Global Identity Management of Virtual Machines Based on Remote Secure Elements

Hassane Aissaoui, Pascal Urien, Guy Pujolle

Global Identity Management of Virtual Machines Based on Remote Secure Elements

Hassane Aissaoui-Mehrez & Pascal Urien
IMT-TELECOM-ParisTech Institute : LTCI CNRS
Laboratory Network and Computer Science Department,
46 rue Barrault 75634 Paris France
{hassane.aissaoui, pascal.urien}@telecom-paristech.fr

Guy Pujolle
Pierre and Marie Curie University
CNRS LIP6/UPMC Laboratory
4 Place Jussieu, 75005 Paris, France
guy.pujolle@lip6.fr

***Abstract*—The work presented in this paper is part of the cooperative project Security for Future Networks (SecFuNet[1]), which aims to develop a security framework for Cloud Computing. This framework introduces, among many services, authentication and authorization controls for Cloud Computing environments.**

The objective is to develop a highly secure identification scheme based on Authentication and Authorization Infrastructures. A particularly innovative aspect of SecFuNet is related to the global Identity Management (IdM) system. The proposed scheme is important for the security of the entire framework.

The identification system is based on secure microcontrollers. One of the ambitious goals of SecFuNet is to demonstrate and experiment these proposals of IdM in order to reach a standard solution for identifying users and nodes in the SecFuNet architecture. The SecFuNet identity model addresses two kinds of elements: users and nodes. For each of them an identity platform is provided dealing with OpenID Server, and grids of secure elements.

Keywords— Microcontrollers; Secure Elements; User-Centric Identity; Virtualization, Cloud Computing.

I. INTRODUCTION

The SecFuNet project proposes solutions for integrating secure microcontrollers in order to develop a security framework for Cloud Computing. This framework introduces, among its many services: authentication and authorization functions for Cloud Computing environments, based on smartcards and user-centric attribute control policies.

The IdM system will be based on several authentication servers composed by secure microcontrollers. The objective of this paper is to describe the IdM architecture of SecFuNet (hardware and software) achieved for the global IdM system, based on the authentication servers and secure microcontrollers. The proposed SecFuNet IdM makes use of OpenID[2] protocols for establishing trust relationships among users, VMs, IdPs and SPs. To eliminate the vulnerabilities like phishing attacks, this IdM is heavily dependent on protected hardware. User/VM authentication is done directly between smartcards (owned by users or VM) and a grid of secure processors arranged in a SecFuNet IdP. The OpenID servers' security is enforced by grids of secure elements.

The rest of this paper is structured as follows. The related work on the main classes of secure microcontrollers has been studied in Section II. In Section III, we present the SecFuNet Global IdM architecture and we will explain the different layers visibility. We will present afterwards, in Section IV, the user and VM interactions with Grid of Secure Elements operations. The conclusion ends this paper.

II. RELATED WORK

This work describes the development achieved in the global IdM system, based on the OpenID authentication servers and secure microcontrollers developed in SecFuNet project [1][2].

A. Secure Microcontrollers

Two classes of secure microcontrollers have been studied, smartcards and TPMs (Trusted Platform Modules). These electronics chips have different computing capabilities, smartcards usually run a Java Virtual Machine (JVM) and therefore are able to execute complex procedures (such as the TLS protocol), while TPMs are dedicated to the RSA algorithm. However these devices may be used in order to enforce trust for the TLS protocol or to guarantee secure storage for cryptographic keys. These security properties are directly provided by smartcards (thanks to dedicated embedded software), but require additional software components for TPMs. In this document we call secure element a device such as smart card or a TPM, which is able to totally or partially handle the TLS protocol and to realize the secure storage or computing of a cryptographic key. To access smartcard grids driven by proprietary protocols, we recently introduced the RACS protocol [3][4], working over a TLS/TCP/IP stacks which provides a standardized way.

B. OpenID

The OpenID concept was initially developed by Brad Fitzpatrick in 2005. About one million of WEB sites were compatible with this standard in 2012. OpenID is based on

[1] *SecFuNet Project*: "*Security for Future Networks*" is a coordinated project supported by the European Commission through the 7th Framework Programme, and by the Brazilian National Counsel of Technological and Scientific Development (CNPq).

