Hindawi Publishing Corporation
Mobile Information Systems
Volume 2015, Article ID 841292, 9 pages
http://dx.doi.org/10.1155/2015/841292

Research Article

V-MGSM: A Multilevel and Grouping Security Virtualization Model for Mobile Internet Service

Hui Zhu, Yingfang Xue, Xiaofeng Chen, Qiang Li, and Hui Li

State Key Laboratory of Integrated Services Networks, Xidian University, No. 2 South Taibai Road, Yanta District, Xi'an 710071, China

Correspondence should be addressed to Hui Zhu; xdzhuhui@gmail.com

Received 29 August 2014; Accepted 1 September 2014

Academic Editor: David Taniar

With the pervasiveness of smart phones and the advance of Mobile Internet, more and more Mobile Internet services migrated to the cloud service platform for better user experience. As one of the most indispensable components of the cloud computing infrastructure, virtualization technology has attracted considerable interest recently. However, the flourish of virtualization still faces many challenges in information security. In this paper, we propose a novel architecture, called multilevel and grouping security model for virtualization (V-MGSM), for the security of resources in cloud computing platform. Specifically, to fulfill the balance between information sharing and privacy preservation, the virtual machines (VMs) are divided into diverse groups based on their corresponding entities, and each VM in the same group is assigned to different security level according to security requirements. Besides, the operation between VMs is based on mandatory access control mechanism. Detailed security analysis shows that the proposed V-MGSM can provide a secure communication mechanism for VMs and implement the synchronous updates of the borrowed data. Ultimately, we implement V-MGSM in Xen for experiments, and the results demonstrate that V-MGSM can indeed achieve data security and privacy protection efficiently for Mobile Internet service.

1. Introduction

In our information society, Mobile Internet which makes Internet connection accessible and ubiquitous is becoming increasingly adopted by ordinary consumers. With wide popularity and broad application, Mobile Internet service performs adding huge income to business communities and the Mobile Internet industry has recently started to take off [1–3]. By the end of March 2013, there were more than 81 million Mobile Internet users in China which generated beyond 20 billion dollars Mobile Internet market scale.

With tremendous advancement in mobile technology, people expect more services whenever and wherever possible. Although current mobile technologies and improvements allow shopping online, chatting online, or any other mobile applications through mobile terminals, several certain issues which hinder the communication process need to be addressed. For example, people tend to use mobile terminals such as smart phones to do anything realizable which causes series of problems including the following: the calculation load is magnified, the battery standby time is shortened, and the storage is limited. For the presented problems of mobile terminals, especially the limited computing ability and the limited storage discussed above, it is necessary to migrate Mobile Internet services to cloud platform to decrease the amount of computation and extend the standby time. A main advantage of cloud computing is to provide large storage and powerful computing ability by cloud server [4–6]. From this prospective, the cooperation between cloud computing and the Mobile Internet which is promising for cloud computing can transport applicable computing ability and storage from terminals to the cloud server. In short, the emergence of cloud computing is significant for the continued development of the Mobile Internet.

Figure 1 illustrates the necessity for traditional Mobile Internet services to be migrated to cloud service platform. Specifically, the larger the service scale is, the more the physical devices and instruments it requires to execute implementation are. In particular for Mobile Internet services, since it is constrained for the capacity of mobile terminals,

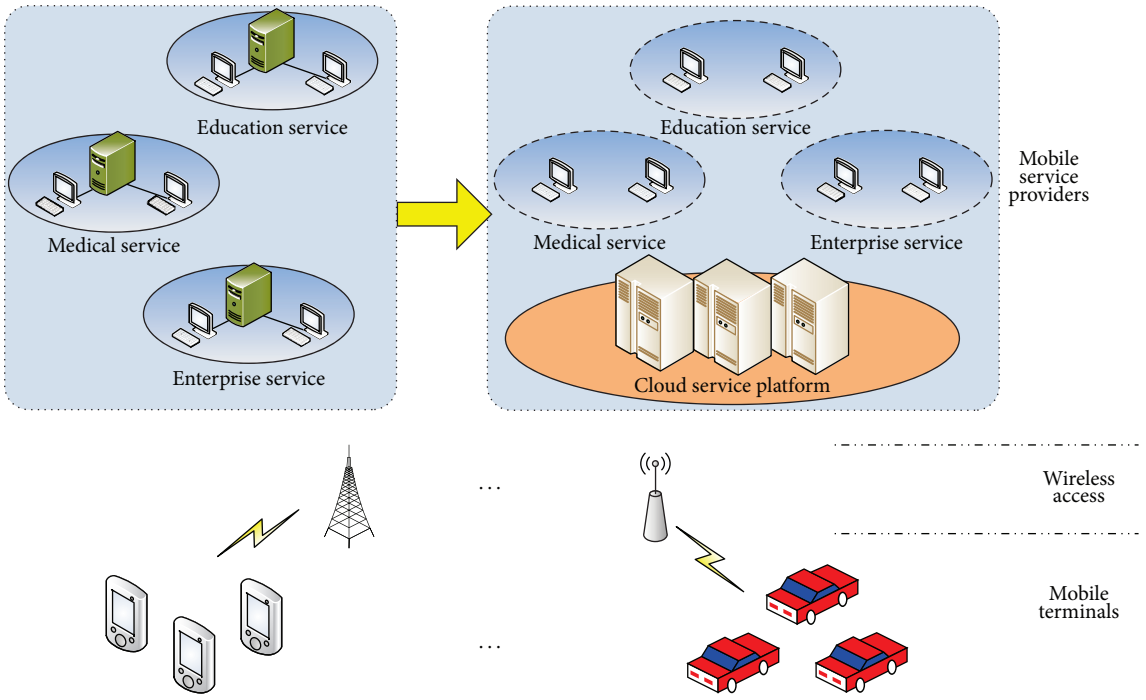

Figure 1: Cloud platform for Mobile Internet service.

