VMITLP : A Security Protocol Towards a Trusted Launch Process of a User Generic Virtual Machine Image on a Public Cloud IaaS Platform

Chawki EL BALMANY, Ahmed ASIMI, and Zakariae TBATOU

***Abstract*—The Infrastructure-as-a-Service (IaaS) cloud model is a component of the cloud architecture which allows provisioning user's virtual machines. The Iaas model offers a pool of computing resources in the form of services so that cloud user has the ability to run their own virtual machine images (VMIs). The main problem identified in this approach is that users are still reluctant to admit the security policy of IaaS Cloud Service Provider (CSP), which does not guarantee the confidentiality and integrity of the user VMI. In this article, we thoroughly cover the process of a generic user VMI instance launch on a trusted cloud platform based on Trusted Cloud Computing (TCC). For this reason, we have designed a VMITLP VMI Trusted Launch Protocol which aims to ensure a secure connection of user VMI. Our protocol only runs on a trusted platform that has been booted in a trustworthy state. In order to strengthen the robustness of our protocol, we have ensured essential security requirements, such as trust and authentication throughout the launch process.**

***Index Terms*—IaaS , Cloud Computing , Virtual Machine , Security , TPM , Trust.**

I. Introduction

With the normalization of the Internet, the development of broadband networks, the computer world has experienced the exploitation of a new paradigm being a solution that meets the needs of the evolution of the industry, the Cloud Computing (CC). the architecture of the cloud is defined as a distributed system that provides a powerful computing and storage, defined by three mainly types of services:

- IaaS (Infrastructure as a Service) which defines the hardware infrastructure.
- PaaS (Platform as a Service) being the provider of the cloud service which administers the operating system and its tools.
- SaaS (Software as a Service) which represents the applications.

The Infrastructure-as-a-Service (IaaS) Cloud Delivery Model defined by NIST (National Institute of Standards and Technology) [1], according to which, this model holds a wide variety of resources delivered as aggregated and managed services under full control of its end-users. These services come with advanced features that are most relevant in terms of form of storage, network, compute, pay-per-use and on-demand provisioning. Market leaders, such as Amazon EC2 and Microsoft Azure, give the IaaS model design a symbolic rating for configuring virtualized hardware and software services designed to allocate user's operating systems and applications to the desktop within virtualization.

Based on [2], the security of user's virtual machine image (VMI) is considered a mutual accountability between the user and the CSP for decreasing the vulnerabilities amount and applying security implications in order to improve the essentiel security properties such as the confidentiality, integrity and availability of the VMI data and its related applications. To do so, the major problem discussed in this article is given an encrypted stored VMI onto a cloud physical disk, we address the security of the generic VMI instance launch process until to be mounted on a cloud host identified and trusted that meets the security requirements defined by the cloud user and the supported policy by CSP with no violation of Service Level Agreement (SLA).

Recently, Trusted Computing (TC) have been emerged to secure the IaaS model infrastructure. TC's aim is to promote the trustworthiness of computer system and guarantee the behaviors of computer in expected ways. TC supports the technology of Trusted Platform Module (TPM) sustained by Trusted Computing Group (TCG) [3]. TPM is a hardware module (namely a ship) that can be used as a trust anchor for software integrity verification in open platforms that also offers protected storage for sensitive parameters. Trusted Cloud Computing (TCC) represents a combination between CC paradigm with TC. It provides a sealed trustworthy environment for user VMI based on TPM and Remote Attestation (RA) to prove for other parties the trustworthiness of the cloud platforms. TCC is able to establish trust instance boot, security isolation, key exchange management and remote attestation which leads to enforce the IaaS security. Thus, Google recently encourages this research field by supporting TC in its own recent CC industry.

Moreover, launch VMI process security depends not only on preserving a trustworthy platform but it also requires applying convenient cryptographic techniques to preserve the identification and access control of only granted CSC credentials to related VMI instance and preserving its confidentiality and data privacy. This part remains undiscussed in most researches broaching TCC.

To address the aforementioned issues, as depicted in

Manuscript received April 07, 2021; revised October 6, 2021.

M. EL BALMANY is a Ph.D member of the Laboratory of Computer Systems and Vision (LabSIV), Faculty of Sciences, University Ibn Zohr, Agadir, Morocco. e-mail: (chawki.elbalmany@gmail.com).

Prof. ASIMI is a full professor and coordinator of the Laboratory of Computer Systems and Vision (LabSIV), Faculty of Sciences, University Ibn Zohr, Agadir, Morocco. e-mail: (asimiahmed2008@gmail.com).

Dr.TBATOU is a full professor in Universiapolis and member of the Laboratory for Sustainable Innovation and Applied Research, Technical University of Agadir, Qr Tilila, Agadir 80000, Morocco e-mail: (tbatou.zakariae@gmail.com).

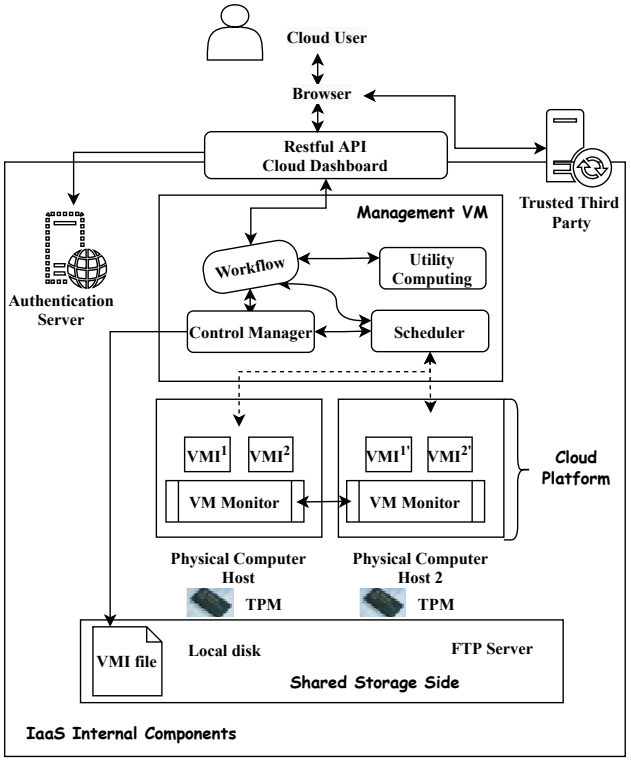

Fig. 1. The Proposed architecture of IaaS model layers

Figure 1, this paper proposes a VMITLP which is a security protocol designed to cover thoroughly a secure launch VMI process onto a TCC for public IaaS model. In addition, VMITLP has been exposed to discuss the aforementioned techniques and handle with known recent attacks that hamper the discussed process. CSP is responsible for maintaining secure communication and cryptographic exchange of encryption keys between internal cloud entities, as long as TTP which is the user interim in the bidirectional communication between In/Out-Cloud.

The remainder of this paper is represented as follow: Section II defines a wide view of the recent researches concerning TCC, authentication and security properties related to IaaS model launch VMI process. Section III briefly describes the principle of the conceived VMITLP. Section III-B covers widely the VMITLP life-cycle from the generation of the VMI instance to its launch. Moreover, a security analysis is discussed in Section IV to compare VMITLP with recent related alternatives. Finally, this paper is achieved in section V by a conclusion defining our future perspectives.

