ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Continuous User Identity Verification Based on Secure Internet Services

[1]R.Sai priyadarshini, [2]A.Swetha and [1]S. Jayanthi

[1]*Agni college of Technology, Anna University, Computer Science and Engineering, S.Jayanthi, Chennai. India*
[2]*Agni College of Technology, Anna University, Computer Science and Engineering, S.Jayanthi, Chennai. India*

ARTICLE INFO

Received 10 March 2015
Received in revised form 20 March 2015
Accepted 25 March 2015
Available online 10 April 2015

Keywords:
Security, Web Servers, Mobile Environments, Authentication

ABSTRACT

In banking applications, user authentication is traditionally based on username and password, come forth biometric solutions allow biometric data during session establishment. But in Unimodal biometric approach still a single verification is considered and the identity of the user is permanent during the entire session. A secure protocol is defined for constant authentication through continuous user verification. Biometric techniques offer solution for secure and trusted authentication. The user's identity has been verified, the system resources are available for fixed period of time and identity of the user is constant during whole session. The proposed system detects misuses of computer resources and prevents malicious activities based on multi-modal biometric continuous authentication. Biometric and user data's are stored in smart phones.

To Cite This Article: R.Sai priyadarshini,A.swetha and S. Jayanthi, Continuous User Identity Verification Based on Secure Internet Services ***Aust. J. Basic & Appl. Sci.,*** *9(15): 1-6, 2015*

INTRODUCTION

Session management in distributed Internet services is traditionally based on username and password, explicit logouts and mechanisms of user session expiration using classic timeouts. Session time out may occur during unperformed working sessions or it expires when user is in idle activity period. Security of web-based application is very important as there is increase in complexity of cyber attacks. Biometric application provides more security for authentication process than proving the username and password. Bio-metric user authentication is typically formulated as a "single shot" 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 is sufficient, and that the identity of the user is constant during the whole session.

For example: consider a user is already logged into the critical service and then and leaves the PC in the work area as a while. This problem is even risky when it is used in mobile phones in public and crowded areas as the device can be lost while the session is active. The users are authenticated and it can be misused easily. To detect the misuses of the computer resources and prevent that from the unauthorized user replaces an authorized one by providing the solution based on the multimodal biometric continuous authentication turning the user authentication as the continuous process rather than the one time occurrence. 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. Face can be acquired by using the front camera but not purposely for the acquisition of the biometric data for example the user may be reading a textual SMS or watching a movie on the mobile phone. Key-stroke data can be acquired whenever the user types on the keyboard, for example when writing an SMS, chat-ting, or browsing on the Internet. 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) 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., smart phones and Desktop PCs.

CASHMA is used for highly secure, a user session is a *continuous sequential multi-modal*

Corresponding Author: R.Sai priyadarshini, Agni college of Technology, Anna University, Computer Science and Engineering, S.Jayanthi, Chennai. India

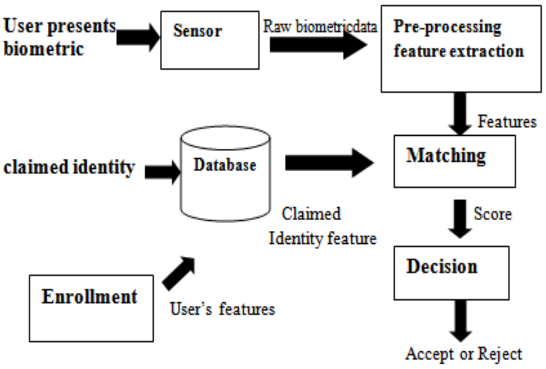
biometric authentication protocol which computes and refreshes session time outs based on the client. In the CASHMA context, each subsystem comprises of all the hardware/software elements necessary to acquire and verify the authenticity of one biometric trait, including sensors, comparison algorithms and all the facilities for data transmission and management.

2. *Preliminaries:*

2.1 Continuous Authentication:

Continuous Authentication (CA) systems represent a new generation of security mechanisms that continuously monitor user behavior and use this as basis to re-authenticate periodically throughout a login session. A problem in continuous authentication is that it aims to tackle the user device (smart phone, laptop, etc.) when it is used, stolen or forcibly taken after the user has already logged into the services. The proposed approach assumes that first the user logs in using a strong authentication procedure, and then a continuous verification process is started based on multi-modal biometric. Similarly, when a multi-modal biometric verification system is presented, it 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.

