JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Cloud Computing Accessibility through User Behaviour Trust with Triple Dynamic Window

Prem Ranjan,Dr. Shailja Sharma

Computer Science and Engineering Rabinadranath Tagore University Bhopal MP India

Abstract

Human-centered computing models, such as Cloud Computing (CC), have emerged as a result of the advancements in computer hardware, software, and communication technologies. Consequently, the identity-based access control system does not effectively prevent malicious users from gaining access regarding information security difficulties, system stability problems and trust issues between cloud service users and cloud service providers are all occurring. It's not yet mature enough to solve issues such as UBT (User Behaviour Trust) evaluation complexity, trust dynamic update efficiency, and evaluation correctness, all of which can pose security challenges for identity-based access control systems. An improved UBT-based CC access control model compared to the current state of the art is the goal of the research. As a result, SLR was used to identify the structure of the UBT-based access control model, summarise CSU behaviours that can be collected as evidence for UBT evaluation, identify trust

attributes that will affect the assessment of UBT, and evaluate the current state of technology and its potential advantages, opportunities, and weaknesses. There are eight current state-of-the-art UBT-based access control models, as well as 23 evidence items from CSU's behaviour that have been grouped into three categories: four important trust qualities, influencers, and countermeasures. The simulation findings reveal that the TDW model performs effectively when it comes to trust fraud and trust expiration while using a prototype of the Triple Dynamic Window Access Control model. This work developed an access control model based on TDW to address the flaws II of trust fraud protection and the trust expiration problem.

Keywords: Internet of Things, Triple Dynamic Windows-based Access Control Model, BTR (Behaviour Trust Record), S3, NetLogo

1. Introduction

In the cloud computing approach, an extensive system pool provides customers with a wide range of storage and computational capabilities[1]. Today's IT development has grown to be a prominent trend. Many personal and commercial customers were drawn to its simplicity, scalability, and on-demand features[2]. However, safety is a serious concern with its use. There is the issue of finding a trustworthy cloud provider, and there is also the issue of cloud users being concerned about security risks. Consequently, cloud computing security will be a long and major area of research in both the application and research process. Models like Infrastructure as a Service and Platform as a Service are used for the cloud

platform, which is service-oriented[3]. An ever-increasing focus is being placed on maintaining the platform's security as the number of people using it and the content it contains grows at an exponential rate. Furthermore, the cloud platform's security issues are exacerbated by various users and their behaviour[4]. In addition, the user's absence will result in irregular access to and privacy for the data in question. As a result, safeguards for data security and aberrant user behaviour must be devised.

When it comes to supporting Internet-connected devices at the periphery, cloud computing faces significant difficulties[5]. There will always be a lengthy network distance between clients and remote clouds because cloud data centres are usually located at the heart of the Internet, which causes significant network delays for end users. Many application scenarios do not lend themselves to latency sensitive and IoT-cloud applications, where the client device is a mobile device such as a smartphone or an Internet of Things (IoT) device[6].

Cloud computing's meteoric rise can be attributed largely to the widespread adoption of virtualization. There are several benefits of extending cloud computing with edge computers such as reducing cloud computing's drawbacks and keeping many cloud computing's features. When it comes to edge computing, system-level virtualisation (SLV) is a common method of implementation. Virtual machines (such as CPUs, GPUs, and so on) can be shared among multiple users using SLV techniques in the same way as individual users can[7]. SLV techniques, on the other hand, remain prevalent and are also effective in most current models of edge computing where

nodes at the edge are designed to be general-purpose devices that the public can use.

This study was inspired by the idea that the user's actions might gauge the trustworthiness of a cloud user. Let's say you have an abnormal user. It's going to cause some weird behaviour. A framework of this type may handle any application situation involving user access.A simple kind of anomaly detection can be viewed as a form of anomaly detection in our trust model. Based on trust, a person's ability to be detected is determined. Trust value is computed and stored on the cloud platform. As a result, it is safe to say that the trust value is reliable.

The following are some of the advantages of our proposal. This is a simple concept that can be implemented in the cloud with ease. Second, trust's worth is dynamic and can change with the times. Lastly, the user's behaviour significantly impacts the trust value that can be calculated.Next, we'll take a look at the content itself. The first step in understanding cloud computing security research is to read this article. After that, a quick look at how people use the cloud Third section will focus on developing a trust evaluation model based on cloud user behaviour data. Then, the digital book cloud platform simulation is used to verify the model, and the findings and analysis are presented in the fourth section. Finally, the essay is summarised, and the project's future is outlined.

