Chapter 4

TRUSTWORTHY FRAMEWORK FOR INSIDER ATTACK PREVENTION AND DETECTION

4.1 INTRODUCTION

This chapter introduces a trustworthy cloud computing architecture that uses the security properties offered by a virtual machine monitor that enforces the principle of least privilege. These security properties are a strong building block to provide trustworthy cloud computing services to cloud consumers. In this section briefly explained about proposed system to prevent insider attacks in cloud environment from cloud consumer and cloud service provider perspectives. The proposed framework is initiating how virtual machines are providing most reliable security materials of the cloud computing architecture but these method is applying for least privilege in that case one of the method is discuss in chapter 3. For cloud consumers, the proposed architecture allocating the well-built security materials of the reliable cloud computing services.

For the reference of the cloud consumers about the cloud computing services to viewed as a how much reliable service they are providing in that case its require to publish the integrity measurement. To reliable the services these integrity measurements can utilize to choose easily for the cloud computing to reliable the services. They should be clearly explained about the security property.

Transparency is a requirement that a remote untrusted system presents integrity measurements to consumers by trustworthy mechanisms [84]. The cloud consumer can
easily analyze the system integrity or remote system reliability for these results of the measurement. In this segment, they are defining about the utilization of the reliable virtual machines monitor and computing technology for the cloud computing architecture to build a strong reliability.

4.2 METHODOLOGY

This section describes the detailed information about the cloud infrastructure, which is considered for the proposed system. First, Cloud computing server is establishing the security critical software component. Secondly, in clearly they are describing to build a strong reliability cloud computing ecosystem of the architecture requirements. Finally, we discuss how these components and mechanisms come together to create a trusted virtualization environment and give an example scenario where this trusted environment is used to offer trustworthy cloud management operations.
Tradition approaches is different from the components of the cloud server and distinct security related tasks are overseeing for every component. As early we are describing about the virtual machines monitor different for virtual machines are make sure to execute and share requirements of the same physical server is the virtualization layer. The Figure 4.1 shows the abstract view of the proposed framework. In the proposed architecture, in addition to those tasks, a virtual machine monitor or hypervisor (TrustVisor) is also the policy decision and enforcement point for memory access. In security basis, it is very tough about the management virtual machines in previous the whole memory space was assigned to the consumer’s virtual machine. In some situations, consumer’s virtual machines are directly connected with the hardware requirements it is responsible for the management virtual machines to produce the hardware drivers, virtual storage and network access.

To manage and monitor the virtual machines the present cloud architecture are utilizing the management virtual machines, for the sake of malicious insiders can assess to open in simple way and direct attack vector. In the architecture, we propose in this chapter, however, the privileges of the management virtual machines were reviewed, and the operations that allowed attacks on consumer’s data were moved to an isolated special-purpose virtual machine. This isolation implies that the virtual machines monitor must be compromised for an attacker to obtain access to the whole memory range. For consumer’s data, they are providing unique requirements for their operations of executing the security purposes for example launching and migrating a virtual machine. The cloud service provider should handle these operations very carefully because consumer virtual machines entire memory spaces are engaged to access. These virtual machines can have a reduced
trusted computing base if they use solutions such as unikernels, e.g., Mirage OS and OSv. The execution of a single application to accept the development of the kernel.

To implement the single application for growing the kernels at the peak point of the virtual machine, that why we can easily minimize the computing reliability. It could be zero or more controlled by the more request because the consumer virtual machines is the final component. Dumbing down of the management virtual machines is the main distinction of the earlier approaches to minimize the advantage operations. It is main thing to maintain the best security to choose the suitable requirements to the uniqueness of the virtual machine. These servers are required to support trustworthy computing such as Intel’s Trusted eXecution Technology (TXT)[22].

A sensitive group NC client maintains TPM enabled hardware support for sensitive information in the cloud infrastructure. In this cloud environment, VM can be size up to 2GB and it poses makes computational overheads while performing encryption and decryption. In general, VM consists of three components: kernel image, boot disk image and an initial ram disk images. From the security perspective, the VM kernel image is most fundamental component of VM and it interacts with the user level application through the system calls. Such that, it is enough to encrypt the kernel instead of entire VM image to maintain confidentiality and integrity of user sensitive data and from the performance perspective.

