A hybrid security model for virtual machines in cloud environment

Zhaogang Shu*, Xiangmin Ji and Yaohua Lin

College of Computer and Information,
Fujian Agriculture and Forestry University,
Fuzhou, 350002, China
Email: zhaogang.shu@gmail.com
Email: jixm168@126.com
Email: linyaohua018@126.com
*Corresponding author

Abstract: In cloud environment, the communication between virtual machines (VMs) residing on the same physical system is inevitable, which brings convenience for malicious software to attack other VMs through the communication channels between VMs. In order to reduce the security risk of the VMs in cloud environment, a novel hybrid security model (HSMVM) is presented, which combines the features of confidentiality, integrality, and organisation isolation of generic security model. The limitations and the advantages of the existing security models are thoroughly analysed in this paper. Afterwards, the basic definitions of HSMVM, judgment method for the system security state, and the state transformation rules are described in detail. The prototype system of HSMVM is developed and tested on Xen system. Experimental results indicate that the security model can effectively prevent the attacks between the VMs with reasonably low resource utilisation.

Keywords: security model; access control; virtual machine; cloud computing.

Reference to this paper should be made as follows: Shu, Z., Ji, X. and Lin, Y. (2017) 'A hybrid security model for virtual machines in cloud environment', *Int. J. Autonomous and Adaptive Communications Systems*, Vol. 10, No. 2, pp.236–246.

Biographical notes: Zhaogang Shu received his MSc in Computer Science from ShanTou University in 2005 and PhD in Automation Engineering from South China University of Technology in 2008. Until September 2012, he was a Senior Software Eengineer and post-doc researcher at the Ruijie network company. Currently, he is the Director of Cloud Computing Laboratory, Fujian Agriculture and Forestry University (Fuzhou, China). His recent research interests include cloud computing and information security.

Xiangmin Ji received his MSc in Computer Science (2005) from University of Chinese Academy of Sciences and currently a PhD candidate at School of Computer Science, Wuhan University (Wuhan, China). Since 2008, he has been a Lecturer of Computer Science at Fujian Agriculture and Forestry University (Fuzhou, China). His recent research interests include cloud computing and information security.

Yaohua Lin received his MSc in Computer Science from Fuzhou University (2009). Since that, he has been a Lecturer of Computer Science at Fujian Agriculture and Forestry University (Fuzhou, China). His recent research interests include wireless sensor network and mobile ad hoc network.

1 Introduction

Virtualisation is considered as one of the key technologies in cloud computing (Wan et al., 2014; Duan et al., 2012; Chen et al., 2011). From the perspective of end users, the virtualisation technology enables multiple logical virtual machines (VMs) to run on a single physical machine at the same time; where, different operating system and application software can be independently installed on each of the VMs. Although these VMs run independently, sometimes the communication between these VMs is inevitable, which may bring a potential security risk to the system. For example, the private data stored in one VM can be intercepted or illegally modified by another VM on the same physical machine (Xu et al., 2014). The communication mechanisms between VMs include virtual CPU, shared memory, shared files, event channel, etc. Malicious software running on a VM may exploit these communication mechanisms to attack other VMs on the same physical machine (Wei et al., 2010, 2014). In recent years, VMs have been deployed widely in cloud environment to support various applications (Wan et al., 2013; Lai et al., 2013; Liu et al., 2014). The security of VMs has become a serious problem to be solved; therefore, industrial developers and the academic researchers have shifted their focus on the development of secure and optimal systems for cloud computing environment (Suo et al., 2013; Zhang et al., 2013).

In order to strengthen the security of VMs in cloud environment, the major goal is to design a security model to control the process of creation, deployment, deletion of the VMs, and the communication between VMs on the single physical machine. The conventional security models include BLP model (Bell and Lapadula, 1973), Biba (1977) model, Chinese wall (CW) model (Brewer and Nash, 1989). However, these models were not originally designed for VMs in cloud environment. Therefore, they can hardly satisfy the specific security requirements of the VM system. Wei et al. (2009) presented a security framework to manage the VM images in cloud environment; however, it cannot control the communication between the VMs. Wu et al. (2010) presented certain policies to protect the VMs from attack in cloud computing; however, it focuses only on network transmission security of the VM system. Liu et al. (2010, 2011) proposed a VM security model, named Virt-BLP model, which defines a series of transformation rules of the system state for the VM system. Virt-BLP inherits the features of BLP model, however, it does not consider the isolation of the data integrity and organisation for the end-users of different organisations.

