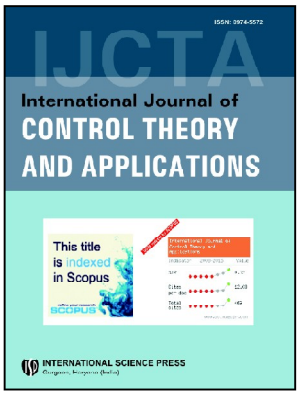

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 28 • 2017

Framework for Prevention of Insider attacks in Cloud Infrastructure through Hardware Security

T. Gunasekhar and K. Thirupathi Rao

Abstract: In this paper, we proposed a framework that effectively prevents and detects the abnormal activities in Cloud Infrastructure. The cloud user requests the Cloud admin for VM, through VM request through TPM module. The TPM module provides a hardware security for the information stored in physical system through sealed storage and remote attestation. The Sealing of memory performed using Storage Root Key(SRK) and Remote attestation via the trusted third party. The sealing, remote attestation and late launch techniques are used on top of the TrustVisor. The TrustVisor provides an information integrity and isolation using Simple Secure Code Block(SSCM), those are formally called as Piece of Application Logic(PAL). The PAL provides a security by isolating the execution environment and performs better when compared to Flicker [2], vTPM [4]. The experiments are conducted on eucalyptus cloud software for private cloud and performance analysis shows us that proposed framework works well and efficient.

Keywords: TPM, Flicker, TrustVisor, SSCM, PAL TPM, Flicker, TrustVisor, SSCM, PAL

1. INTRODUCTION

Cloud computing providing various service models through Internet with various platforms to deliver their resources via minimal overheads in provision process. The Cloud Service Provider (CSP) manages provisioning process with automatic routines on their storage. In this paper we concentrating on IaaS service model and we demonstrated attack patterns that are initiated by CSP or cloud user. The virtualization is a semantic and systematic process of executing one or more guest Operating System (OS) in a logical environment of host OS. The guest OS is termed as Virtual Machine (VM) and virtualization management console in cloud allows CSP to perform following operations: Create, Copy, Save, Read, Write, Live migration, share and restore(rollback) to bring previous state of VM's. These processes remove the management overheads and immensely pose great threat to sensitive data [1]. Let's assume that CSP is compromised, all VM's are under the vulnerable environment that is VM image files can be modified and altered. The insider attack was taken from [2] and to perform these attacks, CSP needs to obtain credentials from the client resources. In [2], the CSP used *string* command to obtain the information from kernel image of VM that is hosted in cloud. The *string* command thoroughly search kernel image file to match the password strings, those are already stored by cloud clients or owner of VM. Once obtained the password, the attacker or compromised CSP can steal or copy information from the cloud resources those are allocated to the cloud client while provisioning. To demonstrate the attack pattern we used the xen hypervisor on Linux environment. The following section describes about Xen hypervisor architecture and security

considerations for proposed protocol and Trusted client with untrusted CSP. Framework for Prevention of Insider attacks in Cloud Infrastructure through Hardware Security.

2. XEN HYPERVISOR AND DOMAIN

Xen hypervisor is a Virtual Machine Monitor (VMM), it lies between host OS and physical hardware and contains direct access to the hardware resources. The Xen hypervisor is designed to provide virtual environment through guest OS by allowing direct access of hardware resources [3]. Generally, xen supports two architectures: those are type 1 and type 2. Type 1 architecture follows full virtualization; it provides virtualization environment to run multiple VMs as logical entities in physical system or host OS.