Our framework uses hardware based security called TPM (Trusted Platform Module), which is proposed by consortium group called Trusted Computing Base (TCB). The main purpose of TPM is to protect sensitive data from external stealing as well as from internal parties in mobile device, PC and PDA’s. In the proposed system, various TPM
services are used to secure data of client VM. Our framework works on Infrastructure as a Service (IaaS) model, so we concentrate on VM and its associated assets in cloud environment. As explained earlier, we used eucalyptus cloud software for implementing IaaS based cloud.

Fig 4.2 Conceptual View of proposed system

Figure 4.2 briefly shows how proposed system works in cloud. Eucalyptus is open source cloud software and it is API- compatible with EC2. We used eucalyptus design for implementing cloud infrastructure, which is an answer for commercial Amazon EC2. Node Controller (NC) should install on TrustVisor hypervisor and VM’s are installing on NC (bare metal hypervisor). To implement the proposed framework, we used open source
cloud software called eucalyptus to establish an IaaS private cloud for our testing. Eucalyptus [94] consists well defined components such as NC, CC, walrus, SC and Cluster Controller (CSC), those provides efficient communication among resources using web-service. Eucalyptus is an EC2 API-compatible and which is an answer to commercial Amazon EC2 cloud infrastructure. Eucalyptus supports libvirt hypervisor, it consists most popular hypervisors Xen and KVM hypervisors. The eucalyptus components are well defined with web-service based interfaces and those components are developed using high-level and standard packages such as Axis2, Apache, and Rampart [95]. Our proposed architecture utilizes these components to prove that our concept is secure from insider attacks.

To provide a trustworthy cloud infrastructure, it has some major components and requirements. Here, we provide set of requirements those are considered as paramount for the trustworthy cloud architecture. These requirements considered and addressed from the NIST BIOS integrity measurement [37]. These requirements are key to evaluate the trustworthiness of the cloud infrastructure by the cloud consumer. A trusted third party is responsible for generating and maintaining golden integrity measurements of the cloud platform resources.

- Cloud components are capable to generate and collect integrity measurements.
- Trusted Platform Module provides tamper evident or tamper resident storage for integrity measurement.
- A secure protocol must be established to provide a secure communication between cloud consumer and provider to transfer the integrity measurements of the cloud components.
• A protocol must be developed for trustworthiness of software executing in a remote platform.

To satisfy the above requirements, the details are provided in the following sections. The following section elaborates how cloud provider ensures the platform integrity of the cloud infrastructure to assure the trustworthiness of the cloud resources of the cloud ecosystems.

4.3 INTEGRITY MEASUREMENTS

The basic requirement for golden integrity measurement is the trusted third party responsible for hosting and managing the genuine software components of the cloud ecosystem, and performs the security tests before collecting and generating the integrity measurements of the cloud infrastructure. A golden integrity measurement can be defined as “it is a cryptographic hash code of the cloud resources which is verified by the trusted third party of the cloud ecosystem to ensure the trustworthiness to the consumer”.

The consumer expects this cryptographic hash before using the cloud resources when verifying the trustworthiness of the cloud components such as Virtual Machine Monitor or hypervisor. The proposed framework, golden measurements are intended to offer functionality similar to the one possible using reference integrity metric (RIM) elaborated in the mobile trusted platform specification [78].

The reason to use the cryptographic hash is that it is impractical to generate same for the other components and cryptographic hash use is a logic layout choice because cryptographic hash capabilities are collision-resistant, this means that it is computationally impracticable to find a distinctive enter that gives the equal output/hash. Therefore, if a certain block of code has a hash h1, it's far impossible to use a different block of code, as
input to the cryptographic hash characteristic, and obtain the same hash h1. This belonging is fundamental in assuring that an integrity measurement is explicitly certain to a single block of binary code and/or data. However, a golden integrity measurement does now not have the objective of presenting information on the presence or absence of vulnerable properties, e.g. Buffer overflows, within the measured binary code and/or data.

The proposed framework foresees the generation of golden integrity measurements as a robust addition to the security development lifecycle. The security improvement lifecycle is composed in a hard and fast of traditional software development lifecycle phases with the addition of protection steps that intend to attain a final product as resilient as feasible in opposition to malicious insider attacks or breaches. The levels include necessities, layout, implementation, verification, and launch. The verification segment already calls for an impartial team to perform a final security review of the product [79]. Therefore, it has to be trivial to deliver within the era of valid golden integrity measurements.

