 CMC, vol.56, no.2, pp.273-283, 2018

# Research on Trust Model in Container-Based Cloud Service

**Xiaolan Xie[1, 2], Tianwei Yuan[1, *], Xiao Zhou[3] and Xiaochun Cheng[4]**

**Abstract:** Container virtual technology aims to provide program independence and resource sharing. The container enables flexible cloud service. Compared with traditional virtualization, traditional virtual machines have difficulty in resource and expense requirements. The container technology has the advantages of smaller size, faster migration, lower resource overhead, and higher utilization. Within container-based cloud environment, services can adopt multi-target nodes. This paper reports research results to improve the traditional trust model with consideration of cooperation effects. Cooperation trust means that in a container-based cloud environment, services can be divided into multiple containers for different container nodes. When multiple target nodes work for one service at the same time, these nodes are in a cooperation state. When multi-target nodes cooperate to complete the service, the target nodes evaluate each other. The calculation of cooperation trust evaluation is used to update the degree of comprehensive trust. Experimental simulation results show that the cooperation trust evaluation can help solving the trust problem in the container-based cloud environment and can improve the success rate of following cooperation.

**Keywords:** Security, cloud service, trust model, container, cooperation.

## 1 Introduction

With the rapid development of cloud computing, virtualization technology is continuously developing as a key technology in cloud computing. Virtual machine technology is a hardware virtualization technology based on a virtual machine management program. Virtual machine technology uses software to simulate a complete hardware system, implements the allocation and isolation of computing resources, and provides resource management and multi-user support for cloud computing [Borisova, Schenderlein and Shchukin (2013)]. Independence and resource contention between applications is a major problem of virtual machine technology.

Container technology is a virtualization technology. Perfectly solves the problem of program independence and resource sharing. And compared with other traditional

[1] College of Information Science and Engineering, Guilin University of Technology, Guilin, 541004, China.

[2] Guangxi Universities Key Laboratory of Embedded Technology and Intelligent Information Processing, Guilin University of Technology, Guilin, 541004, China.

[3] College of Mechanical and Control Engineering, Guilin University of Technology, Guilin, 541004, China.

[4] Department of Computer Science, Middlesex University, London, NW4 4BT, UK.

[*] Corresponding Author: Tianwei Yuan. Email: yuantianwei@glut.edu.cn.

systems, container-based cloud platform is more flexible. The implementation technologies and security mechanisms of different cloud platforms are different, undoubtedly raising higher and broader security requirements for container cloud cross-platform applications. Container virtualization technology provides a lightweight solution that allows bundled applications. This virtualization approach achieves horizontal scalability.

Compared with traditional virtualization, traditional virtual machines have difficulty in resource and expense requirements. The container technology has the advantages of smaller size, faster migration, lower resource overhead, and higher utilization. Containers running on a single machine share that machine's operating system kernel; they start instantly and use less compute and RAM. Images are constructed from file system layers and share common files. This minimizes disk usage and image downloads are much faster.

Containers isolate applications from one another and from the underlying infrastructure. It provides the most powerful default isolation. You can limit application problems to a single container instead of the entire machine.

Containers have some advantages that virtual machines cannot match, and these advantages can be used on specific occasions. For example, the annual "double eleven" Ali, Jing Dong and other *e*-commerce promotions, Spring Festival train ticket sales and so on. When the application providing these services runs in a container, the service provider can instantaneously expand the number of service units to eliminate the peak and guarantee the user experience. Among multiple container resource nodes, some resources are necessarily unreliable, which can greatly affect the execution and scheduling of jobs. There are many insecure factors in the container cloud environment. If the nodes of the network resource in the container cloud are attacked, it will directly affect the task execution on the node. Therefore, some security verification work can be ignored only when tasks and resources trust each other. Therefore, the proposal of a safety mechanism is indispensable [Fu, Liu and Chu (2016)]. The distributed dynamic trust management model is applied in a container cloud environment to ultimately implement trusted management in a trusted container cloud environment.

The general trust model consists of a central node that manages domain-wide entity trust information. Problems with the general trust model: There is no distinction between the credibility of the evaluation; lack of time applicability; single point of failure; not easy to extend, etc. The distributed trust model is based on the trust relationship established in human society [Geng, Zeng and Hu (2017)]. Network nodes independently maintain their own trust data and do not need to manage the central node. This model is applied to the container cloud environment in order to achieve the desired management goals and build a trusted container cloud environment.

