Continuous Biometric Verification for Non-Repudiation of Remote Services

E. Schiavone
Department of Mathematics and Informatics, University of Florence
Viale Morgagni 65, 50134, Florence
Italy
enrico.schiavone@unifi.it

A. Ceccarelli
Department of Mathematics and Informatics, University of Florence
Viale Morgagni 65, 50134, Florence
Italy
andrea.ceccarelli@unifi.it

A. Bondavalli
Department of Mathematics and Informatics, University of Florence
Viale Morgagni 65, 50134, Florence
Italy
bondavalli@unifi.it

ABSTRACT

As our society massively relies on ICT, security services are becoming essential to protect users and entities involved. Amongst such services, non-repudiation provides evidences of actions, protects against their denial, and helps solving disputes between parties. For example, it prevents denial of past behaviors as having sent or received messages. Noteworthy, if the information flow is continuous, evidences should be produced for the entirety of the flow and not only at specific points. Further, non-repudiation should be guaranteed by mechanisms that do not reduce the usability of the system or application. To meet these challenges, in this paper, we propose two solutions for non-repudiation of remote services based on multi-biometric continuous authentication. We present an application scenario that discusses how users and service providers are protected with such solutions. We also discuss the technological readiness of biometrics for non-repudiation services: the outcome is that, under specific assumptions, it is actually ready.

KEYWORDS

Non-repudiation, biometrics, security, authentication, continuous authentication, protocol, biometric signature

1 INTRODUCTION

Information and Communication Technology (ICT) pervades modern society to the extent that we massively rely on it from private life to business. Today, users and operators can share confidential data, perform financial transactions, or remotely execute critical operations in real-time. However, the need for security has gone hand in hand with the technological progress, and our reliance on ICT strictly depends on it.

Between others, authentication and non-repudiation are two of the security services that may be considered essential in many contexts and applications.

Authentication is the process that provides assurance in the claimed identity of an entity [7]. Traditionally, it is a verification process based on pairs of username and password, and it is performed as a single-occurrence during the login phase. However, some critical ICT systems and applications need to be secured for the entire session, possibly without overly disturbing the user. Therefore, solutions based on *biometric continuous authentication* have been studied in literature [16]-[21]. Those approaches shift user identity verification from a single-occurrence to a continuous process; relatively recent solutions are able to transparently acquire biometric data to perform authentication without significantly reducing the system usability [3], [17], [18].

The ability to undeniably demonstrate that users or entities requested specific services or performed certain actions is also useful. In fact, when a dispute arises or an error occurs people may attempt to deny their involvement and to repudiate their behavior. For instance, customers may disclaim a withdrawal from their account, or a payment with their own credit card, that they actually did. According to The New York Times, 0.05 percent of MasterCard transactions worldwide are subjects of disputes, that probably means around 15 million questionable charges per year [22].

In addition to users denying their usage of a service, there have been cases of malicious operators or disreputable service providers. One example is a fraudulent or inaccurate web service for online stock trading: a broker is instructed by a customer to buy or sell stocks, or to follow a particular investment strategy, but retards the process in order to help other investors have their trades executed quickly [39], causing a consistent loss of money for the client. Then, the trader denies the involvement.

The two above are both examples of *repudiation,* which can be defined as the denial of having participated in all or part of an action by one of the entities involved [4]. Consequently, *non-repudiation* is the ability to protect against such denial. A non-repudiation mechanism has to provide evidences in order to clarify responsibilities and guarantee the establishment of the facts even in front of a court of law. Therefore, a non-repudiation service can be useful both as a mean to obtain accountability as well as a deterrent for deliberate misbehaviors. In the two situations reported above, a non-repudiation mechanism would prevent that the fraudulent customer succeeds in denying a transaction or would help an innocent client to protect his investments.

In literature, the problem of repudiation has been mainly tackled in the electronic transactions context. A transaction is a *one-shot* information exchange which can be defined as transferring of a message from A to B [2]. Transactions' security is useful in digital contract signing, e-commerce, or electronic voting, in which communicating parties may try to cheat each other. The issue of transactions' non-repudiation is usually addressed exploiting *digital signatures* and many protocols have been proposed so far [1], [2]. In order to perform a digital

ARES '17, August 29-September 01, 2017, Reggio Calabria, Italy

ACM ISBN 978-1-4503-5257-4/17/08..$15.00
http://dx.doi.org/10.1145/3098954.3098969

signature, users need to keep secret their private key. It can be stored on a computer, or more frequently on a smart card or USB device, and protected by a secret (password, or PIN). Items and secrets must not be stolen, lost or forgotten. But what happens if the information which needs to be signed is not a single transaction but a *continuous information flow*? For instance, the access to the private area on a web service for online banking, VoIP [23], [24] calls in case of verbal contracts signing, etc. may also benefit from non-repudiation. Asking for the password or secret for each single transaction would drastically reduce the usability of the service, and may end up frustrating the user. Conversely, requesting the secret only at the first login does not guarantee authenticity of the user for the whole session. For applications in which the communication is confidential, or involves information regarding assets which need to be protected with high assurance, authentication has to be repeated frequently. Otherwise, insiders may be able to interfere, taking advantage of the parties, or causing errors and consequent disputes hard to be solved.

In those cases, new solutions are necessary in order to provide a non-repudiation service for the entire duration of the session: there is a need for a *continuous non-repudiation* service. Two research questions arise here. Can continuous authentication mechanisms be complemented with non-repudiation? Does biometric authentication provide sufficient and undeniable evidence of user's participation in an action?

Our contribution. In this paper we address the problem of continuous non-repudiation through multi-biometric continuous verification of identities, searching an answer to the above research questions.

We choose the approach of [3] for multi-biometric continuous authentication as a starting point, because it is sufficiently general and is one of the few solutions in literature which improve security of the user session without reducing usability [38]. The idea is to introduce in its architecture and in its protocol the modifications needed to provide also continuous non-repudiation.

We propose two alternative solutions: one called *DS-CNR (Digital Signature based Continuous Non-Repudiation)*, and another called *BS-CNR (Biometric Signature based Continuous Non-Repudiation)*. They mainly differ in the generation and handling of the cryptographic keys and in the underlying architecture. We assume that the communications between the entities are encrypted, and the involved third parties are trustable. Under the stated assumptions, both solutions offer a non-repudiation service for a continuous information flow scenario between a client and a remote internet service, protecting both parties for the whole duration of the session.

We also show that in literature algorithms already exist that possess high verification accuracy. In our opinion, if those algorithms are properly integrated in a multi-biometric system, and possibly coupled with other security mechanism, biometrics is ready to offer non-repudiation.