II. RELATED WORK

In the literature, CC security remains a typical subject of discussion and research because of its complexity as a modern paradigm that supports multiple critical layers. Some researches [4], [5], [6] dealt with a practical workflow management policy by the CSP to improve the Quality of Services (QoS). The latter remains a typical requirement to establish a secure communication between the cloud layers and also to satisfy the requirements of tenants. With respect to the VMI launch process, Vaquero et al [7] gave another comprehensive survey based on the seven main threats presented in CSA [8]. Some recent works adressing the problem of securing virtual machines from malicious behaviors.

TABLE I
ACRONYMS

Notations	Meaning
VMI	**Virtual Machine Image**
VMIF	**Virtual Machine Image File**
H(VMI)	**Hash of VMI**
CSC	Cloud Service Client
TTP	**Trusted Third Party**
Pk_{TTP}	**TTP's Public Key**
Prk_{TTP}	**TTP's Private Key**
R_{req}	**Requested Resources of VM Instance**
R_c	**Computer Host Resources**
TPM	**Trusted Platform Module**
AIK	**Attestation Identity Keys**
EK	**Endorsement Keys**
CH	**Computer Host**
T-CH	**Trusted Computer Host**
PK_{CH}	**CH's Public Key**
PrK_{CH}	**CH's Private Key**
DK_{VM}	**VM Disposable Key**
N	**Nonce**
IML	**Integrity Measurement List**
S	**Scheduler**
CSP	**Cloud Service Provider**
CAS	**Cloud Authentication Server**

Virtualizing the Trusted Platform Module (vTPM) [9] is the first TC technology used for virtualization. Recent researches relied on the virtualization-type vTPM such as [10], [11] by securing SW and HW respectively. VMM-type based vTPM has been also discussed in [12]. The authors proposed a KVM-based vTPM where vTPM instances are emulated by a QEMU as a stub domain over KVM using HW-assisted virtualization. In addition, some researches broach Dynamic Root Trust of Measurement (DRTM), also called late-launch, which minimizes the Trusted Computing Base (TCB) size by excluding boot loader, OS, and able to execute BIOS update [13]. Authors in [14] proposed an approach based on container virtualization to build trust between CSC and CSP by allowing CSP to provision IaaS without accessing to CSC's data. In [15], authors proposed an End to End (E2E) framework for trusted cloud infrastructure based on vTPM to establish a secure communication. Trusted Execution environment (TEE) gave the CSC the opportunity to secure its related Sensitive Application oN clouD (SAND) [16].

Advanced trusted cloud model alternatives have emerged the use of remote attestation TPM keys to achieve a trustworthy cloud computer node based on sealed keys such as [17]. Authors in [18] proposed a Trusted IaaS Platform (TCCP) to run user's virtual machine on a secure hardware and software stack with a remote, untrusted host and migrate VMIs. TCCP presents a concept for launching and securely migrating virtual machines, especially the use of a TC in a trusted environment between the parties involved. However, the overhead of the management is intolerable for massive scale cloud. For this reason, Excalibur is presented in [19]. Excalibur uses attribute-based encryption, which reduces the overhead of key management and improves the performance of the distributed protocols employed.

The authors in [20] proposed a new security scheme "Encrypted Virtual Disk Images in the Cloud (EVDIC)" for encryption protection of disk images stored in cloud servers.

EVDIC also includes the security of the key management and exchange processes by generating a symmetric key based on the Public-Based-Key-Derivation-Function-2 (PBKDF-2). They integrate EVDIC with OpenStack, an open source cloud platform widely used around the world. This work covers standard cryptographic security techniques for launching and storing user VMI. In [21], the authors described a protocol for launching secure virtual machines on public IaaS using secure computing techniques. To ensure that the requested virtual machine instance is launched on a host with attested integrity, the tenant encrypts the image of the virtual machine (with all data injected) with a symmetric key sealed to a particular configuration of the virtual machine host reflected in the TPM platform configuration register (PCR) values. [22] proposed a trusted launch protocol for VM, which uses binding and sealing to provide integrity guarantees to CSC. The protocol does not require secure prepackaging of VMI on CSC side. [23] proposed a work on vulnerability assessment and patching by integrating in-VM-assisted agent-based malware detection (AMD) framework for securing high-risk VMIs in cloud.

Authentication and access control remain a typical security requirements in launch VMI instance process mainly in the IaaS platform. In [24], authors proposed a model for identity authentication and access control, and TC is used to strengthen them. Service in this model may need multiple VMs. These VMs have to register their identities into a Service Authentication List (SAL). If VM can no longer support one service, VM will inform Configuration Update Module (CUM), which will select a new VM to replace the old one before updating SAL. [25] proposed a Cloud Verifier, which play the role of a verification proxy, to verify the integrity and access control enforcement abilities of CNs. Their verification protocol can be summarized as follows: (1) CSC sends quote request to Cloud Verifier. (2) Cloud Verifier verifies itself to CSC, and forwards quote request to CH. (3) CH sends quote reply to Cloud Verifier, which forwards it to CSC. (4) CSC starts its VM after verifying quote reply.

III. VMITLP ARCHITECTURE DESCRIPTION

A. VMITLP *Overview*

The proposed VMITLP constitutes a security protocol designed for Public IaaS cloud model. VMITLP is dedicated to preserve and guarantee the properties and security requirements namely: authentication, confidentiality, integrity, availability and trust throughout the generic VMI instance launch process by deploying existing security and cryptographic techniques besides TCC.

The entities involved in this proposed protocol are divided into three main phases as defined in Table II.

User: The cloud client which accesses the remote guest operating system and is responsible for verifying the security of its environment by verifying the integrity of the data.

Scheduler: An internal cloud entity that is responsible for choosing the right cloud host, able to launch the image of the user's virtual machine and redirect the cloud manager to encrypt / decrypt the requested VMI.

Computer Host (CH) : A physical cloud machine with a Trust Platform Module provisioned by a VMM. The CH is responsible for launching the user VMI with specific security requirements and policy.

Cloud Authentication Server (CAS): A cloud authentication server that is responsible for generating a auth-token on every VMI instance launch request.

Trusted Third Party (TTP): has expertise and capabilities that user may not have. TTP is considered as an interim of the cloud user nearby all internal communication with CSP. It is trusted to assess and expose risk of the launch process upon the user's request.

VMI instance launch process's main aim is to be able to mount a user VMI (O/S Operating System) on a trusted cloud platform that meets the user's needs. This VMI is beforehand stored encrypted on a cloud physical disk. First, based on the security principle discussed in [2], the security of the IaaS model is a mutual responsibility between CSP and the cloud user through the quest to pinpoint security vulnerabilities and critical issues hindering the scalability and flexibility of the IaaS model. Thus, as described in Table III, VMITLP covers the classic VMI launch process by subdividing the linearity of the process into phases describing and completing each appropriate security property:(i) Authentication phase of CH nearby TTP, divided into 4 steps. ii) Establishment of trust phase between TTP and T-CH. iii) Pre-launch phase of generic VMI instance. In meanwhile, maintaining the availability or the consistency of the process has been implemented by setting up recovery or regeneration points in a safe state in case of the event of expiry or unapproved cryptographic verification or unauthorized access failure.