## 2 Based on container-based cloud dynamic trust management model

### *2.1 Basic concept definition*

To synthesize various documents, we first give some descriptive definitions related to trust.

Definition 1. Trust is the belief in each other. It is a kind of judgment based on one's own knowledge and experience. It is a kind of subjective behavior. In this paper we define A Trust B as $T(A \rightarrow B)$.

Definition 2. Satisfaction refers to the completion of the interaction. The demand node A evaluates the service quality (service response time, service operating efficiency, completion degree, etc.) of the interaction. Calculate the degree of confidence for later calculations. The range of satisfaction is [0, 1]: 0 means very dissatisfied and 1 means very satisfied.

Definition 3. The degree of trust describes the demand node's expected judgment of the service capability of the target node. The degree of trust is only affected by the degree of satisfaction and represents the evaluation of the node's service capabilities to other nodes. The degree of trust is in the range [0, 1]: 0 indicates absolute distrust, and 1 indicates absolute trust.

Definition 4. Direct trust degree means that a node makes a unilateral trust assessment to the target node based on the historical service data that has interacted before. In this paper we define A direct trust B as $DT(A \rightarrow B)$.

Definition 5. The recommended trust degree indicates the degree of trust formed by the indirect recommendation of other nodes between nodes. In this paper we define A recommendation trust B as $RT(A \rightarrow B)$.

Definition 6. The cooperative trust degree indicates the degree of trust formed after evaluation by each cooperative node after the multi-target nodes work together. In this paper we define a cooperative trust B as $CT(A \rightarrow B)$.

Definition 7. Comprehensive trust degree is the weighted average of direct trust, recommended trust, and cooperative trust.

### *2.2 A trust management model for container-based cloud environment*

In container-based cloud environment, services can be split into multiple containers distributed over multiple node environments. Different traditional cloud computing, a service can only be communicated and deployed on one node. Container cloud is a lightweight service solution with smaller instance size, faster migration, and lower resource overhead. During service operation, task distribution can run on different nodes. The demand node selects multiple target nodes for trust calculation. In this way, it is possible to avoid the existence of malicious spoofing in the calculation of direct trust in the selection of a single target node, and to avoid co-deception of the target node. We extracted the multi-objective nodes that we worked together and evaluated each other. In the local resource store for this service, each node also evaluates other target nodes in the same team [Kale and Chirchi (2017)] as a basis for cooperative trust calculations.

In this paper, the trust management system is shown in Fig. 1.

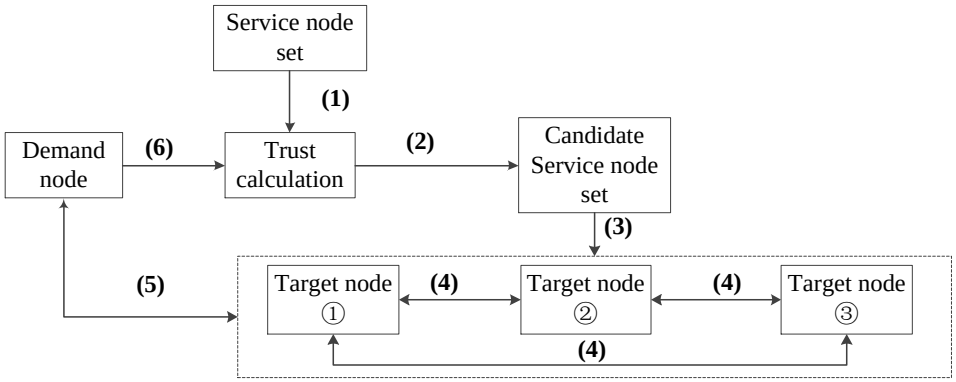


**Figure 1:** Diagram of trust management system

In this model, the demand node seeks the target service through the following steps.

(1) Trust calculation on the service node set. Calculate their direct trust, recommended trust, collaborative trust, and comprehensive trust, respectively.

(2) Filter candidate service node sets by calculation results.

(3) Select n nodes as the final interactive service node according to the requirements and comprehensive trust.