more resources are required in the server of service provider [7, 8]. Benefiting from the increasing of computing ability and the decreasing of costs, virtualization technology brings many advantages over general technology in IT industry, such as raising utilization of hardware and promoting quality of service. For full use of infrastructure and reduction of the cost, more and more Mobile Internet service providers tend to migrate their services into a cloud service platform. However, the migrations of Mobile Internet service providers from conventional model to cloud services platform result in security issues which has become a critical concern. With virtualization, multitenant including medical service, education service, and enterprise service offers different services upon the same platform and may incur new vulnerabilities to the cloud platform, especially privacy violation between different service providers and internal access by VMs within one service provider. Even though virtualization security has attracted considerable interest in recent years and several secure solutions have been proposed [9, 10], the flourish of secure solutions for virtualization still faces many challenges in the balance between information sharing and privacy preservation. Security issues on virtualization technology have been the dominating barrier to the development and widespread use of Mobile Internet.

(i) A VM illegally accesses to another relatively important VM in a virtualization system without authorization, which may cause the leakage of confidential information.

(ii) To improve the efficiency of an organization, it is necessary to guarantee the information sharing among VMs which belong to the same organization.

(iii) Virtual machine monitor (VMM) has the highest priority in a virtualization system, and the virtualization system will be broken if VMM is accessed illegally.

In order to construct a secure communication mechanism for virtualization [11] and fulfill the balance between information sharing and privacy preservation, a new multilevel and grouping security access control model (V-MGSM) is proposed in this paper. And our main contributions are threefold.

(i) In V-MGSM model, to implement the efficient isolation for VMs corresponding to different organizations, all VMs have been divided into several groups, which can avoid the unnecessary access and provide the privacy preservation among different groups.

(ii) For reasonable and secure access control, different security levels are assigned to VMs in the same group. In particular, a low-level VM is not able to access other VMs which have higher security levels; one VM could not access other VMs which are divided into other groups; and VMs with high level could manage VMs with low level. Then, the confidential information in a high-level VM will not be obtained by some malicious low-level VMs.

(iii) In the same group, a high-level VM could borrow data from a low-level VM, and when the borrowed data is modified in the low-level VM, the corresponding data in the high-level VM will be updated synchronously.

The remainder of this paper is organized as follows. In Section 2, we survey the related works. In Section 3, we

present the architecture of our proposed V-MGSM model. Then, we provide the correctness and security proof of V-MGSM model in Section 4, followed by implementation in Section 5. Finally, we draw our conclusions in Section 6.

2. Related Works

2.1. Security Mechanisms for Virtualization. With the prosperity of virtualization, the security of communications between VMs has attracted considerable interest recently. The necessity of communication between VMs in the specific application scenario was firstly pointed out in [12], which mentioned that the implementation of access control mechanism for VMs' communication was also essential. Furthermore, as a mandatory access control (MAC) mechanism, sHype [13] was one of the most well-known security architectures for VMM. Based on Xen, both the communication between VMs and the hardware resource access by VMs could be controlled well by sHype, which determine the kind of VMs that should run simultaneously according to the interest conflicts. However, the type of communication between VMs was not defined in sHype. Recently, some attention has been devoted into communication types between VMs. A Prioritized Chinese Wall (PCW) policy [14] was proposed, constructing a set of VMs which could communicate with each other dynamically. A role-based access control policy [15] was proposed later, which focused on the communication between guest VMs and VMM layer. However, they all ignored inter-VMs communication. A Virt-BLP model [16] based on BLP [17] model was proposed, which met the requirements of multilevel security for virtualization. As a security communication mechanism for virtual machine system, Virt-BLP model secured the communication between VMs, while the memory taken by access control matrix was so large.

2.2. Development of Multilevel Access Control Model. At present, there are several security models and methods used in nonvirtualized system to guarantee the access control. Bell-LaPadula Model [18], SeaView model [19], and multilevel relational (MLR) [20] model have been introduced after thirty years of the research on multilevel access control [21]. Compared with other models previously proposed, MLR was clear in semantics and perfect in function after years of research on the security access control model and has been widely used in several areas and was pretty secure as a multilevel security model. However, in a multilevel user system, due to the fact that a high-level user may want to borrow data from a low-level user when it is reasonable and necessary, a high-level user can borrow data from low-level tuple in MLR and the high-level tuple which does have this borrowed data will be updated or deleted synchronously when the borrowed data has been modified by the low-level user. Meanwhile, a deadly secure risk caused by data borrowing in MLR is that the change of a high-level user's perspective by a low-level user may result in the leakage of confidential data from high-level user if a borrowed relation exists between these two users.

3. Proposed V-MGSM Model

In this section, we redesign the architecture, elements, data explanations, security theorem, and state transition rules for better use in virtualization and propose a multilevel and grouping access control model on the basis of MLR model to construct a secure communication mechanism for VMs. We first review the architecture of virtualization, which serves as the basis of V-MGSM model.

