Accessing Cloud Services with Biometrics Authentication

Pietro Ruiu*, Giuseppe Caragnano*, Giovanni L. Masala†, Enrico Grosso†
* Istituto Superiore Mario Boella (ISMB), Torino, Italy
† Department of Political Science, Communication, Engineering and Information Technologies
Computer Vision Laboratory, University of Sassari, Sassari, ITALY
E-mails: *{ruiu,caragnano}@ismb.it, †{gilmasala,grosso}@uniss.it

***Abstract*—The adoption of Cloud computing involves many advantages in terms of flexibility, scalability and reliability, but also implies new challenges on security, data privacy and protection of personal data. Since more and more sensitive applications and data are moved to the cloud, to verify that each of the participants in the electronic communication is really who he claims to be has become a crucial challenge. Currently, the use of biometric techniques can be considered as an effective solution to ensure a significant increase of security in the authentication protocols managed by modern authentication servers. However the use of biometric data for the logical access to IT services is a more challenging and still unsolved problem. The project Cloud for SME integrates a biometric authentication based on fingerprints with a cloud computing platform, investigating how highly secure authentication methods can increase the adoption of cloud computing technologies among small and medium enterprises.**

***Keywords*—cloud computing, Biometrics, orchestration, security, OpenStack.**

I. Introduction

The development of service-oriented architectures (SOA) and WEB services are bringing to migration from local to web applications, sharing critical data and resources and giving support to multi-user/multi-tenancy scenarios. SOAs support designing and developing in terms of services with distributed capabilities, which can be under the control of different ownership domains. The extension of the web application paradigm to the cloud computing model is denoted as software as a service (SaaS). In particular Cloud computing, involves many advantages in terms of flexibility, scalability and reliability, but also implies new challenges on security, data privacy and protection of personal data [1]. Cloud is a distributed architecture, this implies an increased use of networks and data communication flows compared to traditional architectures. Simple storage operations can involve communication between central systems and cloud remote clients or data must be transferred for the synchronization of images of the same virtual machine among various and distributed hardware infrastructures. From these simple examples it is possible to highlight that there are several possible risks in which to incur as sniffing, spoofing, man-in-the-middle and side channel attacks.

In this way the system has to verify the digital identity but it is not easy to provide a formal definition of identity. Basically, we can define the essential and unique characteristics of an entity are what identify it. Each identity maps to a unique set of characteristics. Two people may share some of the same characteristics, such as being old enough to drive or having the same hair color, but that does not mean that they have the same identity. Therefore, when someone perceives that two identities as the same, he should search for new information that adds details that distinguish the identities from each other (characteristics). Identity also evolves over time, with more characteristics becoming evident everyday or characteristics may change their state. The distinction between characteristics and identity is not firm. Often, a unique characteristic serves as the representation or identifier for identity. Currently, the most common authentication mechanisms to verify digital identity use passwords and private tokens with well known security threats; for example, they can be easily stolen or intercepted and used fraudulently. Tokens are more difficult to be reproduced and for this reason they are often used in banking services. However, being more expensive and difficult to manage, and physical card or device can be easily shared with different people. The use of biometric techniques [2] [3], can be considered as one way to ensure a significant increase of security in the authentication protocols managed by modern authentication servers. The use of multiple biometric features for identity verification minimize the typical risks of traditional authentication systems, in applications that require a high level of security like border control.

The project described in this paper has the primary goal of supporting basic web applications shared by small and medium companies; candidate platforms for Cloud computing should be, therefore, oriented to scalability, to be implemented according to the public or private Cloud models. Furthermore the integration of biometric recognition with cloud computing platform is made. In the following sections we show that the aim of such proposed integration of biometric authentication in Cloud Services is to solve the main key issues in terms of performance (end to end), infrastructural issues, interoperability and human factors (usability, acceptability, environment, enrollment).

II. Related Works

Cloud security is a major concern that may delay its widespread adoption. User access control (UAC) is the core component of security in cloud computing environment, aim-

ing to ensure that stored data are allowed to be accessed only by authenticated/authorized users. At present, authentication is done in several ways: such as, textual, graphical, biometric, 3D password and third party authentication. Password are reinforced in several algorithms introducing multi-level authentication technique [4] or password with additional information as in [5] where for mobile device authors propose additional identication using number entries, basic ID/password type authentication methods, or using the authentication of diverse user bio-information. Biometric Authentication as a Service is an innovative approach for strong authentication in web environments based on the Software as a Service model. However, for [6] both the adoption of SaaS systems and biometric technologies negatively correlate with perceived privacy and data protection risks .

