Research Article

Wei Ma*, Huanqin Li, and Deden Witarsyah

A cloud computing separation model based on information flow

https://doi.org/10.1515/phys-2019-0013
Received Oct 28, 2018; accepted Jan 28, 2019

Abstract: Separation is the primary consideration in cloud computing security. A series of security and safety problems would arise if a separation mechanism is not deployed appropriately, thus affecting the confidence of cloud end-users. In this paper, together with characteristics of cloud computing, the separation issue in cloud computing has been analyzed from the perspective of information flow. The process of information flow in cloud computing systems is formalized to propose corresponding separation rules. These rules have been verified in this paper and it is shown that the rules conform to non-interference security, thus ensuring the security and practicability of the proposed rules.

Keywords: Information flow, separation, cloud computing, non-interference security

PACS: 07.05.-t, 07.05.Kf, 07.05

1 Introduction

Cloud computing is an emerging computing model based on separation and sharing. Sharing is one of the characteristics of cloud computing, while separation is a cornerstone of cloud computing security. If effective separation in a cloud computing environment failed to be implemented, a series of problems would arise, such as virtual machine escape, virtual machine stealing, and covert channels. Therefore, the issue of separation should be one of the primary security issues for a CSP (Cloud Service Provider). By ensuring reliable separation in a cloud computing environment, the confidence of users for the CSP can be enhanced and the acceptance of a CSP by users can be improved.

The study of separation can be discussed from two aspects: technologies and theories. For the technology part, the study of separation includes traditional network isolation, access control, encryption technology, isolation of virtualization level, as well as covert channels and side channels in cloud computing environments, etc. [4–6]. For the theory part, some studies have formalized description, analysis and verification on partial components in cloud computing. However, the cloud computing system is too large and too complex, thus making an overall formalized description and analysis difficult.

Rushby [1] and Kelem [2] have proposed proof conditions for secure separation for raw virtual machine systems. Rushby pointed out that the separation between virtual machines can be considered as "when a virtual machine performs an operation, there should be no perception in other virtual machines". Meanwhile, Kelem described the separation requirements of virtual machine systems as "an internal operation of the virtual machine will not produce systematic state variation visible to other virtual machines". However, with today's point of view, these two conditions are more like achievable results of successful separation, rather than of proof conditions for separation.

The "perception" and "influence" between virtual machines fit the semantic of information flow provided by non-interference. Non-interference theory is an important method in the study of information flow. It provides a semantic to describe information flow within a system. In addition, it can also be applied to analyze the security of information flow in a system. Since the interaction process of the system can be effectively described by information flow, information flow has been widely applied to demonstrate the security of complex systems. In information flow analysis of cloud computing, there exists a series of related work, including critical information flow based on the Xen virtualization system discussed with DFL (Data Flow Logic), the issue of "conflict-of-interests" among different tenants analyzed in [3], with the applica-

***Corresponding Author: Wei Ma:** School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou, Henan, 450045 China; Email: nsjdafi@163.com
Huanqin Li: Elementary Education Department, Zhengzhou Normal University, Zhengzhou, Henan, 450044 China;
Email: ZZSZ2005@126.com
Deden Witarsyah: School of Industrial Engineering, Telkom University, 40257, Bandung, West Java, Indonesia;
Email: dedenw@telkomuniversity.ac.id

tion of a Chinese wall model. Meanwhile, there is also literature [4] describing a non-deterministic system with non-interference theory, indicating that non-interference theory is able to provide modeling and analysis for more complex systems (such as cloud computing systems). Thus, in this study, from the perspective of information flow based on the semantic theory of the separation concept and non-interference theory, we have proposed a separation relationship and requirements for a cloud computing system from a global point of view. The main contributions of this paper are as follows: the formal description of information flow in cloud computing system is given; a separation model of cloud computing system level has been proposed based on information flow; and it is proved that the security of a cloud computing system constructed based on the above separation model can be assured.