The practical architecture of virtualization is shown as in Figure 2(a). With a software layer VMM inserted between hardware and operating systems, several VMs could run on one single physical machine at the same time. In addition, for secure communication between VMs, a mandatory access control (MAC) module is constructed to control the flow of communication between VMM and VMs. However, from different VMs' perspectives, VMM is regarded as different entities. As shown in Figure 2(b), a virtualization system could be divided into several groups according to VMs' corresponding organizations such as User System 1 and User System 2, each of which consists of VM1 and VM2, VM3 and VM4, respectively. What is more, in user system's perspective, they are managed and controlled by VMM1 and VMM2 separately, although there is only one VMM in virtualization system indeed.

3.1. Elements of V-MGSM Model. Before introducing the formal definitions of the multilevel relational model and multilevel relation for virtualization, in this part, we define the basic elements of V-MGSM model as follows.

(i) Subject and Object. Subjects represent active entities such as processes and users; objects represent passive entities such as data and files. In V-MGSM model, one VM is considered as a subject or an object based on the information flow of a communication process.

(ii) Security Level. Assuming that there are two VMs in virtual machine system, if and only if the subject and the object are in the same organization and the security level of subject is higher than or equal to that of the object, then the object can be accessed by the subject.

(iii) Data Attributes. In V-MGSM model, with virtualization system grouped on the basis of entities, in the scenario of communication between VMs, a VM could be queried by another VM only when they are corresponding to the same entity.

Definition 1 (independent multilevel relational model for virtualization). Let D denote an attribute domain, let A_i denote a data attribute from domain D, let C_i denote the security level of A_i, and let RC denote the security level of a VM. The relational instances with different levels are described as $(A_1, C_1, A_2, C_2, \ldots, A_n, C_n, RC)$. The domain of security levels could be specified by the set $\{L, \ldots, H\}$, in which all of security levels of RC and A_i from the infimum L to the supremum H are involved.

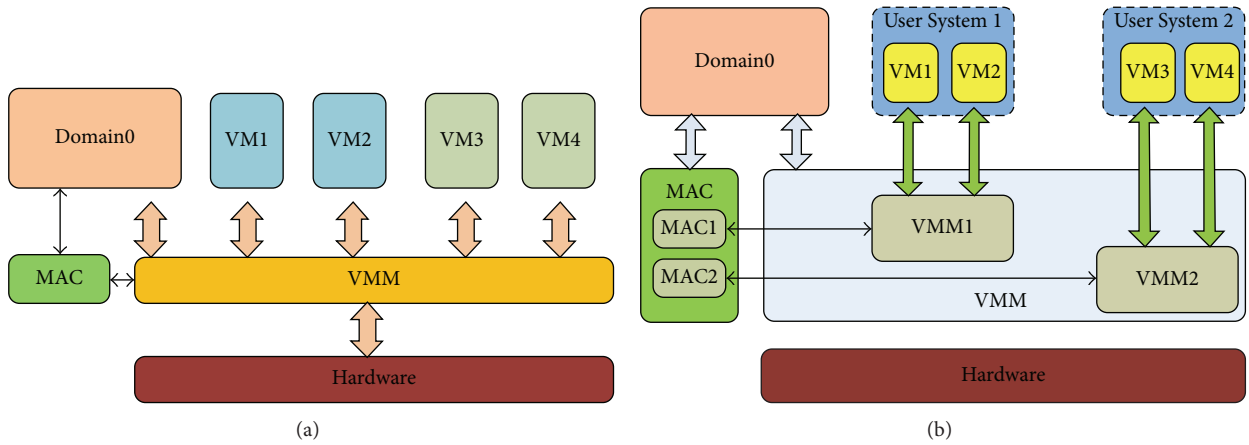

Figure 2: Architecture of virtualization: (a) piratical architecture; (b) architecture viewed by VMs.

Definition 2 (multilevel relation for virtualization). Let $r[rc]$ denote the security level of a VM, let $r[c_i]$ denote the security level of a_i, and the records of a VM in the multilevel relations are denoted by $(a_1, c_1, a_2, c_2, \ldots, a_n, c_n, rc)$. Then, all data of data attributes and security levels in multilevel relation meet the following expression:

$$a_i \in D_i, \qquad r[rc] \leq r[c_i]. \tag{1}$$

Concretely, each relational instance $(A_1, C_1, A_2, C_2, \ldots, A_n, C_n, RC)$ consists of many records $(a_1, c_1, a_2, c_2, \ldots, a_n, c_n, rc)$, which are the records of VMs. All data and security levels in the relational instance meet the following expression:

$$a_i \in D_i \vee a_i = \text{null}, \qquad c_1, rc \in \{L, \ldots, H\}, \quad rc \geq c_1. \tag{2}$$

3.2. Data Explanations of V-MGSM Model. To avoid the fuzziness in the querying result which may occur in accessing a VM in MLR, two important properties, Entity Multi-Instance and Record Multi-Instance, are introduced. And Table 1 is an example taken to describe the definition of Entity Multi-Instance and Record Multi-Instance. Two entities (*Gun, U*) and (*Gun, S*) are described in Table 1, which are created by *U*-subject and *S*-subject, respectively; *U* level is the lowest level, *U*-record is a basic record and could only access *U*-record, and it could only be deleted by *U*-subject as well; *TS* is higher than *U* in security level. Thus, both *TS*-record and *U*-record could be accessed by *TS*-subject; *TS*-subject has borrowed the value of attribute Range "2" which is authorized by *TS*-subject. However, the value of the attribute quantity in *TS*-subject is set by *TS*-subject as the value of attribute quantity in *U*-subject is not authorized by *TS*-subject. Then, two kinds of multi-instance including Record Multi-Instance and Entity Multi-Instance are mentioned in the example shown above.