The rest of the paper is organized as follows. Section 2 introduces the background of our work, presenting the basic definitions related to the subjects of non-repudiation and biometric authentication. In Section 3, we first discuss the possibility of providing non-repudiation through biometrics, surveying related works which show that it is still an open question; then we illustrate the CASHMA multi-biometric continuous authentication system which is our reference to build the solutions presented in this paper. Section 4 and Section 5 describe the protocols, the architectures and exposes some security considerations of DS-CNR and BS-CNR respectively. Section 6 presents a sample scenario in which the solutions can be applied; then discusses the impact of False Acceptance Rate (FAR) for non-repudiation in that case study, and analyzes the security provided by BS-CNR solution for it. Finally, in Section 7 we draw conclusions and discuss future works.

2 BACKGROUND

In this section, we present some basic concepts dealing with non-repudiation, its goal and the standard mechanism to provide it. We also give preliminary definitions about biometric authentication systems that will be used through the remaining of the paper to describe the protocols and the related architectures.

2.1 Basics on Non-Repudiation

Non-repudiation provides the capability to determine whether a given individual or entity took a particular action [5].

Typically, non-repudiation protects individuals against: (i) authors, repudiating having authored particular documents; (ii) messages senders, denying having transmitted messages; (iii) messages receivers, denying messages reception; or (iv) signatories, repudiating their signature on documents. Thus, non-repudiation services can be used to determine if information originated from a particular individual, or if an individual took specific actions (e.g., send an email, sign a contract, approve a procurement request) or received specific information [5].

The goal of the non-repudiation service is to generate, collect, maintain, make available and validate *evidence* concerning a claimed event or action in order to resolve disputes about the occurrence or not occurrence of the event or action. Such evidence is an essential object, and may be produced either directly by an *end entity* or involving a *Trusted Third Party* (TTP) [6]. In order to obtain non-repudiation services, organizations employ various techniques or mechanisms; the most common is digital signature [25].

Regarding transactions, assuming that an entity A wishes to send a message M to entity B, typical disputes that may arise are (Figure 1):

- *Repudiation of Origin*. B claims that it received M from A, while A denies sending it;
- *Repudiation of Delivery*. A claims that it has sent M to B, while B denies having received it.

Then, we also mention two additional disputes that may exist in case a delivery authority is involved: *Repudiation of Submission* and *Repudiation of Transmission* [6].

According to the existing standard, two different mechanisms for non-repudiation are distinguished [6]. An NRO (Non-Repudiation of Origin) token is used to provide

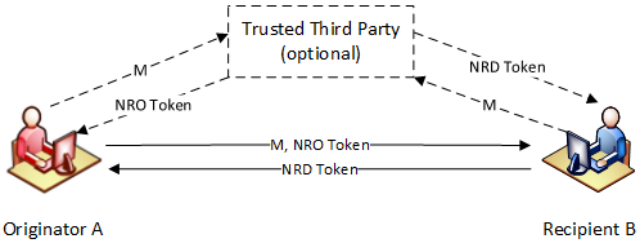

Figure 1: Standard non-repudiation approaches for the transaction scenario, image adapted from [6].

protection against A's false denial of having originated the message. It is generated by A (or by the TTP), sent by A to B, and stored by B after verification of its validity. An NRD (Non-Repudiation of Delivery) token, instead, is used to provide protection against the B's false denial of having received and recognized the content of the message. It is generated by B (or by the TTP), sent to A, and stored by A after verification of its validity. If TTP is involved (optional), it must keep all NRO tokens generated and record whether or not each of NRO token is used to generate a NRD token.

For electronic transactions, the evidences (non-repudiation tokens) are created using digital signatures. A knows its own public key certificate and the associated private key, B knows its own public key certificate and associated private key, and the corresponding public key certificates are available to all the entities concerned [6].

2.3 Basics on Biometric Authentication

A *biometric authentication system* is a system for identity verification of individuals based on their biometric characteristics (also called *traits*). The authentication process consists of two steps: *registration* and *verification*.

During the registration (also called *enrollment*), one or multiple user's biometric traits are presented. The system extracts a digital reference from those characteristics, and generates a *template*, which is saved in a DB [8].

Verification is the validation of a user's identity obtained by comparing the captured biometric with the template(s). Thus, an individual who wants to be recognized claims an identity, for instance with a user name, and the system conducts a one-to-one comparison to determine whether the claim is true or not [8].

A biometric system can be *unimodal* if the verification exploits only one type of characteristics, or *multimodal* (also referred as multi-biometric) if it uses multiple sources of biometric information. A multimodal biometric system can be obtained integrating two or more unimodal *subsystems*, and the fusion can happen at different levels of the verification process [8]. Typically, subsystems 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.

The usage of multiple traits can avoid authenticating an impostor in case a single biometric trait is compromised. In addition, it is well-known that using multiple biometric characteristics combined with an appropriate rule can yield a higher performance than using only one trait [28], [29]. We refer to performance of a biometric system as the achievable recognition accuracy and speed, the resources required to achieve them, and environmental factors that affect them [8].

During the verification, the features extracted from the new traits are compared with the stored templates to generate a matching score, which is exploited to decide on a user's identity. However, the decision can sometimes be wrong, and the error can belong to two categories: false accept or false reject. Thus the two main types of errors metrics are: *False Acceptance Rate (FAR)*, that is, the proportion of verification attempts with wrongful claims of identity that are incorrectly confirmed, and *False Rejection Rate (FRR)*, that is the proportion of verification attempts with truthful claims of identity that are incorrectly denied. FAR and FRR are generally the basic measures of the accuracy of a biometric system [9].

A multi-biometric continuous authentication system integrates two or more subsystems that iteratively and transparently acquire biometric data used for the continuous verification of the user identity.

We define the *subsystem trust level* $m(S_k,t)$ as the probability that the unimodal subsystem S_k at time t does not authenticate an impostor (a non-legitimate user) considering the quality of the sensor, the accuracy of the underlying verification algorithm (i.e., FAR_k , FRR_k), and the risk that the subsystem is intruded or the biometric trait is forged [3].

The *user trust level* $g(u, t)$ indicates the trust placed by the 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 interaction with a sensor) and the time since last acquisition of biometric data [3].

We define the trust level $trust(u,t)$ as a numeric value, computed by continuously evaluating the trust both in the user and the (biometric) subsystems used for acquiring biometric data. It describes the belief that at time t the user u is actually a legitimate user of the system, considering the combination of all subsystems trust levels and of the user trust level [3].

Finally, the trust threshold $trust_{min}$ is a lower threshold on the trust level required by a specific application or web service: the higher is $trust_{min,}$ the higher are the security requirements. If the resulting trust level at time t is smaller than $trust_{min,}$ (i.e., $trust(u,t) < trust_{min}$), the user u is not allowed to access to the service. Otherwise if $trust(u,t) \geq trust_{min}$ the user u is authenticated and is granted access to the service [3].

3 RELATED WORKS

This section presents an analysis of non-repudiation through biometrics, showing that it is an open question; then it describes the CASHMA multi-biometric authentication system that we use as reference to build the solutions presented in this paper.