Although a secure hash characteristic offers certain security properties, it isn't always sufficient to ensure the authenticity of golden integrity measurements. Let us consider a scenario in which a cloud provider develops his very own infrastructure management software and generates himself the golden integrity measurements. In such cases, if customers agree with the provided golden integrity measurements then they're nonetheless susceptible to attack originating from an insider threat within the cloud provider infrastructure. Hence, we propose using trusted authorities together with the ones within the properly-set up public-key infrastructure (PKI). This technique builds a chain of trust, in which the consumer can believe to verify the authenticity of golden integrity
measurements. In what follows, we explain in detail how such chain of trust is executed. The number one intention of imposing a public-key infrastructure is to permit at convenient, secure, and efficient distribution of public keys. A public-key infrastructure (PKI) is defined as the set of physical or logical entities and techniques required to obtain such a goal. The required procedures address the creation, control, storage, distribution, and revocation of digital certificate primarily based on asymmetric cryptography [80].

The general public-key certificates are the important component in a public-key infrastructure due to the fact it is a scalable and secure way for the distribution of public keys [81]. Earlier algorithms depended on a primary public-key authority which would keep each public key and distribute them to cloud consumer using a direct request method. Therefore, it'd constantly be involved in the transactions turning into a single-point of failure and a bottleneck. The digital certificates are exchanged without interference of the certificate authority. Even though it is secure the trusted party is entitled as certificates authority (CA) is required. A certificates authority is commonly a security company, government, or financial organization trusted within the user network.

The generation of a digital certificate is an essential for the consumer to transmit its public key to the certificate authority in a reliable and secure manner. An unsigned certificate is essentially the cloud consumer public key and its information that uniquely identifies the consumer, and certificate authority identification information. The certificates authority makes use of a secure hash feature to achieve a hash code of all this information and encrypts that hash code using its private key to create a digital signature. This signature is then forward to the certificate. The user can then make the certificates public for anyone to apply or transfer it as a respond to consumer requests.
Figure 4.3 illustrates how the party using the certificate can verify the user’s public key through the digital signature attached to it. This signature is verified using the certificate authority’s public key to decrypt the signed hash code and comparing it to a hash code the verifying party generates from the certificate data. A match between the hash codes guarantees the user’s public key can be trusted. For a more detailed description on the contents of a digital certificate were further readers to the X.509 specification [82].

Let us consider as an example the generation of a golden integrity measurement for a specific hypervisor version. During the verification phase of the security development life-cycle an independent and trusted authority would perform security checks on the software. In case those verifications were successful, it would generate a golden integrity measurement for the software and sign it (i.e., encrypt the hash code of the software) using its private key. The public key of this trusted authority can be obtained through its public-key certificate.

It is important to clarify those using golden integrity measurements to offer trustworthy software does not mean only a software version is valid. A version is considered valid as long as the golden integrity measurement in use as a valid signature attached to it. For a cloud consumer to verify the trustworthiness of the hypervisor it just needs to have in his possession the correspondent golden integrity measurement and the trusted authority’s public-key certificate.

When communications are initiated with the cloud provider’s infrastructure, the consumer receives a fresh signed integrity measurement of the hypervisor in use together with public-key certificate from the cloud provider. For the purpose of this discussion it is enough to know there is a digital certificate linked with the cloud provider, more details
about which certificate and why we can trust the provided integrity measurement are given in [85].

![Diagram of Public Key Signing]

Fig 4.3 Public key signing

The consumer can then check if the received integrity measurement matches the golden integrity measurement. If there is a match, then the consumer can trust the hypervisor version execution in the cloud infrastructure. The public keys of both trusted authority and cloud provider allow the client to verify the authenticity of the measurements used in this verification.

### 4.4 PLATFORM INTEGRITY MEASUREMENTS

This section discusses how our architecture possesses components capable of generating and collecting integrity measurements, and offers tamper resistant or tamper evident storage for integrity measurements. These requirements are satisfied through the support for hardware-based security such as Intel’s Trusted Execution Technology (TXT). Therefore, the servers require the presence of a Trusted Platform Module (TPM).
The TPM-chip provides the foundation for hardware-based security, and contains cryptographic functional units (e.g., random number generator), non-volatile, and volatile storage. It is tamper-proof and the hardware root of trust in a trustworthy environment. A root of trust is a hardware or software mechanism that the user implicitly trusts [26]. The definition of trust considered in this document comes from the Trusted Computing Group (TCG): “An entity can be trusted if it always behaves in the expected manner for the intended purpose”. The TPM generates, collects, and offers the tamper resistant storage for integrity measurements. However, using the TPM directly to manage integrity measurements has proved to be a scalability and performance issue [88].