In patent [7] a web-based authentication system comprise at least one Web client station, at least one Web server station and an authentication center. The Web client station is linked to a Web cloud, and provides selected biometric data of an individual who is using the Web client station. The Web server station is also linked to the Web cloud. The authentication center is linked to at least one of the Web client and Web server stations so as to receive the biometric data. The authentication center, having records of one or more enrolled individuals, provides for comparison of the provided data with selected records.

Independent authors propose biometric authentication based on fingerprint through an authentication server and a client software [8] or fingerprint features in [9] are not only used for biometric verication but also for cryptographic key generation; in [10] author involves a fingerprint biometric and password to enhance the security level of the remote authentication scheme for mobile device; paper [11] proposes two-factor authentication scheme based on Schnorr digital signature and feature extraction from fingerprint. As a typical behavioral biometrics, some author [12] [13] use keystroke dynamics to obtain a UAC solution. The advantage of using behavioral biometrics such as keystroke dynamics is that it can be collected even without the knowledge of the user. Authors [14] improve the security of voiceprint storage and transmission, using an approach with homomorphic encryption; their system supports to calculate distortion measurement of voiceprint without disclosing the raw voiceprint data under the tele-biometric functional model in the open network; while an example of iris recognition is implemented in [15] where they have used Hadoop , an open source cloud computing environment, to develop this model.

In this paper, we present a complete Cloud system that uses biometric authentication based on fingerprints integrated with the OpenStack cloud platform. In particular several services are added or improved with respect to the first release [8] and we make a focus on the novel orchestration service, the management of cloud users and the secure routing (VPN).

III. Cloud per le PMI Project

add general description of the project, aim, concept, partner

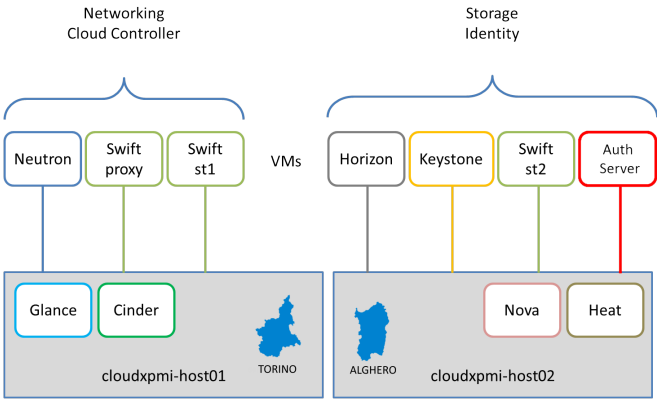

Fig. 1. Subdivision of Open Stack functions between our two Italian data centers of Alghero and Turin. Services Nova and Heat have a physical machine on the server of Turin and all other services are arranged on virtual nodes.

A. Architecture of the System

The architecture of the solution implemented is based on the concept of "sharing nothing" that make components independent and self sufficient, avoiding the sharing of memory or storage. Communication between the different modules are asynchronous and are managed by queue managers (message brokers) that implement the Advanced Message Queuing Protocol (AMQP). The various services communicate with each other through specific Application Programming Interfaces (APIs) that implement the REST model. The cloud services are represented by a suite of enterprise applications (an e-commerce portal, an on line document editor, a project management tool and a personal cloud storage platform) running on isolated VMs.

All the solution has been implemented using open source tools, running on Linux Operating System.

- **OpenStack** [16] is an open source project that many identify as the first true Cloud Operating System. OpenStack allows to control every aspect of a cloud environment, from datacenter resources to virtual deployments. It is developed by the OpenStack Foundation supported by all the major ICT vendor.
- **KVM** [17] is a Linux kernel module that allows control and manage virtualized instances running on bare metal. It is integrated into the kernel, thus improving performance and reducing the impact on existing Linux systems.
- **OpenVPN** [18] is a full-featured SSL VPN which implements OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or username/password credentials.

In the design phase has been decided to use the open source OpenStack as cloud platform, which has a modular architecture and includes native identity management and orchestration functionality. OpenStack consists of different modules, each with a specific function. NOVA is used for

hosting and managing cloud computing systems, NEUTRON is the system that allows to manage the network connectivity and to address the VMs in the cloud. SWIFT is a distributed storage system that can accommodate data of users of the platform or VMs. The service KEYSTONE provides authentication and accounting for the entire platform and it is installed on a dedicated virtual machine; this module is direct connected to our authentication system [8]. HORIZON is a graphical interface platform accessible via the web, GLANCE is the Virtual Machine Image Repository and CINDER allows to provide storage that can be used by nova to serve the VMs. These functions are spread on two bare metal servers located in the site of the two partners of the project: Alghero for the University of Sassari and Torino for Istituto Superiore Mario Boella. It has been chosen to geographically distribute the system for understanding challenging of orchestrate services across a real WAN scenario and moreover test the potential of Openstack's network virtualization functionalities.