2. Literature Survey

The Access Control Model is a key part of ensuring system security, integrity, and availability. As a result, computer

security researchers are increasingly focusing on access control technology. However, access control traditionally used large-scale resource host access control, which can't be used in the cloud computing environment and doesn't address the current security issue. Therefore, J. Almutairi et al. [8] analysed the Centralized and Decentralized Access Control Models in Cloud Computing. Standard RBAC Reference Model standards for access control, enlarged RBAC model to meet the CC complicated access control and management needs[9].

On the other hand, RBAC can only be used in a closed and centralised network environment and cannot match the security requirements of CC's multi-domain environment, which requires RBAC[10]. Some researchers incorporated a trust management mechanism into the RBAC model to solve the methodology's shortcomings. Based on Blaze's "trust management" concept, Tang et al. presented the TRBAC (Trust Role-Based Access Control Model), a trust-based access control model, in response to the RBAC model's shortcomings[11]. When it comes to providing authority to users, this model identified specific requirements, computed the many trust features of the user, and achieved a fine-grained authorization process that is safer and more reasonable.

Intrusion detection and classification models were proposed by Zina et al. To ensure a secure network, use TIDCS and TIDCS-A, two types of trust-based intrusion detection and classification systems. Using a new technique for selecting features, TIDCS trims down the amount of information in the input data. In the beginning, the features are grouped randomly to enhance the chance of them being included in the generation of distinct groups and then sorted depending on their accuracy ratings. Nodes in the network can only classify packets they receive based on their prior performance if they have high-ranking characteristics. In addition, trust links between nodes in TIDCS are examined and refreshed regularly to keep the system running smoothly.

According to [12], CNN is shaping the future generation of cloud security since it can give automatic and responsive techniques to boost security in the cloud. Furthermore, using machine learning (ML), it is possible to develop solutions that combine comprehensive algorithms for secure enterprise data across all cloud apps.

According to Tan et al. [13] and Wenhui [14], a dynamic RBAC model built on trust was developed for use in cloud computing environments to address the shortcomings of the TRBAC paradigm. As outlined in the research [13], the CC environment's security risks can be minimised by using a systematic approach to assigning different levels of access authority based on a user's role and trust degree.

3. Proposed Methodology

This paper has introduced the UBT-based access control model, the Triple Dynamic Windows-based Access Control Model(TDWACM). The design principle, model features, trust updating process and trust

evaluation method, were also discussed in this section. The following are the fundamentals: The trust definitions and restrictions we used to create our model are presented in this section.

Basic Design are based on the trust traits

1. The trust assessment will not include BTRs that have expired.When it comes to assessing trustworthiness, valid BTRs' attachment is inversely related to their age.Users' BTRs influence the credibility of trust assessments.Trust rises slowly to avoid the user quickly obtaining a high trust score with fewer encounters.To punish a user who engages in malicious conduct, trust rapidly decreases.

3.1 An Introduction to Trust: Definition and Limits Trust Value is a double-precision floating-point value between 0 and 1. In BTRs, user behaviour is documented and analysed. The following is a breakdown of customer trust and the service approach that goes with it: A user's access is disallowed if the Trust Value is less than 0.15. With a distrust level of (0.15 Trust Value 0.35), the service strategy offers few essential services and has little authority. Be on the lookout for people that fit this description. To provide essential services and general authority, the service strategy must have a trust level of (0.35 – Trust Value – 0.65). A high level of authority and a significant number of services can be provided if the trust value is (0.65 Trust Value 0.85). It's possible to supply core services and superior authority with a trust level of 0.85. With an initial trust value of 0.5, good behaviour is associated with an increased Trust Value; lousy behaviour is associated with a decreased Trust Value.

A TDW-based approach to assessing trustworthiness Next the focus will be on model's architecture and the unique characteristics of its components.