The organization of this paper is as follows. After a general introduction to separation issues and related work in cloud computing, background of cloud computing service architecture and non-interference theory are briefly discussed in Section II. The formal description of information flow in cloud computing is discussed in Section III, including the corresponding separation rules. In Section IV, security of the separation rules is proved with non-interference theory. Section V offers additional discussion. And finally, the paper is concluded in Section VI.

2 Background

2.1 Cloud computing service architecture

A typical architecture of cloud computing services has been described (Figure 1). Based on virtualization technology, the hardware resources (including CPU, memory, network and storage, etc.) will be encapsulated by a management center within the CSP and allocated to different virtual machines. Through the cloud computing services interface, the resources will provide service to end users in the form of virtual machines. At the same time, there are corresponding regulatory organizations in the CSP, managing and controlling the users, virtual machines and virtual machine interface, as well as virtualization layer.

Three types of information flow in a cloud computing environment are presented here (Figure 1):
(i) *Information flow between virtual machines*. Since a lot of hardware resources are shared between virtual machines, there is a certain information flow between virtual machines, *i.e.*, information flow at the virtual machine level. For this part of information flow, the information is mainly

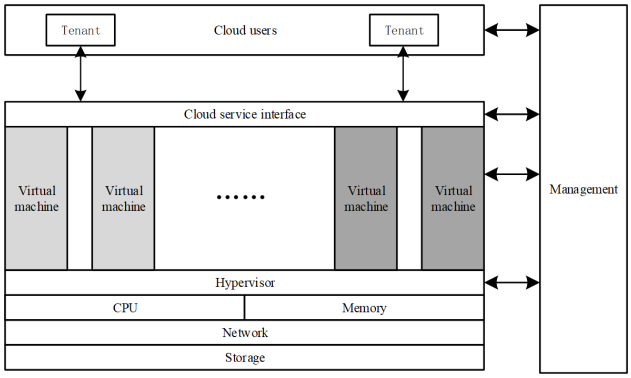

Figure 1: A typical cloud computing system

exchanged through the hypervisor; (ii) *Information flow between users and CSP*. Users send operating instructions to the infrastructure they have rented through the cloud services interface and receive corresponding feedback. Thus the information flow between user and CSP is formed, which is information flow at the cloud computing level; (iii) *Management information flow*. The management center of the cloud computing system responds to management instructions of the cloud computing system, thus generating the management information flow. This type of information flow is similar to that between virtual machines, while there are also great differences.

For the three types of information flow, different levels of separation are required. The information flow between virtual machines brings separation requirements for the virtual machine level. Separation for the virtual machine level is mainly reflected in the "interference" relationship between virtual machines, namely whether the state of a virtual machine would have observable effects on other virtual machines. For example, after the end of the life cycle of a virtual machine, the sharing resource may be released and there may be residual information in the released resources. Another example is whether the preemption of resources in one virtual machine would have effects on response speed of other virtual machines running in the same physical host. Furthermore, the information flow between users and the CSP bring separation requirements on the cloud computing level. Compared with the virtual machine level, the cloud computing level is more dynamic and there should be an independent centralized resource management mechanism. Therefore, separation on the cloud computing level is mainly reflected in the process of uniformly dynamical allocation and retrieving of resources for users, such as whether different tenants access the same resources and how communications are performed between different tenants.

2.2 Non-interference theory

Non-interference theory is a formal semantic for describing the relationship of information flow policies. It was originally applied to describe transitive information flow policies in a deterministic system, such as H $\rightarrowtail$ M and M $\rightarrowtail$ L, and then H $\rightarrowtail$ L, which makes information able to flow from H to L transitively. Later, Rushby [4] extended non-interference theory to intransitive policies, such as H $\rightarrowtail$ D, D $\rightarrowtail$ L, but H $\not\rightarrowtail$ L which indicates that the information cannot flow from H to L. In a cloud computing environment, the direct and indirect influences between components can both be regarded as intransitive information flow. For example, resources are retrieved by the management center and reassigned to a new virtual machine, and this process can be considered as information flow from original virtual machine to the management center, and then from management center to new virtual machine. Therefore, in this study, the proposed separation model has been demonstrated and validated with intransitive non-interference theory.

