Research on an Effective and Secure Cloud Tenants Mechanism

Hui Xia[1] and Weiji Yang[2]
(Corresponding author: Weiji Yang)
Shenyang Normal University, Shenyang 110034, China[1]
Zhejiang Chinese Medical University, HangZhou 310000 , China[2]
(Email: yangweiji@163.com)
(Received May 2, 2018; Revised and Accepted Dec. 20, 2018; First Online July 30, 2019)

Abstract

Tenant separation mechanism paly an important role for cloud computing to be offered as a third-party service, so the tenants' confidence in security and effectiveness is critical to the promotion of cloud comput services.However,tenants in a third party can hardly participate in the the construction and management in cloud conputing service, which make it difficult for the tenants to establish trust sense of separation mechanism in cloud.The paper manily proposes the transparent separation mechanism of trusted cloud tenants,and transforms tenant separation mechanism and tenant transparency requirements into information flow between different security domains in the cloud computing system, and then defines the cloud tenant separation mechanism and the inter-domain information flow policy control mode,finally, proves that the defined cloud tenant separation mechanism is secure and effective by non-interference theory.

Keywords: Cloud Computing Systems; Inter-Domain Information Flow; Tenant Separation Mechanism; Transparency

1 Introduction

Cloud computing provided by cloud service provider (CSP) for multiple tenants, is a kind of resource-reusing service [6,14] to share the computing resources, so the effectiveness and security of tenant separation service will be key premise of acceptance by tenants.The so-called tenant separation refers to prohibiting every information flow from flowing in the tenant security domain, to ensure data against the interference or detection [1, 7] by other tenants in cloud computing system. There are many measures to deal with the issues of tenant separation so far, such as network separation mechanism, virtual machine technology, access control, security audit, security monitoring, storage and communication encryption and so on [2, 5, 6, 9].

The tenants' confidence in the cloud security and effectiveness is critical to accept and adopt to cloud computing services. To some extent tenants separation measures may enhance the confidence of the tenants, however, tenants can not fully build enough confidence only rely on those measures, beacuse they have few opportunities to take part in the construction,operation and management of the infrastructure of cloud computing. If we can establish a transparent mechanism that tenants can also comprehend the principle,implement and manager the tenant separation mechanism [8, 10, 12, 15–17, 21], they may be more willing to believe the mechanism in cloud computing system. Therefore, many researchers have focused on how to achieve credible cloud services through a transparent mechanism.

However, those transparency mechanisms mainly focuse on some attributes evaluation, measurement, reputation and verification to establish a sense of trust. It is is essentially kind of black box testing not to involve any principle and details of performance and functions, but to evaluate those attributes of the cloud computing services. So those mechanisms can not meet security requirements of the tenants.

The paper proposed a method that provides internal details of measurement and verification of cloud tenant separation mechanism for tenants to achieve security and effectiveness. The purpose of the transparency of cloud tenant separation mechanism is to make the tenants get sufficient information about the mechanism, such as the related policy and running process information, which is essentially a kind of information flow among the security domains, so that tenants can measure and validate its security at any time.

In order to achieve the above target, the paper studies on cloud tenant separation mechanism and the requirement of tenants transparency based on the information flow of different domains, and integrates them based on the inter-domain information flow control policy in cloud computing systems, then establishes cloud tenant separation mechanism for transparency requirements; moreover, this paper also uses the theory of non-interference

of information flow, and proves that the proposed cloud tenant separation mechanism is secure and effective by non-interference theory.

2 Related Research

The credible assurance of cloud tenant separation mechanism can be studied from three aspects: Trusted computing platform technology; Software architecture and code size, and Tenant transparency and control requirements.

1) Trusted computing platform technology is based on the hardware password module (TPM), the trusted software stack (TSS), and trusted network connection (TNC), to protect the integrity of the cloud tenants separation. In this scheme, the cloud computing system start from a credible initial state to ensure the operation state in the whole service process in line with expectations [3, 10, 11, 15, 19–21] by the chain of trust, attestation, trusted storage and trusted network and credible assurance mechanism. It does not regard tenants as the object of credible assurances based on trusted cloud tenants separation mechanism, and it is implemented in CSP.

2) Software complexity is an important factor affecting the software reliability, and it include the complexity of structure and the software code size. Generally, more scale of software code will lead to more defects and security vulnerabilities, and its credibility will be low. Therefore, reducing the size of the code is an important method to improve the credibility of cloud computing [18]. Howerver, the size of the code can not be decreases infinitely for cloud computing with integrated service platform, but this set in one of the various components, therefore, reducing the complexity of the software structure has become a meaningful research direction. According to the Murray [13], the size of software interface and code is a major reason of more software errors. Like the trusted tenant separation mechanism, the cloud tenants separation mechanism based on software structure and code size is also not related to tenants. It only uses unilateral credibility guarantee of CSP to improve the security operation of cloud service systems.

