Scientific Research Publishing

Engineering, 2022, 14, 479-496
https://www.scirp.org/journal/eng
ISSN Online: 1947-394X
ISSN Print: 1947-3931

Trust Assessment Model Based on a Zero Trust Strategy in a Community Cloud Environment

Rodrigue N'goran[1], Jean-Louis Tetchueng[2], Ghislain Pandry[1], Yvon Kermarrec[3], Olivier Asseu[1]

[1]Lastic, ESATIC, Abidjan, Ivory Coast
[2]University of Rennes 1, Rennes, France
[3]Lab-STICC, IMT-Atlantique, Brest, France
Email: rodrigue.ngoran@esatic.edu.ci, jean-louis.tetchuengfoping@univ-rennes1.fr, ghislain.pandry@esatic.edu.ci, yvon.kermarrec@imt-atlantique.fr, olivier.asseu@esatic.edu.ci

How to cite this paper: N'goran, R., Tetchueng, J.-L., Pandry, G., Kermarrec, Y. and Asseu, O. (2022) Trust Assessment Model Based on a Zero Trust Strategy in a Community Cloud Environment. *Engineering*, **14**, 479-496.
https://doi.org/10.4236/eng.2022.1411036

Received: October 18, 2022
Accepted: November 4, 2022
Published: November 7, 2022

Abstract

The adoption of Cloud Computing services in everyday business life has grown rapidly in recent years due to the many benefits of this paradigm. The various collaboration tools offered by Cloud Computing have eliminated or reduced the notion of distance between entities of the same company or between different organizations. This has led to an increase in the need to share resources (data and services). Community Cloud environments have thus emerged to facilitate interactions between organizations with identical needs and with specific and high security requirements. However, establishing trust and secure resource sharing relationships is a major challenge in this type of complex and heterogeneous environment. This paper proposes a trust assessment model (SeComTrust) based on the Zero Trust cybersecurity strategy. First, the paper introduces a community cloud architecture subdivided into different security domains. Second, it presents a process for selecting a trusted organization for an exchange based on direct or recommended trust value and reputation. Finally, a system for promoting or relegating organizations in the different security domains is applied. Experimental results show that our model guarantees the scalability of a community cloud with a high success rate of secure and quality resource sharing.

Keywords

Trust Management, Resources Sharing, Community Cloud, Zero Trust

1. Introduction

Cloud Computing (CC), described as the fifth utility service, provides on-demand computing resources (hardware and software) via the Internet [1]. The signifi-

cant gains in terms of financial revenues linked to the use of Cloud services and to rich and diversified service offerings have favored its adoption by companies [2]. However, organizations with high security requirements and legal considerations are reluctant to use Cloud services. This distrust of CC by these companies is due to the dependency on Cloud service providers and the security of sensitive data [3]. One solution to this problem is the use of the Community Cloud (3C) deployment model. The 3C is defined as an infrastructure shared by several organizations and supported by a specific community for the purpose of exchanging resources [4]. Each organization can offer services or make its excess or unused resources available to the community. As an example, a Community Cloud for the agricultural sector can provide relevant services with specific requirements (seed orders, crop rotation, stakeholder investments, soil management techniques, product exposure, etc.) and a required level of security (authentication, confidentiality, communication security, data protection, denial of service protection, supply chain traceability, etc.) for farmers and companies in the sector. In such an environment, managing trust between different entities is a major challenge to meet security requirements and encourage resource sharing [5]. Trust is a prerequisite for building sustainable relationships [6]. Several works related to trust management in CC have been done. L. Guo *et al.* presented in [7], a trust management model based on mutual trust with a reward and punishment mechanism. The special feature of this system is that it considers the opinions of the user and the provider by expressing mutual trust between them. InterTrust, a trust management technique based on subjective logic was introduced in [8]. It shows an improvement of the Trust Network Analysis with Subjective Logic (TNA-SL) trust management algorithm [9] in terms of the significant reduction in execution time. In addition, work has been done to ensure trust in federated cloud environments. Performance-based Risk driven Trust (PRTrust) was presented in [10]. This model allows the establishment of performance and risk-based trust for secure service sharing. It is an extension of the EigenTrust model [11] and is an effective tool for recommending services to users. A study in [12] presented TrustyFeer, a trust management system for improving service quality using subjective logic. This technique shows better results in terms of reducing non-SLA compliant services compared to TNA-SL and EigenTrust models. Most of the models mentioned above only address trust from a cloud service provider and or user perspective. Moreover, most of these trust assessment models are based on feedback from previous exchanges that may be biased by malicious entities [13] [14]. Furthermore, these assessments are made without considering the specificities of an environment such as the Community Cloud. It is important to address trust management in 3C by considering the social and community aspects on the one hand and the security threats internal and external to the system on the other. Therefore, this article proposes, a trust management model (SeComTrust) based on Zero Trust strategy principles in a community cloud. Zero Trust is an architectural concept that

aims to enhance the security of resources and services of an information system [15] [16]. Our strategy is based on the subdivision of our 3C into security zones as in [17]. These security perimeters are groupings of organizations providing resources with levels of sensitivities established based on the common vulnerability assessment system (CVSS) [18]. The contributions of our approach are as follows:

A community cloud architecture model segmented into security domains for sharing resources with well-defined levels of vulnerabilities;

- A technique for evaluating and selecting a trusted organization;
- A mechanism for updating trust values allowing the promotion or relegation of organizations in the security domains.

In the rest of this paper, Section 2 presents the model and its operation. Section 3 describes the experiments and the associated results. Finally, section 4 concludes the article and proposes perspectives for the improvement of our model.

2. Community Cloud Trust Management Model (SeComTrust)

2.1. Research Hypothesis

The SeComTrust, is based on a community cloud consisting of organizations interacting with each other for the purpose of sharing resources. Our 3C is subdivided into three security domains: the Low Security Domain (L_{sd}), the Intermediate Security Domain (M_{sd}) and the High Security Domain (H_{sd}). A security domain is a grouping of organizations that demonstrate the ability to provide resources of a given sensitivity level. Exchanges can be made between organizations of the same or different security domains. From these interactions, trust relationships can be deduced. These trust relationships are described by opinions expressing the level of trust between the organizations. An opinion is a subjective belief based on trust and allows one to express the trust value given to an organization [19] [20]. **Figure 1** below represents a trust network overlay (TON) to our community resource sharing cloud like the proposal in [21]. The vertices or nodes of this network illustrate organizations and the edges represent interactions between them. A trust relationship between two entities is represented by an arrow whose source is the requester and the tip is the resource provider. The label of an edge expresses the trust opinion of the requester towards the supplier. An organization requesting a resource will be referred to as a partner or applicant.