(4) Evaluation of cooperation satisfaction: $n * (n - 1)$ satisfaction evaluation is evaluated between n nodes that work collaboratively after completing a service. Finally, the evaluation is stored in their local trust store.

(5) Interaction. After the service is completed, the demand node and the target node mutually evaluate their satisfaction according to the information of the service.

(6) Demand node performs trust calculation based on interaction records.

## 3 Trust calculation

### *3.1 Local trust store*

A trust model for the construction of container-based clouds in the network. Any node in the container-based cloud is not only a service provider but also a user. The model uses a non-centered construction model [Liu, Datta and Rzadca (2013)]. In the process of calculating the degree of trust, no matter whether the direct degree of trust is calculated, whether the recommended degree of trust or the cooperative trust requires the participation of a local database, two types of data are stored in each node.

The interactive history sequence $H_{all}$, where each record H in the sequence contains target node information, interaction satisfaction data, and interaction time.

The cooperative work history sequence $C_{all}$, each record C in the sequence, contains cooperative work node information, cooperative satisfaction data, and cooperative time.

### *3.2 Direct trust calculation*

The direct trust is influenced by the local trust data, and the time factor also affects the calculation of trust.

Step 1. Read node A's local store satisfaction

Node A reads the satisfaction degree of target node B from the local storage sequence Hall, which is recorded as sequence H. The sequence H={h1, h2,...,hn}, n is the number of interactions. In sequence H, each element hi contains the time of service satisfaction $sat_i$ and interaction time.

Step 2. Calculate the decay coefficient over time of local satisfaction

Trust has timeliness, and the degree of trust will decay with time.

$$T_i = \theta(t - t_i) \tag{1}$$

$sat_i$ represents the satisfaction of the history service; $\theta(t - t_i)$ is the time influence function;t represents the current time; $t_i$ is the time when the $h_i$ was recorded.

Step3. Calculate trust in direct trust DT(A→B)

$$DT(A \to B) = \begin{cases} \sum_{i=1}^{n} \frac{T_i}{\sum_{j=1}^{n} T_i} sat_i, & n > 0 \\ 0.5 \quad , & n = 0 \end{cases} \tag{2}$$

A trust calculation between AB after an interaction is defined as $dt(A \to B) = \frac{T_i}{\sum_{j=1}^{n} T_i} sat_i$.

When $(t - t_i) \to \infty$, $T_i \to 0$. $dt(A \to B) = \frac{T_i}{\sum_{j=1}^{n} T_i} sat_i = 0$.

Indicates that the interactive information is not reliable and has no reference value. $dt(A \to B)$ has no effect on $DT(A \to B)$ . The number of interactions is reduced once. $n \to n - 1$.

When $(t - t_i) \to 0$, $T_i \to 0$. $dt(A \to B) = \frac{T_i}{\sum_{j=1}^{n} T_i} sat_i = \frac{sat_i}{\sum_{j=1}^{n} T_i}$.

Indicates that the information is reliable and has reference value.

Because $sat_i \in [0,1]$ , when $n > 0$, $sat_i = 1$, $DT_{max} = 1$. Indicate absolute trust.

When $n > 0$, $sat_i = 0$, $DT_{min} = 0$. Indicate distrust.

When $n = 0$, $DT = 0.5$. There is no history, indicating neither "trust" nor "distrust".

### *3.3 Recommended trust calculation*

When the demand node is looking for the target node, it will consider the recommendation of other nodes in addition to direct interaction. Recommended trust means that the demand node can understand the target node's credibility more comprehensively and extensively in other ways [Lang (2010)]. To prevent fraud in a single interaction, the reliability and success rate of interaction can be improved by calculating the overall trust degree. In the container-based cloud trust model, recommendation trust is calculated by iterating the recommended chain.

Step 1. Build recommended chain

In a container-based cloud environment, when an interaction occurs, a path formed from the demand node to the target node is called a recommended chain. The recommended chain requires interaction history between two adjacent nodes [Shi, Liu and Wang

(2010)]. In other words, there is a direct trust relationship between two adjacent nodes. In the process of calculating the recommended trust degree, the recommended trust level in the recommended chain will decrease as the number of layer increases. The probability of selecting the node with which the recommended chain is selected as the recommended trust is also reduced. The reason for adding recommended trust in the trust calculation is mainly to predict the distrust of the interactive node through direct trust calculation alone [Tian, Jiang, Zhi-Guo et al. (2010); Mejia, Peña and Muñoz (2011)]. Adding recommended trust can improve the overall stability of the trust model and increase the success rate of interaction.