Therefore security model of the VM system needs further improvements. This paper aims to combine the advantages of BLP model, Biba model, and the CW model to design a hybrid security model that can better satisfy security requirements of VM system. The main contributions of this study include:

1 a novel hybrid security model for VMs in the cloud system is presented, and theoretically proves the efficacy and feasibility of this model

2 a prototype system for the hybrid security model based on Xen system is implemented, and the experimental results indicate that the security model can effectively prevent the attack on the VMs with reasonably low resource utilisation and higher performance.

The rest of the paper is organised as follows. Section 2 analyses the advantages and disadvantages of the existing security models, and highlights the areas that can be improved. Section 3 presents formal description and proof of the proposed hybrid security model. Section 4 describes the implementation and evaluation of the prototype system for the security model. Section 5 summarises the contributions of this study and describes the future work.

2 Analysis and improvement of security model

In this section, the features of BLP model, Biba model, and the CW model will be analysed, and their advantages and disadvantages will be discussed. Moreover, a summary of existing problems in the practical applications will be provided. Finally, the improvement policies for security model of the VM system will be descried.

2.1 Analysis of related models

BLP model is a multi-level access control security model that mainly focuses on the data confidentiality. In BLP model, various security level (such as top secret, secret, confidential, unclassified) are assigned to different object (such as users, process, or file). However, BLP model does not consider the data integrity; furthermore, its strict security level cannot be directly applied to the VM system. For example, a privileged VM (called VM monitor) can access the resources of all other VMs without any restriction of security level.

On the contrary, Biba model mainly aims to solve the problem of data integrity. In Biba model, all the objects are assigned certain integrity levels to guarantee that data flows from the object of higher integrity level to the object with lower integrity level. For example, Biba (1977) model defines the access rules of forbidding 'up to write, down to read'. When this rule is applied to a VM system, the VM of higher level cannot 'read' from the VM of lower level that contradicts the access requirement of the VM system.

The CW model mainly focuses on the isolation of the shared resources existing on a single physical machine, which can be used to alleviate the security risk of the VM system. If two VMs belong to two competing users, the CW model can guarantee that the two VMs will not be deployed on the same physical machine. However, this policy is just a precaution at the creation phase of the VM; it cannot guarantee the secure access between the VMs after they are created and deployed.

2.2 Improvement of model for VM

According to analysis of the described existing security models, mainly three improvement policies for the security model of VM system are devised in this study:

1 Combining the concepts of subject/object of BLP model with the profit collision class (PCC)/organisation group (OG) of CW model. Every VM is treated either as a subject or object, which must belong to only one PCC and one OG. The VMs treated as subject or object conform to the security transformation rules of BLP, and the deployment rules of the CW model.

2 In order to keep both the confidentiality of BLP model and the integrity of Biba model, the VMs can be assigned with both the security level and the integrality level. Since the transformation rules of BLP are conflicted with that of the Biba model, eclectic modifications of transformation rules are necessary to eliminate these conflicts. The proposed solution follow this principle: when two VMs belong to different PCCs, the communication between them will comply with the transformation rules of the BLP; otherwise, the communication between them will comply with the transformation rules of Biba.

3 Most of the VM systems have a special VM, such as domain0 in the Xen environment (Sailer et al., 2005) that has the highest authority to manage other VMs. This privilege VM is considered as trusted subject in the proposed security model.

3 Design of hybrid security model for VM

Based on the improvement policies mentioned in the preceding section, a novel security model, named as hybrid security model for virtual machine (HSMVM) is proposed. This section will describe the basic definition, judgment method of system security state, and the state transformation rules of HSMVM.