Our approach to address the performance issues related to intensive use of the TPM is combining TPM functionality with a minimal software implementation of the TPM standard denominated micro-TPM. The concept of a micro-TPM was introduced in TrustVisor as a solution to offer fast integrity measurements of pieces of application logic (PAL) [89]. For the use of a software-based micro-TPM to be successful it must be combined with the specific functionality from the hardware base TPM. In what follows, we discuss in detail how these two requirements allow for trustworthy management of a platform’s integrity measurements.

In this section, the software and components of our proposed system capable of collecting and generating integrity measurement, and offers tamper evident or resistant storage for platform integrity measurements. These properties can be achieved by using AMD’s Secure Virtual Machine (SVM) and Intel’s Trusted eXecution Technology (TXT). Therefore, the servers require the presence of Trusted Platform Module (TPM).

The TPM-chip provides the foundation for trustworthy, contains cryptographic
functional units, volatile and non-volatile storage. The TPM is tamper-proof and provides a root of trust cloud environment. A root of trust is a hardware or software mechanism that the user implicitly trusts the resources, which are hosted in cloud environment [26]. According to the Trusted Computing Group: “It can be anything, which acts or behaves in a intended manner” [90]. The TPM can generate, collect and offers the tamper resistance storage with volatile and non-volatile memory. Though it provides durable security for sensitive information, TPM ensures the performance and scalability issues [88].

Our approach uses the micro-TPM concept to ensure the data integrity and data confidentiality of the consumer virtual machines. The direct impact of the TPM is high and generates less performance while generating and collecting integrity measurements. The micro-TPM was implemented in TrustVisor hypervisor as an alternative to improve performance of computing system while generating integrity measurement of cloud components. The micro-TPM concept generates and collects the integrity measurement of the Piece of Application Logic (PAL) [89]. To make this as a possible to implement this with the software based micro-TPM, it must be collaborative with physical hardware of the computing system or cloud server. We can discuss how the requirement allows trustworthy management of the given platform’s integrity measurements.

One, the physical TPM has direct impact during the generating and collecting the integrity measurement of hosted components. To overcome this issue, we need to implement the micro-TPM concept to incur the high-performance issues as resulted in Flicker [88]. The software based micro-TPM concept was implemented generates the high performance in hosted primary computing components, while implementing restricted set of functionalities by physical TPM. The micro-TPM implementation should offer the
remote attestation, data sealing, randomness and measurements. Every consumer virtual machine will have the micro-TPM to generate and collect the platform’s integrity measurements. The TPM property tamper evident can be achieved by using the concept of micro-TPM or physical TPM storage locations. The second property is to ensure the trustworthy computing environment while executing the virtual machine. The trustworthy computing can be achieved by using Static Root of Trust for Measurement (STRM) and Dynamic Root of Trust for Measurement (DRTM). In the initial stages of the trusted environments, the BIOS were used as STRM. The STRM’s trusted computing base includes Boot loader, OS, BIOS, applications and ROMS [27, 90]. The STRM ensure the secure execution environment by using the secure boot, where every line in the software involved in the boot sequence is trustworthy. By combining all these components, it creates a large trusted computing base. If the size of the trusted computing base increases, then the possibilities of vulnerabilities also increases.

The recent advancements in the technology, the computer vendors developed new technology to create a trustworthy environment, such as Intel’s Trusted eXecution Technology (TXT) and AMD’s Secure Virtual Machine (SVM) to ensure the Dynamic Root of Trust for Measurement. These are used during computation to ensure the trust over the hosted components in cloud environment. The second generation of DRTM provides a concept called late launch, where the applications can be executed at any time by generating and collecting platform’s integrity measurement. The advent technologies from the major vendors provides the dynamic execution environments by using the Intel’s SENTER and AMD’s SKINIT instructions that allows to execute the application in the trustworthy execution environment and these instructions can reset. These secure
initializing instructions can initialize the CPU, load the Secure Loader Block (SLB) code, sends the SLB contents to the physical TPM, and enables the Direct Memory Access for the SLB and control will be transferred to the SLB. Then the SLB can launch the application from trustworthy state and this process reduces the TCB by eliminating the ROMS, BIOS and Boot loader.

