the adversary; the set of what have been achieved by the adversary defines the current state of the model. ADVISE attack steps have also additional properties, which allow creating executable models for quantitative analysis. The adversary profile defines the set of items that are initially owned by the adversary, as well as his

proficiency in attack skills. The adversary starts without having reached any goal, and works towards them. To each attack goal it is assigned a payoff value, which specifies the value that the adversary assigns to reaching that goal. Three weights define the relative preference of the adversary in: i) maximizing the payoff, ii) minimizing costs, or iii) minimizing the probability of being detected. Finally, the planning horizon defines the number of steps in the future that the adversary is able to take into account for his decisions; this value can be thought to model the "smartness" of the adversary.

The ADVISE execution algorithm evaluates the reachable states based on enabled attack steps, and selects the most appealing to the adversary based on the above described weights. The execution of the attack is then simulated, leading the model to a new state. Metrics are defined using reward structures [14]. By means of the Rep/Join composition formalism [15] ADVISE models can be composed with models expressed in other formalisms supported by the Möbius framework, and in particular with Stochastic Activity Networks (SAN) [16] models.

6.2.3 Modeling approach

The model that is used for the analysis combines an ADVISE model, which takes into account the attackers' behavior, and a SAN model, which models the evolution of trust over time due to the continuous authentication protocol. Both models include a set of *parameters*, which allow evaluating metrics under different conditions and performing sensitivity analysis. Protocol parameters used for the analysis are reported in the upper labels of Fig. 13 and Fig. 14; parameters describing attackers are shown in Table 1 and their values are discussed in Section 6.2.4.

ADVISE model. The AEG of the ADVISE model is composed of 1 attack goal, 3 attack steps, 3 attack skills, and 5 access domains. Its graphical representation is shown in Fig. 11, using the notation introduced in [12]. The only attack goal present in the model, "RenewSession" represents the renewal of the session timeout by submitting fresh biometric data to the CASHMA server.

To reach its goal, the attacker has at its disposal three attack steps, each one representing the compromise of one of the three biometric traits: "Compromise_Voice", "Compromise_Face", and "Compromise_Fingerprint". Each of them requires the "SessionOpen" access domain,

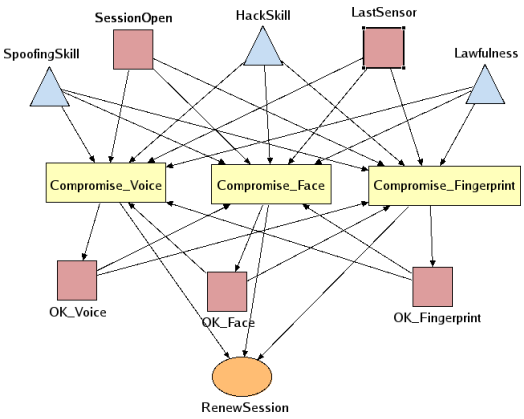

Fig. 11. AEG of the ADVISE model used for security evaluations.

which represents an already established session. The three abilities of attackers are represented by three attack skills: "SpoofingSkill", "HackSkill" and "Lawfulness".

The success probability of such attack steps is a combination of the spoofing skills of the attacker and the false *non-match* rate (FNMR) of the involved biometric subsystem. In fact, even if the attacker was able to perfectly mimic the user's biometric trait, reject would still be possible in case of a false non-match of the subsystem. For example, the success probability of the "Compromise_Voice" attack step is obtained as:

FNMR_Voice*(SpoofingSkill->Mark()/1000.0),

where "FNMR_Voice" is the false non-match rate of the voice subsystem, and SpoofingSkill ranges from a minimum of 0 to a maximum of 1000. It should be noted that the actual value assigned to the spoofing skill is a relative value, which also depends on the technological measures implemented to constrast such attack. Based on the skill value, the success probability ranges from 0 (spoofing is not possible) to the FNMR of the subsystem (the same probability of a non-match for a "genuine" user). The time required to perform the attack is exponentially distributed, and its rate also depends on attacker' skills.