Step 2. Calculate trust in recommended trust RT(A→B)

Average the recommended trust degree of the iterative trust values of all recommended chains. The recommended trust degree represents the degree of trust between the demand node and the target node without considering the direct interaction, or when there is no direct interaction experience between the demand node and the target node. The demand node uses the recommended trust degree as one of the criteria for selecting the target node for interaction. The definition formula is as follows:

$$RT_{ij} = \sum_{a=1}^{m} \sqrt[n]{DT_{ik_1} \cdot DT_{k_1k_2} \cdots\cdots DT_{k_nj}}/m \quad (3)$$

RT is a recommended trust evaluation value for the target node j, m denotes m recommended chains, $DT_{ik_1} \cdot DT_{k_1k_2} \cdots\cdots DT_{k_nj}$ is the recommendation credibility of a single recommendation chain. Because of the trust between two adjacent nodes, there is a direct trust relationship. $DT_{ik_1}$ shows the direct trust between the demand node and the first recommended node. $DT_{ik_1} \cdot DT_{k_1k_2} \cdots\cdots DT_{k_nj}$ is the recommended trust degree of the recommended node k for the target node j. The value of the result of direct trust is in the range of [0, 1]. Multiplied by the indirect credibility of multiple values that are less than one, the calculation results will be smaller and smaller. This also complies with the law of attenuation in the recommended chain with the increase of the number of recommended layers and the smaller the indirect credibility [Hada, Singh and Meghwal (2011); Can and Bhargava (2013)].

### *3.4 Container cloud-based cooperative trust calculation*

Cooperative trust means that in a container-based cloud environment, services can be divided into multiple containers for different container nodes, that is, there are multiple target nodes. When multiple target nodes work for one service at the same time, these nodes are in a cooperative state [Kozhirbayev and Sinnott (2017); Liu, Datta and Rzadca (2013)]. The target nodes are in the same team, and each node also evaluates other target nodes in the same team. This evaluation we call cooperative evaluation.

Step 1. Read the satisfaction of other nodes in a collaborative work

$$sat_i = \sum_{j=1}^{n} sat_j /n \quad (4)$$

Step 2. Calculate the decay coefficient over time of local satisfaction

Trust has timeliness, and the degree of trust will decay with time.

$$T_i = \theta(t - t_i) \quad (5)$$

$sat_i$ represents the satisfaction of the history service; $\theta(t-t_i)$ is the time influence function;t represents the current time; $t_i$ is the time when the $h_i$ was recorded.

Step 3. Calculate trust in direct trust CT(A→B)

$$CT(A \to B) = \begin{cases} \sum_{i=1}^{n} \frac{T_i}{\sum_{j=1}^{n} T_i} sat_i, & n > 0 \\ 0.5 \quad , & n = 0 \end{cases} \tag{6}$$

A trust calculation between AB after an interaction is defined as $ct(A \to B) = \frac{T_i}{\sum_{j=1}^{n} T_i} sat_i$.

When $(t-t_i) \to \infty$, $T_i \to 0$. $ct(A \to B) = \frac{T_i}{\sum_{j=1}^{n} T_i} sat_i = 0$.

Indicates that the interactive information is not reliable and has no reference value. $ct(A \to B)$ has no effect on $CT(A \to B)$. The number of interactions is reduced once. $n \to n-1$.

When $(t-t_i) \to 0$, $T_i \to 0$. $ct(A \to B) = \frac{T_i}{\sum_{j=1}^{n} T_i} sat_i = \frac{sat_i}{\sum_{j=1}^{n} T_i}$.

Indicates that the information is reliable and has reference value.

Because $sat_i \in [0,1]$, when $n > 0$, $sat_i = 1$, $CT_{max} = 1$. Indicate absolute trust.

When $n > 0$, $sat_i = 0$, $CT_{min} = 0$. Indicate distrust.

When $n = 0$, $CT = 0.5$. There is no history, indicating neither "trust" nor "distrust".