1) Phase 1: Authentication of CH by TTP
   - Step 1: Generation of VMI Instance
   - Step 2: Selection of CH
   - Step 3: Identification of CH nearby CAS
   - Step 4: Identification of CH nearby TTP
2) Phase 2: Trust Settlement between TTP and Trusted-CH.
   - Step 5: Trusted-CH and TTP based on TPM
3) Phase 3: Pre-Launch VMI Instance
   - Step 6: Decrypted VMIF and Launch Instance.

The generic VMI launch process has been described in detail in this section. The launch process constitues the generation of the launch instance request by the user to the instance mounting on a T-CH. The architecture depicted in Figure 3 comprises three basic phases for VMITLP. The basic phases have been applied to deploy a trusted environment on which the instance will be mounted. The first mentioned phase "Authentication of CH by TTP" is represented by 4 steps. The primary purpose of this phase is to first identify the candidate cloud CH locally in the vicinity of known Cloud Authentication Server (CAS), and then verify its identification through the TTP. After that, the second phase titled "Establishing trust between TTP and T-CH" is deployed to verify the reliability of the Cloud platform based on TPM integrity certification near the TTP. Therefore, the pre-launch phase is involved in challenging the secure cloud platform within the user to finally approve the accuracy and validation of the entire process which requires a recovery phase to maintain the scalability and accuracy of the process. As shown in Table III and Figure 2, once the verification function is not approved, the launch process

TABLE II
PROPOSED VMITLP COMPONENTS AND LAYERS DESCRIPTIVE

Entities	Components	Description
User / Browser	Device (Laptop, Mobile...)	Launch/Store VM Image
Cloud IaaS Model	Scheduler	Schedule User Request & VM Instance
	Computer Host	Launch VM Instance
	Cloud Authentication Server	Auth-Token Generator
External Server	Trusted Third Party	Attests T-CH

TABLE III
LAUNCH INSTANCE PROCESS ON VMITLP DESCRIPTIVE

N_Phase	*Phase*	*N_Step*	*Objective*	*Verif_Fct()*	*Failure_P*	*Recovery_P*
1	***Authentication of CH by TTP***	***1***	***Generation of VM instance***	**-**	**-**	**-**
		2	***Selection of CH***	**-**	**-**	**-**
		3	***CH Identification by CAS***	***Verif_1()***	✓	***Step 2***
		4	***CH Identification by TTP***	***Verif_2()***	✓	***Step 2***
2	***Trust Settlement between TTP & T-CH***	***5***	***Trust between TTP & T-CH***	***Verif_3()***	✓	***Step 2***
3	***Pre-Launch of VMI***	***6***	***Launch of VMI instance***	***Verif_4()***	✓	***Step 1***

regenerates into a decisive point of recovery to reconstruct the process in a previous reliable phase. The recovery feature allows the proposed VMITLP launch process to attend a trusted CH without involving the human intervention in case of failure.

B. Detailed Proposed VMITLP

In order to preserve a trustworthy cloud platform to mount user VMI, the main aim in this section is to describe in detail the life cycle of the VMI launch process from the generation of VMI instance to its mount on a trusted cloud host.

For this purpose, the proposed VMITLP is divided into three phases:

1) Authentication Phase of CH nearby TTP: The authentication phase between CH and TTP is a key phase for VMI instance pre-launch. In other words, a cloud CH candidate identifies itself to the TTP with respect to the CAS in order to mount the user instance on a trusted platform. TTP represents a sealer (verifier) in order to ensure the authenticity and integrity of the user instance. In addition, this phase remedies two major vulnerabilities throughout the process, the CH considered an outsider is verified nearby TTP based on mutual authentication with CAS. Thus, the CH considered as a malicious internal CH equipped with TPM is authenticated nearby TTP (i.e. considered as a Certification Authority) during the generation of the instance boot process. This phase is characterized by 4 steps as well as failure points or check-functions. The launch process regenerates automatically to a previous safe step in case of any break or false verification or unauthorized access.

Furthermore, before the launch of VMI instance, it is considered that three parameters have been already established for key exchange communication beforehand in the current running instance session. This step is refered as **StepX**.

- Sharing two secret keys and per session. The first secret K_{Ss} key between TTP and the user and the second $K_{Ss'}$ key between TTP and the CAS.
- Generation of an "auth-token" token per session by the CAS to guarantee the freshness and authenticity of the user session.
- The CAS now computes $U_{idf} = E_{K_{Ss}(auth-token)}$, which is the auth-token encrypted by the K_{Ss} key.

Step 1: Generation of the VMI Instance.

- The browser (i.e. user) :
  - Generates a nonce N, a data structure T via a script that represents the encryption of $N \| H(VMI)$ with TTP's public key.
  - Sends U_{idf}, T, URL_{TTP}, and the process launch request to the cloud.
  - Thus, sends the data structure T to TTP.

Step 2 : Selection of CH.

- The Scheduler after receiving the converted request, it:
  - Selects the appropriate available CH that meets the required properties of the VMI. The selected CH resources must match the resources required by the user (CPU, hard disk, and RAM) with some advanced selection methods ;
  - Sends a request for generation of attestation keys to the "intended cloud platform" (CH), in parallel sends T and U_{idf};
  - Sends back in parallel U_{idf} and the identifier of the candidate CH to the CAS for first internal prior identification.
- CH then sends the value U_{idf} to the CAS for the verification of its authenticity.

Step 3 : Identification of CH nearby CAS.

- CAS:
  - First compares the value U_{idf} with the value $CH - U_{idf}$ (i.e. U_{idf} emitted by CH). This comparison is noted by **Verif_1()**;
    - IF **Verif_1()** returns FALSE: CH is considered as an external malicious. CAS :

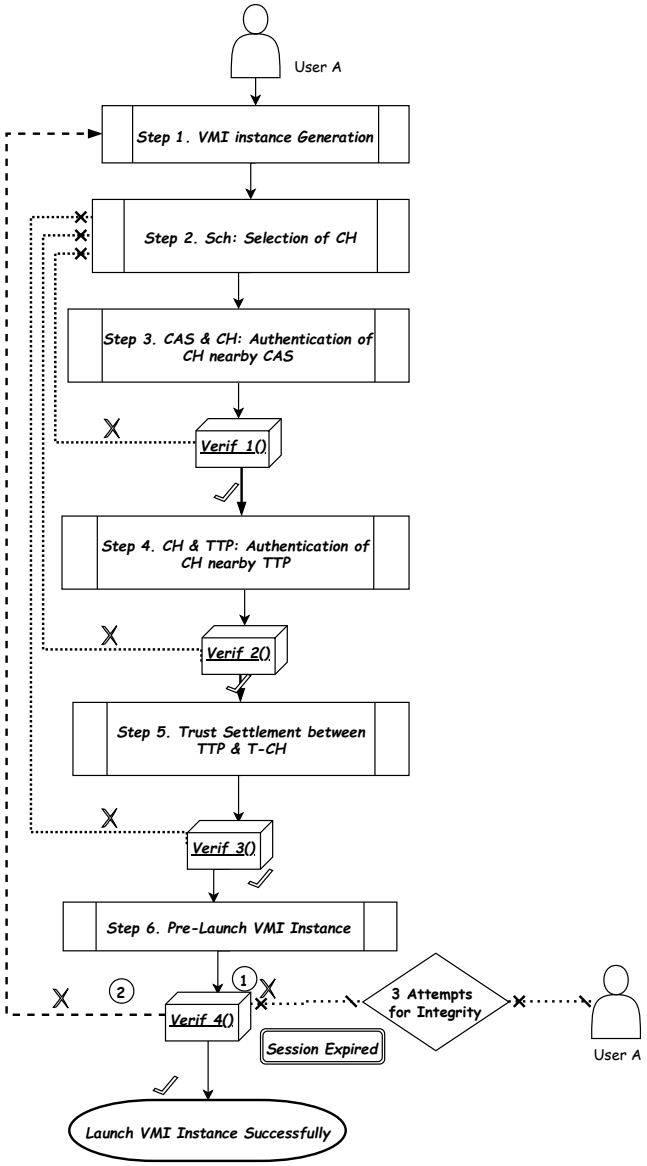

Fig. 2. Launch VMI Instance Process Architecture

· Enquires the scheduler to select another CH candidate again. The recovery phase regenerates from the **Step 2**.