3.1 Multi-Biometric Verification for Non-Repudiation of Remote Services

For biometrics, as for all the other authentication mechanisms, non-repudiation depends on [10]: (i) the ability of the selected traits to discriminate between individuals; (ii) the strength of binding between the trait and the individual in question; (iii) technical and procedural vulnerabilities that could undermine the intrinsic strength of the binding. The discrimination capabilities of biometrics depend on the technology used and on other application-related factors, that are quantified in terms of error rates (FAR and FRR). According to [9], unlike passwords and tokens, biometrics -because of its strong binding to a specific person- is the only authentication factor capable of guaranteeing that authentication cannot subsequently be denied by a user.

Despite biometric traits are sometimes presented in the computer security literature as an authentication factor that may solve the repudiation problem [9], [10], other works like

[11] and [12] draw a completely different conclusion: according to their authors, biometrics is not a security mechanism able to provide non-repudiation. Analyzing the state of the art, we can state that answers to this research question are contradictory.

However, the situation changes [13], [27], [30] if biometric authentication is coupled with another security mechanism like digital signature, which is commonly considered as the standard approach to achieve non-repudiation [6]. In fact, public key infrastructure, or PKI, and biometrics can well complement each other in many security applications, giving birth to biometric cryptosystems [12], [15] and to the so-called *biometric signature* [27], [30].

Biometric signature is defined as the process to derive a private key from a biometric trait and use the private key to sign an e-document. According to [13], this eliminates the problem of vulnerability of private key storage, which resolves the key management issue. The dynamically generated private key lets the user to perform signatures without carrying a disk or smart card [13]. One example is [30], in which the authors propose a fingerprint based signature scheme that uses a biometric trait to generate a key string, and then exploits the string to create a public key and the corresponding private key.

The analysis of the state of the art shows that no solutions capable of providing multi-biometric continuous non-repudiation of remote services exist. To our knowledge, our paper is the first in this field. It is based on:

- *Multi-biometrics*, which as discussed in Section 2.3 solves the problem of authentication factor loss or steal. It can also avoid authenticating an impostor in case a single biometric trait is forged or compromised, and yield a higher performance than using only one trait.
- *Continuous authentication*, to guarantee the actual presence of the user of the system/device. Noteworthy, we exploit authentication as a requirement for non-repudiation: our solution binds the generated evidence of an action to the user identity only if identity verification is successful.
- *Digital signature* (for both the solutions we propose in the paper) and *biometric signature* (for the BS-CNR), continuously applied for the whole duration of session in which a user remotely accesses an Internet service.

3.2 The CASHMA Approach for Continuous Multi-Biometric Authentication

The CASHMA [3] approach for multi-biometric continuous authentication is our starting point. The motivation of this choice is the generality of the solution: it is applicable to several kinds of internet services. Moreover, while alternatives do exist in literature [16] - [21], CASHMA is one of the few approaches which realizes remote and transparent multi-biometric authentication improving security of the user session without significantly reducing usability [38].

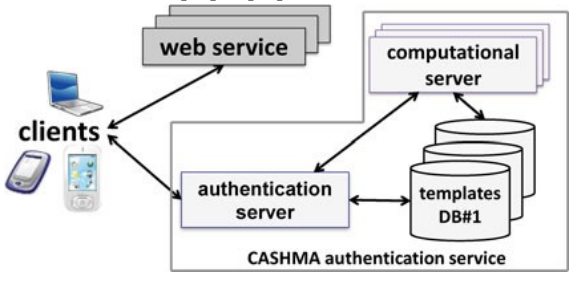

Figure 2: Overall view of the CASHMA architecture [3].

The protocol computes an adaptive timeout (of length T_i) on the basis of the trust posed in the user activity –considering *trust (u,t)* and $trust_{min}$-, and in the quality and kind of biometric data transparently acquired during their activity. The overall system, shown in Figure 2, is composed of the CASHMA authentication service, the clients and the web services, connected through secure communication channels.

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 subscribed to 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.

Clients are the users' devices that acquire the biometric 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.

The CASHMA authentication server is in charge to transmit a certificate to the client. The certificate is composed by the following information: i) *Timestamp* and *sequence number* useful to univocally identify each certificate, and to protect from replay attacks; ii) *ID* is the user ID, e.g., a number; iii) *Decision* represents the outcome of the verification procedure carried out on the server side; iv) *Expiration time* of the session - the absolute instant of time at which the session should expire-, dynamically assigned by the CASHMA authentication server.

The execution of the protocol is composed of two consecutive phases: the initial phase (see Figure 3, [3]), and the maintenance phase.

Initial phase. This phase is structured as follows:

Step 0 - 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.

Step 1 - 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. The application explicitly indicates to the user the biometric traits to be provided and possible retries.

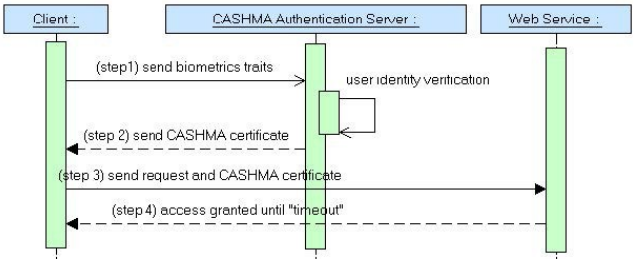

Figure 3: Initial phase of CASHMA protocol in case of successful user authentication [3].

Step 2 - 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 trust level is below the trust threshold $trust_{min}$), new or additional biometric data are requested (back to Step 1) until the minimum trust threshold $trust_{min}$ 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, sets the expiration time at T_0+t_0, creates the CASHMA certificate and sends it to the client.

Step 3 - The client forwards the CASHMA certificate to the web service coupling it with its request.

Step 4 - The web service reads the certificate and authorizes the client to use the requested service until expiration time.

The *maintenance phase* is composed of three steps, analogous to Step 1, Step 2 and Step 3, repeated iteratively [3].

All the channels between the components of CASHMA architecture are encrypted with *Secure Sockets Layer* (SSL/TLS). Thus, the communication is protected against replay attacks using the *Message Authentication Code* (MAC) [31]. Biometric data, even if it is transmitted in raw format from client to the CASHMA authentication service, is not stored on the client: the templates are stored on the CASHMA authentication service side (Figure 2) [3].

4 DS-CNR: DIGITAL SIGNATURE FOR CONTINUOUS NON-REPUDIATION

In order to continuously provide non-repudiation, our idea is to introduce digital signature in the biometric continuous authentication system designed in [3]. We believe that biometrics and digital signature are two security mechanisms that well complement each other. In fact, biometric continuous authentication contributes to strengthen authentication. Vice versa, as discussed in Section 3.1, it's not so clear if, and to which extent, biometric authentication provides non-repudiation if employed alone. As remark, the qualitative risk assessment of CASHMA system conducted in [14] highlighted that the introduction of digital signature is an effective countermeasure to several threats, -as spoofing, forgery, or message corruption- not only for repudiation.