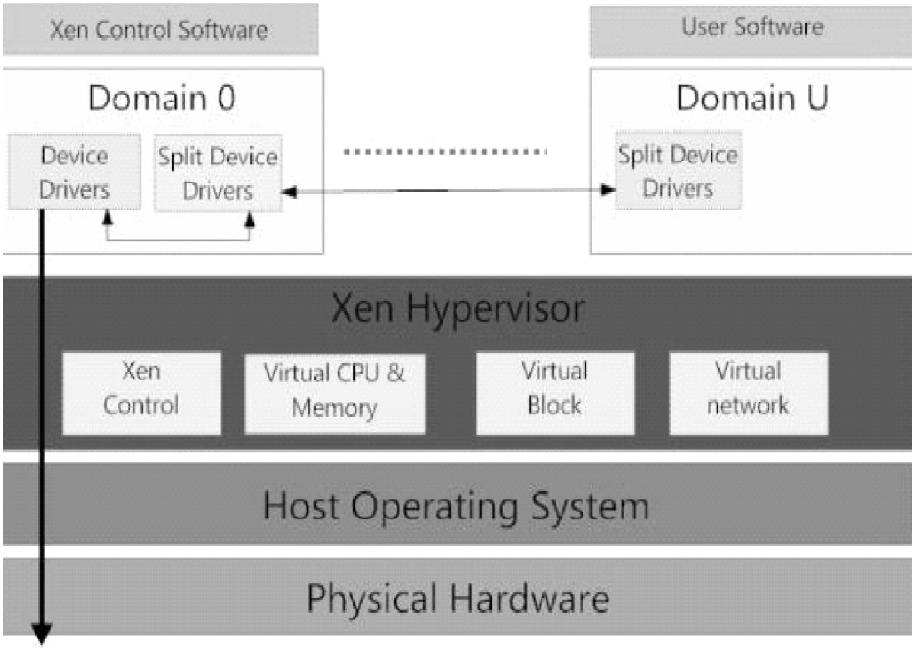

Figure 1: Xen Hypervisor

In Type 1, Virtual machines can access hardware without any interaction with host OS and guest OS can execute the instruction as they required. When illegal instructions or critical instructions can be executed without any interruption of service by using binary translator. The binary translators [4] are largely used in implementation of VMM and for execution in host OS kernel. These binary translators generate huge overheads while executing the critical instruction on host OS kernels. Example of full virtualization applications are Virtual Box and VMware. Type 2 architecture is para virtualization; it provides the virtual environment that doesn't allows virtual machines to access physical resources of host OS. So that, the critical instructions are not allowed to execute directly by guest OS but instead of direct access, guest OS communicates VMM and execute an instruction. As a phenomenon, guest OS need to be modified slightly to execute in Para virtualization environment [3]. An example for this virtual environment is Denali and Xen. Hence, xen is type 2 virtualization architecture. The Xen hypervisor runs on host OS that holds root user as a Domain 0 and other VM's as Domain U0 …. Domain Un.

2.1. Security considerations on proposed protocol

In general, an attacker aims following components in cloud paradigm: Cloud client, Hypervisor or VMM, infrastructure that was used to implement a cloud, guest OS, and applications [6]. In malicious insider environment following security aspects are needed to implement the proposed protocol.

- **Untrusted cloud client**: As per the CERT definition [5] insider itself is an attacker to obtain cloud user sensitive information by using their ring 0 privileges and CSP might turnoff security protection to reduce load on their servers [6]. Major victims of these attacks are private and public clouds such as nebula clouds, volunteer clouds, social clouds etc.
- **Malicious hypervisor**: A compromised hypervisor can read or write the virtual disks, those are assigned to cloud clients. A malicious hypervisor can undo mapping process, it leads that the modifications those actually made by client can't be updated and hypervisor can map to other client virtual disks that leads to data corruption and data misleading [7].
- **Trusted guest OS**: To implement the proposed protocol, we greatly assuming that guest OS is trustworthy and it can launch any sort of attacks on other VM's.
- **Infrastructure:** An infrastructure that used to implement proposed protocol is assumed as secure and scalable to provide service to as many as numbers in the cloud environment.