Rushby [4] also indicated that the "unwinding relation" in non-interference theory can not only be applied to prove the security of a system, but also be applied to construct a secure system by designing access control rules based on this theory. In recent years, Meyden [5] has pointed out and treated some deficiencies in Rushby's intransitive non-interference theory. The "unwind theorem" in non-interference theory has been further improved and applied to prove system security. In this study, the semantic of non-interference defined by Meyden has been followed to make formalized description of a cloud computing system. Here are some basic definitions from Meyden.

A deterministic state machine $M = \{S, s\_0, A, D, O, step, dom, obs\}$ is applied to describe one system in which S is the set of states; represents the initial state of the system; A is the action set of system, $a \in A$ represents that a is a single action, while $\alpha \in A^{\star}$ indicates α is a series of action sequences; D represents domains in the system; O represents the set of observed values observable to the domain; the function $step$: $S \times A \rightarrow S$ represents the system in one state performing an action and then going forward to the next state; dom: $A \rightarrow D$ represents the domain which performs a specified action; and obs: $S \times D \rightarrow O$ represents the observed value of one domain in one state. In addition, $\rightarrowtail$ is applied to represent information flow policy between domains; thus we can simply use M $(D, \rightarrowtail)$ to represent a system composed with D and $\rightarrowtail$. The theorem has been expanded with the relationship family $\sim_u$ in a single D by Meyden:

If $s\sim_u t$, then obs(s, u)=obs(t, u) (the consistency of output, referred to as OC;

If $s\sim_u t$ and $s\sim_{dom(a)} t$, then $s \bullet a \sim_{dom(a)} t \bullet a$ (weak single-step consistency, referred to as WSC);

If $dom(a) \not\rightarrowtail u$, then $s\sim_{dom(a)} t \bullet a$ (local consistency, referred to as LR).

3 Description of cloud computing separation model

3.1 Formal description of information flow in cloud computing

On the basis of the definitions from non-interference theory introduced in Section 2, some supplementary definitions are required. The resources in cloud computing systems are represented by $D = \{PM, R, T_1, T_2, \ldots, T\}$, in which PM indicates the resource address space retained by the management organizations, R represents the resource address space which can be allocated (remaining) in the system, and T_i represents the resource address space of tenant i, which is the address space of the virtual machine owned by tenant i. H is applied to represent the entire address space of cloud computing systems so clearly $H = \{PM \cup R \cup T_1 \cup \ldots \cup T_n\}$. V denotes the set of values of the address space, with A as the set of actions of the system, and S as the set of states of the system. We define the relevant functions as follows:

- The address space mapping function addr: $S \times D \rightarrow H$ represents an actual address space which a virtual machine corresponds to in a specific system state;
- The tenant resource (virtual machine) membership function host: $S \times H \rightarrow 2D$ represents the tenant resource (virtual machine) which an address space belongs to in a specific state;
- The address space value function value: $S \times H \rightarrow V$ represents the value of an address in a specific state in the system. Specifically, value(s,h)=0 means that the address h in state s is filled with 0, which is also the initialization and emptying operation of address h; when address space can be allocated, it must be initialized, namely $\forall h \in R, s \in S$, value (s, h) = 0;
- The tenant resource (virtual machine) observation function $\forall d \in D$, obs (s, d) = $\{(t, value(s, t))|t \in D\}$represents that the observable content of d in state s is a two-tuple composed of one address space and the value in this address space.

With these definitions, a series of actions can be described by the access of address space from correspond-

ing subject to object, such as the access of tenant to their leased infrastructure, management actions from management organization, as well as resources allocation and retrieving. In another words, the operation of an entire cloud computing system can be described with information flow. Follow the definition of non-interference theory, $\rightarrowtail$ is applied to represent information flow in a cloud computing system, such as $T1 \rightarrowtail T2$ indicating address space T1 of Tenant 1 flowing to address space T2 of Tenant 2, and $T1 \not\rightarrowtail T2$ indicating that the information cannot flow from T1 to T2. Hence, the cloud computing system can be expressed as M (D, $\rightarrowtail$). The information flow between virtual machines in the system, as well as the information flow from user or tenant to virtual machines can both be described with value variation within the address space of the tenant resources [6–10].