Definition 3 (Record Multi-Instance). In a multilevel relation, the records which have the same value of A_1 and different values of *RC* are called Record Multi-Instance.

Table 1: Entity with multi-instance.

Name	C_1	Range	Quantity	*RC*
Gun3	*S*	1	Null	*S*
Gun3	*U*	2	Null	*TS*
Gun3	*U*	2	5000	*U*

Definition 4 (Entity Multi-Instance). In a multilevel relation, the records which have the same value of A_1 and different values of C_1 are called Entity Multi-Instance, meaning that the levels of entities are different. Then, the meaning of $r(A_1, C_1, A_2, C_2, \ldots, A_n, C_n, RC)$ can be explained as follows.

(i) Primary Key A_1 and Security Level Attribute C_1 in V-MGSM Model. Let A_1 denote the value of the primary key as well as the name of a VM, and let $r[C_1]$ denote both the identity of entity corresponding to VM and the security level of the entity in an instance r. $r[C_1] = c_1$ implies that the security level of the entity is c_1 and the entity with security level c_1 is called c_1-entity. In V-MGSM, with VMs divided into different groups according to their corresponding entities, different security levels are assigned to these entities. Given the value of C_1 at the specific moment, only one entity could be authorized by subjects who have the same value of C_1.

(ii) Security Level Attribute RC in a Record. $r[RC] = rc$ represents that record r is inserted by a VM with level *rc*, called *rc*-VM, and the data in this record is authorized by *rc*-VMs. If no corresponding record existed, it means that the entity is not authorized by *rc*-VM. The security level of a record, called *rc*, is used to judge whether the access is successful. Specifically, the data of VMs could be accessed by a subject who has a higher level than *rc*. In other words, a *rc*-VM could access other VMs r' if $r'[c_1] = r[c_1]$, $r'[RC] \leq rc$. The data in a record r could only be deleted or updated by its owner and a borrowed relation exists between two VMs r and r' if and only if they are divided into the same group and meet $r'[RC] \geq r[RC] \wedge r'[A_i, C_i] = r[A_i, C_i]$, $r \in$ relation R.

3.3. Security Theorem of V-MGSM. In this part, we first elaborate the modified Simple-security-*_property, which is selected as one of the basements for V-MGSM model and was initially proposed in BLP model. Then, four integrated properties designed for V-MGSM are introduced.

Definition 5 (Simple-security-*_property). In V-MGSM model, the security level of a subject is not dominated by the security level of an object. VMM has the highest priority in a virtualization system; VM could only read data from other VMs with lower levels in the same group and could not read data from VMM or other VMs with higher levels, eliminating the phenomenon of reading upward. Neither VMM nor other VMs with higher security levels could update or delete any data in low-level VMs, eliminating the phenomenon of writing downward. Moreover, two basic secure theorems defined in BLP model claimed that the phenomena of reading upward and writing downward should be avoided absolutely.

Definition 6 (Entity Integrated property). Entity Integrated property can be formally represented as $r[A_1], r[C_1] \neq \text{null} \wedge r[RC] \geq r[C_1]$. The modified Entity Integrated property is designed to confirm that every VM is divided into a unique group. A_1 is the value of the primary key as well as the name of a VM and C_1 is the security level of an entity corresponding to VM in an instance r.

Definition 7 (Multi-Instance Integrated property). Multi-Instance Integrated property can be formally described as $A_1, C_1, RC \rightarrow A_i, 2 \leq i \leq n$. The modified Multi-Instance Integrated property is designed for the inexistence of fuzziness in the querying process, which may be caused by accessing the object by a subject in MLR. All instances with the same value of primary keys are allowed in any relation. However, given the security level, only one entity is authorized by the subject at most. For simplicity, only one entity is authorized by VMM according to the value of C_1 when processing commands at specific moment.

To demonstrate the Multi-Instance Integrated property briefly and clearly, an example shown in Table 2 is used to describe the multi-instance conflict. Two entities with the same value of primary key "*Gun3*" and different values of C_1 are presented in this example, which is not allowed in V-MGSM model. What is more, there is only one entity authorized to avoid the fuzziness in the querying result for a given security level. Compared with MLR, the advantage of V-MGSM model is that entity with multi-instance is permitted by crossing security levels. Meanwhile, only one entity is authorized when VM accesses other VMs.

Definition 8 (Data-Borrowing Integrated property). To all records $r \in$ relation $R, r[C_i] \leq r[RC]$, for $2 \leq i \leq n$, if $r[A_i] \neq$ null, $\exists r' \in$ relation R, and $r'[C_1] = r[C_1] \wedge r'[C_i] = r[C_i]$, then $r[A_i] = r'[A_i]$. In V-MGSM, a high-level user cares more about the borrowed relation with a low-level user, rather than the detailed value of the borrowed data. The borrowed data is vindicated by the low-level VM and the security level of the borrowed data should be maintained as that of the low-level VM.

Table 2: Multi-instance integrity conflict.

Name	C_1	Range	Quantity	RC
Gun3	S	1	Null	S
Gun3	U	2	5000	S
Gun3	U	3	5000	U

(i) The Main Thought of the Improved Data Borrowing. The thought of data borrowing in MLR model is relatively unreasonable, since amount of confidential information may leak from a high-level user because a low-level user can change the view of a high-level user. Compared with MLR which is designed for nonvirtualized system, V-MGSM model is more reasonable and could be applied to a virtualization system and the information in high-level VMs will not be leaked by a low-level VM's *DELETE* or *UPDATE* operation, overcoming the disadvantage of MLR model.