A security sensitive code blocks or piece of application logic should be register for node controller to put in place the appropriate security measures for its execution. This piece of code assures the data confidentiality and integrity of the consumer data and applications in the cloud infrastructure. The execution integrity can be ensured that when application A executes with A inputs, it always returns A outputs outputs. The life cycle of PAL includes registration with node controllers, triggering or invocation, termination, and deletion or removal [88]. The Fig 4.4 demonstrates the three execution modes of system and how it provides memory protection in each mode. The highest privilege level is granted in host mode when the virtualization layer is executing and has control of the system.

![Fig 4.4 PAL system execution modes](image-url)

Fig 4.4 PAL system execution modes [88]
The virtualization layer has full control over the node controller memory, which means it can manipulate guest operating system, memory regions of PAL, and the applications on top of the consumer guest operating system. When the system is in guest mode, the virtualization layer must protect its memory regions as well as memory locations of PAL. The secure application mode is most restrictive mode, when PAL is under the execution. The secure application mode virtualization layer isolates the PAL’s memory locations from all the node controller resources.

The registration process of PAL is already secure from the malicious insider attack as illustrated in the proposed framework. However, the security sensitive operations are performed in the special purpose virtual machine and the remaining operations are concern with physical system over which those are executed. The registration can be done by using application level hyper calls from the hosted hypervisor on the node controller of the proposed architecture. This registration process consists of set functional entry points with expected input and output. The virtualization layer is responsible for verifying that the providing entry points to executing application, while ensuring that the node controller or operating system does not perform any malicious tasks to the memory regions of the guest operating system as specified as security sensitive. The security sensitive code components should have the integrity measurement, which allows the cloud consumer to verify the integrity of the cloud resources in cloud infrastructure. The PAL can invoke as if affected application or code is running normally on top of the node controller. However, after registration the memory locations of the PAL are no longer available to the application and hosted operation system.
The virtualization layer is responsible for handling the security sensitive operations of the PAL, when the PAL is invoked within the applications. When the PAL is invoked, virtualization layer hypothesized the control and setup the environment for execution of PAL as follows: (1) locate the registered PAL to which the executing sensitive code belongs to, (2) change from guest mode to secure application mode so the memory access is restricted to the memory pages of the executing PAL, and (3) prepare the execution environment so control can be passed to the executing PAL.

After finishing PAL’s execution, the virtualization system should attain control over the applications hosted in the operating system. When the system is executing in the secure application mode any attempt to execute code those belongs to the PAL memory locations returns control to the virtualization layer.

The virtualization layer process the PAL output results and make them made available to the calling application. The execution mode switched from secure application mode to guest mode, in which PAL memory space is no longer accessible to the guest operating system. An application any can PAL at any time with different input parameters within the guest operating system perimeter. The PAL removal can be originated from the application that requested for the registration.

After removing the PAL, security sensitive list associated memory pages are emptied or zeroed and which is made available to the node controller. This security measures assure data confidentiality and integrity, and code and execution integrity by isolating the execution environment from the host platform. The PAL provides the application isolation and integrity measurements from virtual machines.
4.5 CLOUD INFRASTRUCTURE TRUSTWORTHINESS

In this section, we present how the trustworthiness of a cloud platform can be shared with cloud consumers increasing the level of transparency when compared with current approaches. To protect the confidentiality and integrity of data kept in the cloud, the cloud platform has to prevent certain attacks and give consumers the ability to assess that this protection is in place. The latter requirement may seem excessive, but it arises from the concern we are dealing with: the insider threat. A malicious insider is in a sense part of the cloud, so he or she can provide false information to the consumer.

4.3.1 Trusted Virtualization Environment

Our solution for protecting consumers’ data (and applications) in the cloud assumes that the attacks against the virtual machine come through the infrastructure and not from targeting vulnerabilities in the consumer VMs themselves [103].