- C the community cloud shown in **Figure 1** below.

$$C = \{O_3, O_4, O_5, O_6, O_7, O_8, O_9\} \tag{1}$$

- $\mathcal{R}$ The set of sharing relations, such as:

$$\begin{aligned}\mathcal{R} = \{&(O_4, O_7), (O_8, O_4), (O_3, O_9), (O_9, O_6), (O_9, O_8), (O_9, O_5),\\ &(O_8, O_6), (O_8, O_9), (O_5, O_3), (O_6, O_3), (O_3, O_6)\}\end{aligned} \tag{2}$$

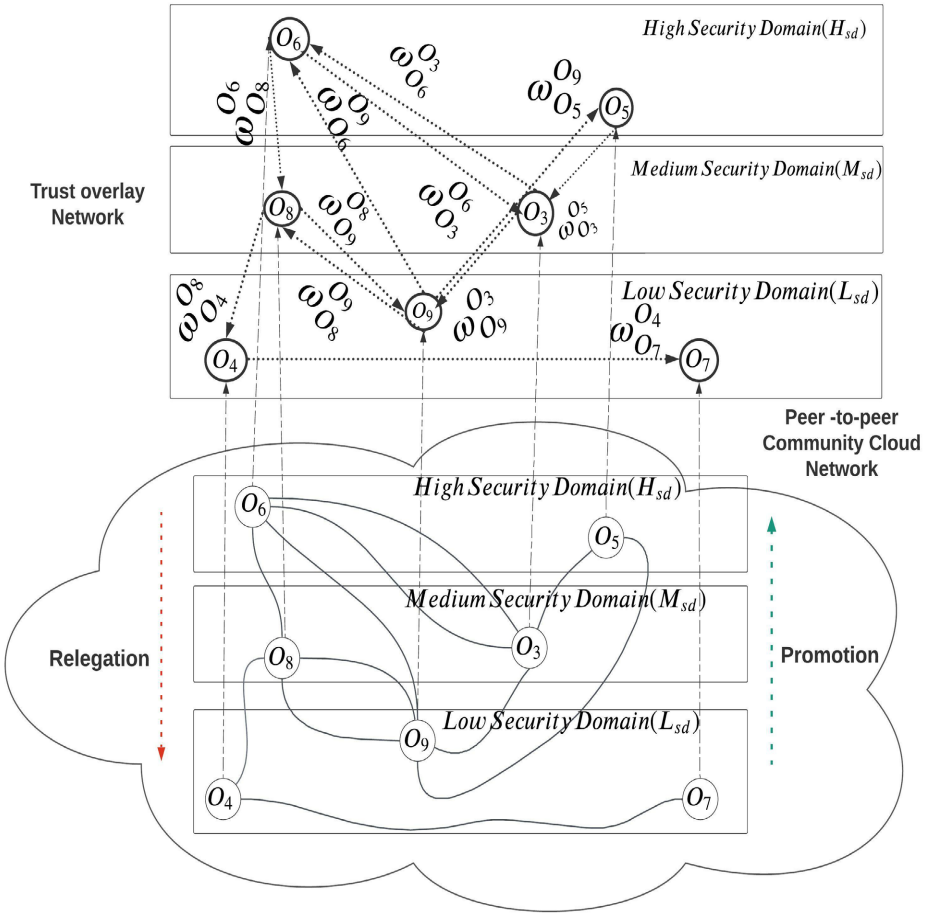


Figure 1. A trust overlay network for a community cloud computing multi-domain.

- Confidence opinions are deduced from the different interactions between organizations in Figure 1. Thus, $\mathcal{R}_O$ the set of confidence opinions is expressed as follows:

$$\mathcal{R}_O = \left\{ \omega_{O_7}^{O_4}, \omega_{O_4}^{O_8}, \omega_{O_9}^{O_3}, \omega_{O_3}^{O_5}, \omega_{O_5}^{O_9}, \omega_{O_9}^{O_8}, \omega_{O_6}^{O_9}, \omega_{O_8}^{O_6}, \omega_{O_8}^{O_9}, \omega_{O_6}^{O_3}, \omega_{O_3}^{O_6} \right\} \tag{3.1}$$

- Based on this set, the opinion matrix is obtained M_{RO}:

$$M_{RO} = \begin{array}{c} \begin{matrix} O_3 & O_4 & O_5 & O_6 & O_7 & O_8 & O_9 \end{matrix} \\ \begin{bmatrix} \omega_{O_3}^{O_3} & 0 & 0 & \omega_{O_6}^{O_3} & 0 & 0 & \omega_{O_9}^{O_3} \\ 0 & \omega_{O_4}^{O_4} & 0 & 0 & \omega_{O_7}^{O_4} & 0 & 0 \\ \omega_{O_3}^{O_5} & 0 & \omega_{O_5}^{O_5} & 0 & 0 & 0 & 0 \\ \omega_{O_3}^{O_6} & 0 & 0 & \omega_{O_6}^{O_6} & 0 & \omega_{O_8}^{O_6} & 0 \\ 0 & 0 & 0 & 0 & \omega_{O_7}^{O_7} & 0 & 0 \\ 0 & \omega_{O_4}^{O_8} & 0 & 0 & 0 & \omega_{O_8}^{O_8} & \omega_{O_9}^{O_8} \\ 0 & 0 & \omega_{O_5}^{O_9} & \omega_{O_6}^{O_9} & 0 & \omega_{O_8}^{O_9} & \omega_{O_9}^{O_9} \end{bmatrix} \end{array} \tag{3.2}$$

- The low, intermediate, and high security domains are represented by *L*, *M*, *H* respectively. The security domains are formulated below as subsets of community organizations:

$$\begin{aligned} L &= \{O_4, O_7, O_9\} \\ M &= \{O_3, O_8\} \\ H &= \{O_5, O_6\} \end{aligned} \tag{4.1}$$

- Based on the sets of security domains and the opinion matrix, the following safety domain matrix M_s is obtained:

$$M_s = \begin{array}{c} \\ M/M \\ 0 \\ H/M \\ H/M \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} \begin{array}{ccccccc} O_3 & O_4 & O_5 & O_6 & O_7 & O_8 & O_9 \end{array} \\ \begin{bmatrix} 0 & 0 & M/H & 0 & 0 & M/L \\ L/L & 0 & 0 & L/L & 0 & 0 \\ 0 & H/H & 0 & 0 & 0 & 0 \\ 0 & 0 & H/H & 0 & H/M & 0 \\ 0 & 0 & 0 & L/L & 0 & 0 \\ M/L & 0 & 0 & 0 & M/M & M/L \\ 0 & L/H & L/H & 0 & L/M & L/L \end{bmatrix} \end{array} \tag{4.2}$$

The security domain relationships in this matrix allow for thresholds of required opinion values for vendor selection. These thresholds are defined by the following governance matrices:

- T_{op}: the supplier governance matrix that expresses the minimum threshold of the supplier's opinion of an applicant (overall reputation of an applicant). $O_{L_{sd}}$, $O_{M_{sd}}$ and $O_{H_{sd}}$ represent respectively of the low, intermediate, and high security domain organizations.

$$T_{op} = \begin{array}{c} \\ O_{L_{sd}} \\ O_{M_{sd}} \\ O_{H_{sd}} \end{array} \begin{array}{c} \begin{array}{ccc} O_{L_{sd}} & O_{M_{sd}} & O_{H_{sd}} \end{array} \\ \begin{bmatrix} \lambda_{\max} & \lambda_{med} & \lambda_{\min} \\ \lambda_{\max} & \lambda_{med} & \lambda_{\min} \\ \lambda_{\max} & \lambda_{med} & \lambda_{\min} \end{bmatrix} \end{array} \tag{5.1}$$

- T_{ou}: the partner governance matrix that expresses the minimum threshold of the applicant's opinion of the supplier.