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 Mobius multi-formalism , and iii) the ability to define ad-hoc metrics for the system we were targeting.



2.2Comparing the Fusion Methods:

2.2.1 Legitimate user using the system:

The biometric observations for 15 minutes. The individual probabilities are not consistently high; they occur in a sporadic manner. This means that any value for the threshold T safe will result in significant False Accept and False Reject rates. In continuous verification, a False Accept is a security breach, while a False Reject inconveniences the legitimate user because he must reauthenticate himself. Ideally, Psafe should not fluctuate, but be equal to 1 as long as observations are available. Of the four fusion methods, Holistic Fusion comes closest to this ideal It computes a Psafe value close to 1, except for periods in which there are no observations from both modalities (around 300s and 600s). At such time, Psafe decreases gradually according to the decay function. By comparison, the Psafe computed by Naïve Integration fluctuates wildly because only a single modality is used any at time. Again, this means no Tsafe value will make both FRR and FAR small. As for Modality-first and temporal first Integration, the plots are similar. The Psafe values are not close to 1. Moreover, in the absence of observations, Psafe drops abruptly to zero, resulting in sudden lock outs. From these plots, it is clear that Holistic Fusion is superior to the other fusion methods.

2.2.2Imposter taking over the system:

The observations when an imposter takes over the system at some time instant (at around 38s). The probabilities of individual biometrics as well as Psafe for all integration methods drop to near zero after the attack. The goal here is to detect the attack as soon as possible so that damage to the system is minimized. Both Holistic Fusion and Naive Integration detect this situation sooner than the other two methods. However, Psafe for Naive Integration does not remain consistently low; it fluctuates widely. This implies that FAR > 0 for most values of Tsafe. For Modality-first and Temporal-first Integration, the system takes longer to detect the imposter (when Tsafe ¼ 0:5). Choosing a larger value for Tsafe can reduce the time to detection, but at the expense of a higherFRR.

2.2.3Imposter successful in faking one of the biometric (Partial impersonation):

The individual probabilities contradict each other, and result in wildly fluctuating plots in both Holistic Fusion and Naïve Integration. This gives us a way to detect partial impersonation: We may just take two thresholds, one high and one low (say, 0.8 and 0.2) and simply count the number of times within a fixed time interval that Psafe jumps between these thresholds. However, comparing we see that Naive Integration cannot distinguish between partial impersonation and the legitimate user. Fluctuating Psafe values seem to be an inherent property of Naive Integration. The plots for Modality-first Integration are relatively flat and are in fact similar to those in (except when there are completely no biometric observations). Again, this means these two methods cannot distinguish partial impersonation from legitimate usage. Only Holistic Fusion provides a way to detect partial impersonation that is different from detecting the real user. We remark that this fluctuating behavior of Holistic Fusion may be intuited from examining.

2.3 Basic Definitions:

Given *n* unimodal biometric sub-systems *Sk*, with *k= 1, 2, …, n* that are able to decide independently on the authenticity of a user, the False Non-Match Rate, *FNMRk*, 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). It is the probability that the unimodal system *Sk*wrongly rejects a legitimate user. Conversely, the False Match Rate, *FMRk*, is the probability that the unimodal subsystem *Sk*makes a *false match* error 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 sys-tem would lead to authenticate a non legitimate user. To simplify the discussion but without losing 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 *user trust level g*(*u, t*) indicates the trust placed by the CASHMA authentication service in the user *u* at time *t*, 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*(*Sk=1,…n, t*) and of the user trust level *g*(*u, t*).

The *trust threshold gmin*is a lower threshold on the glob-al trust level required by a specific web service; if the resulting global trust level at time *t* is smaller than *gmin*(i.e., *g*(*u*,*t*) <*gmin*), the user *u* is not allowed to access to the ser-vice.

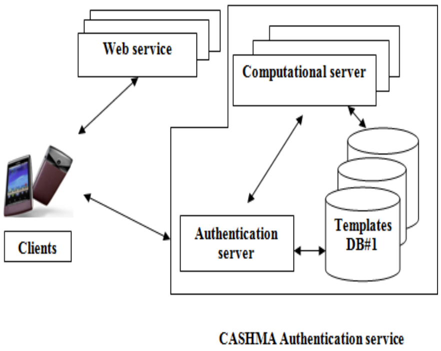
Otherwise if *g*(*u*,*t*) ≥ *gmin* the user *u* is authenticated and is granted access to the service.