∗ IF TRUE : CH is successfully identified nearby CAS. This latter:

· Computes the Val' value which represents the auth-token encryption by the $K_{Ss'}$ key ;

· Sends Val' to CH.

Step 4 : Identification of CH nearby TTP

- CH:
  - Computes the current state of the PCR during the "boot process";
  - Signs the value of the PCR and Val' for its freshness with the TPM SK_{AIK} certification key generated by the command TPM_QUOTE();
  - Sends the signed value and AIK-Certif to TTP (optionally known by its URL).
- TTP considered as CA :
  - Identifies the appropriate public key to the CH via its AIK-Cert;
  - Decrypts the value signed by the public signature key of previously registered CH to find the value PCR and Val' ;
  - Now decrypts the value Val' by the key $K_{Ss'}$ to find the auth-token;
  - Compares the decrypted auth-token value with the shared auth-token value with CAS when accessing the cloud; This comparison is noted **Verif_2()**:

∗ IF **Verif_2()** returns FALSE: CH is an internal malicious.

· TTP enquires the CAS to choose another CH candidate again.

∗ Otherwise: CH is successfully identified with TTP. This last :

· Returns to the CH a request for attestation and generation of PK_{BIND}. CH thus passes to the validity of dignity of trust.

2) Trust Settlement Phase between TTP & T-CH: The scheduler is conceptually dedicated to intercept the user's launch request and choose the corresponding CH capable of meeting the user VMI required resources. After been identified neaby TTP, CH passes to the phase 2 which could be trusted to mount the requested instance. Whilst CH using TPM-based remote certificate as adopted in [18], it sends a certificate to the TTP for validation. CH certificate public key is registered within TTP, which represents a integrity and confidentiality Certification Authority (CA). TTP issues an identity certificate for each subscribed cloud CH, called the AIK certificate in accordance with the principles of the attestation key of TPM. To make this CH more meaningful, previously approved by TTP, a secure boot of VMI start-up in which the hash, using sha-1 (implemented in the TPM) of each code component loaded before execution is recorded in the Platform Configuration Registry (PCR) [17]. The PCR hash values calculated in each measurement code are stored in an event log file called the Integrity Measures List (IML). In addition, PCR contains the linked hash of all the measures in the IML. Any changes or alterations during the boot process can be detected by comparing the running code measure against the saved value.

Step 5: Establishment of trust between T-CH and TTP.

- CH:
  - Computes the list of IML integrity measures;
  - Retrieves a data structure TPM_CERTIFY_INFO by calling the TPM_CERTIFY_KEY() command to the boot process containing the non-migrable, PK_{BIND} certified TPM key and the PCR-INFO value;
  - Computes Verif which represents the encryption of PK_{BIND} and IML with the public key of TTP Pk_{TTP};
  - Forwards AIK, TPM_CERTIFY_INFO, Sign (TPM_CERTIFY_INFO)SK_{AIK}, $Verif$ and T to TTP
- TTP:
  - Validates the AIK certification;
  - Decrypts $Verif$ value by its private key $Prk_{TTP}$to have PK_{BIND} and IML;
  - Computes PCR-INFO based on the IML;

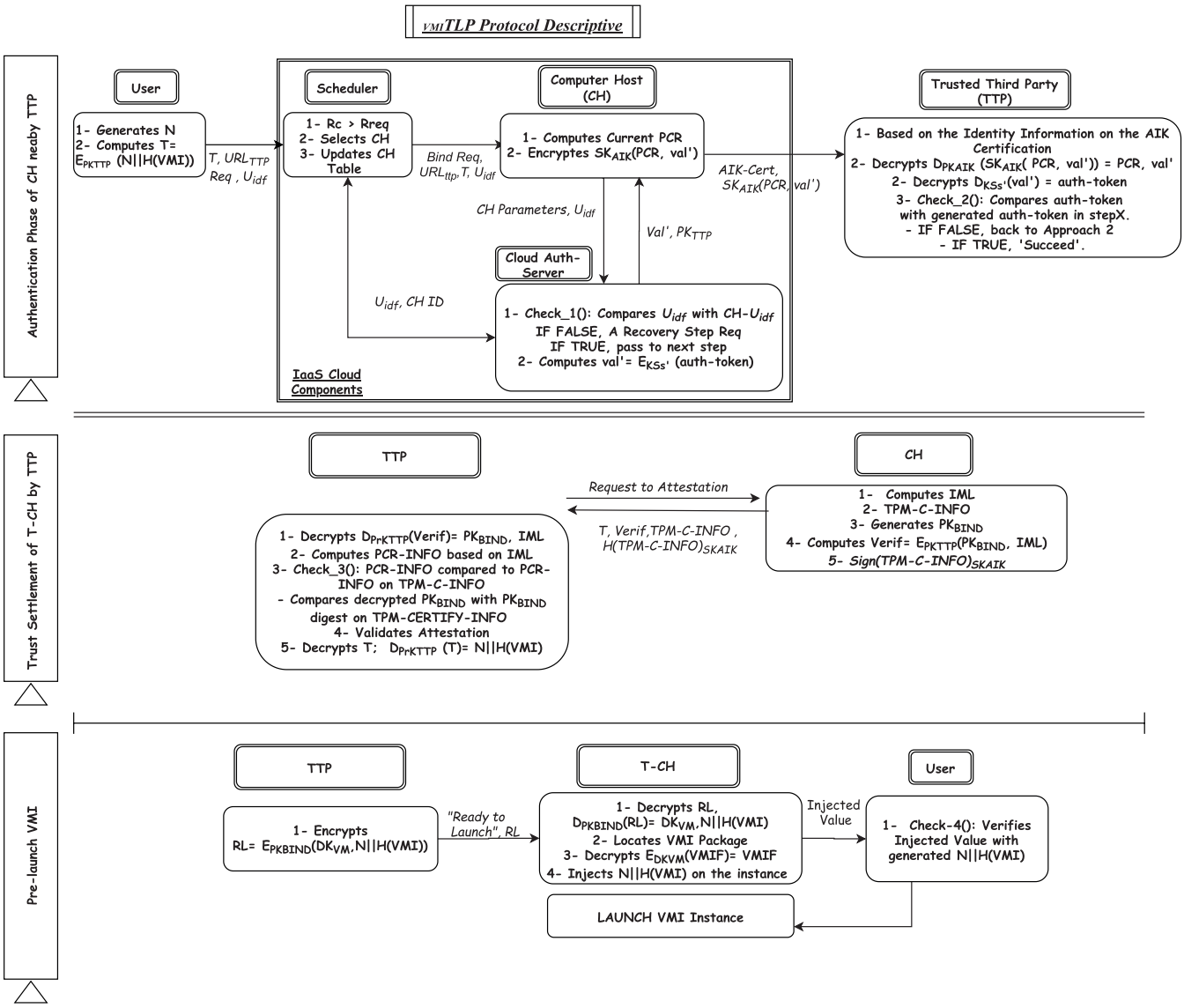

Fig. 3. VMITLP Process Description

– **Verif_3()** : First compares PCR-INFO with the PCR hash value deduced in TPM_CERTIFY_INFO and PK_{BIND} decrypted with PK_{BIND} deduced from TPM_CERTIFY_INFO .
  * IF FALSE: Resumption of selection of a new CH candidate host. The process is regenerating since **Step 2**;
  * IF TRUE, T-CH is now trustworthy and ready to launch the VMI.