$$T_{ou} = \begin{array}{c} \\ O_{L_{sd}} \\ O_{M_{sd}} \\ O_{H_{sd}} \end{array} \begin{array}{c} \begin{array}{ccc} O_{L_{sd}} & O_{M_{sd}} & O_{H_{sd}} \end{array} \\ \begin{bmatrix} \epsilon_{\max} & \epsilon_{\text{med}} & \epsilon_{\min} \\ \epsilon_{\max} & \epsilon_{\text{med}} & \epsilon_{\min} \\ \epsilon_{\max} & \epsilon_{\text{med}} & \epsilon_{\min} \end{bmatrix} \end{array} \tag{5.2}$$

As an example, a share is allowed between a high security domain O_{pjH} provider and a low security domain O_{uiL} requester, if:

$$\omega_{O_{pjH}}^{O_{uiL}} \geq \epsilon_{\min} \text{ and } \omega_{O_{uiL}}^{O_{pjH}} \geq \lambda_{\max} \tag{5.3}$$

2.2. Workflow of SeComTrust

The SeComTrust operating process consists of first identifying and selecting the trusted provider for a given resource, then updating the trust values and transaction lists, and finally applying the promotion relegation protocol to update the security domains. The different selection phases are presented below:

- Supplier selection from the supplier transaction list: This phase consists in

first determining the suppliers of a requested resource. This process is described by (**Algorithm 1** in the **Appendices**). Once the suppliers have been identified, the ideal supplier is selected from the supplier's transaction list (TraM). This selection is done based on direct interaction (**Algorithm 2** of the **Appendices**) at first. Then, if no supplier is found during the direct selection, the choice is made based on the recommendation of friendly suppliers. The types of indirect interactions are presented in the form of Friend of a Friend (FoF) or Friend of Multiple friends (FoM) relationships [8] described in Figure 2 below. The recommendation-based selection procedure is outlined in **Algorithm 3** in the **Appendices**.

- Selection from the reputation lists: a recourse to the base of specific reputations is triggered if no supplier is found following the TraM searches. If the resource request is not satisfied, a final selection operation from the global reputation list is performed. **Algorithm 4** in the **Appendices** describes the different phases of selection through the reputation lists.

2.3. Components of SeComTrust

The architecture of our model is shown in Figure 3 below. It consists of the following components: the Resource Manager (ResM), the Transaction Manager (TraM), the Trust Value Calculator (TruC), the Update Manager (UpdM), and the Reputation Value Manager (RepM). The different components are described in the sections below.

2.3.1. The Resource Manager (ResM)

The Resource Manager consists of a registry that contains a list of organizations in the community and the resources they offer. This list is expressed in the form below:

$$L_{resm} = \left\{\left(O_{p1}, r_{p1}, g_{rp1}\right), \left(O_{p2}, r_{p2}, g_{rp2}\right), \cdots, \left(O_{pj}, r_{pj}, g_{rpj}\right)\right\} \tag{6}$$

with O_{pj} a resource provider organization r_{pj} of sensitivity degree g_{rpj}.

2.3.2. The Transaction Manager (TraM)

The transaction manager is the local repository of an organization's shares. It records and references all the shares made. As such, it is the priority consultation element in the trusted provider selection process. The information in the TraM is presented as follows:

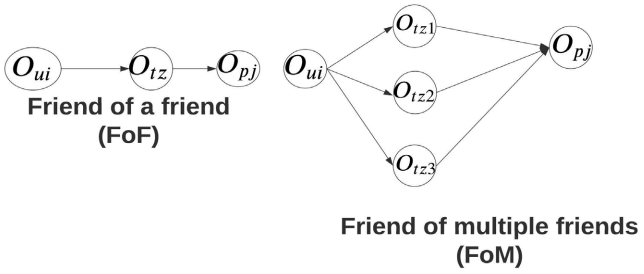
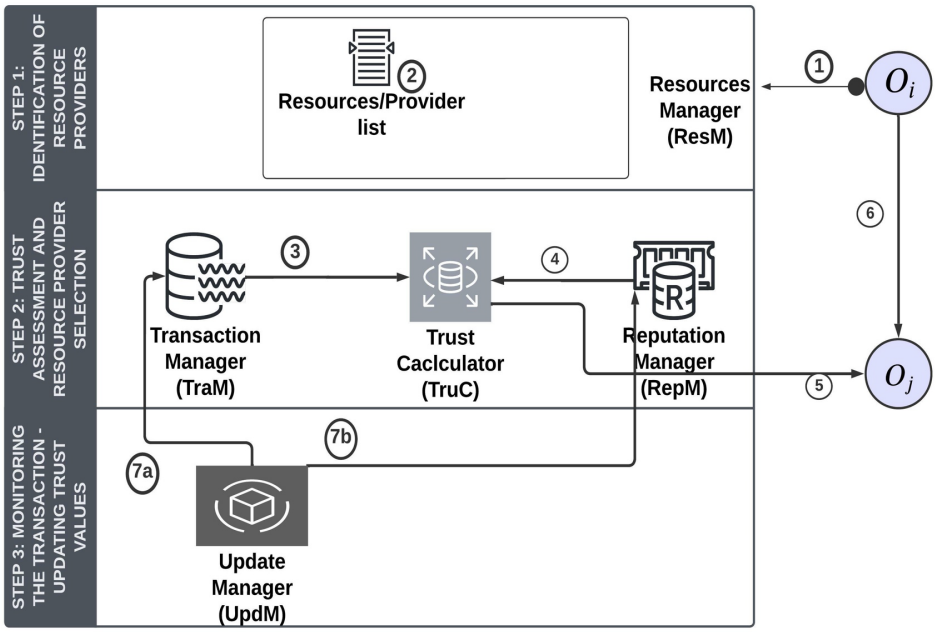


Figure 2. The FoF and FoM relation.



1. Sends information (type, quantity,...) of resource request by the organization O_i
2. Identification of the organizations providing the requested resource
3. Provision of information on the latest transactions of each provider
4. Provision of reputation information of each provider
5. Trust value calculation and provider selection O_j
6. Sharing and use of the resource
7a. Updating the transaction directory
7b. Updating the reputation manager

Figure 3. SeComTrust Architecture.

$$L_{tram} = \left\{ \left(O_{p1}, r_{p1}, g_{rp1}, \omega_{O_{p1}}^{O_{ui}}, sr_{O_{p1}}, sd_{O_{p1}} \right), \left(O_{p2}, r_{p2}, g_{rp2}, \omega_{O_{p2}}^{O_{ui}}, sr_{O_{p2}}, sd_{O_{p2}} \right), \cdots, \left(O_{pj}, r_{pj}, g_{rpj}, \omega_{O_{pj}}^{O_{ui}}, sr_{O_{pj}}, sd_{O_{pj}} \right) \right\} \tag{7}$$

with O_{pj} the resource provider, r_{pj} the resource provided, g_{rpj} the degree of sensitivity of the resource, $\omega_{O_{pj}}^{O_{ui}}$, the partner's opinion value of trust, $sr_{O_{pj}}$ the specific reputation of the provider and $sd_{O_{pj}}$ the supplier's security domain.