The virtual cluster composed by the istances running the four appications, is configured automatically using the native orchestration module (HEAT). In Figure 1 the Subdivision of OpenStack functions between our two Italian data centers is shown. The great modularity of the system and the use of the REST API, according to the SOA model, guarantees a great flexibility and enables interoperability with other systems. CloudXSME can be easily updated or integrated with other platforms by replacing or integrating modules. This peculiarity makes the solution fully compatible with European and national eID initiatives, such as Stork and SPID.

B. Security Mechanisms

The security of the system is ensured at different layers implementing well established mechanisms. The VPN technology is adopted to create a secure and protected channel between client and servers through encrypted tunnels which obfuscates data during the transfer. The VPN selectively enables the services that can be accessed by the users. Before authentication the user client can only communicate with the API server. After being authenticated the system automatically creates a route, enabling the link to the GUI. Furthermore a temporary session ID is generated. The Session ID is enabled only for the user who requested it and it is destroyed as soon as the session ends. All the other components of the system (GUI, AutSrv, cloud services, etc.) have privete IPs and can be accessed only through the VPN. The APISrv is the gateway for the private network, it has a public IP and is configured as the VPN server. Inside the cloud the security is managed by the native OpenStack modules. User related data (models, passwords) are never transferred out the cloud.

C. Authentication System

The recognition system is implemented in a authentication server (AS) that exposes the necessary API to its integration with the rest of the system. The APIs are defined in the operating system minimum set required, then there are the functions for:

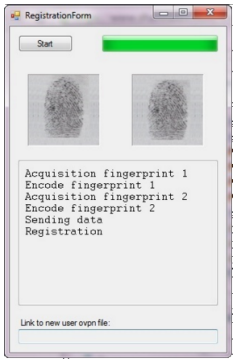

Fig. 2. GUI for the enrollment of new users; it is possible to make two different fingerprints acquisition; during the registration the model is created and the original feature deleted.

- Register a new user in the system
- User identification
- Reset of all users

The authentication system is designed to be scalable in horizontal on multiple computing nodes (currently the system runs on a single node, but can be propagated on N virtual machine twin of that currently in service) and vertical optimizing the CPU performance, through the parallel computation inside the node in which it operates. To improve processing time, at start-up of the computing node, the whole set of information related to the users is copied directly into RAM in order to cancel the disk access times. With the current service configuration (1 node calculation with 4 vCPU) the total time of identification is calculable, on average, in 3/10 of a second per registered user.

The desktop client application is composed by a software for the enrollment of new users and an authentication application (Figure 2). During enrollment the new user's fingerprint is converted into a compact representation, called model; this model will be used to recognize the user. It is not necessary to store the fingerprints in the AS database; only the models are recorded. The features to produce the model are obtained by using the Scale Invariant Feature Transform (SIFT) representation [8]. If the VPN encrypted tunnels is enable, the user starts the session simply touching the fingerprint scanner.

D. Biometrics

The fingerprint scanner used for the purpose of the project has a 1 inch x 1 inch sensor and is certified by FBI according to PIV (Personal Identity Verification) Image Quality Specifications. The fingerprint image of the user is not transferred to the cloud, but it is converted in a model. The features to produce the model are obtained by using the Scale Invariant Feature Transform (SIFT) representation that captures the main local patterns of an image working on a scale-space decomposition of the image [8]. This technologies ensure a good quality and performance level, currently unreachable with most commercial devices. Nevertheless, soon, with the progress of the technology, it will be possible to use biometrics devices already equipped on PC or mobile phones.

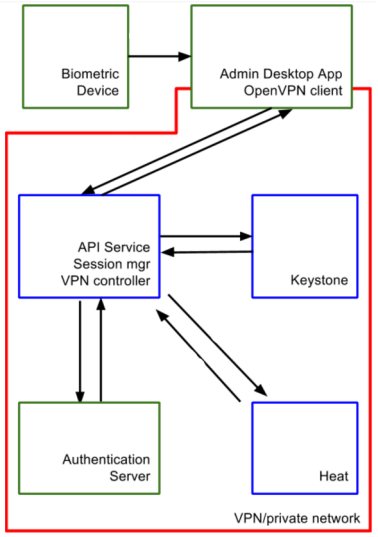

Fig. 3. Components involved and workflow of the Registration procedure.