Remote attestation

The remote attestation is a process of verifying trusted virtualization layer of consumer hosted environments. In the remote attestation, the set of measurements are collect from the resources, which are located at cloud provider side. These measurements are used to represent present state of the cloud resource which are held by cloud consumer. The Dynamic Root Trust Measurement based remote attestation, the consumer receives the integrity measurements of the cloud assets such as hypervisor, those controls the cloud server. The Figure 4.5 provides an illustration of the remote attestation process, that allows the remote party to verify the platform integrity of the of the cloud resources. The communication must be established the consumer and cloud service provider i.e., cloud
server. Let us consider the scenario where the consumer uses the computer without the scope of the security aspects of the cloud security.

Fig 4.5 Remote attestation

The remote attestation process starts from the consumer request. First, the cloud consumer requests the integrity measurement from the cloud server, here the hypervisor send the platform integrity measurement of the hypervisor. Second, the TPM sends the quote operation result to the software agent by receiving the request. Third, the TPM sends the signed platform configuration register that holds the all the integrity measurements of the components that are hosted in the cloud server side. The cloud consumer generates the nonce, which is included with the signed values of integrity measurements. Four steps, the message forwarded to the respective consumer that holds the signed integrity measurement. Fifth, the cloud consumer verifies the signature and if the measurements are matches, then the hypervisor is concluded as a trusted component. This remote attestation process allows the cloud consumer to verify the hypervisor on cloud provider platform.

The cloud consumer behavior based on the trusted party with the trusted cloud components by using above five step remote attestation process.
• The result of matching the integrity measurements tells the cloud consumer to trust the components or not.

• An activity of requesting an integrity measurement from the software agent which is located at cloud provider side. The cloud consumer has to trust the software components otherwise they have to notify the provider about the malicious activity of the software component.

• If the integrity measurement is matched with old or stored integrity measurement with malicious activity of the software component at the cloud service provider side then the cloud consumer has to notify about the malicious component and terminate the communication.

The above explanation gives brief information about the traditional remote attestation process, where the physical machine’s trusted platform module is used for generating platform integrity measurements. The proposed architecture, the transitive trust must be established among the cloud consumer and cloud components which are hosted at cloud server. Let us assume the scenario, where the cloud consumer agent added the hypervisor to trusted computing base to the cloud environment and micro-TPM can generate and collect integrity measurements for future purpose to verify the components and speedup the verification process. The micro-TPM acts like cache memory during the attestation process of the cloud components. This scenario clearly states that consumer is implicitly trust the cloud components which are hosted at cloud provider side. The remote attestation verification can be performed between the cloud consumer, cloud server and trusted verified. The micro-TPM can collect the integrity measurement of the virtual machine of consumer on hypervisor platform.
4.3.2 Critical Management Operations

This subsection provides information about the virtual machine launch, backup, migration and termination and how these operations are performed to ensure the trustworthy architecture, which is presented [91]. This information doesn’t provide the entire critical operations of the cloud architecture, it simply provides some of the operations with illustration of how proposed architecture provides the more transparency during the protection of the cloud consumer sensitive data. This process and operation allows the cloud consumer to verify the cloud components which are located at cloud server.

The remaining critical operations are protected in the same way. These kinds of operations are also an example of the micro-TPM that measures the software components at cloud provider side. The boot process will load all the components integrity measurements while loading the on top of the hypervisor, these integrity measurements are loaded into the physical TPM of cloud server. These integrity measurements are used to establish the initial trust among the major parties in the cloud such as cloud consumer and cloud provider. In this process, the trusted hypervisor plays a major role while collecting the integrity measurements and comparing with old integrity measurements. Then it established the trust among the parties with the help of the trusted third party. The remotely executing hypervisor can establish a trust using the remote attestation process, which is presented in the section 4.3.1 and Figure 4.5. The remotely executing software integrity measurements are passed to the micro-TPM component for the trust over the hosted software components at cloud server.
In this section, we assumed that software communication agent is located at both ends of the communication parties to make remote attestation process more successful and verification of the computing components. The communicating software party is responsible for making the verification at both ends of the cloud consumer and cloud provider side. This process is divided into sub-operations, which are clearly described in this section numerical in word. These steps make the communication even better than the conventional process of remote attestation process.

**Virtual Machine Launch**

In the proposed architecture, the virtual machine launch operation objective is that the consumer should trust the software components those are hosted at cloud provider side. The virtual machine operations are handled by software agents in cloud environment, where the cloud consumers outsource their sensitive data.