[2] *OpenID Project*: http://openid.net/specs/openid-authentication-2_0.html.

common Web technologies (basically URLs), which makes the protocols light and relatively simple to implement. The OpenID is a set of specifications that define protocols for IdM on the Web.

There are many ways to perform authentication between the user and the OpenID server. The most popular is the simple password mechanism. In the SecuFuNet context, we propose a new model of IdM to authenticate the nodes in Cloud, also we use a strong mutual authentication based on a TLS session running in the EAP-TLS device, in which both (client/VM) and server are identified by their X509 certificates and associated private keys.

III. GLOBAL MODEL OF IDM PROPOSAL

As mentioned above, The SecFuNet architecture distinguishes two classes of entities: Nodes and user. The SecFuNet Users are equipped with two classes of secure elements: EAP-TLS smartcards or PKCS#1 smartcards, used to open TLS secure channels with strong mutual authentication.

The SecFuNet node is divided into three layers of visibility, whose components are identified by a certificate. The first one (the infrastructure layer) directly uses the services of a secure element. Upper layers (N) delegate their authentication to the layer N-1 (i.e. layer N-1 identity and private keys) but need a cryptographic token establishing the logical link between the N and N-1 layers. The interaction between SecFuNet components dealing with identities and tokens, implies the existence of a root entity that delivers certificates, and which is referred to as the Identity and Access Management (IAM). The IAM is the root Certification Authority of the SecFuNet architecture. The Global IdM architecture of SecFuNet is illustrated by "Fig. 1,".

The IAM delivers certificates to users, administrators of the infrastructure, platform, and applications layer. According to the OpenID architecture, services are interfaced by service providers, which require authentication from OpenID servers and tokens. OpenID authentication servers are equipped with certificates provided by the IAM entity.

The SecFuNet Users (or Administrators) are associated with certificates, which instantiate their identities (ID_{user}). They are equipped with secure elements, able to establish TLS connections to remote servers. Both entities (client and server) hold a certificate and a private key in order to realize a strong mutual authentication. This mechanism establishes trusted TLS channel between users and SecFuNet WEB servers. Trusted TLS channels are also used for strong mutual authentication with OpenID authentication servers; these mechanisms are detailed in sections VI.

According to the OpenID terminology SecFuNet services are accessed through service provider (SP) portals, which need an identity provider server (IdP) in order to achieve user's authentication.

It is important to notice that EAP-TLS and PKC#11 smartcards may be merged in the same physical device, in one or two JavaCard applications [5][6]. Tokens are delivered by dedicated SP and IdP entities. Tokens may also be stored in secure elements for services that need a user centric model, such as the Microsoft Cardspace paradigm.

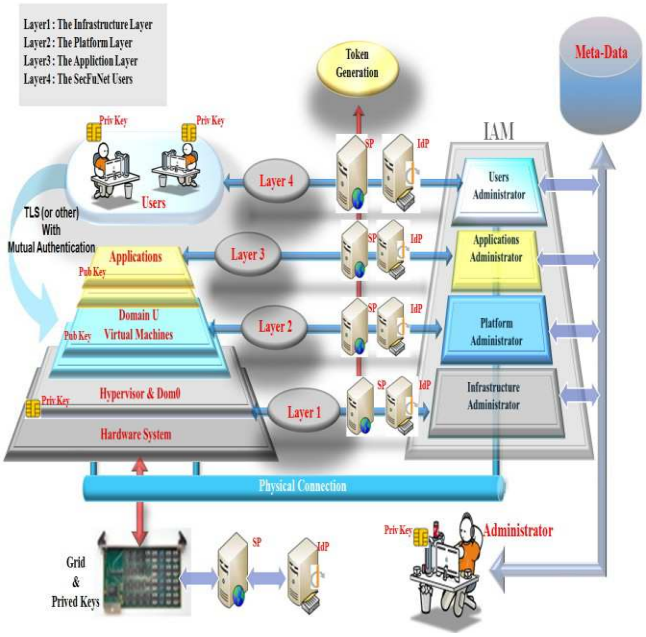

Fig. 1. SecFuNet : Global Model of IdM Architecture

A. The SecFuNet Nodes

Nodes are composed of three main layers of visibility, the infrastructure layer, the platform layer and the application layer. Each of them is associated with an administrator in order to manage it.