In addition, the management operations from a cloud computing management center on a cloud computing system resource constitute special information flow. Some special rules are required to describe this type of information flow. Intuitively, if a configuration change happened to a virtual machine in a cloud computing system (according to a tenant's request or administrator adjustment), it would mean the resource it processed would vary. It is apparent that the new resources could be obtained only from idle resources, or a portion of the resources would be returned to the resource pool. Therefore, the following rules can be defined to accommodate the special management information flow.

Rule 1 (**Resource management rule**)**:** In a cloud computing system $M\ (D, \rightarrowtail)$, $\forall t \in \{T_i | i = 1, 2, \ldots, n\}$, $\forall h \in H$, $\forall a \in A$, there are:

$$\mathrm{addr}(s, h)\, t \wedge \mathrm{addr}(\mathrm{step}(s, a), h) \subset t \Rightarrow \mathrm{addr}(s, h) \subset R$$

$$\mathrm{addr}(s, h) \subseteq t \wedge \mathrm{addr}(\mathrm{step}(s, a), h)\, t \Rightarrow \mathrm{addr}(\mathrm{step}(s, a), h) \subset R$$

The two rules indicate that there is no direct overlap between the resources of different tenants; the resources cannot be directly transited from one tenant to another tenant, while there will always be a process of retrieving and re-allocation. Based on the resource management rule, we can make an independent description of management information flow:

In a cloud computing system $M\ (D, \rightarrowtail)$, $\forall t \in \{T_i | i = 1, 2, \ldots, n\}$, $\forall h \in H$, $\forall a \in A$, dom(a)$\Vdash$ t is applied to indicate that dom(a) produces a management information flow on t, if one of following conditions is met:

$$t \notin \mathrm{addr}(s, h) \wedge t \in \mathrm{addr}(\mathrm{step}(s, a), h)$$

$$t \in \mathrm{addr}(s, h) \wedge t \notin \mathrm{addr}(\mathrm{step}(s, a), h)$$

Although resources between tenants cannot be directly exchanged, there is still need for communication or sharing. Therefore, an appropriate pipe should be defined to describe the information exchange between tenants. For information exchange through a pipe, the fact is that a tenant reads data from one address and then writes data to another address space. Thus, a pipe can be considered as a passage of space address for which the tenant of the sender has read permission, while the tenant of the receiver has write permission. P represents the pipe in $M\ (D, \rightarrowtail)$, the occupied address space is expressed with a pipe address function rs: P$\rightarrow 2^D$, a pipe beginning function begin: $P \rightarrow D$ and a pipe terminating function ter: $P \rightarrow D$ which are all applied to describe the address space connected by the pipe. Obviously, begin(p)≠ter(p). In this paper, for simplicity, the following rules are defined to normalize the pipes:

Rule 2 (**Pipe creation rule**)**:** In a cloud computing system $M\ (D, \rightarrowtail)$, for $\forall u, t \in \{Ti | i = 1, 2, \ldots, n\}$, and $u \neq t$:

$$u \cap t \neq \emptyset \Rightarrow \exists p \in P \wedge s(p) = u \cap t$$

$$\forall p_1, p_2 \in P, p_1 \cap p_2 = \emptyset$$

This rule describes a pipe between two tenants. Meanwhile, it also points out that there exists only zero or one pipe between different tenants, and in addition, no overlap is allowed between two pipes [11–14].