3.1 Basic definitions of HSMVM

Basic definitions of the HSMVM include subject, object, trusted subject, PCC, OG, access attribute, access matrix, security level, integrity level, and the system security state.

3.1.1 Subject, object and trusted subject

In cloud environment, both the terminal systems in the communication are VMs; therefore, both subject and object in the HSMVM refer to the VMs. It is decided on the data flow direction of the communication whether a VM is a subject or object. For example, when VM-A reads data from VM-B, then VM-A is subject, and VM-B is object. These are defined as follows:

Definition 1: Subject set $S = \{s_1, s_2, \ldots, s_n\}$, object set $O = \{o_1, o_2, \ldots, o_n\}$, trusted subject set $S_T \subseteq S$.

Subject set (S) and object set (O) have equal number of elements that is the total number of current VMs on system. If there are n VMs, the subject s_i and object o_i ($1 \leq i \leq n$) denotes the same VM. The privilege VM belongs to the trusted subject set.

3.1.2 PCC and OG

Every subject or object must belong to only one PCC, and one OG, it can be defined as:

Definition 2: OG set $X = \{x_1, x_2, \ldots, x_n\}$, PCC set $Y = \{y_1, y_2, \ldots, y_m\}$.

$X(s_i) \in X$ denotes the OG of the subject s_i, and $Y(s_i) \in Y$ denotes the PCC of the subject s_i.

3.1.3 Security level and integrality level

Every subject or object must be assigned a specific security and integrity level that are used to control the communication between VMs.

Definition 3: Security level set $K = \{k_1, k_2, \ldots, k_n\}$, integrity level set $I = \{i_1, i_2, \ldots, i_m\}$.

$K(s_i) \in K$ denotes the security level of the subject s_i, $I(s_i) \in I$ denotes the integrity level of the subject s_i.

Any two elements of the level set comply with the relationship of strict partial order. For example, two elements satisfy $k_i \geq k_j$, which means that security level k_i is higher than k_j.

3.1.4 Access attribute and access matrix

The communication type between VMs is called access attribute that includes read-only (*r*), write-only (*a*), read-write (*w*), create (*c*), destroy (*d*), modify security level (*ms*), and modify integrity level (*mi*).The access matrix is the collection of all the current access attributes between the subjects and the objects.

Definition 4: The access attribute set $A = \{r, a, w, c, d, ms, mi\}$. Access matrix $M = \{m_{ij} | m_{ij} \subseteq A, s_i \in S, o_j \in O\}\}$, where m_{ij} represents all the access attributes from the subject s_i to the object o_j.

3.2 Judgment method of the system security state

In order to describe more clearly, several symbols are introduced in this paper. $P(A)$ denotes power set of A, which includes all the subsets (such as universal set and null set) of A. Considering A and B are the sets, A^B denotes the mapping from B to A. Furthermore, $A \times B = \{(a, b) \mid a \in A, b \in B\}$ denotes the Cartesian product of A and B.

The whole VM system can be denoted as $V = (B \times M \times F \times G)$, where $B = P(S \times O \times A)$ is the set that maps the access from all the subjects to the objects, $b \subseteq (S \times O \times A)$ is an element of set B that records the access action from current subjects to objects in one system state; M denotes the access matrix; $F = P(K^S \times K^O \times I^S \times I^O)$ denotes the mapping from all the subjects/objects to security/integrity levels, and $f \subseteq (K^S \times K^O \times I^S \times I^O)$ is an element of set F that records current mapping from subjects/objects to security/integrity levels in one system state; Finally, $G = P(X^S \times X^O \times Y^S \times Y^O)$ denotes the mapping from all subjects/objects to PCC/OG, $g \subseteq (X^S \times X^O \times Y^S \times Y^O)$ is an element of the set G that records the current mapping from all the subjects/objects to security/integrity levels in one system state. Therefore, $V_t = (b_t \times M_t \times f_t \times g_t) \in V$ denotes the specific system state at time index t.