### *3.5 Comprehensive trust calculation*

The integrated trust of a node consists of direct trust, recommended trust, and cooperative trust. The formula is as follows:

$$T(A \to B) = \alpha DT(A \to B) + \beta RT(A \to B) + (1-\alpha-\beta)CT(A \to B) \tag{7}$$

In general cognition, people often believe in subjective experiences. However, in the real environment, other people's suggestions also play an important role. In the article, the attributes of cooperation are added as one of the judging criteria. In an objective environment, collaborating on something in a team is also an interaction. Therefore, this paper believes that when there are multiple target nodes working together, the cooperation attribute should also be used as one of the attributes for calculating comprehensive trust. The interactive nodes selected in this way have higher credibility.

## 4 Simulation experiment

This paper uses simulation experiments to verify the performance of the model, and simulates a trusted management model based on the container-based cloud. The improved model is defined as Cotrust. In the simulation, we compared it with EigenTrust mode and showed better results. The EigenTrust model is a trust model proposed by Stanford University. It is the current mainstream trust model.

### *4.1 Parameter setting*

In the simulation system, the nodes in the network are transformed into node objects. The nodes include, node ID, the historical record of the node's direct interaction, and the history of the node cooperation. The node selects the nodes with high satisfaction as the interactive node through the algorithm of simulation trust calculation.

**Table 1:** Simulation node

| Node | Service | Recommended evaluation | Cooperative evaluation | Decision result |
|---|---|---|---|---|
| Malicious A | Malicious | Malicious | Malicious | Failed |
| Malicious B | Normal | Malicious | Malicious | Success |
| Malicious C | Normal | Normal | Malicious | Success |
| Malicious D | Normal | Normal | Normal | Success |

**Table 2:** Parameter setting

| Preselected point of trust | Number of interactive history records | Time influence function | Cooperating node upper limit | Recommended chain threshold | Test times |
|---|---|---|---|---|---|
| 30 | 20 | $\frac{1}{\Delta t + 1}$ | 4 | 5 | 5 |

### *4.2 Experiment on success rate of transaction under malicious service attack*

The experiment described the effect on the success rate of transactions as the proportion of malicious services increases. In simulation experiments, it is shown that performance is similar when there are few malicious nodes, but the improved model will have better performance when there are more malicious nodes. Simulation results as shown in Fig. 2.

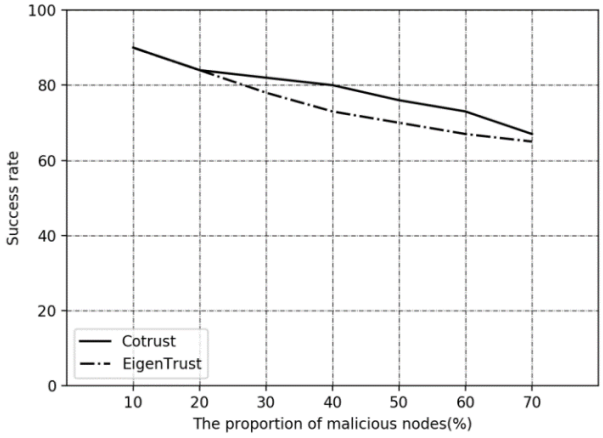


**Figure 2:** Malicious node impact diagram

### *4.3 Experiments on the impact of increasing number of interactions on transaction success*

The experiment describes the change in the success rate of the transaction as the number of interactions increases. When the number of interactions is small, the effect of the co model is not very good. However, as the number of interactions increases, the Cotrust model shows a better trend, and after a certain number of times, the area is stable and shows good feasibility. Simulation results as shown in Fig. 3.

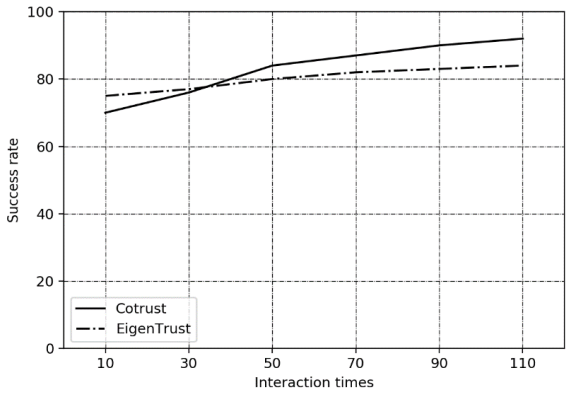


**Figure 3:** Interaction times impact diagram

## 5 Conclusion

This paper proposes a trust model for container cloud environment, which uses direct trust, recommendation trust and cooperative trust to calculate the comprehensive trust degree in three trust ways. The results of the simulation experiments show that the model can effectively solve the trusted problem in the container-based cloud.