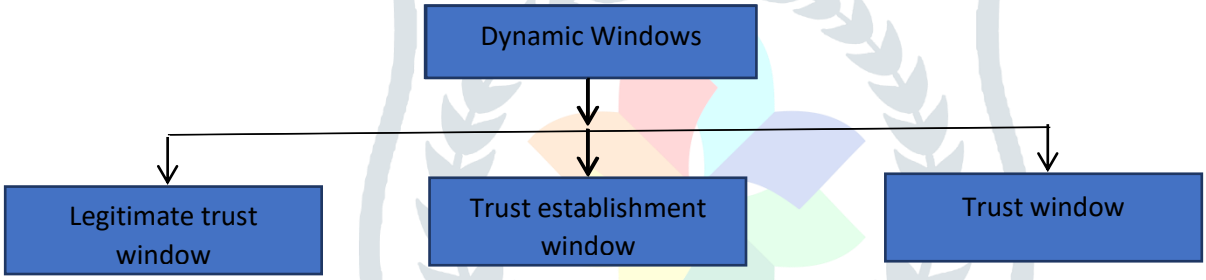

Figure 1. Dynamic Windows

3.2 Double Dynamic Windows –Figure 1depicts three dynamic windows: a legitimate trust window, a trust establishment window, and a most recent trust window. Method Expiration of a trust record will result in it being squeezed out of the Valid Trust Window from the left side of this window. (W Max is the maximum size of the window.) the minimum number of access records should be defined during the trust establishment (W Min is the window's minimum size). For users with BTR counts lower than W Min, the actual trust evaluation uses a "low increase" technique to limit the risk of trust uncertainty and to prevent fraud. It defined the range of the most recent BTRs. (W Rec represents the window's size). To ensure that the final evaluation results match the user's current status, W Rec presents the user's most recent behaviour trust assessment. It is possible to modify the size of each of the three windows shown above to meet varied system needs.

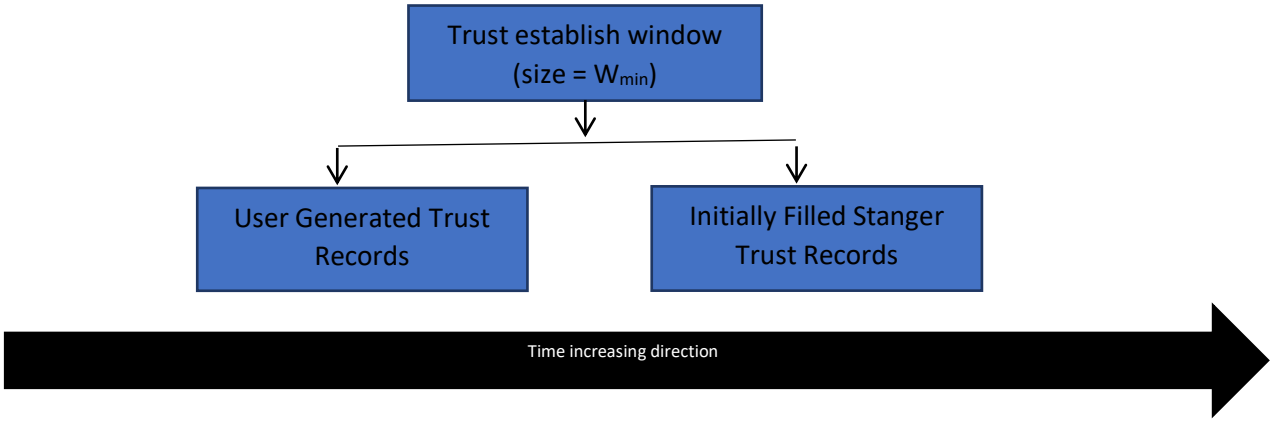

Figure 2: The Trust Establishment Window Is Being Initialized

3.3 Classification of BTRs come in three flavours, and these are labelled with the Flag[15]: In the initialization of the Trust Establish Window, and as BTRs, when a user is inactive for an extended period, the Strange User Record is employed. (Assigned the label "Strange") During the validity period, BTRs were generated based on the user's fundamental interactions with the system. The word "normal" is denoted. Due to the user's malicious activity, the normal BTRs for a portion of the user's normal BTRs were reduced. (BTRs tagged as Punish were affected by the punishment. Window Setup and Initialization Trust Establish Window's BTRs are initialized with unusual user records, the trust value marked as STR, and can be adjusted based on different system needs. The weird BTRs in the Trust Establish Window will be replaced with standard BTRs as the user progresses. Figure 2 depicts the initialization and replacement process. An instance in which the system recorded the BTRs of three different users is shown in the figure. Normal BTRs were used to replace three BTRs that were unusual. The Most Recent Trust Window will store new BTRs once the Trust Establish Window has been filled with standard BTRs.