4.1 The Architecture

The overall system, shown in Figure 4, is obtained from CASHMA (Figure 2) adding a *Certification Authority* (CA), as featured by many PKI schemes. The other main entities that compose the system are: an Authentication Server, the clients and the web services. All of them are connected through SSL. The CA provides digital certificates in order to certify the ownership of a public key by each client and to guarantee that the client has sole control and access to the corresponding private key. We denote the couple of cryptographic keys as K^-_c and K^+_c, being the client's private and public key respectively, certified by the CA. We also assume that the Authentication Server has its own couple of keys: K^-_{AS} and K^+_{AS}.

As in [3], the Authentication Server is in charge of transmitting a certificate, called *DS-CNR certificate*, to the client. As for the CASHMA certificate in [3], the DS-CNR certificate is composed of:

Time stamp, sequence number, ID, Decision, expiration time.

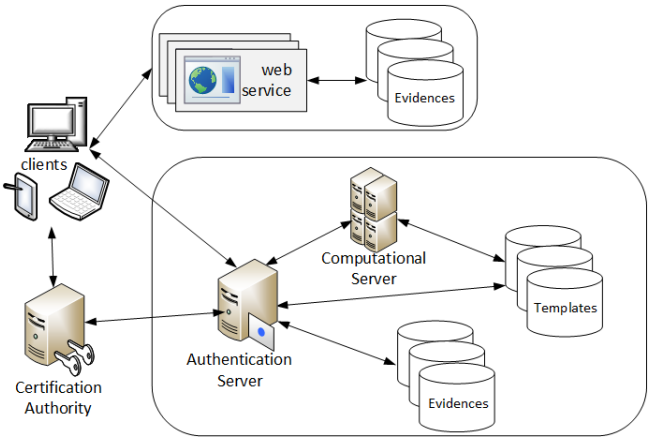

Figure 4: Overall view of the DS-CNR architecture.

The *Decision* represents the outcome of the verification procedure carried out on the server side, and as in [3] the user is not considered legitimate if the trust level is below $trust_{min}$. The higher is $trust_{min}$, the higher are the security requirements of a specific application or web service. Note that, differently from [3], $trust_{min}$ varies during the same session depending on the criticality of the operation that the client is asking to perform on a specific moment.

Further, we introduce two *databases of evidences*: one is for the storage of the messages received by the Authentication Server and the related signatures, while the other contains the digital signatures of all the requests and DS-CNR certificates received by the web service.

The other elements in Figure 4 are: high-performing computational servers that perform comparisons of biometric data for verification of the enrolled users, and databases of templates containing the biometric templates of the enrolled users.

4.2 The Protocol

During the registration phase, the client provides the selected biometric traits that are stored by the Authentication Server.

We propose a continuous non-repudiation protocol (shown in Figure 5) integrating the digital signature in the existing CASHMA protocol. In detail, two different signatures are used during Steps 1 and 3 (and during the maintenance phase on Steps 5 and 7). The following description highlights the differences between the new version of the protocol and the original one, discussed in Section 3.2.

Initial phase. This phase, shown in Figure 5, is now structured as follows:

First of all, exploiting the client's private key K^-_c, the application on client side performs the signature S_1 of the message containing the biometric traits of the user.

Step 1 – The message M_1 with the biometric traits sent at this step is concatenated with its signature S_1. The resulting message is then sent after adding a further encryption layer obtained using the Authentication Server public key K^+_{AS}.

Step 2 - The Authentication Server is the only entity able to decrypt the message, using its private key K^-_{AS}. Then it analyzes the biometric traits received and performs the authentication procedure as usual. As in [3], the user is not considered legitimate if the trust level is below the trust threshold $trust_{min}$. In DS-CNR protocol, depending on the

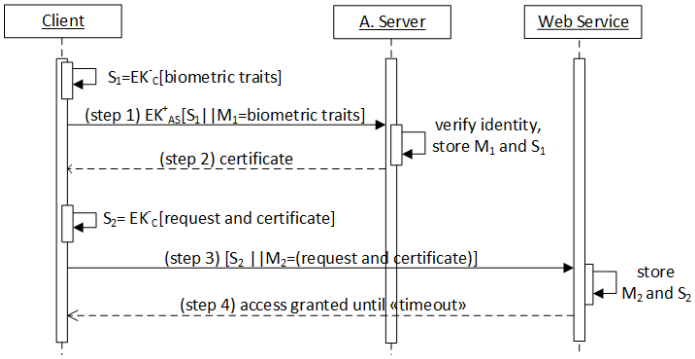

Figure 5: Initial Phase of DS-CNR protocol in case of successful identity verification.

criticality of the action to be performed, the $trust_{min}$ threshold may vary during the same session, and the *Decision* is directly influenced.

If the user identity is successfully verified, the server authenticates the user, computes an initial timeout of length T_0 for the user session, sets the expiration time at T_0+t_0, creates the DS-CNR certificate and sends it to the client. The Authentication Server also stores the received message M_1 and its signature S_1 in its database of evidences: these data can be accessed to solve possible disagreements between client and web service providers. In this way, the Authentication Server can act as a Trusted Third Party.

At the end of Step 2, exploiting the client's private key K^-_c, the client performs the signature S_2 of the message M_2, containing request and DS-CNR certificate.

Step 3 - The client forwards M_2 to the web service concatenating it with its signature S_2, which constitutes the NRO token. In fact, the web service stores M_2 and S_2 (the NRO token) in its database of evidences. This solves disagreements; in particular it protects the web service providers against repudiation of origin.

Step 4 - As in [3], the web service reads the certificate and authorizes the client to use the requested service until expiration time. The client stores the message received on Step 4, which constitutes a NRD token and protects it from disagreements with the web service.

A maintenance phase is then started. It is composed of four steps (Steps 5-8), analogous to Steps 1-4 of Figure 5. This phase is repeated iteratively: the client sends fresh biometric traits (Step 5), the server repeats the identity verification, takes the related *Decision*, renews the certificate and saves the evidences (Step 6); the client sends the new request and certificate to the web service, in order to have the timeout expiration postponed by the web service (Step 7); the web service stores the NRO token and then sends back the "access granted until timeout" confirmation, which is saved by the client and constitutes the NRD (Step 8).

4.3 Security Considerations

As discussed in Section 3.2, we assume that all data exchanged with this protocol are transmitted using SSL, that has been designed to provide privacy and data integrity between communicating parties [31]. However, SSL cannot protect against the insecurities introduced into the client system: if the user leaves his/her computer logged in to a secure web service and someone gets possession of the client computer without permission, the SSL protocol cannot protect him. For this reason, the continuous authentication protocol is necessary to guarantee the authenticity of the user.

In addition, even if SSL -exploiting a MAC- can guarantee that a message has not been changed during its transmission, it cannot provide non-repudiation. In our protocol, this is obtained with the integration of digital signatures in the communications where sensitive data is exposed.