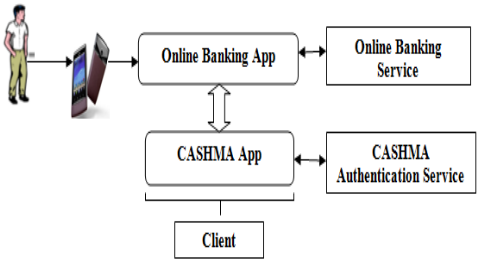
2.4 Face Verifier:

Our Face Verifier is also based on intra and interclass pdfs, except that the score s is now an image distance, rather than asimilarity measure. As in the Fingerprint Verifier, we proceeding two stages. In the training stage, we capture 500 images of each user under varying head poses, using a Canon VCC4video camera and the Viola-Jones face detector. The images are resized to 28 _ 35.

3. *The Cashma Framework:*

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 en-rolled users, and iii) *databases of templates* that contain the biometric templates of the enrolled users. Users have to be registered to the CASHMA authentication ser-vice, expressing also their trust threshold.





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.

4.1 Representation of the Protocol:

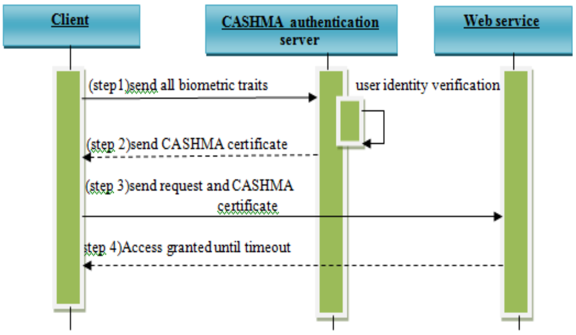
The proposed protocol requires a sequential multi-modal biometric system composed of *n* Unimodal biometric sub-systems that are able to decide independently on the authenticity of a user. For example, these subsystems can be one subsystem for keystroke recognition and one for face recognition.

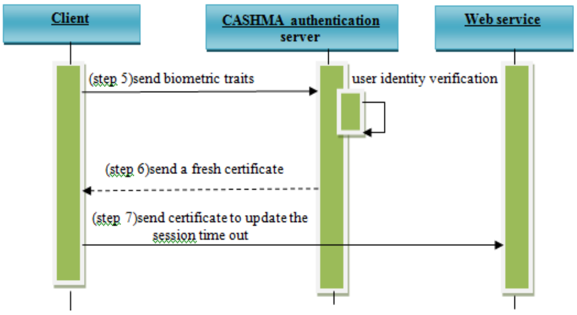
The idea behind the execution of the protocol is that the client continuously and transparently acquires and transmits evidence of the user identity to maintain access to a web service. The main task of the proposed protocol is to create and then maintain the user session adjusting the session timeout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

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.

Initial phase:


Client
CASHMA authentication server
Web service
(step1)send all biometric traits
user identity verification
(step 2)send CASHMA certificate
(step 3)send request and CASHMA certificate
step 4)Access granted until timeout


Client
CASHMA authentication server
Web service
(step 5)send biometric traits
user identity verification
(step 6)send a fresh certificate
(step 7)send certificate to update the session time out

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 *gmin*), new or additional biometric data are requested (back to step 1) until the minimum trust threshold *gmin*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.

4.2 Maintenance Phase:

When at time *ti*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 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. If verification is successful, the CASHMA authentication server applies the algorithm to adaptively compute a new timeout of length *Ti*, the expiration time of the session at time *Ti*+ *ti*and then it creates and sends a new certificate to the client. 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 *ti*+ *Ti*

4.3 Identification:

Given an input biometric sample, identification determines if the input biometric sample is associated with any of a large number (e.g., millions) of enrolled identities. Typical identification applications include welfare disbursement, national ID cards, border control, voter ID cards, driver's license, criminal investigation, corpse identification, parenthood determination, missing children identification, etc. These identification applications require a large sustainable throughput with as little human supervision as possible

5. *Conviction Levels And Timeout Computation:*

Let us assume that the initial phase occurs at time *t*0 when bi-ometric data is acquired and transmitted by the CASHMA application of the user *u*, and that during the maintenance phase at time *ti*>*t*0 for any *i*=1, … *m* new biometric data is acquired by the CASHMA application of the user *u*.