When one of the three attack step succeeds, the corresponding "OK_X" access domain is granted to the attacker. Owning one of such access domains means that the system has correctly recognized the biometric data, and that it is updating the global trust level; in this state all the attack steps are disabled. A successful execution of the attack steps also grants the attackers the "RenewSession" goal. "LastSensor" access domain is used to record the last subsystem that has been used for authentication.

SAN model. The SAN model in Fig. 12 models the management of session timeout and its extension through the continuous authentication mechanism. The evolution of trust level over time is modeled using the functions introduced in Section 4.2; it should be noted that the model introduced in this section can also be adapted to other functions that might be used for realizing the protocol.

Place "SessionOpen" is shared with the ADVISE model, and therefore it contains one token if the attacker has already established a session (i.e., it holds the "SessionOpen" access domain). The extended places "LastTime" and "LastTrust" are used to keep track of the last time at which the session timeout has been updated, and the corresponding global trust level. These values correspond, respectively, to the quantities t_0 and $g(t_0)$ and can therefore be used to compute the current global trust level $g(t)$. Whenever the session is renewed, the extended place "AuthScore" is updated with the global trust level $P(S_k)$ of the subsystem that has been used to renew the session. The extended place "CurrentTimeout" is used to store the current session timeout, previously calculated at time t_0. The activity "Timeout" models the elapsing of the session timeout and it fires with a deterministic delay, which is given by the value contained in the extended place "CurrentTimeout". Such activity is enabled only when the ses-

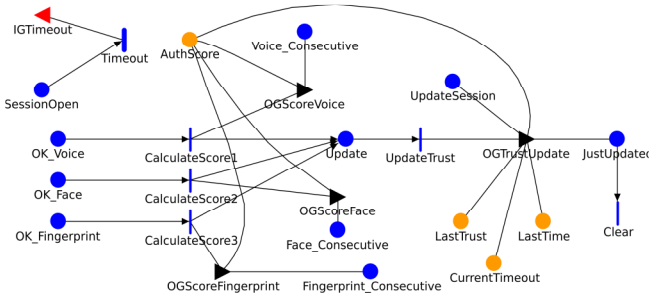

Fig. 12. SAN model for the continuous authentication mechanism.

sion is open (i.e., place "SessionOpen" contains one token). Places "OK_Voice", "OK_Face" and "OK_Fingerprint" are shared with the respective access domains in the ADVISE model. Places "Voice_Consecutive", "Face_Consecutive", and "Fingerprint_Consecutive" are used to track the number of consecutive authentications performed using the same biometric subsystem; this information is used to evaluate the penalty function.

When place "OK_Voice" contains a token, the instantaneous activity "CalculateScore1" is enabled and fires; the output gate "OGSCoreVoice" then sets the marking of place "AuthScore" to the authentication score of the voice subsystem, possibly applying the penalty. The marking of "Voice_Consecutive" is then updated, while the count for the other two biometric traits is reset. Finally, a token is added in place "Update", which enables the immediate activity "UpdateTrust". The model has the same behavior for the other two biometric traits.

When the activity "UpdateTrust" fires, the gate "OGTrustUpdate" updates the user trust level, which is computed based on the values in places "LastTrust" and "LastTime", using (1). Using (3) the current user trust level is then fused with the score of the authentication that is being processed, which has been stored in place "AuthScore". Finally, the new timeout is computed using (4) and the result is stored in the extended place "CurrentTimeout". The reactivation predicate of the activity "Timeout" forces the resample of its firing time, and the new session timeout value is therefore adopted.

Composed model. The ADVISE and SAN models are then composed using the Join formalism [15]. Places "SessionOpen", "OK_Voice", "OK_Face", and "OK_Fingerprint" are shared with the corresponding access domains in the ADVISE model. The attack goal "RenewSession" is shared with place "RenewSession".

6.2.4 Definition of attackers

One of the main challenges in security analysis is the identification of possible human agents that could pose security threats to information systems. The work in [17] defined a Threat Agent Library (TAL) that provides a standardized set of agent definitions ranging from government spies to untrained employees. TAL classifies agents based on their access, outcomes, limits, resources, skills, objectives, and visibility, defining qualitative levels to characterize the different properties of attackers. For example, to characterize the proficiency of attackers in skills, four levels are adopted: "none" (no proficiency), "minimal" (can use existing techniques), "operational" (can create new attacks within a narrow domain) and "adept" (broad expert in such technology). The "Limits" dimension describes legal and ethical limits that may constrain the attacker. "Resources" dimension defines the organizational level at which an attacker operates, which in turn determines the amount of resources available to it for use in an attack. "Visibility" describes the extent to which the attacker intends to hide its identity or attacks.