In other words, the protocol is now able to continuously guarantee user authenticity (thanks to multi-biometric authentication), privacy and data integrity (thanks to the SSL/TLS based communications) and non-repudiation (thanks to digital signature) of the exchanged messages.

Non-repudiation is obtained digitally signing the messages from client to Authentication Server (Steps 2 and 6), and the messages from client to web service (Steps 3 and 7) and back (Steps 4 and 8). With this solution, if a client tries to deny having accessed a web service, a judge can ask to the web service for the client's message M_2 –that contains request and certificate- and the related signature S_2; then retrieve the client public key K^+_c from the Certification Authority and compare the signed message with the original one to verify if the signature is valid or not and. In other words, consulting the evidence (NRO token) the judge establishes if the client is lying or not. Similarly, the NRD token and the help of the Authentication Server, (which has M_1 and S_1 stored) are the evidences that can protect the client from eventual disputes with the web server.

In general, in a direct digital signature scheme, which means without the participation of a TTP, non-repudiation depends on the security of the sender's private key. If a sender wishes to deny the access, the sender can claim that the private key was lost or stolen and that someone else forged his/her signature [26]. The solution we proposed in this section tackles this vulnerability but still does not eliminate it completely. In fact, in DS-CNR the digital signature only works as long as the private key K^-_c remains secret. However, if the key is disclosed (or the client discloses it itself), an attacker can produce the signature S_1 only if it also possesses the biometric traits of the user. Similarly, the signature S_2 can be forged only if the attacker possesses a valid DS-CNR certificate, which contains a new sequence number and in which the *expiration time* has not been reached yet.

With the DS-CNR solution, the problem of usability of continuous non-repudiation is solved. In fact, even if the client has its private key stored on an external device, it does not need to provide a password or secret for each single transaction. It is sufficient to insert the secret only at the first login, without losing security: thanks to the multi-biometric identity verification, a secure authentication is guaranteed for the whole session. If, instead, the private key is stored on the client main computer/device, the multi-biometric continuous authentication will guarantee protection against insiders. In fact, even if an attacker succeeds in violating the client computer or device, it still has to continuously send multiple biometric traits to the Authentication Server if desires to access the web service and perform the operations that it offers.

In other words, without the multi-biometric authentication, the continuous non-repudiation would not be possible at the same degrees of usability and security.

5 BS-CNR: BIOMETRIC SIGNATURE FOR CONTINUOUS NON-REPUDIATION

To address the problem of private key loss [26], we present an alternative solution based on *biometric signature*.

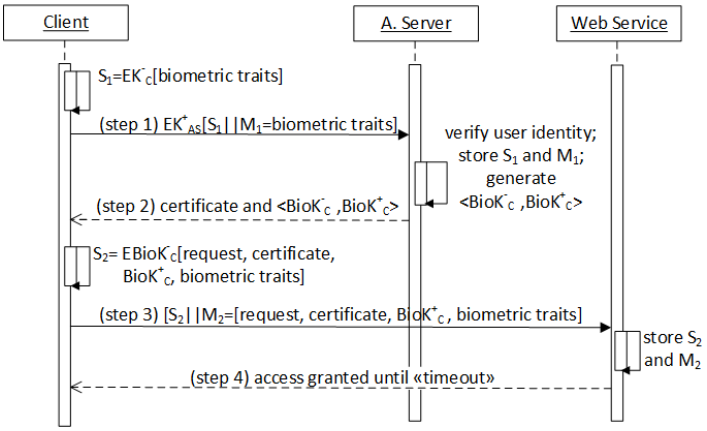

Figure 6: Initial Phase of BS-CNR protocol in case of successful identity verification.

The literature largely investigated [12], [13], [15], [27], [30] biometric signature as a process to derive a private key from a biometric sample and use the key to digitally sign a document or a message. We exploit biometric signature to prevent client's claim of having loss the private key or that the key has been stolen. The idea is to generate a new, additional, private key, denoted as $BioK^-_c$, during the first iteration (Initial Phase) of the protocol. Analogously, the corresponding public key $BioK^+_c$ should be generated concurrently.

The advantage is that the client does not store the private key when the session terminates, because it is valid for only one session, and assuming that the key is securely transmitted through SSL, it cannot claim that it has been lost or stolen. In addition, the $<BioK^+_c, BioK^-_c>$ keys pair is generated only if the user identity is verified (the trust level *trust(u, t)* is above the $trust_{min}$ threshold).

5.1 The Architecture

The overall architecture of the BS-CNR system is shown in Figure 7. The main difference with DS-CNR is that for BS-CNR the Authentication Server is also responsible for the generation of a couple of biometric-derived keys $<BioK^+_c, BioK^-_c>$ for each client during the initial phase of the protocol.

The remainder of the architecture does not change from what discussed in Section 4.1. As in DS-CNR, an external Certification Authority is involved, which provides digital certificates in order to certify the ownership of the couple of cryptographic keys K^-_c and K^+_c by the client. We also assume that the Authentication Server has its own couple of keys K^-_{AS} and K^+_{AS}. The other entities are: an Authentication Server, the clients and the web services. All of them are connected through SSL. Again, the other elements in Figure 7 are a set of high-performing computational servers for the biometric traits comparison useful for verification of the enrolled users, and databases of templates containing the biometric templates of the enrolled users.

As in the solution of Section 4, the BS-CNR certificate is composed of:
Time stamp, sequence number, ID, Decision, expiration time.

Finally, the considerations of Section 4.1 about the *Decision* are still valid here: $trust_{min}$ varies during the same session depending on the criticality of the operation that the client is asking to perform on the web service on a specific moment.

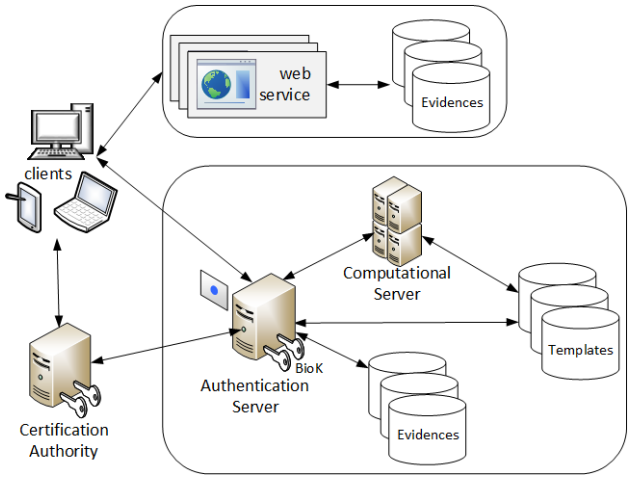

Figure 7: Overall view of the BS-CNR architecture.

5.2 The Protocol

During the registration phase, the client provides the selected biometric traits that are stored by the Authentication Server.