3) Pre-launch phase of VMI instance: In this phase, it is previously considered that the package of the user virtual machine has been already stored encrypted on the storage cloud server. Otherwise, after the T-CH host machine is trustworthy in the previous phase, it becomes ready to launch the VMI instance.

During the pre-launch phase of the VMI, the goal is to use a VMIF decryption key based on the appropriate user ID. This key denoted DK_{VM} is a symmetric and disposable key that is used to encrypt / decrypt the VMIF in each cloud scenario (i.e. Two types of VMI instance life-cycle). i) VMI instance launch scenario, ii) Storage scenario of VMI. DK_{VM} is a disposable key although every T-CH ready to launch VMI, it gets rid of DK_{VM} during the Shut Down of the VMI instance. In other words, disposable key means that no CH can decipher the VMIF even if it becomes malicious.

On the other hand, this key is exchanged between T-CH and TTP via the key PK_{BIND} generated during the current session which gives a freshness to the key DK_{VM}.

VMI instance pre-launch in the trusted selected CH is described as follows:

Step 6: VMI Instance Launch on T-CH.

- TTP:
  – Computes RL (Ready to Launch), RL is the encryption of DK_{VM} is $N\|H(VMI)$ by key PK_{BIND};
  – Then, forwards RL value to T-CH for mounting the requested VMI instance.
- T-CH :
  – Decrypts RL by its own PK_{BIND}, it gets DK_{VM} and $N\|H(VMI)$;
  – Locates the VMIF package by the instance table when

it was created and associated with the properties of the requested VMI;
  - Decrypts the VMIF package by DK_{VM} ;
  - Injects $N\|H(VMI)$ value in VMIF before mounting the VMI instance
- The user now passes to **Verif_4()** the value $N\|H(VMI)$ sent by T-CH with the value $N\|H(VMI)$ generated during the confidence phase.
  - IF FALSE, the CH communicating with the user is not the selected T-CH. This requires another communication attempt between the selected T-CH and TTP. If the 3rd attempt is false, the process is regenerated in **Step 1**. (i.e Session expired);
  - IF TRUE, the user claims the launch of its VMI.
- T-CH launches VMI instance in the trustworthy T-CH.

IV. Performance Evaluation & Security Analysis

As aforementioned, the proposed VMITLP ensures three fundamental security properties throughout the launch instance process namely two-factor authentication which has been ensured for both user instance through the CAS and cloud host platform through the TTP. In addition, for trust settlement TPM stateless keys has been deployed in the proposed scheme to protect the platform configuration and measurement data and provide attestation with regard to the state of the platform configuration.

This section describes thoroughly the performance evaluation of the proposed VMITLP regarding recommended security properties such as confidentiality, integrity, trust and authentication, also the prevention of known attacks which could compromise the effectiveness of the proposed protocol. Thus, the security analysis of proposed protocol regarding recent related works on the basis of performance evaluation, security and severe attacks.

A. Performance Evaluation

In this section, VMITLP has been evaluated with regard to recent alternatives concerned in the IaaS model security. First and foremost, it consists in determining the security vulnerabilities and threats that impede the assurance of the most relevant security properties in form of confidentiality, integrity, availability, trust and authentication as described in Table IV. Moreover, in order to prove essential properties for the effectiveness of the launch VMI instance process, VMITLP is evaluated based on analyzing the security in the cloud which is a shared responsibility between user and CSP by providing a trustworthy cloud platform.

1) Confidentiality: Preserving confidentiality consists in making the information or data unitelligible to only appropriate tenants. Highlighting confidentiality in launch VMI instance process has been analyzed regarding several related alternatives. In [20], proposed EVDIC does not satisfy an environment of trust between the user and the CSP since the latter has full control over the encryption methods of the storage repository and the use of weak generation key to decrypt the requested instance based on PBKDF-2 and user ID. In [21], to ensure that the requested VMI instance is launched on a host with attested integrity, the tenant encrypts the image of the virtual machine (with all data injected) with a symmetric key sealed to a particular configuration of the virtual machine host reflected in the TPM platform configuration register (PCR) values. The proposed solution is suitable for scenarios for launching trusted virtual machines for enterprise customers. In return, confidentiality in VMITLP is established over the entire launch process mainly in two main sites. i) In phase 1 by establishing asymmetric cryptographic keys for identification which cannot expose this phase to any replay attacks and keeps exchanged messages more reliable and effective. In addition, CAS interacts with TTP to mutually identify and authenticate only permissed computer hosts. ii) During pre-launch phase precisely by generating disposable key DK_{VM} to decrypt beforehand stored VMI package into cloud physical disk. This disposable key is generated by TTP based on user credentials and authenticity and can be useful only once for decrypting the requested VMIF. So that, T-CH cannot either be unable to decrypt the mentionned VMIF in an off-session. In meanwhile, isolation between VMs is ensured which every VM contains a unique encryption thread based on unicity of user credentials and shared encryption keys in running session.

2) Integrity: Integrity is considered a primordial security property to determine if the exchanged data has not been corrupted during the communication. In [21], the integrity of the VMI is not applied to maintain trust and corporate responsibility between the user and the CSP. However, integrity in VMITLP is implemented throughout the process by the generation of a dynamic nonce N per session concatenated with $H(VMI)$ by the user challenging the T-CH. In other words, integrity remains a typical mutual responsibility between user and CSP towards a secure launch of the user's VMI. For this purpose, the main aim is to ensure that the user's request has not only been modified by an internal threat, but also that it passes through a prior trust relationship between TTP and the candidate CH . In addition, the integrity of the cloud platform is an important security requirement for a cloud user. It is obtained by using the TCG remote commit mechanism specifically AIK_Certification to identify the CH by TTP considered as a CA, thus making it possible to deal with insider attacks on a awful configured cloud platform [17]. In addition, the communication between the cloud entities is established by IPSec as well as the external communication between the TTP and the CH is established by SSH.

3) Availability: The purpose of availability is to maintain and guarantee access to services. In doing so, VMITLP deploys a recovery phase during each verification. The purpose of deploying checkpoints in the event of a deadline is to maintain the stability and effectiveness of the process. This recovery method ensures the efficiency of each phase and its usefulness, although the recovery time becomes minimal and the session can only be exhausted when an unapproved error and method is detected. In addition, the scheduler deployment which is the responsible entity to select a candidate host based on user requirements. Our paper covers only launch process neither migration nor VMI cloning.