(ii) Conditions for a Successful Data-Borrowing Operation. For secure inner communications between VMs within the same group, the improvement of V-MGSM mainly focuses on the aspect of data borrowing and synchronous update. In specific, we assume that the existence of the borrowed data is the basic condition for a successful data-borrowing operation by a high-level user; then, a low-level VM owns the value of the borrowed attribute; finally, the security level of borrowed data in the high-level VM is still maintained as the original security level, which ensures that the borrowed information is still vindicated by its creator. If none of the conditions is met, the borrowed data should be set null.

(iii) How to Perform Synchronous Update. The security level of the borrowed data is maintained as the original level in high-level VM to perform synchronous update of the borrowed data when it has been modified or deleted by low-level VM.

Therefore, the borrowed data from low-level VMs is set null or updated synchronously in high-level VMs when deleted or modified by the creator, avoiding losing important information.

Definition 9 (Nonborrowed Attribute Integrated property). Nonborrowed Attribute Integrated property can be formally described as $(A_i, A_j \in \text{NBA}), R[C_i] = R[C_j]$. NBA denotes nonborrowed attributes in relation R. For all data in VMs, one instance of multirelation R meets Nonborrowed Attribute Integrated property when the security levels of A_i belonging to nonborrowed attributes are the same.

3.4. State Transition Rules to V-MGSM Model. In this part, four orders (*INSERT*, *DELETE*, *UPDATE*, and *SELECT*) and their corresponding grammars are defined as the state transition rules of V-MGSM model.

Definition 10 (*INSERT* manipulation language). Grammatical form of *INSERT* manipulation language is described as follows:

$$\begin{gathered} INTERT\ INTO\ R\left[\left(A_1, \left[A_2\right], \ldots\right)\right] \\ VALUES\ \left(a_1, \left[a_2\right], \ldots\right) \\ a_1 \in D_1, a_2 \in D_2, \ldots. \end{gathered} \tag{3}$$

Semantics. R denotes the name of a relation, A_1 denotes the name of VM, and [ ] denotes the alternative items. Every *INSERT* operation could create one record with security level c into relation R by a c-subject at most. For all $2 \le i \le n$, if A_i is in the list of "into" attribute, then $r[A_i] = a_i \wedge r[C_i] = r[RC]$. Another case, $r[A_i] = \text{null} \wedge r[C_i] = r[RC] = c$, should be taken into account if the value of A_i is not inserted into record r. *INSERT* operation is permitted only when none of records $r' \in$ relation R meets $r'[A_1] = r[A_1] \wedge r'[RC] = c$. One record could be inserted successfully if the generated state of virtualization satisfies Entity Integrated (EI) property, Multi-Instance Integrated property, and Nonborrowed Attribute Integrated (NBAI) property simultaneously.

Definition 11 (*DELETE* manipulation language). Grammatical form of *DELETE* manipulation language is described as follows:

$$\begin{gathered} DELETE\ FROM\ R \\ \left[WHERE\ p\right]. \end{gathered} \tag{4}$$

Semantics. R denotes the name of a relation, p denotes a check expression which could be security attribute expression or data expression, and the data of any VM could only be deleted by its creator. *DELETE* operation is permitted if and only if $r[RC] = c$ and meets p and then deletes record r. In V-MGSM model, another situation shown as follows is taken into account because of data borrowing. For all records $r \in$ relation R, we can obtain $r'[A_i] = \text{null}$, if $\exists r' \in$ relation R, and $r'[C_1] = r[C_1] \wedge r'[RC] > r[RC] \wedge r'[A_i, C_i] = r[A_i, C_i]$.

Definition 12 (*UPDATE* manipulation language). Grammatical form of *UPDATE* manipulation can be described as follows:

$$\begin{gathered} UPDATE\ SET\ A_2 = s_2\left[, A_3 = s_3\right] \\ \left[WHERE\ p\right] \\ 2 \le i \le n. \end{gathered} \tag{5}$$

Semantics. For *UPDATE* manipulation language, the data of any VM could only be updated by its creator. In V-MGSM model, another situation, synchronous update of the borrowed information, is taken into consideration if high-level VM has borrowed data from low-level VM. The value of A_1 is not allowed to be modified in *UPDATE* operation. For all records r in relation R, if $p \wedge r[RC] = c$, then $r[A_i, C_i] = (s_i, c)$. The value of the updated attribute could only be updated by its creator, so the security level of updated attribute is maintained as that of its owner whether it has been borrowed or not. The value of updated attribute in both of the low-level VM and high-level VM is updated synchronously when borrowed relation exists between these two VMs. Therefore, if $\exists r' \in$ relation R, $r'[A_i, C_i] = r[A_i, C_i] \wedge r'[RC] > c$, and $r'[C_i] = c$, then $r'[A_i] = s_i$.

Definition 13 (*SELECT* manipulation language). Grammatical form of *SELECT* manipulation language is described as follows:

$$\begin{gathered} SELECT\ A_1\left[, A_2\right] \\ FROM\ R_1\left[, R_2\right] \\ WHERE\ p. \end{gathered} \tag{6}$$

Semantics. Let $*$ denote selecting all data from relation R, and let p denote a check expression which could be security attribute expression or data expression. Data of VMs in relation $R_1[, R_2]$ will be calculated in p, and record r will be accessed by a higher-level subject in the same group. If $r[C_1] = r'[C_1] \wedge r[RC] \le c'$, then the information of VM could be queried by c'-subject.