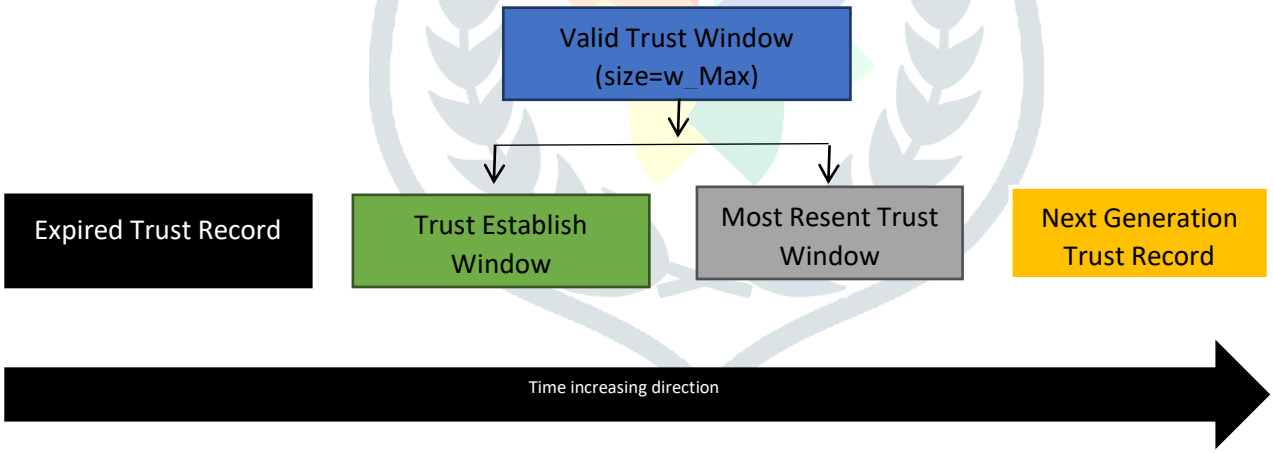

Figure 3. Most Recent Trust Window

An example of this is seen in Figures 3. As a result, the most recent BTRs will always be displayed in the Most Recent Trust Window. As seen in Figure 3. In the most recent window of trust, Valid Trust Window's oldest trust record will be removed from the left side of W Max when there are more BTRs than W Max. Figure 4 shows process 40 in action. New trust records will cause the borders of three windows to shift one step each time

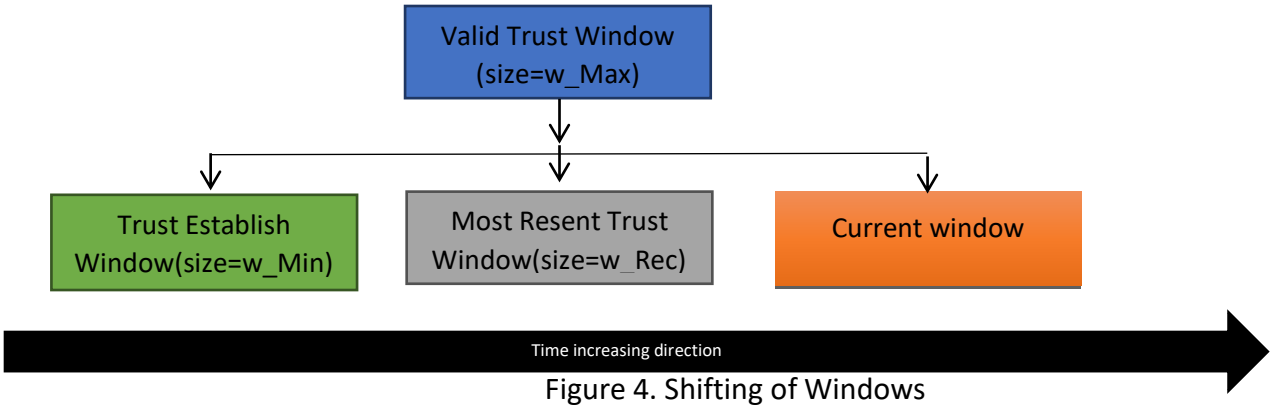

Figure 4. Shifting of Windows

As a result, the system will only keep BTRs that are still valid, and the expired BTRs will be removed automatically. Squeeze out the old BTRs in Figure 4