In this protocol, the Authentication Server exploits the biometric traits not only for user identity verification, but also to generate a couple of cryptographic keys $<BioK^+_c , BioK^-_c>$, following one of the existing approaches e.g., from [13], [27].

Initial phase. This phase, shown in Figure 6, is now structured as follows:

First of all, exploiting the client's private key K^-_c, the application on client side performs the signature S_1 of the message containing the biometric traits of the user.

Step 1 – The message M_1 with the biometric traits sent at this step is concatenated with its signature S_1. The resulting message is then sent after adding a further encryption layer obtained using the Authentication Server public key K^+_{AS}.

Step 2 - The Authentication Server decrypts M_1 with its private key K^-_{AS}. Then it analyzes the biometric traits received and performs the authentication procedure as usual. As in DS-CNR, depending on the criticality of the action to be performed, the security requirements can demand a trust threshold $trust_{min}$ set to significantly high values. So, during the same session, the threshold varies based on the operation that the client is going to accomplish, and the *Decision* is directly influenced.

If the criteria are not completely satisfied (the message in incomplete or the trust is below the threshold), the user is forced to return to Step 1.

Instead, if the user identity is successfully verified, the server authenticates the user, computes an initial timeout of length T_0, sets the expiration time at T_0+t_0, and creates the BS-CNR certificate. Then, the Authentication Server derives the $<BioK^+_c, BioK^-_c>$ couple from the biometric traits. Together with M_1 and S_1, it also stores the key pair in its database of evidences: these data can be accessed to solve possible disagreements between clients and web service providers. In this way, the Authentication Server can act as a Trusted Third Party.

The Authentication Server sends to the client a message containing the BS-CNR certificate, and also the couple of biometric keys $<BioK^+_c, BioK^-_c>$ assigned to the user for this session.

At the end of Step 2, the client, using its new private key $BioK^-_c$, computes S_2. It is the digital signature of the message M_2, which is composed of: the request, the BS-CNR certificate, the public key $BioK^+_c$, and the biometric traits.

Step 3 - The client forwards M_2 to the web service concatenating it with the signature S_2 which constitutes the NRO token. The web service reads the BS-CNR certificate and verifies its validity: it also checks if the $BioK^+_c$ has been already used before the current session. The web service stores M_2 and S_2 (the NRO token) in its database of evidences. This may be used to solve disagreements, in particular protects the web service providers against repudiation of origin.

Step 4 - As in [3] and in DS-CNR, the web service reads the certificate and authorizes the client to use the requested service until expiration time. On its side, the client stores the message received on Step 4, which constitutes a NRD token and protects it from disagreements with the web service.

As in DS-CNR, after the initial phase there is a *Maintenance Phase* composed of four steps (Steps 5-8), analogous to Steps 1-4 of Figure 6. This phase is repeated iteratively: the client sends fresh biometric traits (Step 5), the server repeats the identity verification, takes the related *Decision,* renews the certificate and stores the evidences (Step 6); the client sends the new request and certificate to the web service, in order to have the timeout expiration postponed by the web service (Step 7); the Web service stores the NRO token and then sends back the "access granted until timeout" confirmation, which is saved by the client and constitutes the NRD (Step 8).

Note that the <$BioK^+_c$, $BioK^-_c$> couple is generated (and stored) only during Steps 1-2 of the Initial Phase: these operations are not repeated during the Maintenance Phase.

5.3 Security Considerations

This solution addresses the problem of continuous non-repudiation trying to reduce the weaknesses showed by the DS-CNR, including the potential claim of private key loss by the client. This solution introduces a strong binding between the private key and the client: $BioK^-_c$ is generated in case of successful identity verification only through the biometric traits.

In addition, the key couple can be considered as one-time keys, which remains valid only for one session. However, after the expiration of the <$BioK^+_c$, $BioK^-_c$> keys, the web service or a judge can still use the $BioK^+_c$ to compare the message M_2 and its signature S_2, thus guaranteeing non-repudiation of the access. What cannot be done with these keys is using a private key for more than one session: in this way the client cannot repudiate having accessed the web service. In fact, if a couple of keys <$BioK^+_c$, $BioK^-_c$> is used for more than one session to send a request to the web service (Step 3), the web service would receive a message M_2 with a $BioK^+_c$ key that is already stored, and would deny the access.

Also in this protocol we assume that the transmissions are protected by SSL protocol, so that the privacy and the integrity of the information exchanged are guaranteed. For this solution, privacy is even more important because the message M_2 contains also the biometric traits of the user. This choice has been made to give even more strength to the NRO token stored by the web service. However, to protect the client biometrics, their templates are not sent to the web service in raw format: they are transformed using client specific parameters [40].

Apart from security, we also consider that the computational and storage resources needed are higher with respect to the previous solution: the Authentication Server generates a couple of biometric-based keys for each client, once for each session. Further, it stores the <$BioK^+_c$, $BioK^-_c$> couple with the corresponding certificates in case a TTP is requested to solve future disputes.

6 AN ONLINE BANKING SCENARIO AND ITS SECURITY ANALYSIS

The two solutions presented in this paper are general enough to guarantee a non-repudiation service for many scenarios in which users and remote Internet services are involved in a continuous information flow.

In the following, we present a detailed description of how the BS-CNR is applied in a sample application scenario (Section 6.1) in order to prevent repudiation from a customer (Section 6.2.1) and from an online banking service operator (Section 6.2.2). The same analysis, which we do not report here for space constraints, can be easily performed also for DS-CNR.

6.1 BS-CNR System Design and Assumptions

Let us consider an online banking service which distinguishes between *Risky Actions* (RA_n) and *Basic Actions* (BA_n). An example of risky action is RA_1: transferring an amount of money that exceeds a specified value (e.g., five thousand dollars) to an account of a different bank and/or belonging to a different customer. An example of basic action is BA_1: consulting services. The online banking service exploits BS-CNR to permit *Risky Actions* only to the legitimate client which is highly trusted. That is, in order to get a valid BS-CNR certificate, the $trust_{min}$ threshold required by the Authentication Server is set to a relatively high value. Instead, the $trust_{min}$ threshold is set to a lower value in order to get a BS-CNR certificate valid to perform *Basic Actions*.

We assume a customer intends to perform operations on its online banking account. The customer has been previously enrolled to the BS-CNR system.

On the client side, the customer device is a desktop computer which integrates a webcam for the acquisition of face and iris traits, and a mouse with an optical scanner for the acquisition of fingerprints [43].

We assume that on the Authentication Server the BS-CNR system depends on Sys_1, Sys_2, and Sys_2, being respectively three subsystems with the state-of-the-art biometric verification algorithms:

- Sys_1 with Google FaceNet [32], for face recognition, which showed at FAR 10^{-6} a True Acceptance Rate (TAR) of 86.473% in Megaface challenge [33], and the impressive accuracy result of 99.63%±0.09 in LFW benchmark [34];
- Sys_2 with IRITECH algorithm for iris recognition, which showed at FAR 10^{-6} a FRR of 0.002 in the context of NIST Iris Exchange IREX I [35];
- Sys_3 with Neurotechnology algorithm for fingerprint recognition, which had at FAR$\leq 10^{-2}$ a FRR of 0.083 in the FVC-onGoing online evaluation [36].