4. Correctness and Security Analysis

4.1. Correctness of V-MGSM Model. To prove the correctness of V-MGSM model unambiguously, two steps are necessary. One is the proof of all records r in a relation R that meet four integrated properties; the other one is the proof that any sequence of operations including *INSERT*, *DELETE*, and *UPDATE* transforms an arbitrary legal state into another legal state.

Definition 14. V-MGSM model is correctly equal to the fact that all records in a relation R in a legal state in virtualization system meet four integrated properties defined in Section 3.

Definition 15. A model is correctly equal to the fact that any legal state can be transformed into another legal state by any sequence of *INSERT*, *DELETE*, and *UPDATE* which are previously defined.

It is necessary to stress that *INSERT*, *DELETE*, and *UPDATE* operations change an arbitrary legal state in virtualization system into another legal state. Evidently, any *SELECT* operation does not change any state because no data is inserted, updated, or deleted. Then, we assume that record r has been operated in the following steps.

(i) A generated data by an *INSERT* operation does not break Entity Integrated property, Data-Borrowing Integrated property, or Nonprimary Key Integrated property according to the semantics of *INSERT* operation. Therefore, we just need to prove that an *INSERT* operation does not break Multiple Instance Integrated property. Based on the manipulation regulation of *INSERT* operation, one record r is inserted successfully only when r' which meets $r'[A_1] = r[A_1] \wedge r'[RC] = c$ does not exist in relation R. In a word,

the generated state by an *INSERT* operation does not break four previously defined integrated properties.

(ii) The reason why any *DELETE* operation does not break four integrated properties is explained in detail. Entity Integrated property, Multiple Instance Integrated property, and Nonprimary Key Integrated property are met because no record r' in the original relation R has been inserted or updated after a *DELETE* operation. For $r \in$ relation R, if $\exists r' \in$ relation R and $r'[C_1] = r[C_1] \wedge r'[RC] > r[RC] \wedge r'[A_i, C_i] = r[A_i, C_i]$, then we can obtain $r'[A_i] =$ null, and Data-Borrowing Integrated property is met mandatorily in relation R. Therefore, *DELETE* operation does not break four defined integrated properties.

(iii) Any *UPDATE* operation does not break four integrated properties. Actually, all integrated properties have been met mandatorily based on the semantics of *UPDATE* operation. Therefore, if the initial state is correct, any state transformed from any correct state by an *UPDATE* operation is correct or legal as we defined in Definition 15.

Hence, all related operations can transform a legal state into another legal state without breaking four integrated properties. Eventually, we can prove that the proposed V-MGSM model is correct.

4.2. Security Analysis. To prove the security of V-MGSM model, we define the secure state and explain the reason why the state is still secure after executing series of operations mentioned in the earlier part.

Definition 16. Secure state in V-MGSM model means that no downward information flow exists during communication process between VMs.

The output of any subject s_2 belonging to $SV(c)$ is not influenced by deleting any input of subject s_1 belonging to $SH(c)$. Let S be the set of all subjects including all VMs, and let R be all records with different levels in a virtual machine system. Then, we define $SV(c)$ as a set of all subjects whose security levels are lower than or equal to c, $RV(c)$ denote a set of all records whose levels are lower than or equal to c, and $SH(c)$, $RH(c)$: S-$SV(c)$ are $RH(c)$, R-$RV(c)$, respectively. For any access level c, we have obtained the following equations with great ease according to the meaning of each notation:

$$\begin{aligned} S &= SV(c) \cup SH(c) \wedge SV(c) \cap SH(c) = \emptyset, \\ R &= RV(c) \cup RH(c) \wedge RV(c) \cap RH(c) = \emptyset. \end{aligned} \tag{7}$$

To prove the security of V-MGSM model, we take sequence of operations including *INSERT*, *DELETE*, *SELECT*, and *UPDATE* by VMs as input in V-MGSM and output the result which includes two parts: (1) a set of records or result of failed access is returned to VM by any *SELECT* operation; (2) the successful or failed information returns to the subject after *INSERT*, *DELETE*, or *UPDATE* operation. Upon analyzing the possible access results returned to the subject, the security proof of V-MGSM consists of two situations.

(i) For any access level c, the output to any subject $s \in SV(c)$ is not influenced by changing $RH(c)$ and four cases in the following need to be taken into consideration.

(a) Due to the semantics of SELECT operation, when executing a SELECT manipulation by c'-subject and c' is less than or equal to c, no record in the set RH(c') will be returned to subject s. Therefore, the output to the subject $s \in SV(c)$ is not influenced by changing $RH(c)$ as $RH(c') \supseteq RH(c)$.

(b) The *INSERT* operated by a c'-subject is declined if record $r'' \in$ relation R, and $r''[A_1] = r[A_1] \wedge r''[RC] = c'$. Go a further step; the *INSERT* operation will also be declined whether the generated record breaks EI property, Multi-Instance Integrated property, NBAI property, or Data-Borrowing Integrated property.

(c) The *DELETE* operated by a subject s is declined only when the deleted record r is not created by the subject s, which means that the subject s does not have the right to execute any operation on this record as the subject does not have the ownership of this record.

(d) The *UPDATE* operated by a subject s is declined whether the update operation results in breaking EI property, Multi-Instance Integrated property, and NBAI property or it is not executed by its creator.

Wherein r is the record created by a c-subject and r' is the record created by a c'-subject, only records r and r' are involved and should be taken into account. The successful or failed information, which is delivered to the subject $s \in SV(c)$, is not influenced by changing $RH(c)$ because $r, r' \in RH(c') \supseteq RH(c)$. Then, we can draw a conclusion that $RV(c)$ is not changed by deleting any input of a subject $s \in SH(c)$.