TABLE 1
ATTACKERS AND THEIR CHARACTERISTICS

	ORG	TMA	GEN	INS
Access	External	External	External	Internal
Limits	Extra-legal, major	Extra-legal, minor	Extra-legal, major	Extra-legal, minor
Resources	Government	Contest	Individual	Organization
Skill-Hack	Operational	Adept	None	Minimal
Skill-Spoofing	Operational	None	None	Minimal
Visibility	Covert	Clandestine	Overt	Clandestine

Agent threats in the TAL can be mapped to ADVISE adversary profiles with relatively low effort. The "access" attribute is reproduced by assigning different sets of access domains to the adversary; the "skills" attribute is mapped to one or more attack skills; the "resources" attribute can be used to set the weight assigned to reducing costs in the ADVISE model. Similarly, "visibility" is modeled by the weight assigned to the adversary in avoiding the possibility of being detected. The attributes "outcomes" and "objectives" are reproduced by attack goals, their payoff, and the weight assigned to maximixe the payoff. Finally, the "limits" attribute can be thought as a specific attack skill describing the extent to which the attacker is prepared to break the law. In this paper, it is represented by the "Lawfulness" attack skill.

In our work we have abstracted four macro-agents that summarize the agents identified in TAL, and we have mapped their characteristics to adversary profiles in the ADVISE formalism. To identify such macro-agents we first have discarded those attributes that are not applicable to our scenario; then we aggregated in a single agent those attackers that after this process resulted in similar profiles. Indeed, it should be noted that not all the properties are applicable in our evaluation; most notably, "objectives" are the same for all the agents i.e., extending the session timeout as much as possible. Similarly "outcome" is not addressed since it depends upon the application to which the CASHMA authentication service provides access. Moreover, in our work we consider *hostile* threat agents only (i.e., we do not consider agents 1, 2 and 3 in [17]), as opposed to non-hostile ones, which include for example the "Untrained Employee".

The attributes of the four identified agents are summarized in Table 1. As discussed in [17], names have the only purpose to identify agents; their characteristics should be devised from agent properties. "Adverse Organization" (ORG) represents an external attacker, with government-level resources (e.g., a terrorist organization or an adverse nation-state entity), and having good proficiency in both

"Hack" and "Spoofing" skills. It intends to keep its identity secret, although it does not intend to hide the attack itself. It does not have particular limits, and is prepared to use violence and commit major extra-legal actions. This attacker maps agents 6, 7, 10, 15, and 18 in [17].

"Technology Master Individual" (TMA) represents the attacker for which the term "hacker" is commonly used: an external individual having high technological skills, moderate/low resources, and strong will in hide himself and its attacks. This attacker maps agents 5, 8, 14, 16, and 21 in [17]. "Generic Individual" (GEN) is an external individual with low skills and resources, but high motivation – either rational or not – that may lead him to use violence. This kind of attacker does not take care of hiding its actions. The GEN attacker maps 4, 13, 17, 19, and 20 in [17]. Finally, the "Insider" attacker (INS) is an internal attacker, having minimal skill proficiency and organization-level resources; it is prepared to commit only minimal extra-legal actions, and one of its main concerns is avoiding him or its attacks being detected. This attacker maps agents 9, 11, and 12 in [17].

6.2.5 Evaluations

The composed model has been solved using the discrete-event simulator provided by the Möbius tool [15]. All the measures have been evaluated by collecting at least 100.000 samples, and using a relative confidence interval of ±1%, confidence level 99%. For consistency, the parameters of the decreasing functions are the same as in Fig. 10 (s = 90 and k = 0.003); *FMR*s of subsystems are also the same used in simulations of Section 5 (voice: 0.06, fingerprint: 0.03, face: 0.05); for all subsystems, the *FNMR* has been assumed to be equal to its *FMR*.