Based on the definitions of management information flow and pipe, we can make further description on $\rightarrowtail$, that is:

$$\forall u, t \in \{T_i | i = 1, 2, \ldots, n\}, \quad \text{then } ut \Leftrightarrow u \cap t \neq \emptyset \vee u \,||\models t.$$

3.2 Separation rules for cloud computing system

On the basis of the above formalized description on the information flow of a cloud computing system, we propose the required separation conditions in the cloud computing system:

Rule 3 (**Separation rule for cloud computing system**)**:**

Rule 3-1: Resource separation

$$\forall r \in R, \mathrm{value}(S_0, r) = 0$$

The first rule, Rule 3-1, indicates that the address space of different tenants should not be overlapped. The second rule indicates that allocatable resources in the system (including initial remaining resources and retrieved resources) should be in the emptied state in the resource pool. That is, a new resource obtained by tenants must be initialized, without residual data.

Rule 3-2: Pipe separation

$$\forall r \in R, s \in S, a \in A, addr(step(s, a), r) \neq addr(s, r) \text{ iff } dom(a) = PM$$

$$\forall p, begin(p) \subseteq PM \vee ter(p) \subseteq PM$$

This set of rules reflects the separation rules for creating a pipe. Specifically, the pipe must be created between management organizations and tenants. In addition, resource management can only be performed by management organizations.

Rule 3-3: Tenant separation

$$\forall t \in \{T_i | i = 1, 2, \ldots, n\}, then \exists p, begin(p) = t \vee ter(p) = t$$

Rule 3-3 indicates that each tenant must be controlled by management organizations.

Rule 3-4: Single step consistency in separation

$$\forall s_1, s_2 \in S, h \in H, a \in A, value(s_1, h) = value(s_2, h) \Rightarrow value(step(s_1, a), h) = vaule(step(s_2, a), h)$$

Rule 3-4 indicates that single step consistency should be reached in a cloud computing system. If the values of one address space are the same in two states, when the system sends the same action, the value in this address space should remain the same.

These four sets of rules indicate the separation rules which should be followed in a cloud computing environment. They are closely related to information flow in the cloud computing system, covering the information flow control between virtual machines, information flow control between tenants, as well as the separation roles played by management information flow [15–18].

4 Security analysis

Rule 3 guarantees secure separation in a cloud computing system, and also provides material for implementing separation in a cloud computing system. In this section, based on non-interference theory, the security embodied in Rule 3 will be analyzed.

Lemma 1 (**Observation of single-step consistency**)**:** If the cloud computing system M (D, $\rightarrowtail$) can satisfy the rules in Rule 3, then

$$\forall a \in A, \forall s_1, s_2 \in S, t \in \{T_i | i = 1, 2, \ldots, n\}$$
$$obs(s_1, t) = obs(s_2, t) \Rightarrow obs(step(s_1, a), t) = obs(step(s_2, a), t)$$

Proof: According to the definition of the obs function, the observed result should be a two-tuple composed of address space and value in this address space; thus, action *a* should be discussed in two cases.

CASE: *a* is a simple operation of address writing. For such an action, *a* can only change the value in the address space, while the range of address space will not be changed. Therefore, simply according to Rule 3-4,

$$vaule(s_1, h) = value(s_2, h) \Rightarrow value(step(s_1, a), h) = value(step(s_2, a), h),$$

we have $obs(step(s_1, a), t) = obs(step(s_2, a), t)$.

CASE: *a* is a management operation. For such an action, *a* can change the range of address space, while the value in the original address space will not be changed. When the action *a* is to compress the address space, obviously, the observed value within the address space will not be changed; when the action is to expand the address space, because the newly allocated address space h' is the address space after initialization, there will be

$$value(s_1, h') = value(s_2, h') = value(step(s_1, a), h') = value(step(s_2, a), h') = 0.$$

The value in the original address space will not be influenced, so that .

Lemma 1 has been proved. $\rightarrowtail$

Lemma 2 (**secure separation theorem**)**:** If the cloud computing system M (D, $\rightarrowtail$) satisfies the rules in Rule 3, then M is non-interference secure with respect to "$\rightarrowtail$ ".