3) The cloud tenants separation mechanism based on tenants transparency requirements and controllable requirements overcomes the limitations of the previous two methods. It aims at the tenants' credible requirements and truly improves the tenant's confidence and trust. Cloud computing service is a third-party service mechanism. That is, the construction and management of the system is generally undertaken by the CSP. To improve the tenant's confidence in the cloud service, the tenant must actually participate in the management of the cloud service [4, 17] and let the tenants know as much as possible about the internal strategy and operational details of the cloud tenant isolation mechanism. For example: Kaufman proposes to provide tenants with a secure application programming interface (API) in the cloud computing system, so that tenants can monitor and evaluate the cloud computing service process themselves [17]; Other studies have also given ways and recommendations for improving transparency in cloud computing systems [10, 15, 21]. However, these studies often focus on measuring (including self-assessment or word of mouth) and verifying certain external attributes of cloud services, such as some features and performance, etc. Because the measurement method does not involve the internal structure and strategy of the details of cloud tenants separation mechanism, so it is very difficult to obtain the real structure and operation status of cloud tenants separation mechanism for tenants, and it can not meet the requirements of the cloud tenants' high security.

Different from the above research, this paper regards the transparency requirement as the information flow between different security domains in the cloud computing system. It transmits the internal policy and real-time running informationfrom the cloud management platform security domain to the tenants,which is a method and means for tenants to measure and verify the cloud tenant separation mechanism. At the same time, as the measurement and verification goes deep into the internal principle and real-time status of the cloud separation mechanism, it provides a higher confidence guarantee for the tenant to determine whether the cloud tenant separation mechanism is credible. The main contribution of this paper is to meet the requirements of tenants for data transparency by transferring information flows between different security domains in the cloud service system. A credible cloud tenant separation mechanism for transparency requirements is proposed. In addition, this paper proves the safety and effectiveness of the proposed mechanism through the information flow non-interference theory, and further improves the confidence level of the tenant's separation mechanism for cloud tenants. This is another major contribution of this paper.

3 Tenant Separation Policy Mechanism

If there is information exchange between two different security domains, they must have common accessible address space or communication connection between them. Therefore, to meet the security isolation requirements between tenants in the cloud computing platform,they should ensure there is no cross-overlapping accessible address space between them, and there is no direct communication connection between different tenants. This section proposes the inter-domain information flow policy

in cloud computing to meet the requirements of maximizing cloud resource utilization and security isolation,which is based on the resource reuse requirements and resource management features of cloud computing.

3.1 Cloud Computing Security Domain Division

In the cloud computing system, the computing resources include two parts, Computing time resource and Computing space resource:

- Computing time resource can be simply calculated by CPU computing time, including total computing time and per unit computing time of CPU. Cloud management platform (CMP) allocates the corresponding CPU calculation time to the tenant according to the service level agreement (SLA);
- Computational space resources include physical and logical storage resources such as memory, disk, I/O, and their scope can be identified by the address space in which the resource is located.

In order to simplify the discussion of the problem, this paper does not consider the calculation of time resources, and only uses the computing resource address space to represent the cloud computing resources. Thus, the management of the computing resources of the system is represented by the management of the resource address space. For example, if the system allocates a new virtual machine to the tenant, it means that the computing resource address space owned by the tenant increases; the operation of the computing resource by the system or tenant is expressed as reading and writing the content of the resource address space.

In a cloud computing system, a cloud computing platform consists of multiple security domains, including:

1) CMP: CMP communicates with tenants and provides services to tenants;
2) Tenant (tenant) domains: They are assigned by CMP to the corresponding tenant according to the service contract.
3) System resource pool (SRP): This type of resource is managed by CMP, but may be assigned to tenants as needed.

In any state, these three types of resources are a division of the cloud computing system address space, and there is no overlap between them. This kind of address space division of cloud computing reflects the security isolation feature of cloud computing, but this division is dynamically changed. The system dynamically allocates resources from the SRP to the tenant through the CMP, or recycles the resources in the tenant domain and returns it.

3.2 Tenant Segregation

In this paper, $M(D, \rightarrow)$ is used to represent the cloud computing system:

- $D = \{P, R, T)1, T_2, \cdots, T_n\}$, P is the security domain where the CMP is located, R represents SRP, $T_i (1 \leq i \leq n)$ is the security domain corresponding to the tenant i;
- $>\!\!\rightarrow \subseteq D \times D$ to $\forall u, v \in D$, $u >\!\!\rightarrow v$ indicates that information can flow from the security domain u to the security domain v, or u interferes with v.

Obviously, $src : C \rightarrow D$ satisfies the reflexive relationship. For the convenience, the symbol $>\!\!\vdash\!\!\rightarrow$ means no interference, $u >\!\!\vdash\!\!\rightarrow v$ indicates u does not interfere with v.

Use H to represent address space set of $M(D, >\!\!\rightarrow)$, and S to represent the state set of system $M(D, >\!\!\rightarrow)$, and $s_0 \in S$ indicates the initial state of the system. According to Section 2.1, in any state $P, R, T_1, T_2, \cdots, T_n$ are all a division of H, i.e. $H = P \cup R \cup T_1 \cup \cdots \cup T_n$. Use the function $domh : S \times D \rightarrow D$ represents the actual address space corresponding to the security domain in a specific system state. The function $h : S \times D \rightarrow 2^H$ indicates the security domain to which an address space belongs in a specific state and $domh : S \times H \rightarrow D$ indicates the security domain to which an address space belongs in a specific state, V represents the set of all possible values of the address space H, and the function $val : S \times H \rightarrow V$ represents the value of an address in a particular state in $M(D, >\!\!\rightarrow)$. For simplicity, we use the assignment of "0" to reset or clear an address (or device). For example, $val(s, h) = 0$ means to reset or clear the address h under state s.