Thus the subsystems possess a relatively high *subsystem trust level* $m(S_k,t_0)$ thanks to their low FAR_k.

If during an iteration of the protocol the client wants to perform a *Risky Action* (RA_n) but the Authentication Server does not provide the BS-CNR certificate, the action would not be permitted. If it happens when the operation is already ongoing, because during the previous iteration it has been permitted, the operation will be aborted.

6.2 Repudiation Attempts and Security Analysis

Before describing two repudiation attempts and performing the security analysis, we summarize the assumptions under which the analysis is conducted. The assumptions are: (i) all the communications between the entities are secure (e.g., under SSL/TLS); (ii) the Authentication Server and the Certification Authority are both Trusted Third Parties; (iii) the subsystems Sys_1, Sys_2, and Sys_2 adopt specific anti-spoofing and liveness detection measures so that the biometric traits are not forgeable.

The following analysis considers only attacks in terms of fraudulent behaviors of the parties and describes how the BS-CNR solution neutralizes them guaranteeing non-repudiation. In the future works, starting from the analysis of [14], we are going to consider also cyber-attacks, with a model-based, quantitative security evaluation of both DS-CNR and BS-CNR.

6.2.1 Fraudulent Customer. Consider the situation in which a customer sends money to a second account, then denies the transaction and asks for a refund in order to deliberately fraud the bank.

At time t_i, the Step 2 of the protocol has been just completed, the Authentication Server verified the identity of the user, produced a BS-CNR certificate containing a timeout of T_i=20 seconds (s), thus setting the expiration time at t_i+T_i. The customer sends a RA_1 action request, providing on Step 3 all the data needed. The banking service checks the validity of the certificate, of the $BioK^+_c$ key, and grants the operation until t_i+T_i is reached. The customer spends more than 20s to complete the RA_1 operation, but the BS-CNR protocol continuously verifies the user identity and checks if *trust(u, t)* is always above the $trust_{min}$ threshold requested by the web service to perform RA_1 operations. Otherwise, the operation is aborted. In this example, the expiration time is extended, and the customer is able to complete the operation.

A short time later, the customer repudiates the transaction, claims having been robbed and asks for the judge intervention in order to get a refund.

The online banking service is protected against this attempt of repudiation: it can provide the NRO token received on Step 3 (and Step 7) of the protocol. This protection also invalidates a possible claim of private key loss by the user: the Authentication Server can provide the evidences stored at the end of Step 1, showing that the private key $BioK^-_c$ has been generated on that moment and cannot be used during subsequent sessions. Moreover, even assuming the loss of the private key, if all Sys_1, Sys_2, and Sys_3 subsystems have successfully verified the user, the probability that the multi-biometric system has authenticated an impostor is:

$P(FA_{system}) = P(FA_1) \cap P(FA_2) \cap P(FA_3) = 10^{-14} \approx 2^{-46}$.

Its value is comparable with the probability of inverting 2048-bit RSA used for digital signature, which is $\leq 2^{-60}$[37]. Thus, we believe that biometrics constitutes an important additional layer of security, and the $P(FA_{system})$ value is sufficiently low to cause the customer's attempt of repudiation to be useless.

Besides, such a claim would be counter-productive if an investigation discovers the fraud attempt.

6.2.2 Fraudulent Banking Operator. Let us examine the opposite situation, with a dishonest operator of the online banking service, which somehow obtains access to the money transferring system. We assume the bank employee succeeds in sending 20 thousand dollars to his own account from the one of a customer. The customer who has been ripped off asks for a refund and for the judge intervention. When questioned, the online banking service is not able to produce a valid NRO token (S_2 and M_2 evidences), related to a hypothetical RA_1 operation executed by the customer on that time interval. Thus the judge solves the dispute in favor of the innocent customer and the bank is forced to refund.

7 CONCLUSIONS AND FUTURE WORKS

Nowadays, ICT plays an important role in our society, and security services are getting increasingly fundamental to protect users and entities involved. Non-repudiation is one of these services: it provides evidences of actions, protects against denial of involvement, and helps solving disputes between parties. Traditional non-repudiation mechanisms, as digital signature, are widely applied to prevent denial of past behaviors as having sent or received messages. However, if the information flow is continuous, evidences should be produced for the entirety of the flow and not only at specific points. Further, non-repudiation should be guaranteed by mechanisms that do not reduce the usability of the system or application.

To meet these challenges, in this paper, we proposed two solutions for *continuous non-repudiation* of remote services, based on multi-biometric continuous authentication, respectively coupled with digital signature and biometric signature: DS-CNR, and BS-CNR.

We choose the CASHMA approach [3] as a starting point, which is one of the few solutions in literature which improve security of the user authentication without reducing usability. Moreover, the CASHMA system has been evaluated both from quantitative [3] and qualitative [14] perspectives. In [14], the assessment identified the main threats both for the transmission and the biometric system level. The selected countermeasures, most of which (e.g. digital signature, avoid transmitting raw biometrics, biometric signature) have been integrated in the CNR solutions, are capable of reducing the risk for the detected threats.

The DS-CNR and BS-CNR approaches are a demonstration that, under specific assumptions, continuous authentication mechanisms can actually be complemented with non-repudiation. The proposed solutions are able to provide continuous non-repudiation for the entire information flow between a client and a remote Internet service. The NRO and NRD evidences generated, and the Authentication Server involvement, guarantee protection for both parties.

We also showed that some biometric algorithms in literature already have high verification accuracy. In our opinion, if those algorithms (or different ones equally accurate) are properly integrated in a multi-biometric system, with high quality sensors and anti-spoofing measures, biometric-based continuous non-repudiation is possible. The probability of authenticating an impostor is relatively low and constitutes a first evidence of user involvement. This is particularly valid for traits which inherently possess high distinctiveness [8]. Then, coupling biometrics with other security mechanisms makes continuous non-repudiation very effective.

As a future work, we are going to discuss if some assumptions can be relaxed and to which extent. We are also planning a model-based, quantitative security evaluation in order to assess the likelihood and impact of attacks on the system, under different scenarios, system settings, and attackers' profiles. Models will allow quantifying the threat analysis available at [14]. This study will permit comparisons between our solutions and state-of-the-art approaches for non-repudiation of remote services. The analysis will be performed integrating SAN [41], a well-known formalism for the analysis

of dependable systems, and ADVISE [42], a methodology for security evaluation. Finally, the direction of our future work is also to identify and realize practical case studies to better understand advantages and limitations of the two solutions.

ACKNOWLEDGMENTS

This work has been partially supported by the project DEVASSES, funded by the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no PIRSES-GA-2013-612569, and by the FAR-FAS 2014 TOSCA-FI project funded by the Tuscany Region.