(ii) The state in V-MGSM model is modified only by *INSERT*, *DELETE*, or *UPDATE* operation and we assume that c' is greater than c in following situations.

(a) c'-VM could generate a c'-record by *INSERT* operation.

(b) c'-VM could only delete c'-record.

(c) c'-VM could update c'-record.

Since c' is greater than c, $r' \notin RV(c)$.

Finally, we draw a conclusion that no downward information flow exists during communication process. The results returned to any subject s_2 belonging to set $SV(c)$ which is not influenced by deleting the input of any subject s_1 belonging to set $SH(c)$. Thus, any state of virtualization transformed from a secure initial state is maintained securely after new state transition rules are carried out.

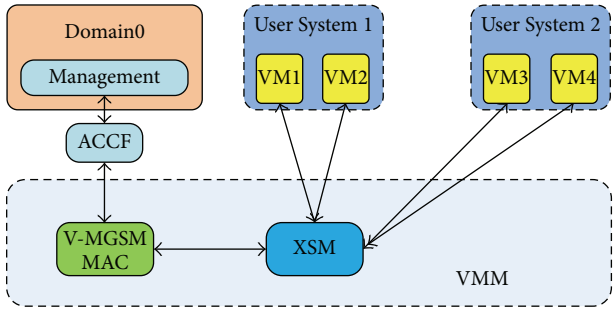

Figure 3: The implementation of MAC framework in Xen.

5. Implementation of V-MGSM Model

As shown in Figure 3, we construct a MAC framework which is composed of VMM, Management, XSM, V-MGSM MAC, ACCF, and VMs. Specifically, ACCF denotes access control configuration file, Management is used to manage the access control over other VMs in Domain0, and V-MGSM MAC is a security module which is implemented as a security hook function based on XSM, provided by Xen, and manages communications between VMs according to the state transition rules in V-MGSM model. When a VM makes a request to access another VM with access attribute *SELECT*, *UPDATE,* or *DELETE*, the XSM intercepts the request; and XSM transfers this access request to V-MGSM MAC; then, ACCF stores the access authority and relation between subjects and objects, which is used to help V-MGSM MAC to determine whether the access request is declined or not; finally, V-MGSM MAC returns the decision. If the returned decision is "Yes," then XSM authorizes the subject to access the object with the access attribute; otherwise, the access is declined.

The implementation is built in a workstation with four quad-core Intel Core i7, 2.5 GHz CPUs, 16 GB memory, and 1 T storage. In specific, the version of Xen is 3.3.1, the operating system in Domain0 is CentOS 5.4, and each VM covered in the test operates on Ubuntu 8.10. A web service is provided by Apache in Domain0, through which each VM could log in and query information of shared memory; in addition, a software used to apply to sharing memory and set the value in shared memory, called Virmem, is executed in every VM.

Step 1 (the initialization of the virtualization in Xen). Five VMs are created in Xen and denoted by VM1, VM2, VM3, VM4, and VM5, respectively, and the security levels of them are $H3, H2, H1, H3,$ and $H2$, where $H3 > H2 > H1$. For efficient privacy preservation between different organizations, VM1, VM2, and VM3 are grouped into User System 1, and VM4 and VM5 are in User System 2; the unique identifications of User System 1 and System 2 are L1 and L2, respectively. Then, the Virmem in VM3, VM4, and VM5 is applied for sharing the memory and sets the first byte of each shared memory by "*e*," "*k*," and "*m*," respectively. And VM2 is applied for sharing the memory to VM1 and sets the first byte's value by "*b*."

Table 3: The test's results of V-MGSM.

(a) The result of VM1's first query

VM name	C_1	Data (first byte)	C_2	*RC*
VM1	L1	b	H2	H3
VM2	L1	b	H2	H2
VM3	L1	e	H1	H1

(b) The result of VM4's query

VM name	C_1	Data (first byte)	C_2	*RC*
VM4	L2	k	H3	H3
VM5	L2	m	H2	H2

(c) The result of VM2's query

VM name	C_1	Data (first byte)	C_2	*RC*
VM2	L1	f	H2	H2
VM3	L1	e	H1	H1

(d) The result of VM1's second query

VM name	C_1	Data (first byte)	C_2	*RC*
VM1	L1	f	H2	H3
VM2	L1	f	H2	H2
VM3	L1	e	H1	H1

Step 2 (VM1 in User System 1 and VM4 in User System 2 query all shared memory in Domain0 by web service, resp.). Then, the results are shown in Tables 3(a) and 3(b). As VM1 owns the highest security level, all data in User System 1 but not User System 2 could be queried by VM1. And as shown in Table 3(a), the values of C_1 in these records are the same which means that all VMs are corresponding to the same group User System 1. As shown in Table 3(b), when VM4 queries all shared memory in User System 2, the value of C_1 is also the same. Therefore, the result shows that our proposed V-MGSM model is correctly and securely applied to the privacy preservation for virtualization.

Step 3 (the value of shared memory is updated by VM2). VM2 in User System 1 updates the first byte's value of shared memory to "f" by Virmem. Then, VM2 queries shared memory by web service in Domain0, and the result is shown as in Table 3(c).

Step 4 (VM1 queries shared memory again). As shown in Table 3(d), the value of data in both first and second records is modified to "f." Hence, the value of borrowed shared memory in a high-level VM could be updated synchronously and successfully if it is updated by its owner, a low-level VM.