Proof: To prove M is non-interference secure with respect to "$\rightarrowtail$ ", we need to respectively prove that, for "$\rightarrowtail$ ", M satisfies OC, WSC and LR according to Theorem 1.

In M (D, $\rightarrowtail$), the relation of M on D is defined $\sim t$, $s_1 \sim s_2$ if $obs(s_1, t) = obs(s_2, t)$. Obviously OC can be satisfied.

According to Lemma 1, we can see

$$\left.\begin{array}{l} s_1 \sim_t s_2 \Leftrightarrow obs(s_1, t) = obs(s_2, t) \\ obs(s_1, t) = obs(s_2, t) \Rightarrow obs(step(s_1, a), t) \\ \quad = obs(step(s_2, a), t) \end{array}\right\}$$

$$\Rightarrow step(s_1, a) \sim_t step(s_2, a)$$

That is, WSC can be satisfied;

Since dom(a) $\not\sim$ t, apparently there is no information flow from dom(a) to t. Thus, there is no pipe between dom(a) and t; according to Rule 3-2, we can see that dom(a) is not PM. Thus dom(a) can neither change the address space range of t, nor change the values in the address space of t. That is, obs(s, t)=obs(step(s, a), t), according to the definition of $\sim t$, $obs(s, t) = obs(step(s, a), t) \Leftrightarrow s \sim_t step(s, a)$. Thus, LR can be satisfied.

Lemma 2 has been proved.

Lemma 2 proves that the separation can be effectively ensured in a cloud computing system constructed based on Rule 3.

5 Discussion

Resources in a cloud computing system, such as computing resources of CPU time slice, storage resources of memory, disk and network etc., can be described with address space. The information flow in a cloud computing system can also be described by writing operations on different address spaces. Therefore, when discussing separation based on information flow, the focus should mainly be on the separation of resources, which is the separation of address space. Based on the separation rules proposed in this study, a separation model has been constructed in a cloud computing system. The role of PM has been emphasized in this model: the actions of retrieving, allocation and scheduling of any resources (address space) must be completed by PM. In this model, PM has been placed in the central position of the entire cloud computing system. As shown in Figure 2, the different tenants T1, T2, T3, communicate with PM through pipes P1, P2, P3, respectively. Through pipe P4, the appropriate resources will be scheduled (allocating or retrieving) to different tenants (dashed arrows in Figure 2) by PM.

Actually, the cloud computing system constructed based on Rule 3 is more suitable to construct a cloud computing system requiring strict separation. In addition, the separation security between this system and an existing actual system can also be analyzed to a certain extent. For example, in the architecture of Xen [19], each virtual machine communicates with VMM through an event channel, while there is no channel between virtual machines. VMM provides corresponding CPU resources for virtual machines, as well as physical memory mapping (or dynamic adjustment such as a balloon driver) and I/O operations. However, in some other actual systems, the collaborative tenants should and must perform job division and collaboration through the network. In this case, besides the rules proposed in this study, some new sharing management rules to accommodate a wider range of application scenes are required.

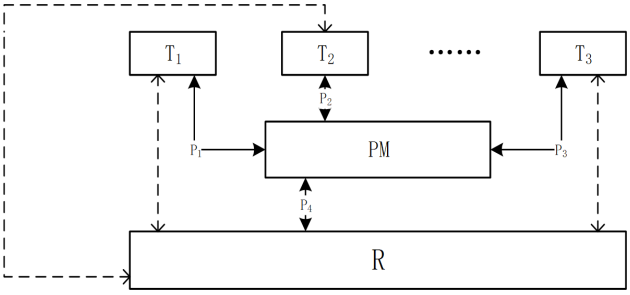

Figure 2: The cloud computing system based on Rule 3 and the allocation of resources scheduled through pipes

6 Conclusion

Security is the core issue of cloud computing, while separation is one of the most important security measures. In this paper, separation in a cloud computing system has been analyzed from the perspective of information flow, and a series of security rules have been proposed which can be proved in theory. Secure separation can be ensured in a cloud computing system constructed based on this security rule. Our study provides guidance for the construction of cloud computing systems and it also provides reference for the consistency between actual systems and strategy. Learning from the assessment and validation method in TCSEC and CC, the rules proposed in this study can also provide a method and guidance for verifying the separation in existing cloud computing systems. Therefore, it also provides references for analyzing security of existing cloud computing systems, as well as designing and constructing future trusted and secure cloud computing systems.