Results in Fig. 13 show the effectiveness of the algorithm in contrasting the four attackers. The left part of the figure depicts the measure $P_k(t)$, while T_k is shown in the right part. All the attackers maintain the session alive with probability 1 for about 60 time units. Such delay is given by the initial session timeout, which depends upon the characteristics of the biometric subsystems, the decreasing function (1) and the threshold g_{min}. With the same parameters a similar value was obtained also in MAtlab simulations described in Section 5 (see Figure 10): from the highest value of $g(u,t)$, if no fresh biometric data is received, the global trust level reaches the threshold in slightly more than 50 time units. By submitting fresh biometric data, all the four attackers are able to renew the authentication and extend the session timeout. The extent to which they are able to maintain the session alive is based on their abilities and characteristics.

The GEN attacker has about 40% probability of being able to renew the authentication and on the average he is able to maintain the session for 80 time units. Moreover, after 300 time units he has been disconnected by the system with probability 1. The INS and ORG attackers are able to renew the session for 140 and 170 time units on the average, respectively, due to their greater abilities in the spoofing skill. However, the most threatening agent is the TMA attacker, which has about 90% chance to renew the authentication and is able, on the average, to extend its

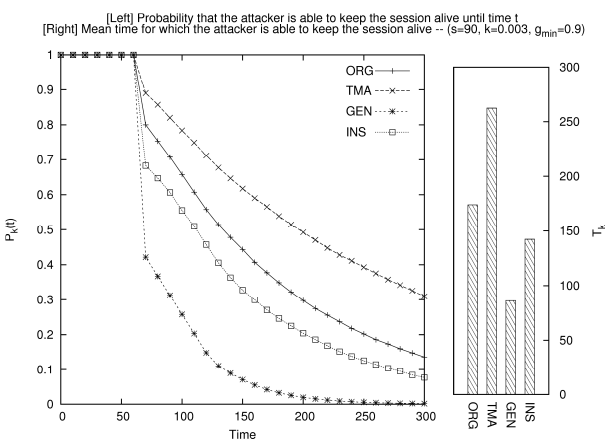

Fig. 13. Effect of the continuous authentication mechanism on different attackers.

session up to 260 time units, which in this setup is more then four times the the initial session timeout. Moreover, the probability that TMA is able to keep the session alive up to 30 time units is about 30% i.e., on the average once every 3 attempts the TMA attacker is able to extend the session beyond 300 time units, which is roughly 5 times the initial session timeout.

Possible countermeasures consist in the correct tuning of algorithm parameters based on the attackers to which the system is likely to be subject. As an example, Fig. 14 shows the impact of varying the threshold g_{min} on the two measures of interest, $P_k(t)$ and T_k, with respect to the TMA attacker. Results in the figure show that increasing the threshold is an effective countermeasure to reduce the average time that the TMA attacker is able to keep the session alive. By progressively increasing g_{min} the measure T_k decreases considerably; this is due to both a reduced initial session timeout, and to the fact that the attacker has less time at his disposal to perform the required attack steps. As shown in the figure, by setting the threshold to 0.95, the probability that the TMA attacker is able to keep the session alive beyond 300 time units approaches zero, while it is over 30% when g_{min} is set to 0.9.

7 Prototype Implementation

The implementation of the CASHMA prototype includes

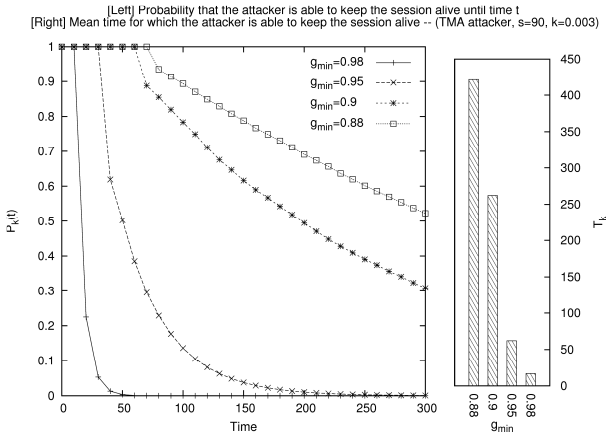

Fig. 14. Effect of varying the threshold g_{min} on the TMA attacker.