**Acknowledgement:** This research work was supported by the National Natural Science Foundation of China (Grant No. 61762031), Guangxi Key Research and Development Plan (No. 2017AB51024), Guangxi key Laboratory of Embedded Technology and Intelligent System, Guangxi Fundamental Laboratory for Embedded Technology and Intelligent Systems.

## References

**Borisova, D.; Schenderlein, M.; Shchukin, D. G.** (2013): Nanocontainer-based anticorrosive coatings: Effect of the container size on the self-healing performance. *Advanced Functional Materials*, vol. 23, no. 30, pp. 3799-3812.

**Can, A. B.; Bhargava, B.** (2013): Sort: A self-organizing trust model for peer-to-peer systems. *IEEE Transactions on Dependable & Secure Computing*, vol. 10, no. 1, pp. 14-27.

**Cheng, X.; Jiang, Y.** (1999): Errors related to substitution and set operations. *Journal of Software*, vol. 10, no. 2, pp. 201-204.

**Cheng, X.; Jiang, Y.; Liu, X.** (1996): Fuzzy logic based on debate semantics. *Science in China*, vol. 26, no. 1, pp. 64-71.

**Cheng, X.; Liu, X.** (1995): The operator fuzzy logic based on belief considerations. *Chinese Journal of Computers*, vol. 18, no. 12, pp. 881-885.

**Cheng, X.; Zhong, S.; Dan, O.** (2004): Abstract operator logic and its natural deduction system. *Journal of Northeast Normal University (Natural Science)*, vol. 36, no. 4, pp. 38-44.

**Chillarón, M.; Vidal, V.; Segrelles, D.** (2017): Combining grid computing and Docker containers for the study and parametrization of CT image reconstruction methods. *Procedia Computer Science*, vol. 108, no. 2, pp. 1195-1204.

**Du, J.; Du, J.; Cheng, X.; Lin, K.** (2016): Degradation and encryption for outsourced png images in cloud storage. *International Journal of Grid & Utility Computing*, vol. 7, no. 1, pp. 22-28.

**Efthimiadou, E. K.; Tziveleka, L. A.; Bilalis, P.** (2012): Novel PLA modification of organic microcontainers based on ring opening polymerization: Synthesis, characterization, biocompatibility and drug loading/release properties. *International Journal of Pharmaceutics*, vol. 428, pp. 134-142.

**Fu, S.; Liu, J.; Chu, X.** (2016): Toward a standard interface for cloud providers: The container as the narrow waist. *IEEE Internet Computing*, vol. 20, no. 2, pp. 66-71.

**Geng, X.; Zeng, X.; Hu, L.** (2017): An novel architecture and interprocess communication scheme to adapt chromium based on docker container. *Procedia Computer Science*, vol. 104, no. 3, pp. 691-696.

**Goldschmidt, T.; Hauck-Stattelmann, S.; Malakuti, S.** (2018): Container-based architect-ture for flexible industrial control applications. *Journal of Systems Architecture*, vol. 84, pp. 28-36.

**Hacker, T. J.; Romero, F.; Nielsen, J. J.** (2012): Secure live migration of parallel applications using container-based virtual machines. *International Journal of Space-Based and Situated Computing*, vol. 2, no. 1, pp. 45-57.

**Hada, P. S.; Singh, R.; Meghwal, M.** (2011): Security agents: A mobile agent based trust model for cloud computing. *International Journal of Computer Applications*, vol. 36, no. 12, pp. 12-15.

**Hu, W.; Wang, H.; Min, Z.** (2014): A storage allocation algorithm for outbound containers based on the outer-inner cellular automaton. *Information Sciences*, vol. 281, pp. 147-171.

**Jiang, J.; Han, G.; Wang, F.** (2015): An efficient distributed trust model for wireless sensor networks. *IEEE Transactions on Parallel & Distributed Systems*, vol. 26, no. 5, pp. 1228-1237.