As in [22], the applicant's overall trust opinion of the supplier is defined as the weighted sum of the supplier's direct or recommended relationship and the supplier's reputation:

$$\omega_{O_{pj}}^{O_{ui}}(r,g) = \beta DRT_{O_{pj}}^{O_{ui}}(r,g) + (1-\beta) sr_{O_{pj}}(r,g) \tag{8}$$

with $sr_{O_{pj}}(r,g)$ specific reputation, $DRT_{O_{pj}}^{O_{ui}}(r,g)$ is the trust opinion based on direct or indirect interactions between the supplier and the partner. This trust opinion value is calculated using subjective logic (SL) and its four basic parameters namely: belief (b), disbelief (d), uncertainty (u) and base rate (a) [23] [24]. The trust opinion of the O_{ui} organization towards the O_{pj} organization for direct

interaction is formulated as follows:

$$DRT_{O_{pj}}^{O_{ui}}(r,g) = b + (\alpha * u) \text{ with } b,d,u \in [0,1] \text{ and } b+d+u=1 \tag{9}$$

with

$$\begin{cases} b = \dfrac{p_t}{p_t + n_t + 2} \\ d = \dfrac{n_t}{p_t + n_t + 2} \\ u = \dfrac{2}{p_t + n_t + 2} \end{cases} \Leftrightarrow \begin{cases} p_t = \dfrac{2b}{u} \\ n_t = \dfrac{2d}{u} \end{cases} \tag{10}$$

p_t the number of previous positive trades between O_{ui} and O_{pj}, and n_t the number of negative trades. The base rate α is paramount for new or inactive community members.

$$\alpha = 0.5 \tag{11}$$

For indirect interactions, several operators are defined by the SL to determine the trust values [23]. For two organizations O_{ui} and O_{pj} without prior direct interactions (Equation (12) below). The trust derived between O_{ui} and O_{pj} is called transitive trust and will be computed as the opinion $\omega_{O_{pj}}^{O_{ui}:O_{tz}}$ using the discounting operator ($\otimes$):

$$\omega_{O_{pj}}^{O_{ui}:O_{tz}} = \omega_{O_{tzj}}^{O_{ui}} \otimes \omega_{O_{pj}}^{O_{tz}} \begin{cases} b_{O_{pj}}^{O_{ui}:O_{tz}} = b_{O_{tz}}^{O_{ui}} b_{O_{pj}}^{O_{tz}} \\ d_{O_{pj}}^{O_{ui}:O_{tz}} = b_{O_{tz}}^{O_{ui}} d_{O_{pj}}^{O_{tz}} \\ u_{O_{pj}}^{O_{ui}:O_{tz}} = d_{O_{tz}}^{O_{ui}} + u_{O_{tj}}^{O_{ui}} + b_{O_{tz}}^{O_{ui}} u_{O_{pj}}^{O_{tz}} \\ \alpha_{O_{pj}}^{O_{ui}:O_{tz}} = \alpha_{O_{pj}}^{O_{tz}} \end{cases} \tag{12}$$

On the other hand, if there are two intermediate organizations O_{tz1} and O_{tz2}, such that O_{pj} has already interacted with O_{tz1} and O_{tz2}, and there is no previous interaction between O_{ui} and O_{pj} (Equation (13) below). The confidence derived between O_{ui} and O_{pj} is parallel confidence and is represented as the opinion $\omega_{O_{pj}}^{O_{tz1}:O_{tz2}}$. It is expressed below using SL and its consensus operator ($\oplus$) [23]:

$$\omega_{O_{pj}}^{O_{tz1}:O_{tz2}} = \omega_{O_{pj}}^{O_{tz1}} \oplus \omega_{O_{pj}}^{O_{tz2}} \begin{cases} b_{O_{pj}}^{O_{tz1}:O_{tz2}} = \dfrac{b_{O_{pj}}^{O_{tz1}} u_{O_{pj}}^{O_{tz2}} + b_{O_{pj}}^{O_{tz2}} u_{O_{pj}}^{O_{tz1}}}{u_{O_{pj}}^{O_{tz1}} + u_{O_{pj}}^{O_{tz2}} - u_{O_{pj}}^{O_{tz1}} u_{O_{pj}}^{O_{tz2}}} \\ d_{O_{pj}}^{O_{tz1}:O_{tz2}} = \dfrac{d_{O_{pj}}^{O_{tz1}} u_{O_{pj}}^{O_{tz2}} + d_{O_{pj}}^{O_{tz2}} u_{O_{pj}}^{O_{tz1}}}{u_{O_{pj}}^{O_{tz1}} + u_{O_{pj}}^{O_{tz2}} - u_{O_{pj}}^{O_{tz1}} u_{O_{pj}}^{O_{tz2}}} \\ u_{O_{pj}}^{O_{tz1}:O_{tz2}} = \dfrac{u_{O_{pj}}^{O_{tz1}} u_{O_{pj}}^{O_{tz2}}}{u_{O_{pj}}^{O_{tz1}} + u_{O_{pj}}^{O_{tz2}} - u_{O_{pj}}^{O_{tz1}} u_{O_{pj}}^{O_{tz2}}} \\ \alpha_{O_{pj}}^{O_{tz1}:O_{tz2}} = \alpha_{O_{pj}}^{O_{tz1}} \end{cases} \tag{13}$$

2.3.3. The Confidence Value Calculator (TruC)

The TruC executes the various algorithms for calculating and selecting confidence values.

2.3.4. The Update Manager (UpdM)

The role of the update manager is to update the trust information. The updated values are the specific reputation of the supplier, the global reputation of the partner and the supplier. In addition, the security domains are also updated through the promotion and relegation mechanism.

2.3.5. The Reputation Manager (RepM)

The RepM is the register of reputations of organizations in the community. We distinguish between two types of reputation: the reputation of a specific provider of a given resource and the overall reputation of an organization derived from its general behavior in the community. The information in this register is formulated as follows:

$$L_{repm} = \left\{ \left(O_{p1}, r_{p1}, g_{rp1}, sr_{O_{p1}}, gr_{O_{p1}}, sd_{O_{p1}} \right), \left(O_{p2}, r_{p2}, g_{rp2}, sr_{O_{p2}}, gr_{O_{p2}}, sd_{O_{p2}} \right), \cdots, \left(O_{pj}, r_{pj}, g_{rpj}, sr_{O_{pj}}, gr_{O_{pj}}, sd_{O_{pj}} \right) \right\} \tag{14}$$

with O_{pj} the resource provider, r_{pj} the resource provided, g_{rpj} the degree of sensitivity of the resource provided, $sr_{O_{pj}}$ the reputation of the organization as a provider of r_{pj} of degree g_{rpj}, $gr_{O_{pj}}$ the overall reputation value, and $sd_{O_{pj}}$ the organization's security domain.

2.4. Updating System Trust Values

2.4.1. Updating the Specific Reputation

The specific reputation of a supplier is updated after each transaction (**Figure 4**). To encourage the sharing of secure resources, a weight is assigned to each exchange according to the resource's sensitivity level. The sensitivity level describes the degree of vulnerability of a resource. The g_{rpj} degrees of resource sensitivities are defined from the common computer resource vulnerability system CVSS v2.0 score range [18].

$$\begin{cases} g_{rpj} \in [7,10], \gamma_i = 0.2 \\ g_{rpj} \in [4,7[, \gamma_i = 0.35 \\ g_{rpj} \in [0,4[, \gamma_i = 0.45 \end{cases} \tag{15}$$

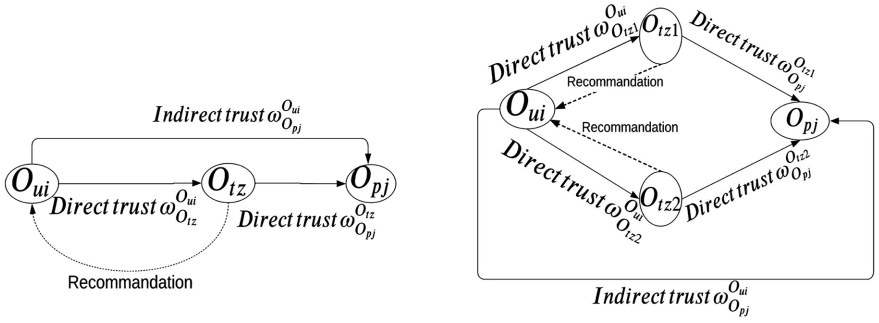


Figure 4. Transitive and parallel combinaison trust.