face, voice, iris, fingerprint and online dynamic handwritten signature as biometric traits for biometric kiosks and PCs/laptops, relying on on-board devices when available or pluggable accessories if needed. On smartphones only face and voice recognition are applied: iris recognition was discarded due to the difficulties in acquiring high-quality iris scans using the camera of commercial devices, and handwritten signature recognition is impractical on most of smartphones today available on market (larger displays are required). Finally, fingerprint recognition was discarded because few smartphones include a fingerprint reader. The selected biometric traits (face and voice) suit the need to be acquired transparently for the continuous authentication protocol described.

A prototype of the CASHMA architecture is currently available, providing mobile components to access a secured web-application. The client is based on the Adobe Flash [19] technology: it is a specific client, written in Adobe Actions Script 3, able to access and control the on-board devices in order to acquire the raw data needed for biometric authentication. In case of smartphones, the CASHMA client component is realized as a native Android application (using the Android SDK API 12). Tests were conducted on smarthphones Samsung Galaxy S II, HTC Desire, HTC Desire HD and HTC Sensation with OS Android 4.0.x. On average from the executed tests, for the smartphones considered we achieved FMR=2,58% for face recognition and FMR=10% for voice. The dimensions of biometric data acquired using the considered smarthphones and exchanged are approximately 500 KB. As expected from such limited dimension of the data, the acquisition, compression and transmission of these data using the mentioned smarthphones did not raise issues on performance or communication bandwidth. In particular, the time required to establish a secure session and transmit the biometric data was deemed sufficiently short to not compromise usability of the mobile device.

Regarding the authentication service, it runs on Apache Tomcat 6 servers and Postgres 8.4 databases. The web services are, instead, realized using the Jersey library (i.e., a JAX-RS/JSR311 Reference Implementation) for building RESTful Web services.

Finally, the example application is a custom portal developed as a Rich Internet Application using Sencha ExtJS 4 JavaScript framework, integrating different external online services (e.g. Gmail, Youtube, Twitter, Flickr) made accessible dynamically following the current trust value of the continuous authentication protocol.

8 Concluding Remarks

We exploited the novel possibility introduced by biometrics to define a protocol for continuous authentication that improves security and usability of user session. The protocol computes adaptive timeouts on the basis of the trust posed in the user activity and in the quality and kind of biometric data acquired transparently through monitoring in background the user's actions.

Some architectural design decisions of CASHMA are here discussed. First, the system exchanges raw data and not the features extracted from them or templates, while cripto-token approaches are not considered; as debated in Section 3.1, this is due to architectural decisions where the client is kept very simple. We remark that our proposed protocol works with no changes using features, templates or raw data. Second, privacy concerns should be addressed considering National legislations. At present, our prototype only performs some checks on face recognition, where only one face (the biggest one rusting from the face detection phase directly on the client device) is considered for identity verification and the others deleted. Third, when data is acquired in an uncontrolled environment, the quality of biometric data could strongly depend on the surroundings. While performing a client-side quality analysis of the data acquired would be a reasonable approach to reduce computational burden on the server, and it is compatible with our objective of designing a protocol independent from quality ratings of images (we just consider a sensor trust), this goes against the CASHMA requirement of having a light client.

We discuss on usability of our proposed protocol. In our approach, the client device uses part of its sensors extensively through time, and transmits data on the Internet. This introduces problematic of battery consumption, which has not been quantified in this paper: as discussed in Section 7, we developed and exercised a prototype to verify the feasibility of the approach but a complete assessment of the solution through experimental evaluation is not reported. Also, the frequency of the acquisition of biometric data is fundamental for the protocol usage; if biometric data are acquired too much sparingly, the protocol would be basically useless. This mostly depends on the profile of the client and consequently on his usage of the device. Summarizing, battery consumption and user profile may constitute limitations to our approach, which in the worst case may require to narrow the applicability of the solution to specific cases, for example only when accessing specific web sites and for a limited time window, or to grant access to restricted areas (see also the examples in Section 3.2). This characterization has not been investigated in this paper and constitute part of our future work.