The above Figure 4.6 illustrates the overall process of virtual machine launch. First, the process of secure virtual machine launch protocol starts with cloud consumer software communicates with software agent which is resided at cloud server side, which is responsible to manage the all cloud resources in the cloud environment. Second, this interaction implies to mean that to launch the virtual machine within the perimeters of the cloud infrastructure with privileged access levels and consumer can access virtual machine using the given physical address of the cloud provider.
Fig 4.6 Secure VM Launch

Now it’s time to make communication among the cloud consumer and target cloud server. Four, the consumer has to request the integrity measurements of the cloud platform to the cloud server, where the consumer virtual machine is supposed to launch. Fifth, the target cloud platform generates and signs the cloud server’s micro-TPM, then sends back to the consumer for the verification purpose of the integrity measurements. Six, the consumer verifies the signature and checks the integrity measurement which is corresponds to the current integrity measurements of the target hypervisor, where the virtual machine is supposed to launch and run.

If the verification is successful, the consumer establishes a secure communication channel with the privileged operations virtual machine and sends the symmetric key necessary to decipher the encrypted virtual machine image (7). An approach for exchanging the key could be using the public portion of an attestation identity key associated with the micro-TPM of the destination server. The virtual machine image can be sent by the consumer or be collected from a local repository (8). Finally, the VM image
is deciphered using the key the consumer provided (9) and the process is concluded with the launch of the virtual machine in the cloud server (10). The communications between consumer agent and privileged operations virtual machine software agent are handled in the destination server agent which can be software part of the management software of the server.

This management software is part of the management virtual machine as seen in Figure 4.1. In step six of Figure 4.6, the secure communications channel between privileged operations virtual machine can use inter-virtual machine communication solutions to avoid bloating the trusted computing base in the privileged operations VM. The advantages and performance of using such channels to handle communications between virtual machines was already discussed in previous work [92].

**Virtual Machine Migration**

From the consumer’s perspective, verifying if the initial server (where his virtual machine is instantiated) is trustworthy is not enough because virtual machines can be migrated to different cloud servers within the cloud infrastructure. However, this problem can be easily solved by having the source cloud server verify if the destination server is trustworthy. The consumer has already verified that the initial server is trustworthy, so that guarantees that the virtual machine migration operation can be trusted to perform the steps represented in Figure 4.7. The communications between the involved servers are handled in the respective management virtual machine. The data can then be sent to the privileged operations virtual machine using inter-virtual machine communication channels in order to keep the trusted computing base to a minimum.
Consider a scenario where an attacker tries to migrate a virtual machine from a trustworthy server to a platform over which he has total control. This attack wouldn’t go through because the source server would reject the migration when the integrity measurements do not match a trustworthy virtualization environment. Furthermore, the privileged operations virtual machine should have software agents responsible for logging and auditing these operations to detect any anomalies.

The remote attestation process can be happens between the cloud server and cloud consumer, to check the integrity of the source platform where the cloud server launches the consumer virtual machine. The Figure 4.7 explains the remote attestation process, the cloud server is placed at remote location. The initial process starts with the cloud server agent sends a migration request. The cloud server sends the remote attestation of target server’s platform integrity, then the destination server request for the quote operation from the virtual machine micro-TPM. Next, the signed platform integrity of the cloud environment is sent to the requested server. Fourth, the platform integrity and signatures
are verification will be happens at trusted party side to decide whether a given platform is trustworthy or not. Fifth, if the verification is successful then the encrypted virtual machine will be sent along with the decryption key to the destination server, where the consumer virtual machine will be launched. Sixth, the destination server decrypts the virtual machine and launches in the verified platform.

**Virtual Machine Backup**

One of the major attractive characteristic of cloud storage is to provide an environment that might almost possible for successful accomplishment of the on-demand network access [3]. The cloud system scenario will be like consumer stores the data and then access the contents by using the interfacing components. The online storage system must be available to the cloud user until consumer stop receiving services. To ensure the data availability, the cloud provider must maintain a backup service from the cloud components. The backing up consumer information is also presents the some security requirement, which are follows and must be satisfied.