The Infrastructure Layer: Includes hardware resources (servers, grid of secure elements GoSE) and hypervisors (HV), running in the Domain0 (in the sense of Xen technology).

Each hypervisor has a certificate, which instantiates its identity (ID_{HV}). It is optionally associated to a secure microcontroller storing its private key. In order to enforce a high trust and security level, TLS sessions with strong mutual authentication between HV and administrators are mandatory for all virtual machines (VM) transfer operations based on a WEB interface. Administrators are equipped with secure elements. However, and for legacy reasons, classical SSH sessions working with a private key on the HV side, and a login/password credential on the administrator side could be still supported.

The Platform Layer: Is a set of Virtual Machine (VM), running in the domain (DomainU in the sense of Xen technologie). Each VM holds a certificate (that instantiates its identity ID_{VM}), but its private key is stored in a GoSE [1][2].

For example each VM is associated to a secure element (smartcard) plugged in a SIM array "Fig. 2,". All cryptographic procedures dealing with secret keys (such as signing, symmetric encryption or decryption) are performed by the secure element linked to the VM.

The smartcard grid is interfaced by a service provider (SP), which may be OpenID compliant or a classical WEB interface. The protection of asymmetric cryptographic key is a critical issue for the SecFuNet Node.

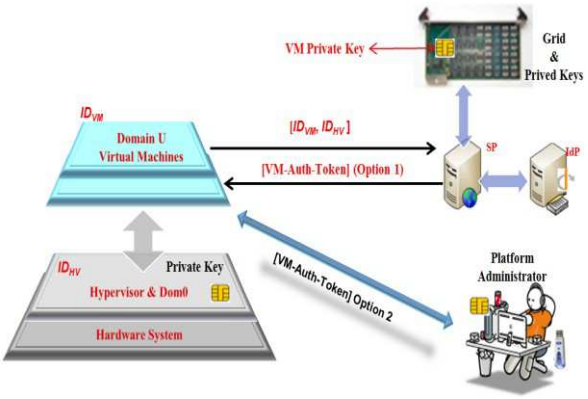

Fig. 2. Operations dealing with the VM private key in SecFuNet

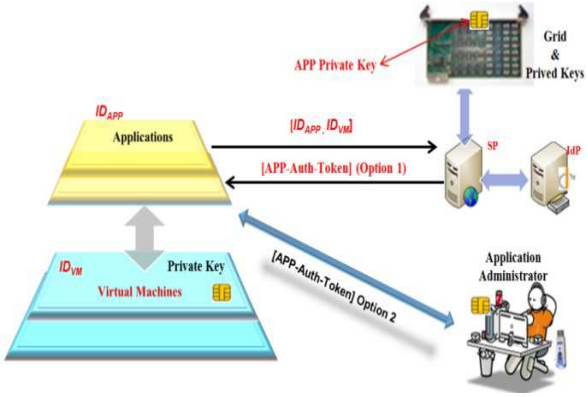

Fig. 3. Operations dealing with the Application private key in SecFuNet

The authentication between the VM and an OpenID authentication server requires the HV private key and VM-Auth-Token (VM Authentication Token). The VM-Auth-Token establishes the proof that the VM (identified by its certificate ID_{VM}) is attached to the HV (identified by its certificate ID_{HV}).

Conceptually the VM-Auth-Token is a signed piece of information (such as certificate) comprising ID_{VM}, ID_{HV} and a validity date; it works with HV public key. Therefore TLS sessions dealing with this certificate imply the use of the HV private key: $VM\text{-}Auth\text{-}Token = \{ID_{VM}, ID_{HV}, Validity\text{-}Date\}Sign\text{-}Priv_{HV}$.