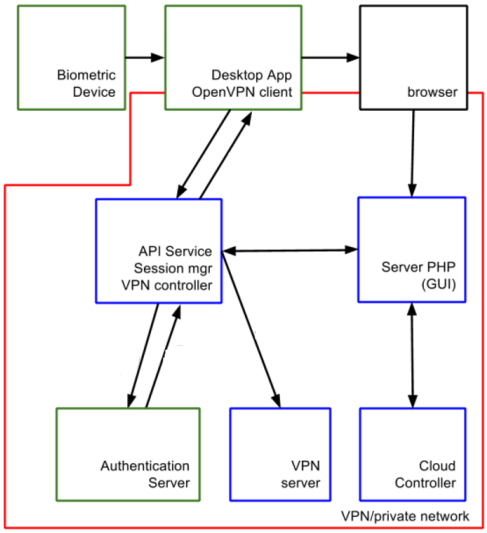

Fig. 4. Components involved and workflow of the authentication procedure.

IV. Automation Engine

add a general description of the engine

All the automation tasks are conduced by the API service. This module translate the registration and authentication requests in a set of automated tasks to be performed on the different components of the system: OpenStack for the management of pre-configured virtual machine and creation of clusters, the VPN server for managing the network connectivity, the AS for the management of biometric data and the GUI for the security sessions creation.

A. Registration Process

Before registration process starts, a secure channel is created through a VPN. Next, the client sends new user's data to the system. The API service receives two files in JSON format, containing the meta-data that are generated during fingerprint acquisition. The JSON object also contains the user's company name. After receiving user data properly, the system initiates an automated procedures for set up the virtualized environment that will host the user's services. A general overview of the process is represented in the Figure 3. During registration, the service API does the following:

1) add user to API service list;
2) add user to AS;
3) add user to OpenStack Keystone
4) add user to OpenVPN Server
5) create a new Stack with OpenStack HEAT

At this stage, automated checks on each component are carried out. Given the complexity and heterogeneity of the resources involved, the system will conduct checks to prevent misalignment between Authentication needed configuration service, Keystone, OpenVPN server and API service. Therefore, the registration process ends when at least one of the operations listed above fails. Initially, the system calculates the new user's ID, password and a network CIDR. The password is generated by random algorithm and it is used by the API service to communicate with Keystone and manage cloud services. Therefore, in order to make the whole system even safer, no other component will possess credentials. Continuing, the API service sends the username and password to the AS that registers the new user. If the registration is not successful, the AS returns an error message, and the whole process stops. Now we describe the core of the automatic procedure. The API service requires a token in Keystone to create a new user. The interactions between Keystone and API service is done through the OpenStack API endpoint, called Identity. To ensure the secure connection between the user and the services, during the registration process, a VPN certificate is automatically generated. In this phase, the automation is in charge of to the OpenVPN server which accepts as input (communication done using a Rest API Interface) the user name and the network's CIDR, and returns the OVPN certificate, ready to be used on the VPN client. The OpenVPN server set up correct routing rules that allow user access to the network and thus to its services. The rules will take effect only if user is authenticated successfully. OpenStack has images of virtual machines, pre-configured and ready to use. The API service sends a request to create a new stack. A Stack consists of a set of virtual machines connected to a new network. The network is then connected to a virtual router. All these operations are carried out automatically by the machine-accessible orchestrator HEAT. This hardest operation of the entire process because it involved almost all OpenStack services: NOVA for the creation of virtual machines, Neutron security groups, ports, subnets and vRouter interfaces. Finally, API service has successfully completed all operations and returns to the desktop client, the Open VPN certificate.

B. Authentication Process

The procedure for authenticating the user is shown in Figure 4. Before performing any operation, the user connects to the system with its OpenVPN certificate. When the client desktop finishes processing and coding of bio-metric data, sends file to the API service through a VPN tunnel. The data is transmitted to the AS and if user is recognized, AS returns a pair of values (that are username and password). The credentials will also be used in this case (as happens in the registration process) by API service, GUI and OpenStack. According to result of authentication stage, the API service creates or not a new session. When the user is correctly recognized the API service generates new *PHP* session by creating a session file in *PHP* session_path containing usernames, passwords, token (OpenStack) and stackID. The username and password parameters are supplied from the AS, while stackID is obtained by consulting a list on the API service and a token is generated by an automatic procedure. The API service connects to OpenStack Keystone requesting the token that will be used to manages the Virtual machines (start, stop, resume, etc.). Finally, the API service has completed its task and returns the generated session ID to desktop client. The desktop client will use it to generate a URL like this:

http://gui_server/index.php?sid=<sessionID>.