**Kale, J.; Chirchi, V. R.** (2017): Result and analysis: Data sharing between peer-to-peer using trust model. *International Journal of Computer Applications*, vol. 157, no. 8, pp. 30-33.

**Kozhirbayev, Z.; Sinnott, R. O.** (2017): A performance comparison of container-based technologies for the cloud. *Future Generation Computer Systems*, vol. 68, pp. 175-182.

**Lang, B.** (2010): A computational trust model for access control in p2p. *Science China (Information Sciences)*, vol. 53, no. 5, pp. 896-910.

**Li, X.; Du, J.** (2013): Adaptive and attribute-based trust model for service level agreement guarantee in cloud computing. *IET Information Security*, vol. 7, no. 1, pp. 39-50.

**Liu, X.; Datta, A.; Rzadca, K.** (2013): Trust beyond reputation: a computational trust model based on stereotypes. *Electronic Commerce Research & Applications*, vol. 12, no. 1, pp. 24-39.

**Liu, Z.; Huang, Y.; Li, Jin; Cheng, X; Shen, C.** (2018): DivORAM: Towards a practical oblivious RAM with variable block size. *Information Sciences*, vol. 447, pp. 1-11.

**Mei, J. P.; Yu, H.; Shen, Z.** (2017): A social influence based trust model for recommender systems. *Intelligent Data Analysis*, vol. 21, no. 2, pp. 263-277.

**Mejia, M.; Peña, N.; Muñoz, J. L.** (2011): A game theoretic trust model for on-line distributed evolution of cooperation inMANETs. *Journal of Network & Computer Applications*, vol. 34, no. 1, pp. 39-51.

**Pérez, A.; Moltó, G.; Caballer, M.** (2018): Serverless computing for container-based architectures. *Future Generation Computer Systems*, vol. 83, pp. 50-59.

**Selvaraj, C.; Anand, S.** (2012): Peer profile based trust model for p2p systems using genetic algorithm. *Peer-to-Peer Networking and Applications*, vol. 5, no. 1, pp. 92-103.

**Shchukin, D. G.** (2013): Container-based multifunctional self-healing polymer coatings. *Polymer Chemistry*, vol. 4, no.18, pp. 4871-4877.

**Shi, J.; Bochmann, G. V.; Adams, C.** (2012): A trust model with statistical foundation. *IFIP Advances in Information & Communication Technology*, vol. 173, pp. 145-158.

**Shi, Z. G.; Liu, J. W.; Wang, Z. L.** (2010): Dynamic p2p trust model based on time-window feedback mechanism. *Journal on Communications*, vol. 31, no. 2, pp. 120-129.

**Tajeddine, A.; Kayssi, A.; Chehab, A.** (2011): Fuzzy reputation-based trust model. *Applied Soft Computing*, vol. 11, no. 1, pp. 345-355.

**Tian, C. Q.; Jiang, J. H.; Hu, Z. G.; Li, F.** (2010): A novel super-peer based trust model for peer-to-peer networks: A novel super-peer based trust model for peer-to-peer networks. *Chinese Journal of Computers*, vol. 33, no. 2, pp. 345-355.

**Tilmans, S.; Russel, K.; Sklar, R.** (2015): Container-based sanitation: assessing costs and effectiveness of excreta management in Cap Haitien, Haiti. *Environ Urban*, vol. 27, no. 1, pp. 89-104.

**Wang, K.; Wu, M.** (2010): Cooperative communications based on trust model for mobile ad hoc networks. *Information Security IET*, vol. 4, no. 2, pp. 68-79.

**Wang, S. X.; Zhang, L.; Wang, S.** (2010): A cloud-based trust model for evaluating quality of web services. *Journal of Computer Science & Technology*, vol. 25, no. 6, pp. 1130-1142.

**Xie, X.; Liu, R.; Cheng, X.; Hu, X.; Ni, J.** (2016): Trust-driven and PSO-SFLA based job scheduling algorithm on cloud. *Intelligent Automation & Soft Computing*, vol. 22, no. 4, pp. 561-566.

**Zhang, J. A.; Guo, X. E.** (2010): Trust model based on dynamic recommendation in p2p network. *Computer Engineering*, vol. 36, no. 1, pp. 174-176.