2.2. Trusted client with untrusted CSP

Suppose a client is running a VM in remote virtualization environment to perform computational tasks by using cloud computing assets. This process involves high security issues at running environment of VM and that pose an impact on integrity and confidentiality of individual or enterprise data. In Xen based environment, Domain 0 have root level access privileges on all other VMs in cloud virtual environment. An untrusted CSP can launch various attacks to steal integrity, confidentiality, availability of cloud user sensitive data.

- **Integrity**: In Xen based architecture, Domain 0 contains full access to all other VM image files. So Domain 0 can perform malicious data actions those leads to loss of data integrity.
- **Confidentiality**: Domain 0 can access any VM pages and can perform required operations to mask their malicious actions. As shown in fig.1, Domain 0 contains device drivers and I/O device those provide direct access to transmission channel. The Domain 0 can view data which is being transmitted over network and stored on physical hard disks.
- **Availability**: As we know Domain 0 has ability to shut down or reboot client VM without permissions/ notices to cloud user and this leads to unavailability of sensitive data for the period of time.

3. METHODOLOGY

This section describes about threat model, security requirements and proposed architecture and then it significance in cloud computing.

3.1. Threat model

Domain 0 contains ring 0 privileges to access any content of cloud vendors and physical resources hosted at cloud data center. To launch insider attack on resources, Dom0 obtains dump of a memory dump of target or specified VM. Initially malicious insider has no idea about password stored in dump of VM snapshot. To obtain a password from dump, an attacker simply devises a method on obtained snapshot of VM. The dump will be filtered using *strings* command, it thoroughly checks dump and returns available strings with name of password. Once Domain 0 obtains credentials from dump of guest OS, the following are expected issues:

- A CSP can access guest OS contents by using their level privileges. With effect of this Cloud client might lose their data confidentiality and integrity. As said earlier, CSP can save, restore, reboot, and shutdown any guest operating system.

- In [9] demonstrated various attack scenarios and those pose great threats in cloud computing virtual environment. The least privileged (Domain U) allows access for the Domain 0; it contains root level or ring 0 access privileges.
- A malicious insider or CSP can change or breach data upon agreed with competitors of the client company. Attackers (insiders) inside the company have great risk to information resources because they are sophisticated about internal structure.
- Malicious Domain 0 cannot access the hypervisor but it can access secondary storage and network I/O. With this maliciousness user can perform any task without any permissions from owner of Domain or Virtual Machine.

The following sections describe prevention and detection methods.

3.2. Secure Runtime Architecture

This section briefly describes about our proposed model to mitigate insider attacks in cloud virtual environment. In our model, trusted base computing is core for ensuring confidentiality, integrity and data privacy with reference of architecture [10].

- Organization has to collaborate with cloud provider to use the cloud resources and organization administrator authorizes their employees to utilize the cloud assets with access control mechanism. After authorize cloud user, they can access resources with their company access privileges and they can launch any number of VM when they actually required. In our approach, we are concentrating on cloud virtual machines.
- In order to ensure the confidentiality, we using Trusted Cloud Computing Platform (TCCP), it provides isolation execution environment by extended version of trusted computing with IaaS service model.
- TPM based attestation: An execution environment platform details are analyzed and stored in tamper free and inexpensive chip is Trusted Platform Module (TPM). The log details are maintained by *Integrity Measurement Module* (IMM). This attestation allows external parties to validate or authorize the cloud clients to ensure trust over cloud resources.
- After verification through attestation process, TrustVisor [10] is used for isolation environment that ensures confidentiality about client data in cloud storage.
- To ensure integrity on VM image files, the proposed method uses hashing technique and figure 3 demonstrate how it works. This framework greatly works with ubuntu virtual environment and guarantees about integrity on VM.