4) Trust: Trust is associated to privacy and integrity techniques which have been deployed by combining the various modules of TPM. Our approach has been compared only with alternatives broaching Remote Attestation (RA) based on TC. Thus, this alternatives have been divided into binary-based RA and property-based RA. Distributed Trusted

TABLE IV
LAUNCH INSTANCE PROCESS ON VMITLP DESCRIPTIVE

	Confidentiality	Integrity	Availability	Trust	Authentication
EVDIC [20]	✓	✗	-	-	-
E2E [15]	-	-	-	✓	-
D-TCCP [18]	✗	-	-	✓	-
TEE [16]	-	✗	-	✓	-
RAA-TCCP [26]	✗	✗	✓	✓	-
TAP [27]	-	✗	-	✓	✓
VMITLP	✓	✓	✓	✓	✓

✓: *Discussed* ✗: *Mentioned* - : *Not mentioned*

Cloud Computing Protocol (D-TCCP) which is an extension to TCCP [18] claims that Trusted Cordinator (TCrd) in TCCP is a bottleneck, as it manages all CNs (Cloud Nodes). They used the CH registration protocol of TCCP, but in VM launch and migration protocols, they just used HVM instead of TCrd. Moreover, [26] proposed a Remote Anonymous Attestation of TCCP (RAA-TCCP) protocol, which realizes the identity and integrity of CN. RAA-TCCP neither uses attribute certifiate nor AIK certifiate, which simplifies certificates management. This is done by using offline TTP and Property-Based Ring Signature (PBRS). RAA-TCCP cannot fulfill a trustworthy environment since VMI is depending on CH.

In the other hand, the settlement of Trust has been deployed in the last phase of VMITLP to determine a T-CH able to securely mount the requested VMI. After identifying the candidate CH, TTP is considered as the user's interim within the cloud entities to obtain and validate the chosen trusted platform on which the user instance will be launched. Since then, in the same way as TCCP and vTPM, validation of the reliability of the candidate cloud platform later called "Trusted-Computer Host" is achieved by collectively encrypting the calculated IML hash values with the PK_{BIND} which is validated by TTP based on the data structure $TPM\_CERTIFY\_INFO$. Therefore, the reliability of the cloud platform is a difficult task for which any failure or unapproved verification detected by the TTP simultaneously neglects the host state of the candidate computer and requests the CAS to select another computer host, as described in section III-A.

5) Authentication: Authentication of the chosen CH nearby CAS and TTP is satisfactorily performed. Mutual authentication can locate and identify the suspicious that CH represents to be a malicious external threat and otherwise an internal threat as discussed in TAP [27]. As a result, another aspect of authentication has been provided by a well-defined key exchange between the entities involved for the current session. In other words, VMITLP resists to the "man-in-the-middle" threat by carefully identifying the cloud platform at each step and regenerating the process in a secure recovery point to maintain the scalability of the user's request.

B. Comparative Performance

TCC has been deployed which CSP provides a sealed box execution environment to cloud user for proving VMI instance integrity and trust. As aforementioned, TC combined RA provides a trust link between CSP cloud platform and user running VMI instance based on AIK Certificate attestation quote and attestation provider. In this section, VMITLP is evaluated with regard to approaches based on binary-based RA such as D-TCCP [18] and property-based RA such as RAA-TCCP and TAP.

This evaluation is based on security ensurance mainly on preserving the confidentiality, trust and authentication as described in 4. Thus, a comparative study regarding performance evaluation for timestamp of VMI instance process and the severe attacks which can tamper the effectiveness of the main process.

Nevertheless, as shown in 4, alternatives based on binary-based RA such as D-TCCP [18] suffers from certain limitations like the need to change the reference hash values even for insignificant changes in the system. Additionally, hash values are measured at boot time, not at demand time, while current systems are constantly updated and can have a very long timestamp. Additionally, binary-based RA reveals platform configuration, which can lead to security threats. In addition, software can be considered reliable, but not could be untrustworthy.

In other hand, VMITLP ensures an encouraging results compared to RAA-TCCP [26] and TAP "Trusted Access Plarform" [27]. RAA-TCCP neither uses attribute certificate nor AIK certificate but deploys offline TTP and Property-Based Ring Signature (PBRS). RAA-TCCP has two phases: i) proof preparation, where CH ensures it has EK_{Cert}, PCRs, and ring signature key; and ii) proof implementation, where ring signature proves CH security properties. However, the proposed scheme lacks a prior phase of CH identification before forming group ring which could lead to integrating a malicious insider attacks such as replay and Man-in-the-Middle attacks. Otherwise, TAP [27] proposed a trusted access protocol for CC based on trusted access authentication framework using RA. The proposed framework consists on two main pahses: i) Registration, where CSP registers at CH. ii) Login and Authentication, where user logins on CH after two factor authentication of identity and platform between user and CH. However, the user running VMI instance cannot surely been mounted on trustworthy cloud platform, since the AIK certification is not efficient to prove the trustworthiness of the CH.

C. Security Analysis

This section addresses the security analysis of proposed method regardless recent related works and several security threats. Moreover, the authentication and trust proof using BAN Logic has been deployed for more comprehensive evaluation.

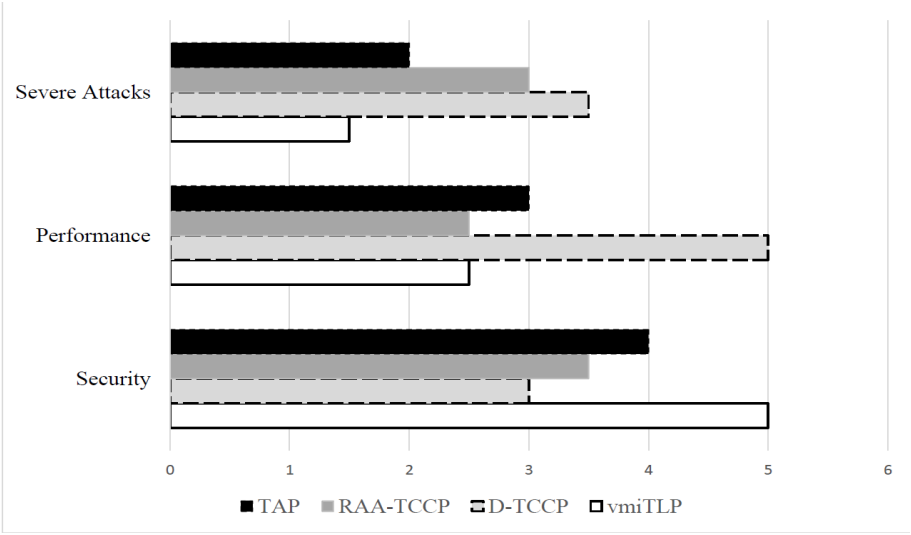

Fig. 4. Evaluation of VMITLP regarding security performance

1) Authentication and Trust Proof using BAN Logic: Most revelant related researches have been addressing the VMI launch process through ensuring trust based on TPM stateless key. In our proposed protocol, the VMI launch process consists not only about preserving trust but also by ensuring the authentication of the user instance within the cloud host and otherwise the identification of the cloud host nearby the TTP (i.e Certification Authority). In order to give a comprehensive evaluation for the authentication phase in VMITLP, BAN logic has been deployed to demonstrate and highlight the effectiveness of the authentication phase in VMITLP. The effectiveness of the authentication phase has been proven by determining the trustworthiness of exchanged information and its privacy against eavesdroping.