It is assumed that $R = \{\text{get, give, release, cancel}\}$ and $RA = R \times S \times O \times A$ denotes all the possible access request actions from subject to object. $D = \{\text{yes, no, error}\}$ denotes the request decision set, where 'yes' means a request is allowed, 'no' means the request is not allowed, and 'error' means unknown request. State transformation of the VM system can be described as $RA \times V \rightarrow D \times V$. If initial state of the VM system and every system state transformation is secure, the whole system can be considered as a secure system. Therefore, the state transformation rules of VM system are very important.

3.3 State transformation rules of HSMVM

In the following state transformation rules, it is assumed that $r_k \in RA$ denotes current request action, $V_t = (b_t \times M_t \times f_t \times g_t)$ denotes the system state before transformation, $V_{t+1} = (b_{t+1} \times M_{t+1} \times f_{t+1} \times g_{t+1})$ denotes the system state after the transformation, and $d_k \in D$ denotes the current system decision. The system state remains unchanged ($V_{t+1} = V_t$) when d_k = no or d_k = error, and changes ($V_{t+1} \neq V_t$) when d_k = yes.

Rule 1: creation of VM: $r_k = \{$ get, s_i, o_j, $c\} \in$ RA represents that subject s_i requests to create an object o_j, if and only if the following two conditions are both satisfied.

$$\begin{aligned} & s_i \in S_T \\ & \neg\exists o_k \in O,\ X\left(o_k\right) \neq X\left(o_j\right) \wedge Y\left(o_k\right) = Y\left(o_j\right) \end{aligned} \tag{1}$$

The first condition implies that the subject VM must belong to the trusted subject set; in other words, s_i must be a privileged VM. The second condition means that the new object o_j cannot coexist with another VM having same PCC, and different OG on a certain physical machine. This condition complies with the organisation isolation feature of the CW model. Finally $V_{t+1} = (b_{t+1} \times M_{t+1} \times f_{t+1} \times g_{t+1})$, where

1 $b_{t+1} = b_t$, $M_{t+1} = M_t$

2 $f_{t+1} = f_t \leftarrow \{K^O \leftarrow K^O \cup (K(o_j), o_j), I^O \leftarrow I^O \cup (I(o_j), o_j)\}$

3 $g_{t+1} = g_t \leftarrow \{X^O \leftarrow X^O \cup (X(o_j), o_j), Y^O \leftarrow Y^O \cup (Y(o_j), o_j)\}$.

Symbol '$A \leftarrow B$' means that A is modified by B.

Rule 2: deletion of VM: $r_k = \{$get, s_i, o_j, $d\} \in RA$ represents the condition when the subject s_i requests to delete the object o_j, and release all the resources owned by o_j; if and only if $s_i \in s_T$. d_k = yes and $V_{t+1} = (b_{t+1} \times M_{t+1} \times f_{t+1} \times g_{t+1})$, where

1 $b_{t+1} = b_t - S \times \{o_j\} \times A - \{s_j\} \times O \times A$

2 $M_{t+1} = M_t \leftarrow \{m_{xj} \leftarrow \varnothing, m_{jx} \leftarrow \varnothing, (1 \le x \le n)\}$

3 $f_{t+1} = f_t \leftarrow \{K^O \leftarrow K^O \cup (K(o_j), o_j), I^O \leftarrow I^O \cup (I(o_j), o_j)\}$

4 $g_{t+1} = g_t \leftarrow \{X^O \leftarrow X^O \cup (X(o_j), o_j), Y^O \leftarrow Y^O \cup (Y(o_j), o_j)\}$.

Rule 3: modify security or integrality level: $r_k = \{$get, s_i, o_j, $ms\} \in RA$ or $r_k = \{$get, s_i, o_j, $mi\} \in RA$ represents that the subject s_i requests to modify the security or integrity level of the object o_j; if and only if $s_i \in s_T$, and d_k = yes. Finally $V_{t+1} = (b_{t+1} \times M_{t+1} \times f_{t+1} \times g_{t+1})$, where $b_{t+1} = b_t$, $M_{t+1} = M_t$, $g_{t+1} = g_t$ and

$$f_{t+1} = f_t \leftarrow \left\{K^O \leftarrow \left(\left(K\left(o_j\right), o_j\right) \leftarrow \left(K'\left(o_j\right), o_j\right)\right)\right\} \text{ or}$$

$$f_{t+1} = f_t \leftarrow \left\{I^O \leftarrow \left(\left(I\left(o_j\right), o_j\right) \leftarrow \left(I'\left(o_j\right), o_j\right)\right)\right\}.$$

$K'(o_j)$ and $\Gamma(o_j)$ represents the modified security and integrity level of o_j.