Acknowledgement: The research is supported by Henan Programs for Science and Technology Development (No. 182102210329).

References

[1] Rushby J.M., Proof of separability a verification technique for a class of security kernels, Int. Symp. Program.. Spring. Berlin Heidelberg, 1982, 352-357.

[2] Kelem N.L., Feiertag R.J., A separation model for virtual machine monitors, Res. Security Priv., 1991. Proc. 1991 IEEE Comput. Soc. Sym. IEEE, 1991.

[3] Wu R.Y., Information flow control in cloud computing. Collaborative computing: networking, applications and worksharing (CollaborateCom), 2010 6th Int. Conf. on. IEEE.

[4] Rushby J., Noninterference, transitivity, and channel-control security policies, SRI Int., Comput. Sci. Lab., 1992.

[5] Van D.M.R., What, indeed, is intransitive noninterference?, Computer security–ESORICS 2007, Spring. Berlin Heidelberg, 2007, 235-250.

[6] Seshadri A., Luk M., Qu N., Perrig A. SecVisor: A tiny hypervisor to provide lifetime kernel code integrity for commodity oses, ACM SIGOPS Operating Syst. Rev, 2007, 41(6), 335-350.

[7] Dai Y.H., Design and verification of a lightweight reliable virtual machine monitor for a many-core architecture. Front. Comput. Sci., 2013, 7(1), 34-43.

[8] Sierra L.A., Yepes V., Garcia-Segura T., Pellicer E., Bayesian network method for decision-making about the social sustainability of infrastructure projects, J. Clean. Prod., 2018, 176, 521-534.

[9] Hoang T.S., Abstractions of non-interference security: probabilistic versus possibilistic, Form. Aspects Comput., 2014, 26(1), 169-194.

[10] Bacon J., Information flow control for secure cloud computing, IEEE Trans, Network Serv. Manage., 2014, 11(1), 76-89.

[11] Jean B., David E., Thomas P., Jat S., Ioannis P., Peter P., Computer system evaluation criteria, IEEE Trans. Network Serv. Manage., 11(1), 76-89.

[12] Xu L.Q., Active and personalized services in an information security engineering cloud based on ISO/IEC 15408. Intelligence and security informatics, Spring. Int. Publishing, 2014, 35-48.

[13] Common criteria for information technology security evaluation (CC), Version 3.1, revision 4, September, 2012.

[14] Sharip Z., Yusoff F.M., Plankton community characteristics of natural and man-made tropical lakes, J. Environ. Biol., 2017, 38(6), 1365-1374.

[15] Costamagna A., Drigo M., Martini M., Sona B., Venturino E., A model for the operations to render epidemic-free a hog farm infected by the aujeszky disease, Appl. Math. Nonlinear Sci., 2016, 1(1), 207-228.

[16] Peng W., Ge S., Ebadi A.G., Hisoriev H., Esfahani M.J., Syngas production by catalytic co-gasification of coal-biomass blends in a circulating fluidized bed gasifier. J. Clean. Prod., 2017, 168, 1513-1517.

[17] Gao W., Wang Y., Wang W., Shi L., The first multiplication atom-bond connectivity index of molecular structures in drugs, Saudi Pharm. J., 2017, 25(4), 548-555.

[18] Dewasurendra M., Vajravelu K., On the method of inverse mapping for solutions of coupled systems of nonlinear differential equations arising in nanofluid flow, heat and mass transfer, Appl. Math. Nonlinear Sci., 2018, 3(1), 1-14.

[19] Barham P., Xen and the art of virtualization, ACM SIGOPS Operating Syst. Rev., 2003, 37(5), 164-177.