A is used to express all the actions set of the system, O is the system output set, function $dom : A \rightarrow D$ represents the security domain corresponding to each action, step: $S \times A \rightarrow S$ represents system transition function, $obs : S \times D \rightarrow O$ represents the system output observed by a particular security domain in a certain state. $s \cdot \alpha$ represents the state reached from the state s via the action sequence $\alpha \in A*$. If ε is used to represent the sequence of empty motions, $\alpha \in A$, then $s \cdot \varepsilon = s$, $s \cdot \alpha a = step(s \cdot \alpha, a)$.

The values corresponding to a particular address space are related to the state of the system, and they may change due to actions in the system. Without loss of generality, we assume $\forall s, t \in S$, $r \in H$, $a \in A$. $var(s, r) = val(t, r) \Rightarrow val(step(s, a), r) = val(step(t, a), r)$. That is: for a specific storage address, the change in its stored value is only related to system actions. The system output observed in a security domain consists of two contents: the address space range and the value corresponding to each address. That is, the system output function can be specifically defined as follows: $\forall d \in D$, $s \in S$, $obs(s, d) = \{(m, val(s, m)) | \forall m \in h(s, d)\}$.

3.2.1 Path (channel)

As $D = \{P, R, T_1, T_2, \cdots, T_n\}$ is a division of H, there is no common accessible address space between any two different security domains in the cloud computing system $M(D, >\rightarrow)$, and only inter-domain communication can be realized through channels. According to the inter-domain communication of the cloud computing platform Isolation requirements, there should be no information exchange between any two tenants, but in order to achieve the dynamic reuse of cloud resources, each tenant should communicate with the CMP to submit resource requests or return unused resources to the system. To avoid the abuse of resources, the cloud computing system prohibits tenants from directly accessing SRP. Tenants can only obtain or return resources through CMP.

To simplify the description of the channel, this paper assumes that one channel only supports one-way communication. $C \subseteq H \times H \times S$ is used to represent the channel set of $M(D, >\rightarrow)$ in a specific state, $c =< h_1, h_2, s >\in C$, h_1 represents the source address of channel c, h_2 represents the destination address of channel c. Use $src : C \rightarrow D$ to indicate the source domain of the channel, that is, the security domain of the write channel; $tag : C \rightarrow D$ represents the destination domain of the channel, that is, read the security domain of the channel. Support one-way communication requirements:

$$\forall c \in C \Rightarrow src(c) \cap tgt(c) = \phi.$$

In order to meet the tenant isolation requirements, the channel is either originated from the CMP or terminated at the CMP, that is $\forall c \in C \Rightarrow src(c) = P \vee tgt(c) = P$.

At the same time, all tenant security domains must be under CMP management, namely: $\forall u \in D - P \Rightarrow \ni c_1 =< h_1, h_2, s >\in C \wedge domh(s, h_1) = u \wedge \ni c_2 =< h_3, h_4, s >\in C \wedge domh(s, h_3) = P \wedge domh(s, h_4) = u$.

3.2.2 Resource Reuse and Remaining Information Protection

The channel proposed in Section 2.2.1 and its rules can prohibit explicit information flow between tenant security domains, but the resource reuse mechanism of cloud computing may still lead to implicit information flow between tenant security domains. For example: If the storage resources returned by a tenant to the system are not cleaned up and assigned to the next tenant, the information remaining on these storage resources will be observed by other tenants.

In order to eliminate this implicit information flow between tenant security domains under the resource reuse mechanism, the system needs to meet the following resource management requirements:

Requirement 1: $\forall r \in H$, $domh(s_0, r) = R \Rightarrow val(s_0, r) = 0$;

Requirement 2: $\forall s \in S$, $\forall r \in H$, $a \in A$, $domh(step(s, a), r) \neq domh(s, r) \wedge domh(s, r) = R \Rightarrow dom(a) = P$.

Requirement 3: $\forall s \in S$, $\forall r \in H$, $a \in A$, then: $domh(step(s, a), r) \neq domh(s, r) \wedge domh(s, r) \neq R \Rightarrow domh(step(s, a), r) = P \wedge val(step(s, a), r) = 0 \wedge dom(a) = P$.

Requirement 1 indicates that all address spaces in the SRP must be emptied when the system is initialized; Requirement 2 states that all resources must be retrieved from the SRP by the CMP and assigned to the tenant; Requirement 3 states that the resources in the cloud computing are either continued to be used by the tenant, or reclaimed by CMP and returned to SRP after being emptied.

3.2.3 Tenant Transparency Mechanism

The tenant transparent mechanism means that the status information in the CMP should be as transparent as possible to the tenant without violating the tenant isolation mechanism. The status information in CMP can be divided into three categories: Type 1 status information is closely related to the privacy protection of all tenants and cannot be open to any tenant. Once opened, the tenant will be informed of other tenants' information; the second type is independent of the specific tenant privacy and can be open to all tenants. For example, the version information of the basic software used in the cloud computing infrastructure; the third type is related to a specific tenant and can only be opened to the corresponding tenant. Use $P_t = \{P_{nr}, P_r, T_{1r}, T_{2r}, \cdots, T_{nr}\}$ to represent a division of P, where:

- P_{nr} represents Type 1 status information and cannot be open to any tenant;
- P_r indicates Type 2 status information, which can be read for all tenants, but no tenant can change it;
- $T_{ir}(1 \leq i \leq n)$ represents the third type of status information, that is, only open to Tenant i, Tenant i can read or change its status.