At this stage, the user can access to services simply connecting to the URL via browser. When the user makes a request for service management, the GUI server interacts with NOVA and other OpenStack services, through the REST API. All the automation layer is run with PHP with a light framework which is able to manage processes quickly. When the user leaves the GUI, the session is destroyed and all environmental variables used for service management are removed.

V. Conclusion

In this work, the authors have discussed about a platform capable of delivering cloud services and a robust user's companies biometric authentication system. The point of strength of the system is its modularity, each module has plays a very specific task and it is able to communicate with others through a HTTP Rest API interface. Another innovative element is the high automatism which the authentication and registration are handled. In the future we will try to improve the customization and flexibility of the platform. This implies improving, and in many cases extend, the automatic functions of service management. There is still a lot of work to do in order to increase delivery speed of virtualized environment. It is therefore necessary to streamline the start up procedures of resources, and interaction with OpenStack.

Acknowledgment

This work was supported in part by Regione Autonoma Sardegna LR 7 2007, N.7: "Promozione della ricerca scientifica e dell'innovazione tecnologica in Sardegna", Piattaforme di Cloud computing per le PMI, Codice: CRP-61647.

References

[1] Madhan Kumar Srinivasan, K Sarukesi, Paul Rodrigues, M Sai Manoj, and P Revathy. State-of-the-art cloud computing security taxonomies: a classification of security challenges in the present cloud computing environment. In *Proceedings of the international conference on advances in computing, communications and informatics*, pages 470–476. ACM, 2012.

[2] Arun A Ross, Karthik Nandakumar, and Anil Jain. *Handbook of multibiometrics*, volume 6. Springer Science & Business Media, 2006.

[3] Claus Vielhauer. *Biometric user authentication for IT security: from fundamentals to handwriting*, volume 18. Springer Science & Business Media, 2005.

[4] H. A. Dinesha and V. K. Agrawal. Multi-level authentication technique for accessing cloud services. In *Computing, Communication and Applications (ICCCA), 2012 International Conference on*, pages 1–4, Feb 2012.

[5] Young-Sik Jeong, Ji Soo Park, and Jong Hyuk Park. An efficient authentication system of smart device using multi factors in mobile cloud service architecture. *International Journal of Communication Systems*, 28(4):659–674, 2015.

[6] Christian Senk and Florian Dotzler. Biometric authentication as a service for enterprise identity management deployment: a data protection perspective. In *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*, pages 43–50. IEEE, 2011.

[7] Yuan-Pin Yu, Stephen Wong, and Mark B Hoffberg. Web-based biometric authentication system and method, July 27 1999. US Patent 5,930,804.

[8] GL Masala, P Ruiu, A Brunetti, O Terzo, and E Grosso. Biometric authentication and data security in cloud computing. In *Proceedings of the International Conference on Security and Management (SAM)*, page 9. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015.

[9] Kai Xi, Tohari Ahmad, Fengling Han, and Jiankun Hu. A fingerprint based bio-cryptographic security protocol designed for client/server authentication in mobile computing environment. *Security and Communication Networks*, 4(5):487–499, 2011.

[10] Chin-Ling Chen, Cheng-Chi Lee, and Chao-Yung Hsu. Mobile device integration of a fingerprint biometric remote authentication scheme. *International Journal of Communication Systems*, 25(5):585–597, 2012.

[11] Ali A Yassin, Hai Jin, Amin Ibrahim, and Deqing Zou. Anonymous password authentication scheme by using digital signature and fingerprint in cloud computing. In *Cloud and Green Computing (CGC), 2012 Second International Conference on*, pages 282–289. IEEE, 2012.

[12] Kai Xi, Yan Tang, and Jiankun Hu. Correlation keystroke verification scheme for user access control in cloud computing environment. *The Computer Journal*, 54(10):1632–1644, 2011.

[13] Salil P Banerjee and Damon L Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1):116–139, 2012.

[14] Hua-Hong Zhu, Qian-Hua He, Hong Tang, and Wei-Hua Cao. Voiceprint-biometric template design and authentication based on cloud computing security. In *Cloud and Service Computing (CSC), 2011 International Conference on*, pages 302–308. IEEE, 2011.

[15] Shelly and N. S. Raghava. Iris recognition on hadoop: A biometrics system implementation on cloud computing. In *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*, pages 482–485, Sept 2011.

[16] Open source software for creating private and public clouds.

[17] Kernel-based virtual machine (kvm).

[18] Openvpn technologies.