REFERENCES

[1] S. Kremer, O. Markowitch, and J. Zhou. 2002. An intensive survey of fair non-repudiation protocols. *Computer communications*, 25, 17, 1606-1621.

[2] J. A. Onieva, J. Zhou, and J. Lopez, Multiparty nonrepudiation: A survey, *ACM Computing Surveys (CSUR)*, vol. 41, no. 1, p. 5, 2009.

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

[4] ITU Baseline identity management terms and definitions, Recommendation ITU-T X.1252, April, 2010.

[5] NIST Special Pub. 800-53 Revision 4: Security and Privacy Controls for Federal Information Systems and Organizations, 2013.

[6] ISO/IEC 13888-3:2009 — Information technology — Security techniques — Non-repudiation — Part 3: Mechanisms using asymmetric techniques, 2009.

[7] ISO/IEC 18014-2:2009 — Information technology — Security techniques — Time-stamping services — Part 2: Mechanisms producing independent tokens, 2009.

[8] A. K. Jain, A. Ross, S. Prabhakar, An introduction to biometric recognition, *IEEE Transactions on circuits and systems for video technology*, vol. 14, n. 1, pp. 4-20, 2004.

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

[10] H. Bidgoli, (2006). Handbook of Information Security, Information Warfare, Social, Legal, and International Issues and Security Foundations (Vol. 2). John Wiley & Sons.

[11] D. R. Kuhn, V. C. Hu, W. T. Polk, and S. J. Chang, Introduction to public key technology and the federal PKI infrastructure, National Inst of Standards and Technology Gaithersburg MD, 2001.

[12] A. Kholmatov, and Y., Berrin, Biometric cryptosystem using online signatures, *Computer and Information Sciences–ISCIS* 2006. Springer Berlin Heidelberg, 2006. 981-990.

[13] H. Feng, and C. Choong Wah, Private key generation from on-line handwritten signatures, *Information Management & Computer Security* 10.4 (2002): 159-164.

[14] E. Schiavone, A. Ceccarelli, A. Bondavalli, Risk Assessment of a Biometric Continuous Authentication Protocol for Internet Services, in *Proceedings of ITASEC the 1st Italian Conference on Cybersecurity*, CEUR workshop proceedings v.1816, pp. 53-65, 2017.

[15] C. Rathgeb, A. Uhl, A survey on biometric cryptosystems and cancelable biometrics, *EURASIP Journal on Information Security*, vol. 2011, n. 1, pp. 3, 2011.

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

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

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

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

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

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

[22] R. Lieber, (2013). Disputing a Charge on Your Credit Card, http://www.nytimes.com/2013/01/26/your-money/what-happens-when-you-dispute-a-credit-card-charge.html [online]

[23] R. Hosie, (2017). 'Can you hear me' phone scam currently defrauding US consumers set to hit British shores, http://www.independent.co.uk/life-style/can-you-hear-me-phone-scam-fraud-us-britain-police-pennsylvania-florida-uk-a7597106.html [online]

[24] N. Kuntze, A. U. Schmidt, and C. Hett, Non-repudiation in internet telephony, *IFIP International Information Security Conference*, pp. 361-372, 2007.

[25] W. Diffie, and M. Hellman, (1976). New directions in cryptography. *IEEE transactions on Information Theory*, vol. 22, n. 6, pp 644-654, 1976

[26] W. Stallings, Cryptography and Network Security, Principles and Practice. 5/E. Pearson Education India, 2011.

[27] P. K. Janbandhu, and M. Y. Siyal, (2001). Novel biometric digital signatures for Internet-based applications. *Information Management & Computer Security*, *9*(5), 205-212.

[28] A. Ross, A. K. Jain, Information fusion in biometrics, *Pattern Recognition Letters*, 24(13), pp. 2115–2125, 2003.

[29] L. Hong, A. K. Jain, S. Pankanti, Can Multibiometrics Improve Performance?, *Proceedings AutoID*, (99), pp. 59-64, 1999.

[30] A. Burnett, F. Byrne, T. Dowling, and A. Duffy, (2007). A Biometric Identity Based Signature Scheme. *IJ Network Security*, *5*(3), 317-326.

[31] T. Dierks, The transport layer security (TLS) protocol version 1.2, (2008).

[32] F. Schroff, D. Kalenichenko, and J. Philbin, (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 815-823).

[33] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, (2016). The megaface benchmark: 1 million faces for recognition at scale. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 4873-4882). http://megaface.cs.washington.edu/ [online]

[34] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. *University of Massachusetts, Amherst, Technical Report 07-49*, October, 2007. http://vis-www.cs.umass.edu/lfw/ [online]

[35] P. Grother, E. Tabassi, G. Quinn, and W. Salamon, (2009). Irex I - Performance of Iris Recognition Algorithms on Standard Images, NIST Interagency Report 7629. *NIST Information Access Division.* https://www.nist.gov/itl/iad/image-group/irex-i [online]

[36] B. Dorizzi, R. Cappelli, M. Ferrara, D. Maio, D. Maltoni, N. Houmani, S. Garcia-Salicetti and A. Mayoue, Fingerprint and On-Line Signature Verification Competitions at ICB 2009, in proc. Int. Conf on Biometrics (ICB), Alghero, Italy, pp.725-732, June 2009. https://biolab.csr.unibo.it/FvcOnGoing/UI/Form/Home.aspx [online]

[37] C. Y. Koo, N. Yakovenko, J. Blank, J. Katz, Digital Signature Schemes, Lecture 16 of Advanced Topics in Cryptography, 2004.

[38] E. Schiavone, A. Ceccarelli, A. Bondavalli, and A. M. Carvalho, (2016). Usability Assessment in a Multi-Biometric Continuous Authentication System. In Dependable Computing (LADC), 2016 Seventh Latin-American Symposium on (pp. 43-50). IEEE.

[39] D. Barboza, Online Brokers Fined Millions In Fraud Case, 2003. http://www.nytimes.com/2003/01/15/business/online-brokers-fined-millions-in-fraud-case.html [online]

[40] A. Nagar, K. Nandakumar, and A.K. Jain, (2010, February). Biometric template transformation: a security analysis. In *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics.

[41] W. H. Sanders, and J. F. Meyer, (2001). Stochastic Activity Networks: Formal definitions and concepts. In *Lectures on Formal Methods and PerformanceAnalysis* (pp. 315-343). Springer Berlin Heidelberg.

[42] E. LeMay, M. D. Ford, K. Keefe, W.H. Sanders, and C. Muehrcke, (2011, September). Model-based security metrics using ADversary VIew Security Evaluation (ADVISE). In *Quantitative evaluation of systems (QEST), 2011 eighth international conference on* (pp. 191-200). IEEE.

[43] SecuGen OptiMouse Plus. http://www.secugen.com/products/po.htm [online]