Like the contribution in [10], the specific reputation is expressed as follows:

$$sr_{O_{pj}} = \begin{cases} sr_{O_{pj}}^{c} + \Delta_i \ \text{ if positive result} \\ sr_{O_{pj}}^{c} + \Delta_i / 2 \ \text{ if positive result with violation} \\ sr_{O_{pj}}^{c} - \Delta_i \ \text{ if positive result} \\ \text{with } \Delta_i = \gamma_i \delta_i \text{ and } \delta_i(j) = \dfrac{\left(I_{l\min}(j) + I_{l\max}(j)\right)/2}{I_{l\max}} \end{cases} \tag{16}$$

$I_{l\min}(j)$ the minimum value of assurance level of security domain j, $I_{l\max}(j)$ the maximum value of assurance level j, $I_{l\max}$ the maximum value of the assurance level, γ_i is the weight of an exchange.

The assurance level I_l is the ability of an organization to provide a resource of a given sensitivity level. Organizations are grouped into security domains based on their assurance level.

$$\begin{aligned} O_{pj} &\in H_{SD} \ \text{ if } I_l \in [7,10] \\ O_{pj} &\in M_{SD} \ \text{ if } I_l \in [4,7[\\ O_{pj} &\in L_{SD} \ \text{ if } I_l \in [0,4[\end{aligned} \tag{17}$$

2.4.2. Updating the Global Reputation

Global reputation is based on the overall results of an organization's interactions as a provider within the community. It is formulated as follows:

$$gr_{O_{pj}} = \alpha + \rho \sum_{k=1}^{n} sr_{O_{pj}}(k), \ \rho = \frac{ST_{O_{pj}}}{ST_C} \tag{18}$$

with α the prime rate (α = 0.5), n the number of organization-specific reputations O_{pj}, $ST_{O_{pj}}$ the total number of exchanges for a given resource provider, ST_C the total number of shares within the community, and ρ the weight of the organization's exchanges. After each update operation, the organization is promoted, relegated, or retained in a security domain.

3. Experiments and Results

3.1. Experimentation Environment

This article proposes a community cloud experimentation environment established in two phases. During the first phase, the 3C architecture is initialized. This involves generating dataset files describing organizations, provided resources, and resource requirements queries. Then, in the second phase, statistical data are produced through simulations of resource sharing between organizations. The information deduced from these experiments is used to evaluate the performance of our trust model. The experiments are conducted on a MacBook Pro (Retina, 15-inch, mid-2015), 2.2 GHz Intel Core i7 quad-core processor, 16 GB 1600 MHz DDR3 memory. Programming is performed in a Pycharm development environment (IDE) and the Python language Python 3.9. We distinguish two types of resource provider organizations. On the one hand, organizations

provide resources in accordance with service level agreements (SLA) established between the actors involved in a transaction. These are referred to as good providers or G organizations. On the other hand, those providing unreliable resources. They are said to be malicious or M organizations. The number of organizations being an essential characteristic in the setting up of a perennial and prosperous community, the experiments are carried out on groups of organizations of the beach [80,250] members. The rate of malicious organizations is 20%. Rounds consisting of 500 resource requests are performed in each experimentation set. The experimentation parameters are summarized in Table 1 below.

The main metrics used to measure the performance of our model are:

$$\text{SRTG} = \frac{\text{number of resources provides by organizations G}}{\text{total number of resources provided by organizations G}} \tag{19}$$

- SRTG: Transaction success rate of G organizations
- RPOT: Rate of participation of organizations in transactions. This rate measures the number of different organizations involved in successful exchanges.

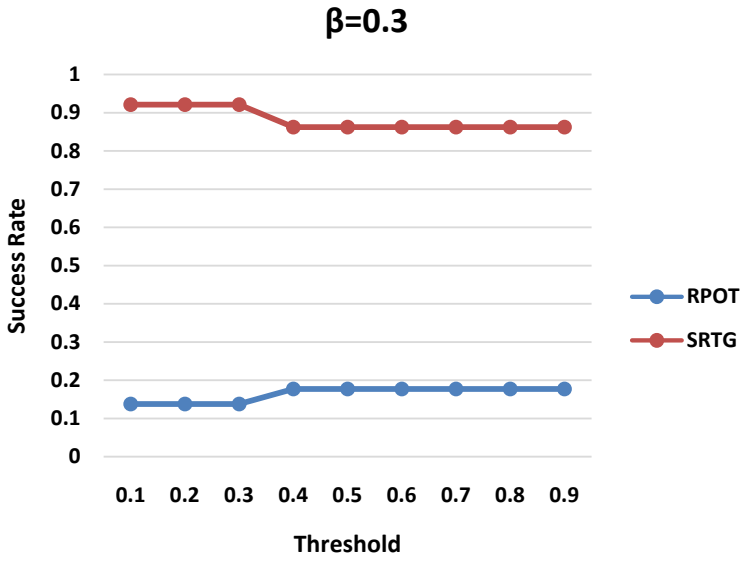
3.2. Selection Threshold and Parameter β Value

The selection confidence value of a vendor from the SeComTrust is calculated through Equation (8). To determine the value of β, representing the weight of the direct or recommended confidence value (DRT) in this equation, we examine the rate of participation of different organizations in transactions (RPOT) and the success rate of G providers (SRTG). RPOT measures the number of providers actively participating in transactions, limiting the possibility of selecting the same organizations repeatedly to increase the SRTG. The GTRS is an important performance indicator for trust models. It expresses the ability of the trust model to resist malicious attacks.

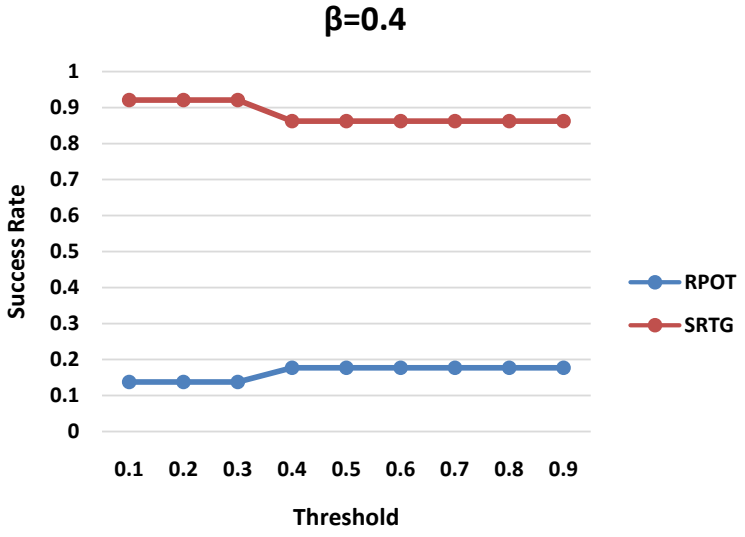
We perform simulations by setting the value of β between 0.3 and 0.7. The experimental results presented in Figure 5 show that the RPOT and STGR values are jointly higher (RPOT = 0.21, STGR = 0.98) when β is equal to 0.6 and the threshold is equal to 0.3. Ultimately, β is set to 0.6 and the selection threshold to 0.3 to provide a model with a high G-supplier transaction success rate and high organization participation.