Channels can be used to implement tenant transparency mechanisms, such as using a source-originated CMP channel to provide tenants with system state information they want to know and allow to know.

The channel can be used to implement the tenant transparent mechanism, for example use a source sent in the CMP channel to provide tenants with the system status information that they want to know and admit to know. Using the function b: $S \times H \rightarrow P_t$ to indicate that an address space in the CMP belongs to a region in P_t, we have the following rules:

Rule 1. $\forall c =< h_1, h_2, s >\in C \Rightarrow b(s, h_1) \notin P_{nr}$;

Rule 2. $\forall c =< h_1, h_2, s >\in C \wedge b(s, h_1) \in T_{ir} \Rightarrow domh(s, h_2) \in T_i$;

Rule 3. $\forall c =< h_1, h_2, s >\in C \wedge domh(s, h_2) \in T_i \Rightarrow b(s, h_1) \in T_{ir} \cup P_r$;

Rule 4. $\forall c =< h_1, h_2, s >\in C \wedge domh(s, h_1) \in T_i \Rightarrow b(s, h_2) \in T_{ir}$;

Rule 5. $\forall i, j, 1 \leq i \leq n, 1 \leq j \leq n, i \neq j \Rightarrow T_{ir} \cap T_{jr} = \phi$.

Rule 1 means that it is not possible to have a channel source in an area of the CMP that is not open to tenants; Rule 2 indicates that a channel can only terminate in the tenant security domain if it originates in an area of the CMP that is only open to specific tenants.; Rule 3 means that if a channel terminates at a tenant, it either originates from an area in the CMP that is open to all tenants, or originates from an area that is only open to that tenant; Rule 4 means that if a channel source originates in a tenant security domain, it must terminate in an area of the CMP that is only open to specific tenants. Rule 5 indicates that there is no intersection in the CMP that is open to different tenants.

4 Feasibility Analysis and Verification

4.1 Feasibility Analysis

The cloud tenant separation strategy stated above is feasible and reasonable in technology. First, the main difficulty of the tenant security domain in isolation mechanism lies in the security isolation between resources occupied by different tenant security domains on the shared platform. For example, virtual machines assigned to tenants, storage resources, and network resources have no overlapping intersections with other tenants. Because virtual machine technology only achieves isolation between virtual machines, however, each tenant may have multiple virtual machines at the same time. Therefore, Virtual Local Area Network (VLANs) and other technologies are required to implement identification and isolation between virtual units of different tenants, take 802.1Q for example, it may need to be implemented in the storage system through security mechanisms for the isolation of tenant storage resources, such as access control and data encryption. In a shared network of a cloud computing platform, VPN is an optional mechanism to achieve separation of different tenants.

Second, it is the feasibility of the remaining information protection. When cloud computing resources are reallocated, we clear the reclaimed resources to avoid indirect traffic between tenant domains in Section 2.2.2, in order to achieving the remaining information protection for computing resources, different types of cloud computing services faces a different difficulty, for example, infrastructure-as-a-service (IaaS) and platform as a service (PaaS), After the tenant returns the virtual machine, the CSP can clear the computing resources by deleting, re-creating, or cloning modes; however, after the tenant returns the computing resources, it is more difficult for the CSP to clear the related resources for software as a service (SaaS), the reason is that these tenants may have an impact on the state of the underlying system platform during using resources, and these effects are difficult to be cleared by system restart, because there may be other tenants using these platforms, it is necessary to provide support at the service-related application level of SaaS, and clean up or empty the relevant status after the tenant returns the service resources.

Again, the performance of the remaining information protection mechanisms. During the process of clearing the storage resources,such as the disk returned by the tenant, the emptying of the disk involves rewriting the returned disk space (otherwise, the information about the former tenant is also saved on the disk). The writing process of ordinary disks (such as SATA and SAS disks) is extremely time consuming; this dynamic multiplexing mechanism of disk resources will result in a large amount of disk rewriting behavior for a large number of tenant services in a cloud computing system, while disk IOPS (The number of reads and writes per second is a major factor affecting the overall performance of the cloud computing system). The system's emptying of the disk during the emptying of storage resources,such as disks returned by the tenant involves rewriting disk space (otherwise, the information about the former tenant is also stored in Disk), ordinary disk (such as SATA and SAS disk) write process is extremely time-consuming; disk resources, this dynamic reuse mechanism will lead to a large number of disk rewriting behavior, and Disk Input and Output Per Second (IOPS) is a major factor affecting the overall performance of cloud computing systems. To solve this performance problem, you can use disk asynchronous clear mode, the so-called asynchronous disk emptying means: after the disk storage resource is returned, the system marks this part of the disk space as "not cleared". All disk storage resources whose status is "not cleared" cannot be reassigned to the tenant. The system processes the "uncleared" disk storage space through a special asynchronous process. The asynchronous process rewrites the corresponding disk space by utilizing the system idle time slice without affecting the overall performance of the cloud computing service. The disk space can be reassigned to the tenant only after being emptied; to ensure that the asynchronous disk storage mode works properly, the CSP needs to allocate a certain amount of redundant disk space.