Rule 4: get read-only-access: r_k = {get, s_i, o_j, r} $\in$ RA represents that the subject s_i requests to access the object o_j with 'read-only' privilege. If one of the following three conditions is satisfied, then d_k = yes.

1 $s_i \in s_T$

2 $X(s_i) \neq X(o_j) \wedge Y(s_i) \neq Y(o_j) \wedge K(s_i) \neq K(o_j) \wedge (r \in m_{ij})$

3 $X(s_i) = X(o_j) \wedge Y(s_i) = Y(o_j) \wedge I(s_i) = I(o_j) \wedge (r \in m_{ij})$.

The first condition ensures that the privilege VM can access any of the other VMs. The second condition implies that a subject with higher level of security than the object is allowed to access the object with 'read-only' privilege, when the subject and object do not belong to the same PCC and ORG. The third condition means: when the subject and object belong to the same PCC and ORG, and the integrity level of the subject is lower than the object, then the subject is allowed to access object with 'read-only' privilege. It can be seen that the second condition is inherited from the BLP model and the third condition is inherited from the Biba model. However, both of these are constrained by considering the PCC and OG features of the CW model. According to Rule 1, it is impossible that the subject and the object belong to the same OG but different PCC, or vice versa. Finally $V_{t+1} = (b_{t+1} \times M_{t+1} \times f_{t+1} \times g_{t+1})$, where $b_{t+1} = bt \cup (s_i, o_j, r)$, $M_{t+1} = M_t$, $f_{t+1} = f_t$, $g_{t+1} = g_t$.

Rule 5: get write-only-access: r_k = {get, s_i, o_j, a} $\in$ *RA* represents the request of the subject s_i to access object o_j with a 'write-only' privilege. If one of the following three conditions is satisfied, then d_k = yes.

1 $s_i \in s_T$

2 $X(s_i) \neq X(o_j) \wedge Y(s_i) \neq Y(o_j) \wedge K(s_i) \neq K(o_j) \wedge (a \in m_{ij})$

3 $X(s_i) = X(o_j) \wedge Y(s_i) = Y(o_j) \wedge I(s_i) = I(o_j) \wedge (a \in m_{ij})$.

Similarly, the second and the third conditions are the combination of the BLP model, Biba model, and the CW model. Finally $V_{t+1} = (b_{t+1} \times M_{t+1} \times f_{t+1} \times g_{t+1})$, where $M_{t+1} = M_t$, $f_{t+1} = f_t$, $g_{t+1} = g_t$, and $b_{t+1} = bt \cup (s_i, o_j, a)$.

Rule 6: get read-write-access: r_k = {get, s_i, o_j, w} $\in$ RA represents that the subject s_i requests to access the object o_j with a 'read-write' privilege. If one of the following three conditions is satisfied, then d_k = yes.

1 $s_i \in s_T$

2 $X(s_i) \neq X(o_j) \wedge Y(s_i) \neq Y(o_j) \wedge K(s_i) \neq K(o_j) \wedge (w \in m_{ij})$

3 $X(s_i) = X(o_j) \wedge Y(s_i) = Y(o_j) \wedge I(s_i) = I(o_j) \wedge (w \in m_{ij})$.

Comparing the Rule 4 and 5, the main difference of the condition is that the security or integrity level of the subject and object must be equal. It also complies with the state transformation rules of the BLP and Biba models. Finally $V_{t+1} = (b_{t+1} \times M_{t+1} \times f_{t+1} \times g_{t+1})$, where $b_{t+1} = bt \cup (s_i, o_j, w)$, $M_{t+1} = M_t, f_{t+1} = f_t, g_{t+1} = g_t$.