Table 1. The experiment parameters.

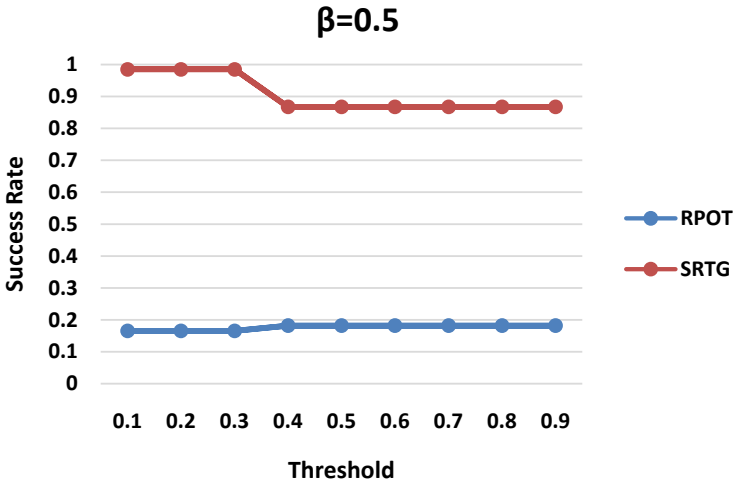
Experiment settings	Values
Number of resource providers	80; 120; 150; 180; 200; 220; 250
Number of resource types	10
Number of sensitivity levels	3
Number of rounds	15, 20
Percentage of malicious providers	20%
Number of transactions per round	500


β=0.3
Success Rate
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Threshold
RPOT
SRTG

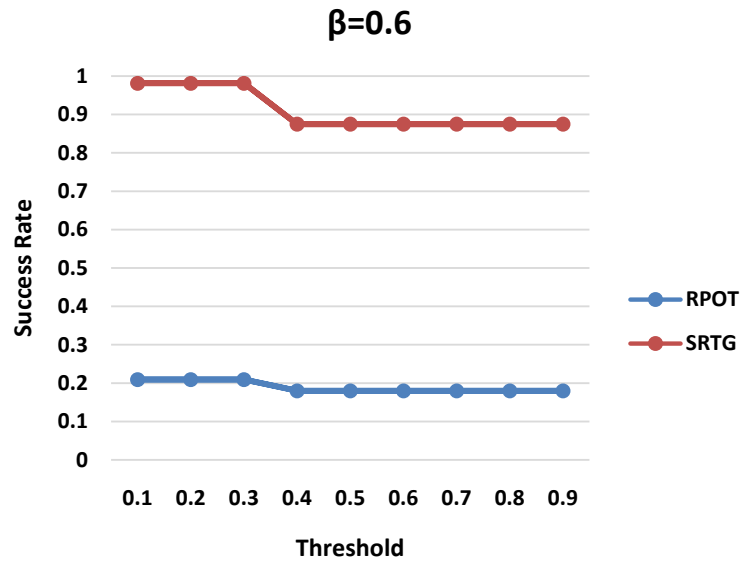
(a)


β=0.4
Success Rate
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Threshold
RPOT
SRTG

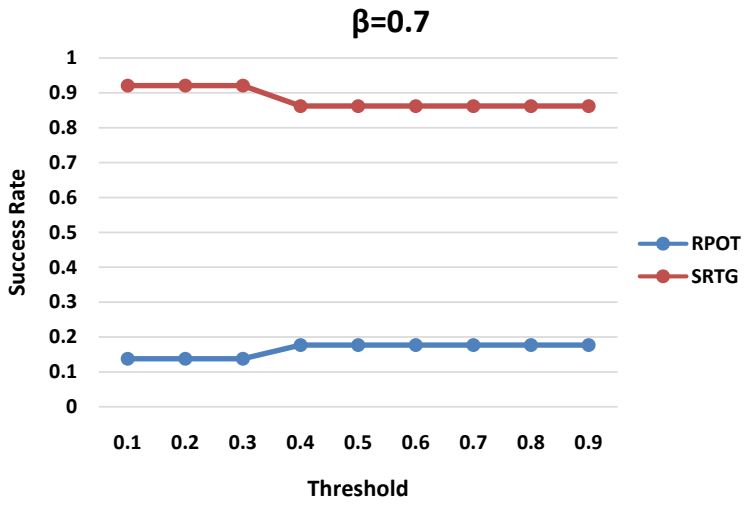
(b)


β=0.5
Success Rate
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Threshold
RPOT
SRTG

(c)



(d)



(e)

Figure 5. Selection of β and threshold.

3.3. Results and Discussion

To analyze the performance of our model, we compared our model to the TNA-SL model [9] [24] and to the Intertrust algorithm [8]. The simulations consisted in evaluating the scalability of the model by increasing the number of organizations in the community. The SeComTrust G-supplier success rate is compared to the other two models mentioned above. Scalability is one of the major characteristics of a Cloud environment [4]. In this paper, the resource sharing framework is modeled around sets of various sizes of organizations. The communities of organizations range from 80 to 250 members. For each group of organizations, 20% of malicious people are integrated. The results in **Figure 6** below show the success rates of the three models (SeComTrust, InterTrust, TNA-SL). Each model is run in 15 rounds of 500 transaction requests per organization

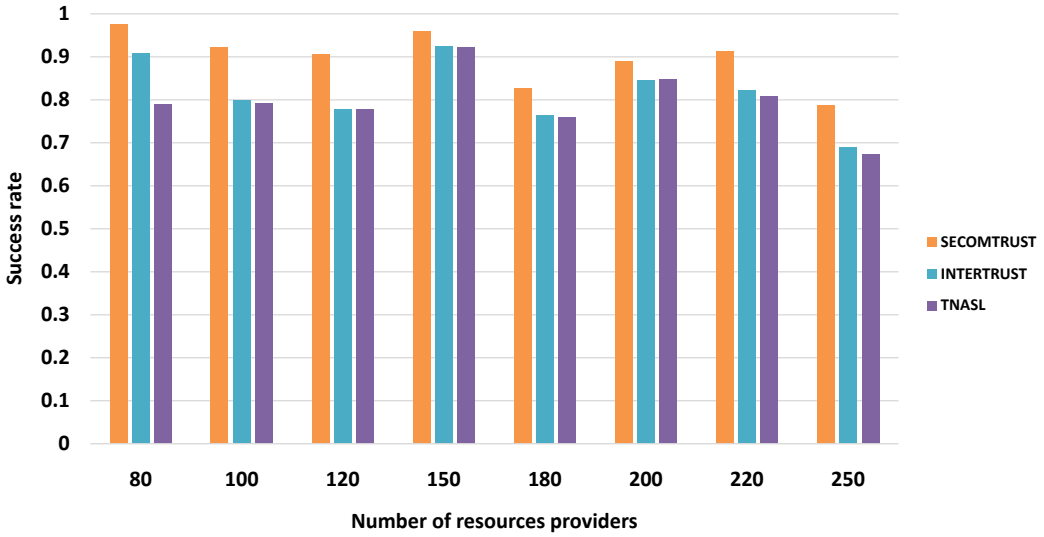


Figure 6. Success rates for different numbers of resource providers of which 20% are malicious providers.

group. From these results, the success rate of SeComTrust is significantly higher than the success rate of the Intertrust and TNA-SL algorithms for all experienced user groups.