It has to be noticed that the functions proposed for the evaluation of the session timeout are selected amongst a very large set of possible alternatives. Although in literature we could not identify comparable functions used in very similar contexts, we acknowledge that different functions may be identified, compared and preferred under specific conditions or users requirements; this analysis is left out as goes beyond the scope of the paper, which is the introduction of the continuous authentication approach for Internet services.

Acknowledgment

This work has been partially supported by the Italian MIUR through the projects FIRB 2005 CASHMA (DM1621 18 July 2005) and PRIN 2010-3P34XC TENACE.

REFERENCES

[1] CASHMA - Context Aware Security by Hierarchical Multilevel Architectures, MIUR FIRB 2005.

[2] L. Hong, A. Jain, and S. Pankanti, "Can Multibiometrics Improve Performance?," *Proc. AutoID'99*, Summit, NJ, pp. 59–64, 1999.

[3] S. Ojala, J. Keinanen, J. Skytta, "Wearable authentication device for transparent login in nomadic applications environment," *Proc. 2nd International Conference on Signals, Circuits and Systems (SCS 2008)*, pp. 1-6, 7-9 Nov. 2008.

[4] BioID, "Biometric Authentication as a Service (BaaS), "BioID press release, 3 March 2011, https://www.bioid.com [online].

[5] T. Sim, S. Zhang, R. Janakiraman, and S. Kumar, "Continuous Verification Using Multimodal Biometrics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 687-700, April 2007.

[6] L. Montecchi, P. Lollini, A. Bondavalli, and E. La Mattina, "Quantitative Security Evaluation of a Multi-Biometric Authentication System," *Computer Safety, Reliability and Security, F. Ortmeier and P. Daniel (eds.), Lecture Notes in Comuter Science, Springer*, vol. 7613, pp. 209-221, 2012.

[7] S. Kumar, T. Sim, R. Janakiraman, and S. Zhang, "Using Continuous Biometric Verification to Protect Interactive Login Sessions," *Proc. 21st Annual Computer Security Applications Conference (ACSAC '05)*, pp. 441-450, 2005. IEEE Computer Society, Washington, DC, USA.

[8] A. Altinok and M. Turk, "Temporal integration for continuous multimodal biometrics," *Multimodal User Authentication*, pp. 11-12, 2003.

[9] C. Roberts, "Biometric attack vectors and defences," *Computers & Security*, vol. 26, Issue 1, pp. 14-25, 2007.

[10] S.Z. Li, and A.K. Jain, *Encyclopedia of Biometrics*, First Edition, Springer Publishing Company, Incorporated, 2009.

[11] U. Uludag, and A. K. Jain, "Attacks on Biometric Systems: a Case Study in Fingerprints," *Proc. SPIE-EI 2004, Security, Steganography and Watermarking of Multimedia Contents VI*, pp. 622-633, 2004.

[12] E. LeMay, W. Unkenholz, D. Parks, C. Muehrcke, K. Keefe and W. H. Sanders, "Adversary-Driven State-Based System Security Evaluation", *Proc. of the 6th International Workshop on Security Measurements and Metrics (MetriSec 2010)*, pp. 5:1-5:9, 2010.

[13] O. Sheyner, J. Haines, S. Jha, R. Lippmann, J.M. Wing, "Automated generation and analysis of attack graphs", *IEEE Symposium on Security and Privacy*, pp. 273- 284, 2002.

[14] D. M. Nicol, W. H. Sanders, K. S. Trivedi, "Model-based evaluation: from dependability to security," *IEEE Trans. Dependable and Secure Computing*, vol. 1 no. 1, pp. 48–65, 2004.

[15] T. Courtney, S. Gaonkar, L. Keefe, E. W. D. Rozier, W. H. Sanders, "Möbius 2.3: An Extensible Tool for Dependability, Security, and Performance Evaluation of Large and Complex System Models," *Dependable Systems & Networks (DSN '09)*, pp. 353-358, 2009.

[16] W. H. Sanders, J. F. Meyer, "Stochastic activity networks: formal definitions and concepts", *Lectures on formal methods and performance analysis*, Springer-Verlag New York, Inc., pp. 315-343, 2002.