![Fig 4.8 Trusted VM Backup](image)

Fig 4.8 Trusted VM Backup
The virtual machine backup process assumes that virtual machine which is launched on cloud server is trustworthy. It can be verified by using the remote attestation process and they will launch if the virtual machine is trustworthy. If the virtual machine is undergoing to the backup server then the virtual machine migration step will takes place before the virtual machine snapshot is backup. The virtual machine backup steps are explained in Figure 4.8. The virtual machine backup process is very simple and process of backup is as follows: first, the consumer agent sends an encryption key and VM backup request for cloud service provider or designated cloud server.

Then, then cloud server takes the VM snapshot of the consumer virtual machine and encrypts it with key, which is provided by the consumer. During the virtual machine launch, encrypt with same key to launch the target virtual machine. Finally the encrypted virtual machine will be storage in the designated storage location.

**Virtual Machine Termination**

The virtual machine termination is simple process and it should be discussed to mitigate the security vulnerabilities in the process. This can be important when the memory space is due during the life cycle of the virtual machine.

Suppose, let us assume that where the application executing in virtual machine persists the private key for the key exchange to sign the values. This clearly states that private key is present in the volatile memory of the virtual machine for signing operation. After termination, the memory space of the virtual machine must be zeroed or randomized, if it is not happening then the attacker creates a special purpose Linux image to corrupt the virtual machine and try to collect the sensitive data from the remainder of the memory.
The virtual machine can be termination process is described and illustrated in the Figure 4.9. At initial stage, the cloud consumer can request to the cloud server for the virtual machine termination. Then, the consumer can encrypt and store the virtual machine snapshot and it must be located in the cloud infrastructure. The cloud consumer can request to launch the virtual machine and decrypt it with the same symmetric key in the cloud premises.

For maintain the security during termination process, the cloud server must terminate the virtual machine with zeroes or randomizes the memory space and they encrypt the virtual machine snapshot using private symmetric key. Then, the virtual machine will be stored in designated location within the cloud premises. The cloud server make sure about the virtual machine state i.e., whether those locations are zeroes or randomized. If it not randomized, then the termination process must happen to protect the security sensitive information.
4.6 IMPLEMENTATION

This section describes the proof-of-concept of our proposed protocol architecture as described in earlier sections. In order to ensure the trustworthiness of cloud platform a remote attestation concept, we used most popular and widely used method called Integrity Measurement Architecture (IMA) [93]. Remote attestation uses IMA; it works based on binary attestation concept. After configuring IMA, it calculates and extends the hashes of all components while boot process into their respective PCRs. To ensure the remote attestation with privacy preserving of NC, we used Attestation Identity Key (AIK) for signing hashes of PCRs while performing quote operation. We used TPM emulator for communication with TPM device using TPM driver.

The NC is the central component of our proposed eucalyptus cloud framework, where CC can launch and execute VMs. Eucalyptus cloud software installed as NC in HP ML150 server with 1.67GHz Xeon processor and 12GB of RAM or primary memory. Initially, NC running on Ubuntu 12.04 with libvirt hypervisor process. Here, cloud clients are divided into two groups: sensitive group users and normal users.

The NC’s are register with the front-end node for user interface. The consumer request NC with public key Pk_c and nonce value for VM execution. The NC verifies the public key Pk_c and nonce then forwards the public key and nonce (which is received from the consumer) to the TrustVisor module through TPM. TrustVisor verifies the Pk_c and then acknowledge with the nonce to consumer for private key pk_c. The pk_c of consumer provided to TrustVisor for decryption of VM and NC decrypt VM with pk_c. Now, consumer can access the VM and every VM consist μTPM for securing the contents by its own. The
μTPM is a part of the TrustVisor on NC CPU, such that it eliminates the unfavorable performance impact on frequent usage of dynamic root of trust [88] (vTPM [95], Flicker [87]). The TrustVisor is a bare metal hypervisor, which provides an isolated execution environment and “micro” TPM concept for securing VM contents through Secure Sensitive Code Block (SSCB). Secure sensitive code blocks are formally known as Piece of Application Logic (PAL). The PAL provides an isolated execution environment by using trusted execution environment technology.

4.7 CONCLUSION

In this chapter, the proposed trustworthy architecture provides the data integrity and confidentiality of cloud consumer virtual machines. The proposed framework provides a trust over the cloud resources using secure virtual machines launch and remote attestation protocols.