4. PROPOSED METHOD

4.1. TPM Remote Attestation

A Trusted Computing Group (TCG) suggested a standard design for Trusted Platform Module (TPM). A TPM provides various services in trusted computing environment such as remote attestation, storage sealing, secure VM launch and etc. Platform configuration details are captured and stored in TPM and these are used for attestation process while verifying the VM. An attestation allows external parties to take trusted decision on software configuration platform state. In remote attestation, some form of Integrity measurement is required called Integrity Measurement Architecture (IBM IMA) [10]. Events are represented and reduced as measurement using cryptographic algorithm SHA-1 hash function. Every measurement is extended in to Platform Configuration Register (PCR) by combining earlier hash code with present/new measurement. TPM 1.2 and TPM 2.0 requires minimum of 16 PCR's and 32 PCR's respectively.

A TPM quote operation usually attest the values of PCR registers and TPM quote consists of set of PCRs information along with a nonce value which is signed with Endorsement Key (EK). A confidential private part of EK is used for signing during quote creation and it tells that which is signed by the secured TPM.

4.2. Memory seal storage

TPM module provides storage sealing concept that maintains user data in encrypted format by using Storage Root Key (SRK). Seal command takes SRK and set of indices of PCR to encrypt the data that is stored in cloud environment. A TPM never leave SRK and it will act as a root key for all the encryption process [11]. As a result of encryption, seal generates cipher *C* along with integrity protected indices and list of PCR's. An *Unseal* takes cipher *C* and PCR list that is used while *sealing* processand now it's time to check integrity of PCR values. TPM takes care of checking integrity of PCR current values with list of PCR values (old), if it matches then TPM generates output the resulting data otherwise returns an error [11].

4.3. Proposed architecture with Trust Visor

In this section, we explained proposed architecture proof -of-concept with implementation. As we discussed earlier, remote attestation used for verification of VM by external trusted parties. This approach mainly used Intel based processor called Integrity Measurement Architecture (IMA) [12].