[17] T. Casey, "Threat Agent Library Helps Identify Information Security Risks", White Paper, Intel Corporation, September 2007.

[18] A. Ceccarelli, A. Bondavalli, F. Brancati, E. La Mattina, "Improving Security of Internet Services Through Continuous and Transparent User Identity Verification," *Proc. International Symposium on Reliable Distributed Systems (SRDS)*, Irvine, USA, pp. 201-206, October 2012.

[19] Adobe, Products List, http://www.adobe.com/products [online].

[20] T. F. Dapp, "Growing need for security in online banking: biometrics enjoy remarkable degree of acceptance", Banking & Technology Snapshot, DB Research, 8 February 2012.

[21] A.K. Jain, A. Ross, S. Pankanti, "Biometrics: a tool for information security," IEEE Transactions on Information Forensics and Security, vol.1, no.2, pp.125,143, June 2006.

[22] L. Allano, B. Dorizzi, S. Garcia-Salicetti, "Tuning cost and performance in multi-biometric systems: a novel and consistent view of fusion strategies based on the Sequential Probability Ratio Test (SPRT)", Pattern Recognition Letters, Volume 31, Issue 9, pp. 884–890, 2010.

[23] S. Evans and J. Wallner, "Risk-based security engineering through the eyes of the adversary," in Proc. of the 2005 IEEE Workshop on Information Assurance. United States Military Academy, West Point, NY, June 2005, pp. 158–165..

[24] M. Afzaal, C. Di Sarno, L. Coppolino, S. D'Antonio, L. Romano, "A Resilient Architecture for Forensic Storage of Events in Critical Infrastructures," International Symposium on High-Assurance Systems Engineering (HASE), pp. 48-55, 2012.

[25] M. Cinque, D. Cotroneo, R. Natella, A. Pecchia, "Assessing and improving the effectiveness of logs for the analysis of software faults," International Conference on Dependable Systems and Networks (DSN), pp. 457-466, 2010.

[26] N. Mendes, A.A. Neto, J. Duraes, M. Vieira, H. Madeira., "Assessing and Comparing Security of Web Servers," IEEE International Symposium on Dependable Computing (PRDC), pp. 313-322, 2008.

Andrea Ceccarelli received the Bachelor, and Master degree in computer science, and the PhD in Informatics and Automation Engineering at the University of Firenze, Italy, respectively in 2006, 2008 and 2012. He is currently a Research Associate at the same department. He published in International conferences and journals and has served in the Program Committee of International Conferences and Workshops.

Leonardo Montecchi received the Bachelor degree and Master degree in computer science from the University of Florence, Italy, in 2007 and 2010, respectively. He is currently a Ph.D. student in the "Computer Science, Systems and Telecommunications" program at the same university.

Francesco Brancati received the M.S. degree in computer science and the Ph.D in Informatics and Applications from the University of Firenze in 2008 and 2012, respectively. During his Ph.D. Francesco's research interests has been in the design and the experimental evaluation of resilient systems with particular focus on monitoring system uncertainties in time perception. After his Ph.D. He joined Resiltech s.r.l., where he currently coordinates the R&D activities.

Paolo Lollini received the laurea degree, and the PhD degree in Computer Science from the University of Florence, Italy, in 2001, and 2005, respectively. Since 2006, he has been a Research Associate at the Mathematics and Computer Science Dept. of the same University. He is currently Publication Chair of IEEE SRDS 2013, and Local Organizing Chair of SAFECOMP 2014.

Angelo Marguglio graduated as Doctor with laude in Computer Engineering from the University of Palermo, he has been working in the R&D Lab for Engineering Ingegneria Informatica S.p.A. since 2006 as part of the Intelligence Systems Unit. He is currently a Senior Researcher at the same laboratory.

Andrea Bondavalli (M'03) received the M.S. degree in computer science from the University of Pisa, in 1986. He has been a Researcher with the Italian CNR, and is currently a Professor at the University of Florence. He is a Member of the Editorial Board of the IJCCBS journal and the chair of the Steering Committee of the IEEE SRDS. He served as Program Chair and as General Chair of the most important conferences in Depenable Computing and the Conference Coordinator of IEEE DSN 2009.