Rule 4, 5, and 6 mainly focus on getting the access to object. The opposite action is to release the access to the object that does not require any conditional check. For example, r_k = {release, s_i, o_j, r} $\in$ RA represents that the subject s_i releases the 'read-only-access' to the object o_j, and $V_{t+1} = (b_{t+1} \times M_{t+1} \times f_{t+1} \times g_{t+1})$; where $M_{t+1} = M_t, f_{t+1} = f_t, g_{t+1} = g_t$, and $b_{t+1} = bt - (s_i, o_j, r)$. The process of releasing the 'write-only-access' or 'read-write-access' is similar; therefore, the 'write-only-access' is not described here.

Rule 7: give or cancel access right: r_k = {give, s_i, o_j, A} or r_k = {cancel, s_i, o_j, A} represents that the privileged VM gives or cancels the access right of the subject s_i to the object o_j; these include 'read-only', 'write-only', and 'read-write' authorisations. The condition is according to the Rules 4, 5, and 6. By giving or cancelling the access rights, only the access matrix (M) is modified without any real access action. For example, if r_k = {give, s_i, o_j, r}, then $V_{t+1} = (b_{t+1} \times M_{t+1} \times f_{t+1} \times g_{t+1})$, where $M_{t+1} = M_t \leftarrow \{m_{ij} \leftarrow m_{ij} \cup (r)\}$, $b_{t+1} = b_t, f_{t+1} = f_t$, and $g_{t+1} = g_t$.

4 Implementation and valuation of HSMVM

In order to verify the applicability of the HSMVM, the software security module based on HSMVM (SMBH), has been developed on Xen system. SMBH is deployed on the domain0 of Xen system, which is treated as a privileged VM. SMBH is mounted on the hook interface of the Xen security modules (XSM) (Sailer et al., 2005). Thus, the functions of SMBH are called with the call to hook interface of the XSM; consequently, the access control rules of HSMVM are initialised. Figure 1 shows the complete software framework of the SMBH on Xen system.

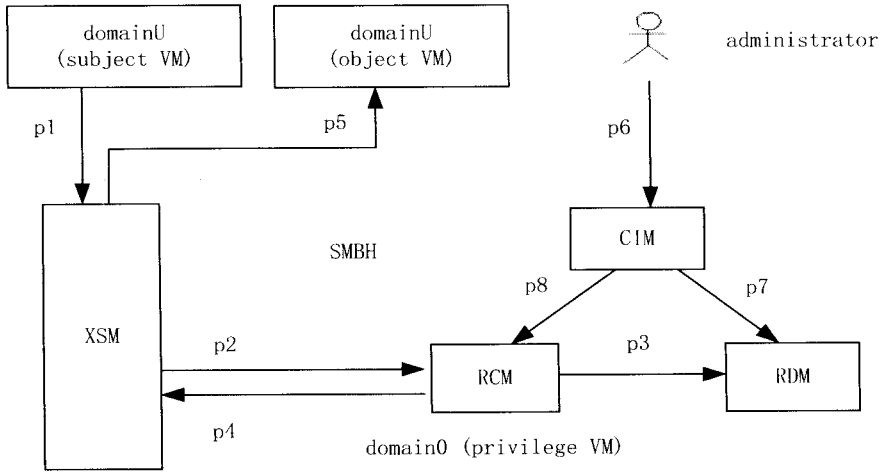
From Figure 1, it can be seen that the SMBH constitutes three sub-modules:

1 rules control module (RCM): is the core sub-module of the SMBH that is in-charge of controlling the communication between the subject VM and the object VM according to the state transformation rules described earlier

2 rules database module (RDM): is responsible for storing the related information of all the currently running VMs, such as security level, integrity level, access matrix, PCC, and OG information