Finally, it is the channel and the corresponding tenant transparency mechanism. As a communication carrier between the security domain and the CMP, the channel needs to undertake the transmission of certain management commands, such as the management commands sent by the Hypervisor to the virtual machine, and also the data transfer between the security domain and the CMP. For the former, it is often reflected in the virtual system, such as Event Channel and Hypercall in Xen; for the latter, the main consideration should be the confidentiality and integrity of data transmission.

Therefore, VPN is a good solution. On the basis of ensuring the confidentiality and integrity of the transmis-

sion information, the feasibility of the transparent mechanism is mainly focused on the relevant information of the CMP, such as the tenant virtual machine running status and current. The encapsulation of the implemented security policy should ensure the reliability and verifiability of this information. In response to this problem, vTPM based on the combination of trusted computing and virtualization technology will be a feasible solution. CMP collects current virtual machine running status information from different virtual machines, passing through the vTPM of the virtual machine. After the AIK is signed, it is aggregated to the CMP for verification, encapsulation and re-signing with the AIK of the CMP, and then sent to the tenant via the channel, ensuring the reliability of these transparency information.

4.2 Prototype System Validation

Figure 1 is a schematic diagram of the prototype system, verifying several key techniques proposed in this paper. In Figure 1, the cloud computing environment provides services for tenants consists of three parts: CMP, compute cluster, and storage cluster. The compute cluster mainly assumes the operation of the virtual machine, while the storage cluster mainly provides storage service for the virtual machine. The separation mechanism proposed in the prototype system is mainly reflected in the following aspects:

1) Virtual machines (groups) that provide services for different tenants are isolated using VLAN technology to meet the separation mechanism of the tenant security domain.

2) Use of access control technology to ensure that different virtual machines (groups) between the physical storage access control, to achieve the separation of storage resources;

3) The tenant uses the VPN to connect to the external interface of the CMP to use the cloud computing service, which is a kind of separation of the tenant space;

4) In the storage cluster, based on the parent-child and COW (copy-on-write) mechanism, the storage cluster can realize the initial allocation, recovery and redistribution of disk resources, and also consider the asynchronous clearing of the disk, and realize protection of the remaining information;

5) The channels in the cloud environment are reflected in the VPN connection between the tenant and the CMP, and the management of the virtual machine by calling the Event Channel.

6) CMP collects the current transparency certificate from the virtual machine, and provides it to the tenant along with the cloud computing service after encapsulation. This is the embodiment of the transparency requirement of the prototype system.

5 Safety Analysis

To prove the security separation and protection capability of tenant's information flow between the security domains, the non-interference theory is undoubtedly a good method. However, when using the non-interference theory tools and methods to prove the security effectiveness of the tenant isolation mechanism in cloud computing services, it is necessary to fully consider the specific characteristics of the cloud computing service, such as the dynamic multiplexing capability of resources.

The definition of non-interference is used in the information security field to describe the interference relationship between security domains, that is, the information flow between different security domains. If any action of the security domain u does not make the security domain v aware, i.e., these actions of u do not change the system output that can be observed by v, then u means no interference to v.

We will show that the tenant separation mechanism given is safe and effective in Section 2. Firstly, Meyden's TA-security theorem [18] is presented before concrete proofs are given.

For $M(D, >\!\!\rightarrow)$, $u \in d$, $a \in A$, $\alpha \in A^*$, function ta_u definition:

1) $ta_u(\varepsilon) = \varepsilon_1$;

2) If $dom(a) >\!\!\vdash\!\!\rightarrow u$, then $ta_u(\alpha a) = (ta_u(\alpha)$;

3) If $dom(a) >\!\!\rightarrow u$, then $ta_u(\alpha a) = (ta_u(\alpha), ta_{dom(a)}(\alpha), a)$.

Meyden gives a system security definition based on above theory: in system If $M(D, >\!\!\rightarrow)$, for $\forall u \in D$, $\forall \alpha \in A^*$ and $\alpha' \in A^*$, if $ta_u(\alpha) = ta_u(\alpha')$, there are $obs(u, s_0 \cdot \alpha) = obs(u, s_0 \cdot \alpha')$, then the system $M(D, >\!\!\rightarrow)$, is 'TA-' safe to the strategy $>\!\!\rightarrow$. The following theorem of system security decision is also given [18].

Theorem 1. *If there is weak unwinding in system $M(D, >\!\!\rightarrow)$ about the strategy $>\!\!\rightarrow$, then $M(D, \rightarrow)$ is 'TA-' safe about the strategy $>\!\!\rightarrow$. Wherein, the weak unwinding of the system $M(D, >\!\!\rightarrow)$ on the strategy refers to the relationship family $\sim_u$ about D that satisfies the following conditions:*

1) If $s \sim_u t$, then $obs(u, s) = obs(u, t)$; (Output consistency, referred to as OC);

2) If $s \sim_u t$, and $s \sim_{dom(a)} t$, then $s \cdot a \sim_u t \cdot a$ (Weak Single-step Consistency, referred to as WSC);

3) If $dom(a) >\!\!\vdash\!\!\rightarrow u$, then $s \sim_u s{\cdot}a$. (Local Recognition, referred to LR).

Unless otherwise stated, the system $M(D, >\!\!\rightarrow)$, mentioned later in this paper refers to the cloud computing system that satisfies the various definitions, requirements and rules of Section 2.

First, we give the definition of inter-domain information flow for cloud computing system $M(D, >\!\!\rightarrow)$,.