The SP, which controls the grid services, needs three elements in order to authorize an operation with the secure element bound to a VM: a successful authentication with the authentication server dealing with the ID_{HV}, the ID_{VM}, and the possession of the VM-Auth-Token attribute. This last parameter may be dynamically provided by the OpenID server (Option1) or statically assigned by the platform administrator during the transfer of the VM (Option2).

- The Application Layer: This layer is a set of guest applications (APPs) executed by a virtual machine (VM), which are managed by dedicated administrators. Applications are identified by certificates (ID_{APP}) and a secret key stored in the secure element grid. The smartcard grid is interfaced by a service provider (SP), which may be OpenID compliant or a classical WEB interface. The authentication between the APP and an OpenID authentication server requires the VM private key and APP-Auth-Token (Application Authentication Token). This token establishes the proof that the APP (identified by its certificate ID_{APP}) is attached to the VM (identified by its certificate ID_{VM}).

Conceptually the APP-Auth-Token is a signed piece of information (such as certificate) comprising ID_{APP}, ID_{VM} and a validity date; it works with VM public key. Therefore TLS sessions dealing with this certificate imply the use of the VM private key: $APP\text{-}Auth\text{-}Token=\{ID_{APP}, ID_{VM}, Validity\text{-}Date\}Sign\text{-}Priv_{VM}$

The SP, which controls the grid services, needs three elements in order to authorize operations with the secure element bound to an APP. A successful authentication with the authentication server dealing with the ID_{VM}, the ID_{APP}, and the possession of the APP-Auth-Token attribute "Fig. 3,".

This last parameter may be dynamically provided by the OpenID server (Option 1) or statically assigned by the application administrator (Option 2).

IV. Grid of Secure Elements Operations

In order to remotely work with its private key, the VM must interact with the SP that interfaces the secure element grid. The VM is authenticated by the ID_{VM}, the ID_{HV} (and the HV private key) and the VM-Auth-Token. Two use cases are described: the use of an OpenID platform, and the use of RACS Protocol [4] compliant smartcard grids.

A. OpenID use case

As illustrated by "Fig. 4," the VM may access to the private key of the hypervisor stored in a secure element. It establishes a session with the grid service provider, and provides the ID_{HV}, ID_{VM} and VM-Auth-Token parameters (Step 1). It also request cryptographic operation with the grid, such as encryption or decryption with the VM private key. It is thereafter redirected toward the OpenID server (Step 2) that performs the authentication process with the HV secure element (Step 3).

Strictly speaking the OpenID server, authenticates the hypervisor, and is not aware of an existing functional link with the VM. At this step the service provider knows that the remote entity has access to the HV private key and is supplied with the appropriate credential (VM-Auth-Token). Step 5-6, the grid service provider realizes the requested operation and returns the result (Step 7).

Obviously the session between the VM and the grid service provider must be secured by the TLS protocol. In order to get a greater security level, the grid service provider may pack the result in a container, which is made of three parts:

- A header, which is the encrypted value of a symmetric AES key, with the HV public key, according to the PKCS#1 standard.
- A body, which is the encrypted value of the result of the requested operation, according to an AES-CBC procedure.
- A trailer, which is the signature by a trusted authority (identified by its public key) of the header concatenated to the body, according to the PKCS#1 standard.