5.1Computation of Trust in the Subsystems:

The subsystem trust level could be simply set to the static value *m*(*Sk*, *t*)=1-*FMR*(*Sk*) for each unimodal subsystem *Sk*and any time *t*. In the initial phase *m*(*Sk*, *t*0) is set to *1-FMR*(*Sk*) for each subsystem *Sk*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*·*h*

Where *x* is the number of consecutive authentication at-tempts using the same subsystem and *h>0* is a parameter used to tune the penalty function. If the same sub-system is used in consecutive authentications, the subsystem trust level is a multiplication of i) the subsystem trust level *m*(*Sk*, *ti*-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(*Sk*, *ti*) = *m*(*Sk*, *ti*-1) · (*penalty* (*x*, *h*))-1.

5.1Computation of Trust in the User:

During the maintenance phase, the user trust level is computed for each received fresh biometric data. The user trust level at time *ti*is given by:

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

Value Δ*ti*=*ti*-*ti*-1 is the time interval between two data transmissions; *trust*(*ti*-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 .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.

5.2Merging User Trust and Subsystems Trust: the Global 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:

$$\mathbf{trust(t_0)} = \mathbf{1} - \boldsymbol{\pi_{k-1,\dots,n}}(\mathbf{1} - \boldsymbol{m(S_K, t_0)})$$

Equation (2) includes the subsystem trust level of all subsystems used in the initial phase. We remind that for the first authentication *m*(*Sk, t*0) is set to 1-*FMR*(*Sk*).

***trust*(*ti*) = 1 - (1 - *g*(*ti*)) (1 - *m*(*Sk, ti*)) =**
= *g*(*ti*) + *m*(*Sk, ti*) - *g*(*ti*) *m*(*Sk, ti*) =
= *g*(*ti*) + (1 - *g*(*ti*)) *m*(*Sk, ti*).

5.3Computation of the Session Timeout:

The last step is the computation of the length *Ti*of the session timeout. This value represents the time required by the global trust level to decrease until the trust thresh-old *gmin.* Starting from a given instant of time *ti*, we consider *ti+1* as the instant of time at which the global trust level reaches the minimum threshold *gmin*, i.e., g(*ti+1*)=*gmin*. The timeout is then given by *Ti*=Δ*ti=ti+1-ti*. To obtain a closed formula for such value we first instantiated (1) for i+1 i.e., we substituted *trust*(*ti*-1) with *trust*(*ti*), Δ*ti*= *Ti*and *g*(*ti*) = *gmin*. By solving for *Ti*, 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 *Ti*of the timeout at time *ti* for the user *u* is:

$$\boldsymbol{T_i} = \begin{Bmatrix} \mathbf{tan}\left(\frac{\mathbf{g_{min}}\cdot\left(\mathbf{arctan}(-\mathbf{s.k}) - \frac{\pi}{2}\right)}{\mathbf{trust}(\mathbf{t_i})} + \frac{\pi}{2}\right)\cdot\frac{\mathbf{1}}{\mathbf{k}} + \boldsymbol{s} \\ \mathbf{0} \end{Bmatrix} \mathbf{if T_i} >$$

0

Conclusion:

The continuous authentication process improves the user authentication in more secure manner and increase the usability of user session, where the biometric data acquired transparently through monitoring the user's action. The client is very simple. And the protocol works with no changes using features, templates or raw data. 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

REFERENCES

Altinok, A. and M. Turk, 2003. "Temporal integration for continuous multi-modal biometrics," *Multimodal User Authentication*, pp: 11-12.

CASHMA, 2005. Context Aware Security by Hierarchical Multilevel Architectures, MIUR FIRB.

Roberts, C., 2007. "Biometric attack vectors and defenses," *Computers & Security*, 26(1): 14-25.

LeMay, E., W. Unkenholz, D. Parks, C. Muehrcke, K. Keefe and W.H. Sanders, 2010.

"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.

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

Hong, L., A. Jain and S. Pankanti, 1999. "Can Multibiometrics Improve Per- formance?," Proc. AutoID'99, Summit, NJ, 59–64.

Ojala, S., J. Keinanen, J. Skytta, 2008. "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.

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

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

Sim, T., S. Zhang, R. Janakiraman and S. Kumar, 2007. "Continuous Verification Using Multimodal Biometrics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(4): 687-700

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