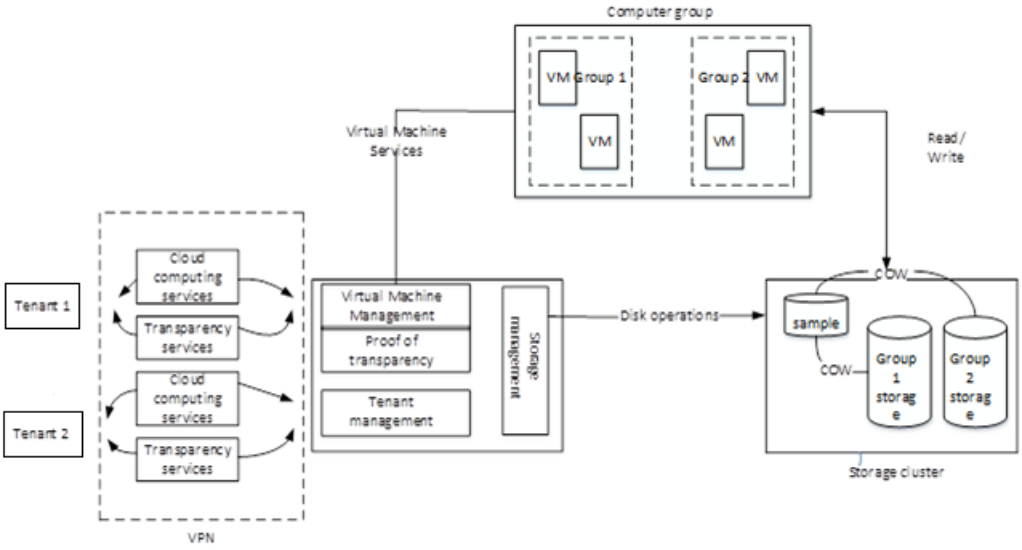

Figure 1: Schematic diagram of the prototype system

Definition 1. *The set of inter-domain information flows in the cloud computing system $M(D, >\!\!\rightarrow)$ is defined as follows:*

1) $F = \phi$;

2) If $\forall c =< h_1, h_2, s >\in C$, then $F = F \cup \{| < h_1, h_2, s > |\}$;

3) If $f_1 = | < h_1, h_2, s > |$, $f_2 = | < h_2, h_3, s > | \in F$, then $F = F \cup \{| < h_1, h_2, s > |\}$.

wherein, $f = | < h_1, h_2, s > | \in F$, $h_1$represents the source address of the information flow f, and h_2 represents the destination address of the information flow f.

Definition 2. *The inter-domain interference relationship in cloud computing system $M(D, >\!\!\rightarrow)$ is defined as: $\forall u, v \in D$, $u >\!\!\rightarrow v$ if and only if $(\ni f)(f = | < h_1, h_2, s > | \in F \wedge domh(s, h_1) = domh(s, h_2) = v)$.*

Lemma 1. *In Cloud computing $M(D, >\!\!\rightarrow)$, for $\forall s \in S$, $r \in H$, if $domh(s, r) = R$, then $val(s, r) = 0$.*

Proof. It can be proved from Requirements 1 to 3 in Section 2.2.2 by the recursion method (The proof is abbreviated). □

Lemma 2. *In Cloud computing system $M(D, >\!\!\rightarrow)$, for $\forall a \in A$, $u \in D - P$, if $dom(a) >\!\!\vdash\!\!\rightarrow u$, then $dom(a) \neq P$.*

Proof. All tenant security domains must be managed by CMP according to the assumptions in Section 2.2.1, i.e.: $\forall u \in D - P \Rightarrow \ni c_1 =< h_1, h_2, s >\in C \wedge domh(s, h_1) = P \wedge domh(s, h_2) = u$. Therefore, if $dom(a) >\!\!\vdash\!\!\rightarrow u$ then there must be $dom(a) \neq P$. □

Lemma 3. *In Cloud computing system $M(D, >\!\!\rightarrow)$, for $\forall u, v \in D - P$, $u >\!\!\rightarrow v \Rightarrow u = v$.*

Proof. $\forall u, v \in D - P$, $u >\!\!\rightarrow v$, $(\ni f)(f = | < h_1, h_2, s > | \in F \wedge domh(s, h_1) = domh(s, h_2) = v)$ according to Definition 2. Suppose there is at least one channel between $u, v, u \neq v$; $\forall c \in C \Rightarrow src(c) = P \vee tgt(c) = P$, u, v must pass the information through P according to channel properties in Section 2.2.1, so there is a flow of information:

$$u >\!\!\rightarrow P >\!\!\rightarrow \cdots >\!\!\rightarrow v.$$

First consider the simplest case, $u >\!\!\rightarrow P >\!\!\rightarrow v$, as $u >\!\!\rightarrow P$, $\ni c_1 =< h_1, h_1', s >\in C \wedge domh(s, h_1') = P$. According to Rule 4 of Section 2.2.3, there is $b(s, h_1') \in T_{ur}$; as $P >\!\!\rightarrow v$, there is $\ni c_2 =< h_2', h_2, s >\in C \wedge domh(s, h_2') = P \wedge domh(s, h_2) = v$; according to Rule 3 of Section 2.2.3, $b(s, h_2') \in T_{ur} \cup P_r$. According to Rule 5 of Section 2.2.3, since $u \neq v$, $T_{ur} \cap (T_{vr} \cup P_r) = \phi$, so $b(s, h_1') \cap b(s, h_2') = \phi$ and $u >\!\!\rightarrow P >\!\!\rightarrow v$ contradict, and so $u = v$. Recursive launch when: $u >\!\!\rightarrow P >\!\!\rightarrow t_1 >\!\!\rightarrow P >\!\!\rightarrow t_2 >\!\!\rightarrow P >\!\!\rightarrow \cdots v$, $t_i \in D - P$, $1 \leq i \leq n$, then $u = t_1 = t_2 = \cdots = t_n = v$.