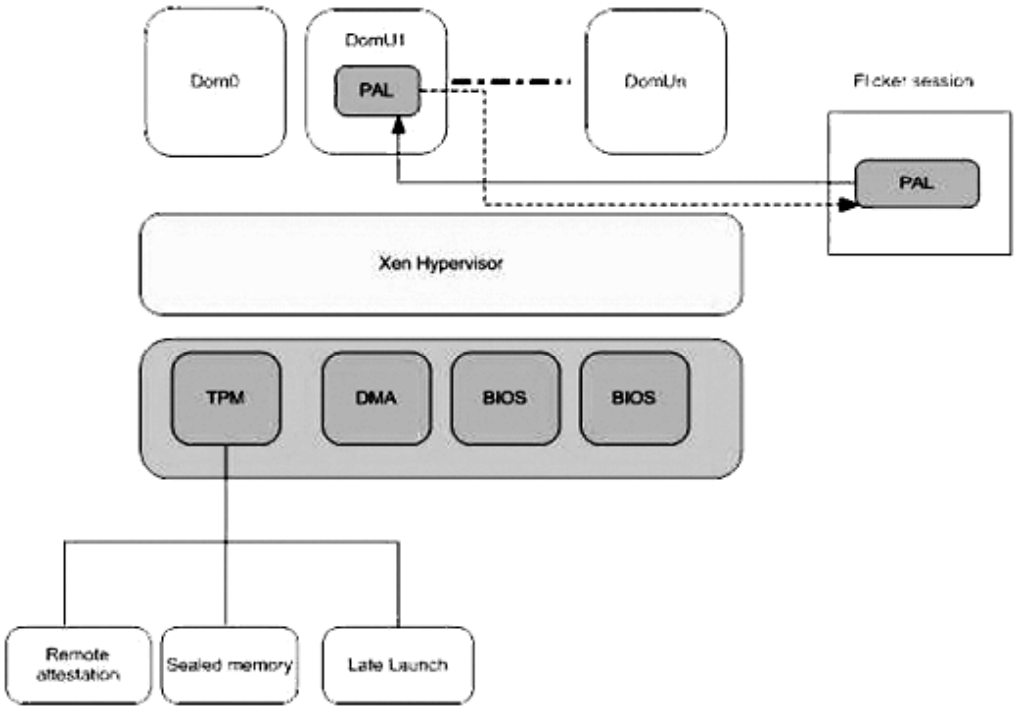

Figure 2: Proposed Architecture

Remote attestation

As described in earlier sections, TPM module stores the measurement in PCR registers and those are utilized for remote verification by extending PCR registers.An actual process starts by invoking nonce to Node controller and the node controller forwards respective PCR value to the third party for verification. The third party matches the PCR hash value with earlier PCR value.

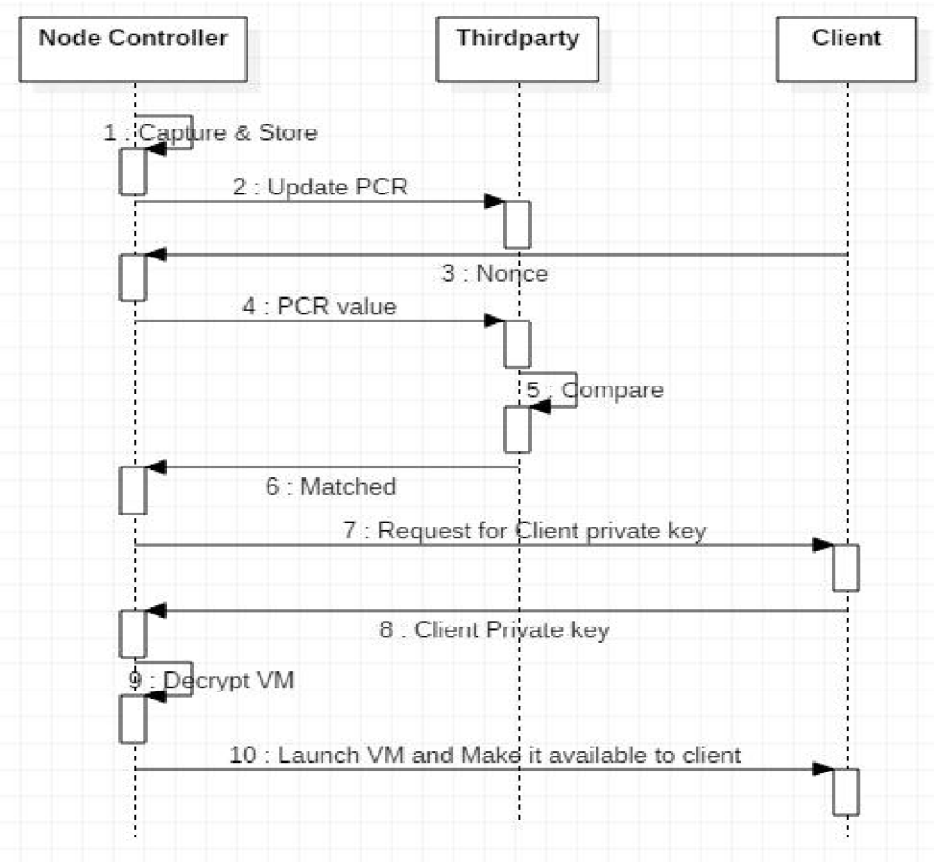

Figure 3: Remote Attestation

Upon matching NC request the client for private key and client send the private key back to node controller for the decryption of VM. After receiving key from client, node controller decrypts the VM kernel and establishes a session in TrustVisor [14] for isolated environment.

Secure VM launch

Our proof of concept proposed algorithm uses TPM secure VM launch protocol for securing the VM in Cloud Storage(CS).The VM kernels are encrypted and stored in cloud storage, so that only trusted or verified NCs can launch the VM. The process of secure VM launch ensures that client private key or decryption key isprovided to NC to decrypt the VM.The proposed protocol proceeds as follows: the client sent VM request to NC and initialize Trustvisor session and extends the PCR register 18 with measurement of Trustvisor.Trustvisorconsists of private and public key for session management and secure VM launch operation.Trust visor sends public key (tpk) to node controller and node controller forward a key along with nonce of client.