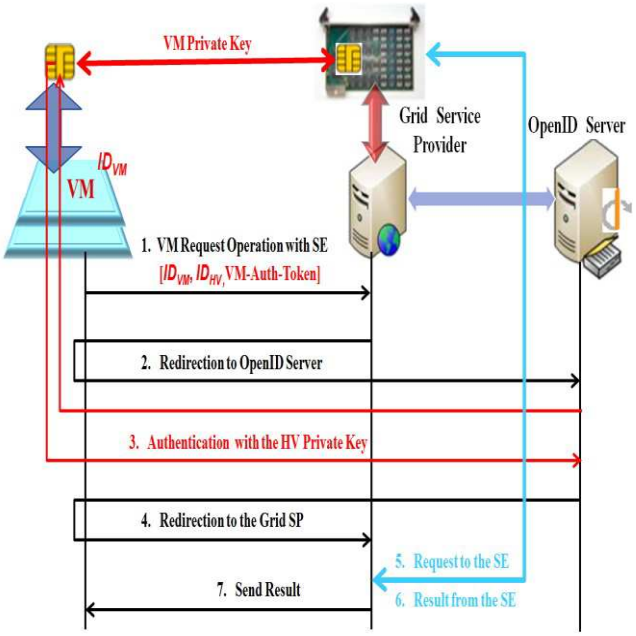

Fig. 4. Grid Service Provider in an OpenID Context

A trust issue may exist between the grid SP and the VM, because they do not necessarily share the same ring of confidence. In that case the container could be directly generated by the VM secure element, previously bound to the ID_{HV}. This operation is performed by secure element administrator thanks to Global Platform mechanisms grid.

B. Grid Administration

The Grid administration is a sensitive topic, because the Secure Elements are not necessarily managed by the grid service provider. Hopefully the Global Platform Standards [9] provide all the mechanisms needed for a remote and secure management.

Secure Elements are equipped with symmetric secret keys used for mutual authentication purposes; secure channels are available for embedded application, loading, activation or deletion.

Thanks to these facilities, secure elements may be bound to an ID (such as ID_{HV} or ID_{VM}) and protect their produced information by containers, which can be only decrypted by the authorized entity (either a hypervisor or a virtual machine).

However, a specific access is needed for allowing these operations. A TLS proxy could be used such as RACS (Remote APDU Call Secure protocol) [3][4], in order to transport ISO7816 commands and to convert them in a format compatible with the grid protocol.

V. Conclusion

The architecture proposed and implemented in this paper presents a solution for the IdM in the context of the SecFuNet project, which aims to develop a highly secure identification scheme, for Cloud Computing, based on the OpenID protocols integrated with secure elements (smartcards and the secure elements grids).

The SecFuNet identity model addresses two kinds of elements and some layers of visibility. For each of them an identity platform is provided dealing with OpenId server, secure elements and grids of secure elements.

This integration provides a higher level of security when compared to traditional password authentication, and establishes trust relationships among the resources (users, IdPs and SPs), and eliminates the vulnerabilities like phishing attacks. This IdM is heavily dependent on those classes of secure elements. User strong authentication is done directly between smartcards (owned by users) and a grid of secure processors arranged in a SecFuNet IdP. The OpenID security is also enforced by these grids.

For the future, we intend to extend the base protocols to enable the federated IdM as well as a user-centric attribute control approach, i.e. establishment of trust relationships and transposition of attribute assertions to access services on all SecFuNet domains. Another ambitious goal of SecFuNet is to demonstrate and experiment with these proposals of IdM in order to reach a standard solution for identifying users and nodes in the SecFuNet architecture.

Acknowledgment

This work has had financial support from CNPQ through process 590047/2011-6 (SecFuNet project) and also through processes 307588/2010-6 and 384858/2012-0. We also thank CAPES for the financial support with PhD scholarship.

References

[1] Secfunet, "Infrastructure of the authentication server", Deliverable 2.1

[2] SecFunet, "Array and software of authentication server", Deliverable 2.2

[3] Urien, P., "Cloud of Secure Elements, Perspectives for Mobile and Cloud Applications Security", First IEEE Conference on Communications and Network Security" IEEE CNS 2013, 16-19 october 2013, DC USA

[4] Urien, P, 'Remote APDU Call Secure », draft-urien-core-racs-00.txt, August 2013

[5] Jurgensen, T.M. et. al., "Smartcards: The Developer's Toolkit", Prentice Hall PTR, ISBN 0130937304, 2002.

[6] Chen, Z., "Java CardTM Technology for Smartcards: Architecture and Programmer's (The Java Series) ", Addison-Wesley Pub Co 2002, ISBN 020170329.

[7] Urien, P., Pujolle, G., "Security and Privacy for the next Wireless Generation", International Journal of Network Management, IJNM, Volume 18 Issue 2 (March/April 2008), WILEY.

[8] IETF draft, "EAP-Support in Smartcard", draft-urien-eap-smartcard -25.txt, July 2013.

[9] Global Platform standards, http://www.win.tue.nl/pinpasjc/docs/CardSpecv2.1.1v0303.pd