This is due to the selection confidence value which is obtained by the weighted sum of the direct or recommended confidence and the specific reputation value, unlike Intertrust and TNA-SL only based on SL parameters. Moreover, the proposed reputation value update mechanism and the selection of the provider from several lists (L_{tram}, L_{repm}) according to a well-defined order justify the success rates of SeComTrust. In conclusion, we can state that our model allows the deployment of scalable 3Cs.

4. Conclusions

Trust management in cloud environments is a major challenge for adoption. However, trust management systems in cloud environments are primarily focused on public deployment types and focused on feedback between users and cloud service providers or between cloud service providers. These techniques do not focus on community cloud architectures. In view of this observation, we propose in this paper the SeComTrust, a model for managing, evaluating, and selecting trusted organizations in a Community Cloud environment based on a Zero Trust strategy. SeComTrust evaluates trusted organizations grouped in security domains by considering the direct interactions between them and their reputation within the community. In addition, this model is associated with a promotion and relegation mechanism to ensure that trust is monitored over time.

Through a series of experiments, we compared the results of our model to InterTrust and TNA-SL algorithms. We have shown that SeComTrust guarantees the scalability of a 3C by presenting success rates (SRTG) largely superior to those of InterTrust and TNA-SL. In future work, we will propose to incorporate

resource quantity attributes and quality metrics into the exchange validation process and evaluate the attack resistance and execution time of SeComTrust.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I. (2009) Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. *Future Generation Computer Systems*, **25**, 599-616. https://doi.org/10.1016/j.future.2008.12.001

[2] Coret, S. (2021) Gartner: Les revenus mondiaux du cloud s'élèveront à 474 milliards de dollars en 2022, contre 408 milliards de dollars en 2021. https://cloud-computing.developpez.com/actu/328766/Gartner-Les-revenus-mondiaux-du-cloud-s-eleveront-a-474-milliards-de-dollars-en-2022-contre-408-milliards-de-dollars-en-2021-le-cloud-sera-la-piece-maitresse-des-nouvelles-experiences-numeriques/

[3] Pearson, S. (2013) Privacy, Security and Trust in Cloud Computing. In: Pearson, S. and Yee, G., Eds., *Computer Communications and Networks*, Springer, London, 3-42. https://doi.org/10.1007/978-1-4471-4189-1_1

[4] Parmar, V. and Bhavsar, C. (2011) Implementing Community Cloud to Overcome the Problems of Complexity and Security in Business Environment. *Indian Journal of Applied Research*, **2**, 1-3. https://doi.org/10.15373/2249555X/MAY2014/214

[5] Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M. (2010) View of Cloud Computing—A Critique. *Computer Communications of the ACM*, **53**, 50-58. https://doi.org/10.1145/1721654.1721672

[6] Coste, B. (2019) Contextual Detection of Cyber-Attacks by Trust Management on Board a Ship. Ecole Nationale Supérieure Mines-Télécom Atlantique, Palaiseau.

[7] Guo, L., Yang, H., Luan, K., Luo, Y., Sun, L. and Zheng, X. (2021) Trust Management Model Based on Mutual Trust and a Reward-with-Punishment Mechanism for Cloud Environments. *Concurrency and Computation*: *Practice and Experience*, **33**, e6283. https://doi.org/10.1002/cpe.6283 https://onlinelibrary.wiley.com/doi/10.1002/cpe.6283

[8] Kurdi, H., Alfaries, A., Al-Anazi, A., Alkharji, S., Addegaither, M., Altoaimy, L. and Ahmed. S.H. (2019) A lightweight trust management algorithm based on subjective logic for interconnected cloud computing. *Journal of Supercomputing*, **75**, 3534-3554. https://doi.org/10.1007/s11227-018-2669-y

[9] Jøsang, A. and Simon, R. (2006) Trust Network Analysis with Subjective Logic. *Angewandte Chemie International Edition*, **6**, 951-952.

[10] Kumar, R. and Goyal, R. (2021) Performance Based Risk Driven Trust (PRTrust): On Modeling of Secured Service Sharing in Peer-to-Peer Federated Cloud. *Computer Communications*, **183**, 136-160. https://doi.org/10.1016/j.comcom.2021.11.013

[11] Kamvar, S.D., Schlosser, M.T. and Garcia-Molina, H. (2003) The Eigentrust Algorithm for Reputation Management in P2P Networks. *Proceedings of the* 12*th international Conference on World Wide Web*, New York, 20-24 May 2003, 640-651.

https://doi.org/10.1145/775152.775242

[12] Kurdi, H., Alshayban, B., Altoaimy, L. and Alsalamah, S. (2018) TrustyFeer: A Subjective Logic Trust Model for Smart City Peer-to-Peer Federated Clouds. *Wireless Communications and Mobile*, **2018**, Article ID: 1073216. https://doi.org/10.1155/2018/1073216

[13] PoojaGoyal and Deora, S.S. (2022) A Review: Trust Management Techniques Used for Cloud Computing. In: Gupta, D., Polkowski, Z., Khanna, A., Bhattacharyya, S. and Castillo, O., Eds., *Proceedings of Data Analytics and Management. Lecture Notes on Data Engineering and Communications Technologies*, Vol. 90, Springer, Singapore, 117-132. https://doi.org/10.1007/978-981-16-6289-8_12

[14] Damera, V.K., Nagesh, A. and Nagaratna, M. (2020) Trust Evaluation Models for Cloud Computing. *International Journal of Scientific & Technology Research*, **9**, 1964-1971.

[15] Mehraj, S. and Banday, M.T. (2020) Stablishing a Zero Trust Strategy in Cloud Computing Environment. 2020 *International Conference on Computer Communication and Informatics*, Coimbatore, 22-24 January 2020, 20-25. https://doi.org/10.1109/ICCCI48352.2020.9104214

[16] Rose, S.W., Borchert, O., Mitchell, S. and Connelly, S. (2020) Zero Trust Architecture. NIST Special Publication, SP 800-207, NIST Computer Security Resource Center, Gaithersburg, 49 p.

[17] Canadian Centre for Cyber Security (2020) Exigences de base en matière de sécurité pour les zones de sécurité de réseau (Version 2.0)-ITSP.80.022. https://cyber.gc.ca/fr/orientation/exigences-de-base-en-matiere-de-securite-pour-les-zones-de-securite-de-reseau-version

[18] FIRST (2020) Common Vulnerability Scoring System SIG. https://www.first.org/cvss/

[19] Cardoso, R.C., Gomes, A.J. and Freir, M.M. (2017) A User Trust System for Online Games-Part II: A Subjective Logic Approach for Trust Inference. *IEEE Transactions on Computational Intelligence and AI in Games*, **9**, 354-368. https://doi.org/10.1109/TCIAIG.2016.2593000

[20] Cardoso, R.C., Gomes, A.J. and Freir, M.M. (2017) A User Trust System for Online Games-Part I: An Activity Theory Approach for Trust Representation. *IEEE Transactions on Computational Intelligence and AI in Games*, **9**, 305-320. https://doi.org/10.1109/TCIAIG.2016.2592965

[21] Zhou, R. and Hwang, K. (2006) Trust Overlay Networks for Global Reputation Aggregation in P2P Grid Computing. 20*th International Parallel and Distributed Processing Symposium*, Rhodes, 25-29 April 2006.