Client responds with nonce, client private key and public key of Trust visor to the Node controller and the Node Controller forwards the same copy of keys to the Trust visor. Upon receiving, the Trust visor decrypt the VM kernel image using private key of client and public key of Trust visor. Now, a VM execution started in isolated environment apart from the Node controller. In the next section protocol III described briefly.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed framework implemented on HP elite Notebook 8540 with configuration of Intel i5 processor, 8GB RAM, 500HDD and Ubuntu 14.04 as a host OS. The Eucalyptus cloud software used for implementing the

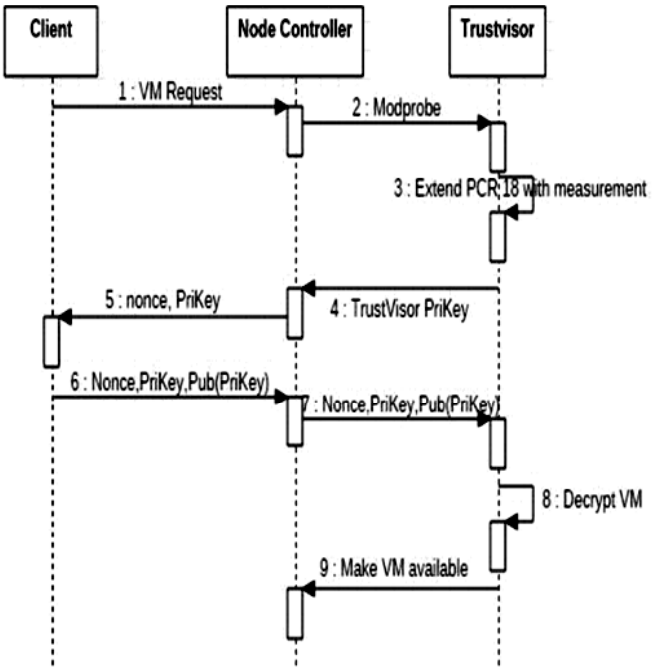

Figure 4: Secure VM launch

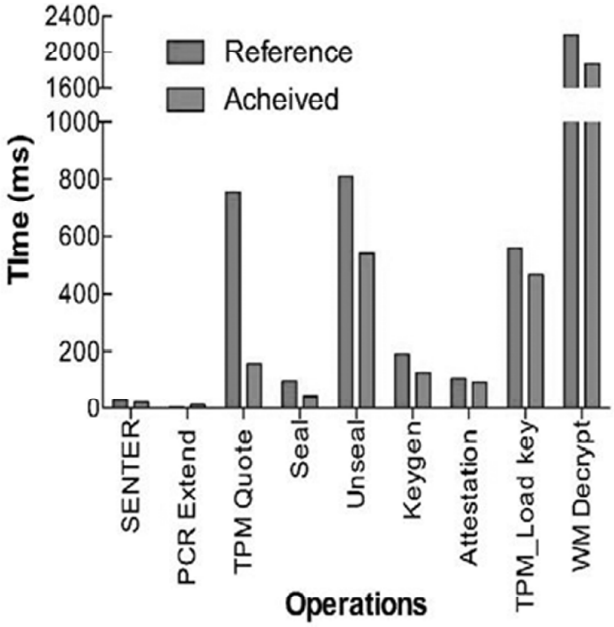

Figure 5: Performance evaluation

private cloud that provides an infrastructure for launching VM's. The experimental results show that proposed framework has greater ability to reduce the TCB minimization and less over heads while communicate with TPM device through the host operating system.