Lemma 3 indicates that no information can pass through any channel between any two tenant security domains. □

Lemma 4. *In Cloud computing system $M(D, >\!\!\rightarrow)$, for $u, v \in D$, then $u \neq v$, $u >\!\!\rightarrow v \Rightarrow u = P \vee v = P$.*

Proof. Suppose $u \neq P \wedge v \neq P$, for $u >\!\!\rightarrow v$, according to Lemma 3, there is $u = v$ and $u \neq v$ contradict, therefore, the hypothesis does not hold. □

Lemma 5. *In Cloud computing system $M(D, >\!\!\rightarrow)$, for $\forall a \in A$, $\forall s, t \in S$, $u \in D$: $obs(s, u) = obs(t, u) \wedge obs(s, dom(a)) = obs(t, dom(a)) \Rightarrow obs(step(s, a), u) = obs(step(t, a), u)$.*

Proof. According to the definition of function $obs(\cdot)$ in Section 2.2, $obs(\cdot)$ is determined by the range of domain address space and its corresponding value, $obs(s,u) = obs(t,u)$ means that the security domain u has the same address space range and each address has the same value under the states s and t.

- When $dom(a) >\!\!\rightarrow u$, i.e.: action a will neither change the address space range of u nor change the value of each address, so there is that is, action a does not change the address space of u, nor change the corresponding value of each address, so: $obs(s,u) = obs(t,u) \Rightarrow obs(step(s,a),u) = obs(step(t,a),u)$;
- When $dom(a) \rightarrow u$, according to Lemma 4, there are three cases:
  1) If $dom(a) = u$, $dom(a)$ is an address of reading and writing operation, a will not change the address space range of u. As assumed in Section 2.2, for $\forall s,t \in S$, $r \in H$, $a \in A$, $va;(s,r) = val(t,r) \Rightarrow val(step(s,a),r) = val(step(t,a),r)$, therefore: $obs(step(s,a),u) = obs(step(t,a),u)$;
  2) If $dom(a) \neq u$, $dom(a) = P$, $dom(a)$ is resource management class, at this time $dom(a) = P$, and a allocates resources for u or reclaims resources from u. In this case, a will change the address space range of u, but will not change the value of each address. According to Lemma 1, if it is a newly allocated resource, its address space has a value of '0'. If the resource is reclaimed, the remaining address space values will not change. Therefore, in the states s and t, after the action a is completed, the address space range of u and the value corresponding to each address remain the same, namely: $obs(step(s,a),u) = obs(step(t,a),u)$;
  3) In Case $dom(a) \neq u$, $u = P$, $dom(a)$ read and write $T_{dom(a)r}$, report the third category of information in the tenant transparency mechanism, as $obs(s,dom(a)) = obs(t,dom(a))$, so action a reports the same state information in $dom(a)$ to $T_{dom(a)r}$ without affecting the values of other address spaces in P_t under the states of s and t, so $obs(step(s,a),u) = obs(step(t,a),u)$.

□

Theorem 2. *Cloud computing system $M(D,>\!\!\rightarrow)$ about the strategy ">→" is 'TA-' safe.*

Proof. To prove that $M(D,>\!\!\rightarrow)$ is 'TA-' safe with respect to the strategy ">→", according to Theorem 1, it must be proved that $M(D,>\!\!\rightarrow)$ has a weak unwinding about the strategy ">→" to satisfy OC, WSC and LR requirements.

- Define the relationship family $D \sim_u$ with respect to D for $M(D,>\!\!\rightarrow)$, which is $s \sim_u t$ if and only if $obs(s,u) = obs(t,u)$, obviously, OC is satisfied.
- According to Lemma 5, there is obviously $s \sim_u t \wedge s \sim_{dom(a)} t \Rightarrow step(s,a) \sim_u step(t,a)$, that is, WSC is satisfied;
- Finally, we need to prove LR, i.e., $dom(a) >\!\!\vdash\!\!\rightarrow u \Rightarrow s \sim_u step(s,a)$. As $dom(a) >\!\!\vdash\!\!\rightarrow u$, and known in Lemma 2 that $dom(a) \neq P$, so $dom(a)$ does not change the u address space range; also for $dom(a) >\!\!\vdash\!\!\rightarrow u$, so $dom(a)$ does not change the value of u's address space. Comprehensive analysis of the above, $obs(s,u) = obs(step(s,a),u)$. In accordance with the definition of $\sim_u$, there is $s \sim_u step(s,a)$, that is, LR is satisfied.