[22] Alunkal, B.K (2003) Grid Eigen Trust: A Framework for Computing Reputation in Grids. Illinois Institute of Technology, Chicago.

[23] Jøsang, A. (2009) Subjective Logic. *Representations*, **171**, 1-8.

[24] Jøsang, A. and Bhuiyan, T. (2017) Optimal Trust Network Analysis with Subjective Logic. 2008 *Second International Conference on Emerging Security Information, Systems and Technologies*, Cap Esterel, 25-31 August 2008, 179-184. https://doi.org/10.1109/SECURWARE.2008.64

Appendices

Algorithm 1 Identification of resource providers

Pre-condition: an organization O_{ui} needs the resource r_{ui} with the degree of sensitivity g_{rui}
Input:O_{ui}, r_{ui}, g_{rui}, the list of resources L_{resm}
Output: The list of resource providers L_{rps}
Assumptions: All requested resources exist and are listed in the list L_{resm} with associated providers

function IDENTIFICATION(Liste $L_{resm}, r_{ui}, g_{rui}$)
  L_{resm} *see equation* 6
  for *each* $O_{pj} \in L_{resm}$ **do**
    if $r_{ui} == r_{pj}$ *and* $g_{rui} == g_{rpj}$ **then**
      $L_{rp} \leftarrow O_{pj}$
    end if
  end for
  if $L_{rp} == null$ **then**
    for *each* $O_{pj} \in L_{resm}$ **do**
      if $r_{ui} == r_{pj}$ *and* $g_{rui} == g_{rpj}$ **then**
        $L_{rp} \leftarrow O_{pj}$
      end if
    end for
  end if
  Sort the organizations in the L_{rp} list by descending security domain and level of assurance. In case of equality of values, the most recent value is considered as the largest.
  $L_{rps} \leftarrow L_{rp}(DESC)$
  return L_{rps}
end function

Algorithm 2 Selecting a resource provider from the direct transaction list

Pre-condition: The list of providers L_{rps} of the sensitivity level g_{rui} resource r_{ui} requested by the organization O_{ui}
Input: The list of resource providers L_{rps}, The requestor's local list of previous transactions L_{tram}, the reputation list of the reputation manager L_{repm}
Output: The selected resource provider O_{pjs}

procedure SELECINDIRECTTRAM($L_{rps}, L_{tram}, L_{repm}, r_{ui}, g_{rui}$,$O_{ui}$)
  $L_{rps} \leftarrow$ Identification($L_{resm}, r_{ui}, g_{rui}$)
  L_{tram} *see equation* 7
  for *each* $O_{pj} \in L_{rps}$ **do**
    if $O_{pj} \in L_{tram}$ **then**
      if $(r_{tram} == r_{ui}$ *and* $g_{tram} == g_{rui})$ *or* $(r_{tram} != r_{ui}$ *and* $g_{tram} \geq g_{rui})$ **then**
        Get trust information from O_{pj}
        Calculate opinion$(\omega_{O_{pj}}^{O_{ui}})$ *of* O_{ui} *on* O_{pj} *using equations* 8 *to* 11
        $\omega_{O_{ui}}^{O_{pj}} \leftarrow gr_{O_{ui}}$ // $gr_{O_{pj}}$ global reputation of O_{ui}
        $valInitTram(O_{pj}, \omega_{O_{ui}}^{O_{pj}}, \omega_{O_{pj}}^{O_{ui}}, r_{ui}, g_{rui}, O_{ui})$
      end if
    end if
  end for
end procedure

Algorithm 3 determination of the trust value by recommendation

Pre-condition: The list of providers L_{rps} of the sensitivity level g_{rui} resource r_{ui} requested by the organization O_{ui}
Input: The list of resource providers L_{rps}, The requestor's local list of previous transactions L_{tram} the reputation list of the reputation manager L_{repm}
Output: The selected resource provider O_{pjs}

procedure SELECTINDIRECTTRAM($L_{rps}, L_{tram}, L_{repm}, r_{ui}, g_{rui}$,$O_{ui}$)
  $L_{rps} \leftarrow$ Identification($L_{resm}, r_{ui}, g_{rui}$)
  L_{tram} *see equation* 7
  for *each* $O_{pi} \in L_{rps}$ **do**
    if O_{pj} *is a Friend of only one Friend* (FoF) *of* O_{ui} *in* L_{tram} **then**
      Get trust information from O_{pj}
      Calculate opinion$(\omega_{O_{pj}}^{O_{ui}})$ *of* O_{ui} *on* O_{pj} *using equations* 8 *to* 12
      $\omega_{O_{ui}}^{O_{pj}} \leftarrow gr_{O_{ui}}$ // $gr_{O_{pj}}$ global reputation of O_{ui}
      $valInitTram(O_{pj}, \omega_{O_{ui}}^{O_{pj}}, \omega_{O_{pj}}^{O_{ui}}, r_{ui}, g_{rui}, O_{ui})$
    else
      //O_{pj} *is a friend of multiple friends* (FoM) *of* O_{ui} *in*L_{tram}
      Get trust information from O_{pj}
      Calculate opinion$(\omega_{O_{pj}}^{O_{ui}})$ *of* O_{ui} *on* O_{pj} *using equations* 8 *to* 11 *and* 13
      $\omega_{O_{ui}}^{O_{pj}} \leftarrow gr_{O_{ui}}$ // $gr_{O_{pj}}$ global reputation of O_{ui}
      $valInitTram(O_{pj}, \omega_{O_{ui}}^{O_{pj}}, \omega_{O_{pj}}^{O_{ui}}, r_{ui}, g_{rui}, O_{ui})$
    end if
  end for
end procedure

Algorithm 4 Selection of a resource provider based on reputation

Pre-condition: The list of providers L_{rps} of the sensitivity level g_{rpj} resource r_{pj} requested by the organization O_{ui}

Input: The list of resource providers L_{rps}, the reputation list of the reputation manager L_{repm}

Output: The selected resource provider O_{pjs}

function SELECTINREP($L_{rps}, L_{repm}, r_{pj}, g_{rpj}$)
  $L_{rps} \leftarrow$ Identification($L_{resm}, r_{ui}, g_{rui}$)
  for *each* $O_{pj} \in L_{rps}$ **do**
    if $sr_{O_{pj}} ! = 0$ **then**
      $L_{sr} \leftarrow O_{pj}$
      Ranked organizations in descending order of specific reputation values
      $L_{srs} \leftarrow L_{sr}(DESC)$
    end if
  end for
  if $L_{srs} ! = null$ **then**
    for *each* $O_{pj} \in L_{srs}$ **do**
      $valInitTram(O_{pj}, \omega_{O_{ui}}^{O_{pj}}, \omega_{O_{uj}}^{O_{ui}}, r_{ui}, g_{rui}, O_{ui})$
    end for
  else
    Ranked organizations in descending order of global reputation values
    $L_{grs} \leftarrow L_{rps}(grDESC)$
    for *each* $O_{pj} \in L_{grs}$ **do**
      $valInitTram(O_{pj}, \omega_{O_{ui}}^{O_{pj}}, \omega_{O_{uj}}^{O_{ui}}, r_{ui}, g_{rui}, O_{ui})$
    end for
  end if
end function