The experimental results show that multilevel security access control for VMs and efficient isolation for VMs corresponding to different groups and synchronous update have been implemented in a virtualization system and perform a secure communication process between VMs.

6. Conclusions

In this paper, based on multilevel relation, a new multilevel and grouping security model, called V-MGSM, is proposed

for virtualization. In specific, on the basis of the security requirements for different VMs, we redesign the elements, integrated properties, and data manipulation languages of V-MGSM model, and all VMs are divided into several groups based on their corresponding entities, which can avoid the unnecessary access and provide the privacy protection among different groups. Besides, different security levels are assigned to VMs in the same group, and the confidential information in a high-level VM will not be obtained by some malicious low-level VM. Then, when the borrowed data is modified in the low-level VM, the corresponding data in the high-level VM will be updated synchronously. In addition, detailed security analysis shows that the proposed V-MGSM can provide a secure communication mechanism for VMs and achieve the synchronous updates of the borrowed data. Eventually, we implement V-MGSM in Xen for experiments, and the results demonstrate that V-MGSM can indeed achieve efficiency for Mobile Internet service.

Disclosure

The abstract of this paper has been presented in the 5th International Conference on Intelligent Networking and Collaborative Systems, pp. 9–16, 2013 [1].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (nos. 61303218 and 61272455); China 111 Project (B08038); and the Fundamental Research Funds for the Central Universities of China (no. K5051301017).

References

[1] H. Zhu, Y. Xue, Y. Zhang, X. Chen, H. Li, and X. Liu, "V-MLR: a multilevel security model for virtualization," in *Proceedings of the 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS '13)*, pp. 9–16, IEEE, September 2013.

[2] A. Flahive, D. Taniar, and W. Rahayu, "Ontology as a Service (OaaS): a case for sub-ontology merging on the cloud," *The Journal of Supercomputing*, vol. 65, no. 1, pp. 185–216, 2013.

[3] D. Taniar and S. Goel, "Concurrency control issues in Grid databases," *Future Generation Computer Systems*, vol. 23, no. 1, pp. 154–162, 2007.

[4] S. Goel, H. Sharda, and D. Taniar, "Replica synchronisation in grid databases," *InternationalJournal of Web and Grid Services*, vol. 1, no. 1, pp. 87–112, 2005.

[5] D. Taniar, C. H. Leung, W. Rahayu, and S. Goel, *High Performance Parallel Database Processing and Grid Databases*, vol. 67, John Wiley & Sons, 2008.

[6] A. Flahive, D. Taniar, and W. Rahayu, "Ontology as a Service (OaaS): extracting and replacing sub-ontologies on the cloud," *Cluster Computing*, vol. 16, no. 4, pp. 947–960, 2013.

[7] S. Shukla and R. Kumar Singh, "Security of cloud computing system using object oriented technique," in *Proceedings of the 3rd International Conference on Computing, Communication and Networking Technologies (ICCCNT '12)*, 9, p. 1, IEEE, July 2012.

[8] H. Zhu, T. Liu, G. Wei, and H. Li, "PPAS: privacy protection authentication scheme for VANET," *Cluster Computing*, vol. 16, no. 4, pp. 873–886, 2013.

[9] F. Sabahi, "Cloud computing security threats and responses," in *Proceedings of the IEEE 3rd International Conference on Communication Software and Networks (ICCSN '11)*, pp. 245–249, 2011.

[10] C. Li, A. Raghunathan, and N. K. Jha, "A trusted virtual machine in an untrusted management environment," *IEEE Transactions on Services Computing*, vol. 5, no. 4, pp. 472–483, 2012.

[11] S. M. Bellovin, "Virtual machines, virtual security?" *Communications of the ACM*, vol. 49, no. 10, p. 104, 2006.

[12] J. E. Smith and R. Nair, "The architecture of virtual machines," *Computer*, vol. 38, no. 5, pp. 32–38, 2005.

[13] R. Sailer, T. Jaeger, E. Valdez et al., "Building a MAC-based security architecture for the Xen open-source hypervisor," in *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC '05)*, pp. 276–285, December 2005.

[14] G. Cheng, H. Jin, D. Zou, A. K. Ohoussou, and F. Zhao, "A prioritized chinese wall model for managing the covert information flows in virtual machine systems," in *Proceedings of the 9th International Conference for Young Computer Scientists (ICYCS '08)*, pp. 1481–1487, November 2008.

[15] M. Hirano, T. Shinagawa, H. Eiraku et al., "Introducing role-based access control to a secure virtual machine monitor: security policy enforcement mechanism for distributed computers," in *Proceedings of the IEEE Asia-Pacific Services Computing Conference (APSCC '08)*, pp. 1225–1230, IEEE, 2008.

[16] Q. Liu, G. Wang, C. Weng, Y. Luo, and M. Li, "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, 2010.

[17] D. E. Bell and L. J. La Padula, "Secure computer system: unified exposition and multics interpretation," DTIC Document, 1976.

[18] D. E. Bell and L. J. LaPadula, "Secure computer systems: mathematical foundations," Tech. Rep., DTIC, 1973.

[19] T. F. Lunt, D. E. Denning, R. R. Schell, M. Heckman, and W. R. Shockley, "SeaView security model," *IEEE Transactions on Software Engineering*, vol. 16, no. 6, pp. 593–607, 1990.

[20] R. Sandhu and F. Chen, "The multilevel relational (MLR) data model," *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 93–132, 1998.

[21] X. Chen, J. Li, and W. Susilo, "Efficient fair conditional payments for outsourcing computations," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1687–1694, 2012.