BAN logic [28] was proposed in 1989 as a formal method for analyzing authentication protocols. Using the BAN logic, we show that a user and a cloud host CH are mutually authenticated through the TTP and CAS. the launch instance request has been proven regarding the two factor authentication to ensure its trustworthiness and the origin message freshness.

Notations of BAN logic:

$P \mid\equiv X$	:	P believes in X.
$P \lhd X$	:	P sees X.
$P \mid\sim X$	:	P once said X.
$\#(X)$	:	The formula X is fresh.
$P \overset{K}{\longleftrightarrow} Q$	:	P and Q share the secret key K.
$\overset{K}{\longrightarrow} P$	:	P has a public key K.
X_K	:	X is encrypted by the key K.
$X_{K^{-1}}$	:	X is encrypted by the public key K.
$< X >_Y$	:	X combined with Y, $X \parallel Y$.

Rule Definitions:

- **Rule (1). Message meaning rule (MMR):** If P believes P and Q share a secret K and if a message X encrypted with K is seen by P, P believes that X was once said by Q.

$$\frac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X} \text{ or}$$

$$\frac{P \mid\equiv P \overset{Y}{\longleftrightarrow} Q, P \lhd \{X\}_Y}{P \mid\equiv Q \mid\sim X}$$

- **Rule (2). Nonce verification rule (NVR):** If P believes that X is fresh and that X was once said by Q, P believes Q believes X.

$$\frac{P \mid\equiv \#\{X\}\, P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$$

- **Rule (3). Jurisdiction rule (JR):** If Q has jurisdiction over X is believed by P and if Q believes X is also believed by P, P believes X.

$$\frac{P \mid\equiv Q, P \mid\equiv X, P \mid\equiv Q \Longrightarrow X}{P \mid\equiv X}$$

- **Rule (4). Freshness-concatenation rule (FR):** If it is believed that a part of a formula is fresh, it is believed that the entire formula is fresh.

$$\frac{P \mid\equiv \#\{X\}}{P \mid\equiv \#\{X, Y\}}$$

Our main goal is to prove the sharing of the keys K_{Ss} and $K_{Ss'}$ and the freshness of generated nonce per session between TTP (i.e. user interim) and CH in order to prove the two factor authentication in the proposed scheme. The main goals of BAN logic are presented as follow :

1) ***Goal G1:*** CAS $\mid\equiv$ CH $\overset{K_{Ss}, K_{Ss'}}{\longleftrightarrow}$ CH.
2) ***Goal G2 :*** TTP $\mid\equiv$ TTP $\overset{K_{Ss}, K_{Ss'}}{\longleftrightarrow}$ CH.

The messages exchanged during authentication phase of CH through the TTP can be expressed in a generic form and we divide the two-factor authentication into two main steps. i) Step 3: Identification of CH nearby CAS. ii) Step 4: Identification of CH nearby TTP.

- **Message 1.** User $\rightarrow$ CAS : (U_{idf}). (Note that U_{idf} is generated in StepX $U_{idf} = E_{K_{Ss}}(auth - token)$).
- **Message 2.** CH $\rightarrow$ CAS : (U_{idf}). (Generated U_{idf} in launch session).

- **Message 3.** CAS $\rightarrow$ CH : $Val' = E_{K_{Ss'}}(auth{-}token)$.
- **Message 4 .** CH $\rightarrow$ TTP : $Val' = E_{K_{Ss'}}(auth{-}token)$.

Using the assumptions mentioned as follow :

- **A1.** User $|\equiv$ (User $\overset{U_{idf}}{\Longleftrightarrow}$ CAS).
- **A2.** CH $\lhd \{U_{idf}\}$.
- **A3.** CAS $|\equiv \#\{auth-token\}$.
- **A4.** TTP $|\equiv$ (TTP $\overset{K_{Ss'}}{\Longleftrightarrow}$ CAS).

The two factor authentication between candidate CH and TTP has been proven as follow :

- S1. From Message 1 and MMR. we get :
$$\text{CAS } |\equiv (\text{User } |\sim U_{idf})$$
- S2. Using A1 and Message 2, we obtain :
$$\text{CAS } |\equiv (\text{CH } |\sim U_{idf})$$
- S3. Based on S1, S2 and NVR, we conclude that: CAS $|\equiv$ User $|\equiv \#\{K_{Ss}\}$ (CH $\lhd \{auth-token\}_{K_{Ss}}$) **Satisfied Goal 1.**
- S4. From Message 3, Message 4, A4 and JR we conclude that:
TTP $|\equiv$ CH $|\sim \#\{K_{Ss'}\}$(CH $\lhd \{auth-token\}_{K_{Ss'}}$)
Satisfied Goal 2.

2) Formal Security Analysis: In this section, we will evaluate the essentiel security properties of proposed VMITLP by analyzing the effectiveness of attack prevention strategies such as Man-in-the-Middle (MitM), replace attacks.

Theorem 1 (Immunity from MitM): *It is impossible for an attacker (malicious cloud outsider or insider) to intercept a network connection or take advantage of "session hijacking" that compromises the web session by stealing the session auth-token.*

Proof. MitM attack game is defined as follow: a malicious attacker inside or outside the cloud can sniff the network or intercept generated auth-token by CAS to be identified nearby TTP in the phase 1 (Authentication of CH by TTP).

After the generation of user request to launch its VMI instance, the scheduler selects an adversary CH which will be informed or intercept scheduler request T, U_{idf}, the CH then can surpass the authentication nearby the CAS and gets back $Val' = E_{pk_{TTP}}(auth{-}token)$. Thus, the adversary CH sends a request for identification in step 5. The identification request holds Val' which will be verified by TTP. In this step, TTP first decrypts Val' by the exchanged secret key $K_{S_{s'}}$ with CAS. Then, TTP compares the decrypted Val' value to get auth-token and compares it with the auth-token already exchanged with CAS in stepX. For this purpose, if the verification function **Verif_2()** returns false, then TTP rejects the adversary CH request for authentication and enquires back the CAS to select another CH. Moreover, VMITLP requires multi-factor authentication with per session secret keys, generated auth-token and user generated nonce N per session. After all, any adversary could not surpass the authentication phase nearby TTP. This conducts that adversary CH or MitM could not win the game.

Theorem 2 (Immunity from replace attacks): *It is impossible for the CH to pass the verification within TTP in the phase Trust Settlement between TTP & T-CH using by replacing the sealed stateless generated key per session Pk_{BIND} during boot process.*

Proof. We define the replace-attack game as follows : A malicious candidate CH after being identified by TTP in the previous phase (i.e. in case of winning the previous game) sends back the value $Verif = E_{PK_{TTP}}(Pk_{bind}, IML)$ and TPM_CERTIFY_INFO in order to be attested and trusted nearby TTP. In this step, the candidate CH can replace the information of the TPM_CERTIFY_INFO from the generated IML to surpass and grant the trust settlement. If this proof can still pass the verification performed by the TTP, then the CH wins the game; otherwise, it fails.