□

6 Conclusion

The research on the trust guarantee of the cloud tenant isolation mechanism includes multiple levels, including integrity guarantee for the system operation process, formal description and proof of the tenant isolation mechanism strategy, tenant transparency and controllability guarantee, tenant trust evaluation and Pass the calculation model and other aspects.Based on tenant transparency, this paper treats tenant transparent requirements as a kind of information flow between the cloud computing platform and the tenant security domain, and integrates this information flow into the tenant isolation mechanism as part of the tenant isolation mechanism. Policy rules to achieve a formal description of the transparency requirements, and the validity of the relevant tenant isolation model.This method of transforming the abstract system's credible requirements into specific formal rules is an innovation, which provides reference and reference for similar research in the future.

Acknowledgments

This work is partially supported by Scientific Study Project for Institutes of Higher Learning,Ministry of Education,Liaoning Province (LQN201720), and Natural Science Foundation of LaioNing Province, China (20170540819). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] D. S. AbdElminaam, "Improving the security of cloud computing by building new hybrid cryptography algorithms," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 40-48, 2018.

[2] Z. Cao, C. Mao, "Analysis of one secure anti-collusion data sharing scheme for dynamic groups in the cloud," *International Journal of Electronics*

and Information Engineering, vol. 5, no. 2, pp. 68-72, 2016.

[3] C. Chen, H. Raj, S. Saroiu, A. Wolman, "cTPM: A cloud TPM for cross-device trusted applications," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, pp. 187-201, 2014.

[4] Y. Chen, V. Paxson, R. H. Katz, *What's New About Cloud Computing Security*, Berkeley Report, No.UCB/EECS-2010-5, University of California, 2010.

[5] M. Chiregi, N. J. Navimipour, "A comprehensive study of the trust evaluation mechanisms in the cloud computing," *Journal of Service Science Research*, vol. 9, no. 1, pp. 1-30, 2017.

[6] Cloud Security Alliance, *Security Guidance for Critical Areas of Focus in Cloud Computing*, July 14, 2019. (`http://www.cloudsecurityalliance.org/guidance/csaguide.pdf`)

[7] D. Contractor, D. R. Patel, "Accountability in Cloud Computing by Means of Chain of Trust," *International Journal of Network Security*, vol. 19, no. 2, pp. 251-259, 2017.

[8] H. Dev, M. E. Ali, T. Sen, M. Basak, "AntiqueData: A proxy to maintain computational transparency in cloud," in *Proceedings of International Conference on Database Systems for Advanced Applications*, pp. 256-267, 2014.

[9] J. Huang, D. M. Nicol, "Trust mechanisms for cloud computing," *Journal of Cloud Computing*, vol. 2, no. 1, pp. 1-14. 2013.

[10] T. Kekkonen, T. Kanstrén, K. Hatonen, "Towards trusted environment in cloud monitoring," in *Proceedings of 11th International Conference on Information Technology: New Generations (ITNG'14)*, pp. 180-185, 2014.

[11] M. S. Kiraz, "A comprehensive meta-analysis of cryptographic security mechanisms for cloud computing," *Journal of Ambient Intelligence & Humanized Computing*, vol. 7, no. 5, pp. 731-760, 2016.

[12] N. Kumar, B. Chakraborti, A. Kumar, S. Giri, "Reduction of cost by implementing transparency in cloud computing through different approaches," in *Proceedings of International Conference on Advanced Communication Control and Computing Technologies (ICACCCT'14)*, pp. 1723-1725, 2014.

[13] D. G. Murray, G. Milos, S. Hand, "Improving Xen security through disaggregation," in *Proceedings of 4th ACM International Conference on Virtual Execution Environments*, pp. 151-160, 2008.

[14] NIST, *The NIST Definition of Cloud Computing*, Sept. 2011. (`http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf`)

[15] A. Patel, P. Dansena, "TPM as a middleware for enterprise data security," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 7, pp. 327-332, 2013.

[16] V. Srinivas, V. V. Kumari, R. Kvsvn, "Perseverance of Uncertainty in Cloud Storage Services through Reputation Based Trust," *International Journal of Network Security*, vol. 20, no. 5, pp. 951-959, 2018.

[17] A. Sunyaev, S. Schneider, "Cloud services certification," *Communications of the ACM*, vol. 56, no. 2, pp. 33-36, 2013.

[18] R. van der Meyden, "What, indeed, is intransitive noninterference?," in *European Symposium on Research in Computer Security (ESORICS'07)*. LNCS 4734, pp. 235-250, Springer-Verlag, 2007.

[19] V. Varadharajan, U. Tupakula, "TREASURE: Trust enhanced security for cloud environments," in *Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'12)*, pp. 145-152, 2012.

[20] J. Wang, J. Liu, H. Zhang, "Access control based resource allocation in cloud computing environment," *International Journal of Network Security*, vol. 19, no. 2, pp. 236-243, 2017.

[21] J. Zhang, S. Yang, T. U. Shanshan, *et al.*, *Research on v TPCM Trust Management Technology for Cloud Computing Environment*, Netinfo Security, 2018.

Biography

Hui Xia is currently an associate professor in Software College of Shenyang Normal University. He received received the B.S. and M.S. degree from XiDian University, China in 2003 and 2006, respectively.He has authored or coauthored more than twenty journal and conference papers. His current Acknowledgments research interests include data mining, privacy preserving and network security.

Weiji Yang works in Zhejiang TCM university, got bachelor's degree of computer and science in 2005, received double master's degrees of engineering and medicine in 2009 and 2014 respectively, the main research area is artificial intelligence, digital medical image processing and analysis, and smart health care, *etc.*