The SENTER instruction takes 20.5ms for the initiation of secure boot along with the TrustVisor hypervisor boot process. The PCR Extend is used to quote particular PCR value and it took 10.68ms. The TPM quote for measuring the PCR values with hash values are calculated and replaced with new hash digest and this operation took 357.68ms. Thus it shows us that Flicker based environment takes long time to respond for the TPM quote. The seal and unseal operation takes 45.29ms and 537.87ms, when compared to other hypervisor performance in both operations TrustVisor has great ability to reduce the overheads in unseal operation. The remote attestation took 100.3ms for trusting the platform using the PCR values with cryptographic techniques those we discussed earlier sections. The results show us that TrustVisor has great ability to reduce the overheads during the TPM operations.

6. CONCLUSION

The information stored in the cloud infrastructure should be secure and reliable in order to provide assurance to cloud user. In this paper, we analyzed and designed a framework that effectively prevents the insider attacks from insider and cloud users. A TPM and TrustVisor are used for implementation of proposed system and taken results. Then the results are compared with the previous systems of insider attacks. Our framework effectively detects and prevents the abnormalities in the cloud infrastructure.

REFERENCES

[1] Li, C., Raghunathan, A. and Jha, N.K., 2012. A trusted virtual machine in an untrusted management environment. *Services Computing, IEEE Transactions on*, *5*(4), pp. 472-483.

[2] Rocha, F. and Correia, M., 2011, June. Lucy in the sky without diamonds: Stealing confidential data in the cloud. In *Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on* (pp. 129-134). IEEE.

[3] Kappes, G., 2012. Installing and Using Xen.

[4] Kedia, Piyus, and Sorav Bansal. "Fast dynamic binary translation for the kernel." In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, pp. 101-115. ACM, 2013.

[5] Kandias, M., Virvilis, N. and Gritzalis, D., 2011. The insider threat in cloud computing. In *Critical Information Infrastructure Security* (pp. 93-103). Springer Berlin Heidelberg.

[6] Xu, L., Lee, J., Kim, S.H., Zheng, Q., Xu, S., Suh, T., Ro, W.W. and Shi, W., Architectural Protection of Application Privacy Against Software and Physical Attacks in Untrusted Cloud Environment.

[7] Jin, Seongwook, *et al.* "H-SVM: Hardware-Assisted Secure Virtual Machines under a Vulnerable Hypervisor." *Computers, IEEE Transactions on* 64.10 (2015): 2833-2846.

[8] Wojtczuk, R., Rutkowska, J. and Tereshkin, A., 2008. Xen 0wning trilogy. Invisible Things Lab.

[9] Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J. and Felten, E.W., 2009. Lest we remember: cold-boot attacks on encryption keys. *Communications of the ACM*, *52*(5), pp.91-98.

[10] McCune, J.M., Parno, B.J., Perrig, A., Reiter, M.K. and Isozaki, H., 2008, April. Flicker: An execution infrastructure for TCB minimization. In *ACM SIGOPS Operating Systems Review* (Vol. 42, No. 4, pp. 315-328). ACM.

[11] Parno B. The trusted platform module (TPM) and sealed storage. TPM Documentation. June 21st. 2007 Jun 21.

[12] Sailer, R., Zhang, X., Jaeger, T. and Van Doorn, L., 2004, August. Design and Implementation of a TCG-based Integrity Measurement Architecture. In USENIX Security Symposium (Vol. 13, pp. 223-238).

[13] I. Khan; Z. Anwar; B. Bordbar; E. Ritter; H. u. Rehman, "A Protocol for Preventing Insider Attacks in Untrusted Infrastructure-as-a-Service Clouds.," in *IEEE Transactions on Cloud Computing*,vol. PP, no.99, pp.1-1, doi: 10.1109/TCC.2016.2560161.

[14] J. M. McCune *et al*., "TrustVisor: Efficient TCB Reduction and Attestation," *2010 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2010, pp. 143-158.doi: 10.1109/SP.2010.17.