According to the properties of TPM stateless keys, each CH equipped with TPM can generate a sealed key PKbind per session from running IML of current instance boot process in order to be trusted nearby TTP which is a CA. For this purpose, TTP after receiving the adversary's request for trust establishment, it extracts first a Pkbind key from the received query then compares the extracted key with generated PCR-INFO based on received IML which could be false only if adversary CH is malicious. For this purpose, the adversary CH could not surpass the **Verif_3()** function. This conducts that adversary CH cannot win the game.

V. Conclusions

The existing vulnerabilities in current researches and implementations of IaaS security policy for running or launching a user VMI instance cause several resources security concerns and hamper user data privacy. Because of the complexity of the IaaS and the dynamic cloud data nature, balancing the adopted security policy with data workflow provision remains a crucial topic of research recently. In this paper, we have proposed a security architecture VMITLP to securely launch a user generic VMI into a trusted IaaS public cloud platform. It applies combined security approaches's effectiveness based on TPM to fulfil data confidentiality, integrity and mutual authentication. Thus, ensuring trust between the cloud user and the selected cloud computer host which mounts the requested instance.

In addition, VMITLP thoroughly covers the full launch instance scenario from its request to the mount on a Trusted cloud computer host. Thus, the launch process is defined by three main phases and six steps, each security failure or uncorrect code in each phase directly regenerates the process into a past recovery point which is improved than resuming the process from scratch. VMITLP also deals with most CSA published and known cloud attacks such as replay attack, DDoS attack and malicious insider / outsider. The use of TPM endorsement and attestation keys have been also used to ensure trust and a well-selective verification of a cloud computer host, thus it is required to generate a disposable key able to decrypt the related stored VMI package inside by the trusted computer host. In the future, the proposed VMITLP is under implementation with Openstack and security services. Thus, it can be extended to use PdP or PoR schemes for a large scale of data file integrity. Moreover, VMITLP will be extended to minimize or to control the data workflow and the consumption of Input/Output performance.

References

[1] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.

[2] C. El balmany, A. Asimi, and Z. Tbatou, "Iaas cloud model security issues on behalf cloud provider and user security behaviors," *Procedia computer science*, vol. 134, pp. 328–333, 2018.

[3] T. I. Group, "Trusted Platform Module," \OT1\textquotedblrighthttps://en.wikipedia.org/wiki/Trusted_Platform_Module., 2006, [Online; accessed 2006].

[4] L. Liu, M. Zhang, Y. Lin, and L. Qin, "A survey on workflow management and scheduling in cloud computing," in *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE, 2014, pp. 837–846.

[5] J. Anupa and K. C. Sekaran, "Cloud workflow and security: A survey," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2014, pp. 1598–1607.

[6] S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly *et al.*, "Resource scheduling for infrastructure as a service (iaas) in cloud computing: Challenges and opportunities," *Journal of Network and Computer Applications*, vol. 68, pp. 173–200, 2016.

[7] L. M. Vaquero, L. Rodero-Merino, and D. Morán, "Locking the sky: a survey on iaas cloud security," *Computing*, vol. 91, no. 1, pp. 93–118, 2011.

[8] T. T. W. Group *et al.*, "The treacherous 12: cloud computing top threats in 2016," *Cloud Security Alliance*, 2016.

[9] R. Perez, R. Sailer, L. van Doorn *et al.*, "vtpm: virtualizing the trusted platform module," in *Proc. 15th Conf. on USENIX Security Symposium*, 2006, pp. 305–320.

[10] H. Rongyu, W. Shaojie, and I. Lu, "A user-specific trusted virtual environment for cloud computing," *Information Technology Journal*, vol. 12, no. 10, pp. 1905–1913, 2013.

[11] S. Hosseinzadeh, S. Laurén, and V. Leppänen, "Security in container-based virtualization through vtpm," in *Proceedings of the 9th International Conference on Utility and Cloud Computing*, 2016, pp. 214–219.

[12] Y. Shi, B. Zhao, Z. Yu, and H. Zhang, "A security-improved scheme for virtual tpm based on kvm," *Wuhan University Journal of Natural Sciences*, vol. 20, no. 6, pp. 505–511, 2015.

[13] Y. Cheng, X.-Y. Li, and M.-Q. Ling, "A trusted cloud service platform architecture," in *2012 International Conference on Information Science and Applications*. IEEE, 2012, pp. 1–6.

[14] Z. Balogh, E. Gatial, L. Hluchỳ, R. Toegl, M. Pirker, and D. Hein, "Agent-based cloud resource management for secure cloud infrastructures," *Computing and informatics*, vol. 33, no. 6, pp. 1333–1355, 2015.

[15] J. Wang, B. Zhao, H. Zhang, F. Yan, F. Yu, L. Zhang, and H. Hu, "Poster: an e2e trusted cloud infrastructure," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 1517–1519.

[16] W. Dai, H. Jin, D. Zou, S. Xu, W. Zheng, L. Shi, and L. T. Yang, "Tee: A virtual drtm based execution environment for secure cloud-end computing," *Future Generation Computer Systems*, vol. 49, pp. 47–57, 2015.

[17] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh, "Terra: A virtual machine-based platform for trusted computing," in *Proceedings of the nineteenth ACM symposium on Operating systems principles*, 2003, pp. 193–206.

[18] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing." *HotCloud*, vol. 9, no. 9, p. 3, 2009.

[19] N. Santos, R. Rodrigues, K. P. Gummadi, and S. Saroiu, "Policy-sealed data: A new abstraction for building trusted cloud services," in *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*, 2012, pp. 175–188.

[20] M. Kazim, R. Masood, and M. A. Shibli, "Securing the virtual machine images in cloud computing," in *Proceedings of the 6th International Conference on Security of Information and Networks*, 2013, pp. 425–428.

[21] M. Aslam, C. Gehrmann, L. Rasmusson, and M. Björkman, "Securely launching virtual machines on trustworthy platforms in a public cloud-an enterprise's perspective." in *CLOSER*, 2012, pp. 511–521.

[22] R. Patil, H. Dudeja, and C. Modi, "Designing in-vm-assisted lightweight agent-based malware detection framework for securing virtual machines in cloud computing," *International Journal of Information Security*, pp. 1–16, 2019.

[23] N. Paladi, C. Gehrmann, M. Aslam, and F. Morenius, "Trusted launch of virtual machine instances in public iaas environments," in *International Conference on Information Security and Cryptology*. Springer, 2012, pp. 309–323.

[24] H. Liu and S. Wang, "The analysis and design of trusted computing applied into cloud," in *2012 IEEE Control and System Graduate Research Colloquium*. IEEE, 2012, pp. 5–9.

[25] J. Schiffman, T. Moyer, H. Vijayakumar, T. Jaeger, and P. McDaniel, "Seeding clouds with trust anchors," in *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, 2010, pp. 43–46.

[26] L. Yan and X. Bin, "Design and implementation of remote anonymous attestation protocol based on trusted cloud computing platform," *The Open Cybernetics & Systemics Journal*, vol. 9, no. 1, 2015.

[27] Y. Chang, Z. Zhang, and J. Wang, "A security protocol for trusted access to cloud environment," *Advances in Electrical and Electronic Engineering*, vol. 8, pp. 135–144, 2015.

[28] A. Bleeker and L. Meertens, "A semantics for ban logic," in *Proceedings of the DIMACS Workshop on Design and Formal Verification of Security Protocols*, 1997.