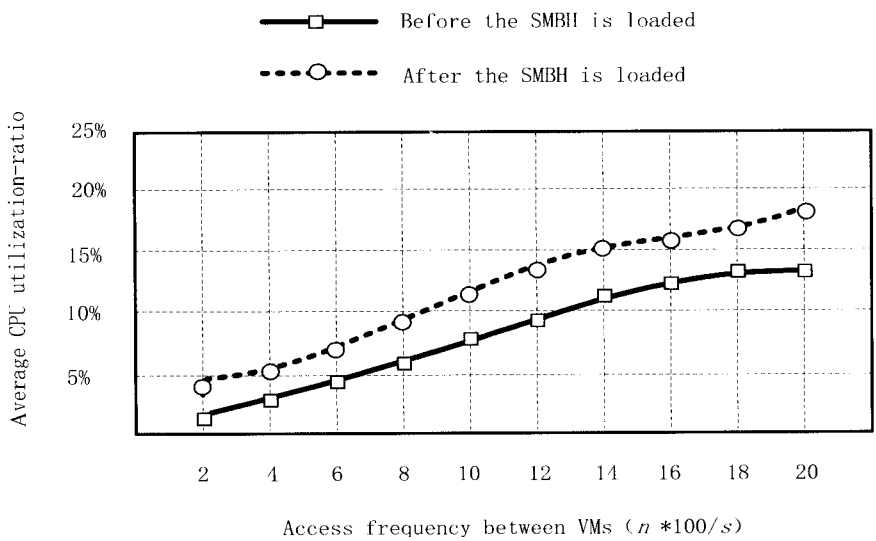
3 configuration interface module (CIM): provides the operation interface for the system administrator to modify the information stored in RDM, or to send the operation instructions to RCM, such as creating a new object VM.

Figure 1 Software framework of SMBH on Xen system (see online version for colours)



The main interactive process of related modules is described as following (corresponding sequence number is labelled in Figure 1). Firstly, when a subject VM makes a request for accessing the object VM with a specific access attribute, the XSM of domain0 will intercept the request by hook interface (p1). Thereafter, the XSM transfers this access request to RCM (p2); RCM looks up the related information from RDM (p3) and makes the decision according to the related information and rules. Finally, RCM returns the decision-making to XSM (p4). If the returned decision is 'yes', XSM authorises the subject to access the object; otherwise, XSM disapproves the access request (p5). Administrator can interact with CIM (p6), and CIM may send instructions to modify the information in RCM (p7), or to control RCM directly (p8).

Figure 2 Comparison of average CPU utilisation-ratios for SMBH



In order to evaluate the performance of SMBH, three client VMs (domainU of Xen system) and a privileged VM (domain0 of Xen system) have been created on the same physical machine (processor: Xeon E7-4820 2.0 GHz, memory: 16 G DDR). Every VM is allocated with a 4 G memory, and CentOS is installed on all of them. The communication process of VMs is simulated by reading or writing from the shared memory of VMs. The CPU average utilisation-ratios for different access frequencies between VMs are recorded. Figure 2 shows the comparison of average CPU utilisation-ratio before and after the SMBH is loaded.

From Figure 2, it can be seen that the increase in the average CPU utilisation-ratio is less than 5%, which indicates that the SMBH performs better with negligible resource utilisation.

5 Conclusions

In cloud computing environment, VMs store private data of users and provide services for users. The VMs of certain competing users can be deployed on the same physical machine that increases the risk of security breach between the VMs. A malicious software from one of the VM can attack other coexisting VMs through the inter-VM communication channels on the same physical machine. This paper presented a novel security model for the cloud computing environment, named HSMVM. The HSMVM is a hybrid security model that combines the confidentiality of BLP model, the integrity of Biba model, and the organisation isolation of the CW model. HSMVM can effectively control the access procedures between the VMs to prevent the malicious attack. Experimental results indicate that the proposed model offers improved security with negligible computational overhead. However, with the development of VM system, unidentified hidden channels between the VMs may appear. The future work involves the detection of these unknown hidden channels between the VMs under the cloud computing environment.

References

Bell, D. and Lapadula, L. (1973) *Secure Computer Systems: Mathematical Foundations*, Technical Report M74-244, The MITRE Corporation, Bedford, Massachusetts.

Biba, K. (1977) *Integrity Considerations for Secure Computer Systems*, Technical Report No. ESD-TR-76-372, Electronic Systems Division, Air Force Systems Command.

Brewer, D. and Nash, M. (1989) 'The Chinese wall security policy', *IEEE Symposium on Security and Privacy*, pp.206–214.

Chen, K., Hu, C., Zhang, X., Zheng, K., Chen, Y. and Vasilakos, A.V. (2011) 'Survey on routing in data centers: insights and future directions', *IEEE Network*, Vol. 25, No. 4, pp.6–10.

Duan, Q., Yan, Y. and Vasilakos, A.V. (2012) 'A survey on service-oriented network virtualization toward convergence of networking and cloud computing', *IEEE Transactions on Network and Service Management*, Vol. 9, No. 4, pp.373–392.

Lai, C., Chao, H., Lai, Y. and Wan J. (2013) 'Cloud-assisted real-time translating for http live streaming', *IEEE Wireless Communications*, Vol. 20, No. 3, pp.62–70.

Liu, Q., Wan, J. and Zhou, K. (2014) 'Cloud manufacturing service system for industrial-cluster-oriented application', *Journal of Internet Technology*, Vol. 15, No. 3, pp.375–382.

Liu, Q., Weng, C., Luo, Y. and Li, M. (2010) 'A mandatory access control framework in virtual machine system with respect to multi-level security I: theory', *China Communications*, Vol. 7, No. 4, pp.137–143.

Liu, Q., Weng, C., Luo, Y. and Li, M. (2011) 'A mandatory access control framework in virtual machine system with respect to multi-level security II: implementation', *China Communications*, Vol. 8, No. 2, pp.86–94.

Sailer, R., Valdez, E., Jaeger, T., Perez, R., Van Doorn, L., Griffin, J. and Berger, G. (2005) *sHype: Secure Hypervisor Approach to Trusted Virtualized Systems*, Technical Report RC23511.

Suo, H., Liu, Z., Wan, J. and Zou, K. (2013) 'Security and privacy in mobile cloud computing', *IEEE International 9th Wireless Communications and Mobile Computing Conference (IWCMC)*, pp.655–659.

Wan, J., Zhang, D., Sun, Y., Lin, K., Zou, C. and Cai, H. (2014) 'VCMIA: a novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing', *ACM/Springer Mobile Networks and Applications*, Vol. 19, No. 2, pp.153–160.

Wan, J., Zou, C., Ullah, S., Lai, C., Zhou, M. and Wang, X. (2013) 'Cloud-enabled wireless body area networks for pervasive healthcare', *IEEE Network*, Vol. 27, No. 5, pp.56–61.

Wei, J., Zhang, X., Ammons, G., Bala, V. and Ning, P. (2009) 'Managing security of virtual machine images in a cloud environment', *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, pp.91–96.

Wei, L., Zhu, H., Cao, Z., Dong, X., Jia, W., Chen, Y. and Vasilakos, A.V. (2014) 'Security and privacy for storage and computation in cloud computing', *Information Sciences*, Vol. 258, No. 1, pp.371–386.

Wei, L., Zhu, H., Cao, Z., Jia, W. and Vasilakos, A.V. (2010) 'SecCloud: bridging secure storage and computation in cloud', *ICDCS Workshops*, pp.52–61.

Wu, H., Ding, Y., Winer, C. and Yao, L. (2010) 'Network security for virtual machine in cloud computing', *IEEE 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, pp.18–21.

Xu, F., Liu, F., Jin, H. and Vasilakos, A.V. (2014) 'Managing performance overhead of virtual machines in cloud computing: a survey, state of the art, and future directions', *Proceedings of the IEEE*, Vol. 102, No. 1, pp.11–31.

Zhang, H., Li, B., Jiang, H., Liu, F., Vasilakos, A.V. and Liu, J. (2013) 'A framework for truthful online auctions in cloud computing with heterogeneous user demands', *INFOCOM*, pp.1510–1518.