3.3 Update Strategy BTRs: This part explains how to keep your behaviour trust records up to date.Strategy Updates for Malicious BTRs BTRs that are found to be engaging in malicious conduct (i.e., the Trust Value is less than half) will have some of their most recently evaluated trust values converted to distrust values (mistrust), lowering the user's overall trust value swiftly as punishment for their bad actions. The trust penalty must consider four factors: The greater the degree of malice, the more severe the penalty. The smaller the Tm, the bigger the punishment for the detected malevolent action. This is based on the user's current comprehensive trust value (Tc): the higher the current trust value, the bigger the punishment. Given that a user's present trust value indicates the level of service they've gotten and their previous authority, harmful action can damage more if their trust value is high. Seventh (Nm): the more malevolent activities, the more lenient the penalty. Malicious activity can be classified as probable misoperation for a tiny percentage of the population. For repeated offences, the severity of punishment will continue to rise. The system's actual security requirements and application scenarios. A penalty factor was established. We can calculate n using formula (1) and mistrust (2).
In this case, N = Tc Tm = min Tc Tm (1)
0.5 Nm of mistrust (2)
Only the actual user's BTRs will be punished, since formula (1) only counts the number of genuine BTRs, excluding expired BTRs and unusual BTRs, which are not included. With =10, Tm=0.3, Tc=0.6, Nm=2, m=30, with formula (1) we obtain N=20, and with formula (2) we get distrust = 0.25. The penalty drops the current trust value from 0.20 to 0.25. 'Punish'-designated BTRs. Figure 5 shows the procedure in action. Update strategy for malicious BTRs.

3.3.2 Strategy for Updating Expired BTRs It is common for BTRs to expire overtime when a user has been absent from the system for a lengthy period. (Current Time – Time I > W Max).). Because there are no new BTRs generated when a user stop using the system, old BTRs will remain in the Valid Trust Window, requiring an expired BTRs update approach. By using Strange BTRs to replace the expired ones, we constructed an effective updating method. BTRs that have been out of date for some time will be replaced by "Strange BTRs" if the user has not been using the system. Since the user's trust value will decrease over time, the user will be perceived as a peculiar user. Untrusted users, like trusted ones, tend to cease using the system for long periods before returning. This trait is consistent with the way people behave when they have lost confidence in others.

4. Results and Discussion

NetLogo utilises an agent-based programming environment to model natural and social phenomena[16]. Development platform NetLogo was derived from the Logo language. As a result, the Logo language may now be used to govern better the behaviour of tens of thousands of simulated individuals. As a result, NetLogo can simulate both individual behaviour at the cellular level and macro-scale evolution. NetLogo is well-suited for simulating both natural and social phenomena, but it excels when dealing with systems

that exhibit time variation. As part of the TDWACM prototype development, we utilised NetLogo to create a simulation environment. We emulated the essential elements found in a CC network in this environment, including the CSP and several different types of CSUs.

There are five categories of CSUs: honest users, dishonest users, cheaters, random users, and intermittent users. Honest users fall towards the first category. The number of "good" users (represented by green turtles) can be adjusted using a slide bar, with a default value of 200. By default, the number of problematic users is set at 200, however the slider can be used to increase or decrease this number. Fraudulent users maintain good behaviour for a period of time before engaging in malevolent behaviour numerous times before returning to good behaviour; this cycle is repeated indefinitely. A slide bar lets you adjust the number of cheaters, which are represented as yellow turtles. The default value is 200. Random Users: Randomly perform nice and harmful actions. In the form of blue turtles, the number of random users can be adjusted using a slider and set at 200. When interacting with a server, intermittent users always exhibit positive behaviour, regardless of how long they are interacting with the service. Interaction time is set to 30 seconds, with a 60-second wait between each exchange. Violet turtles represent the number of intermittent users, which can be set to 100.

Parameters used for the evaluation of the model are Strong Mistrust, Mistrust, General Trust, Trust and Very Trust. The prototypes used during the processing are Valid Trust Window Size, Trust Establish Window Size, Most Recent Trust Window Size, Initial User Trust Value, Penalty Factor, Strange Trust Value.

Table 1. Comparison table of the proposed model with the S3 model

	Types of Users	Mistrust	Strong Mistrust	Neutral Trust	Normal Trsut	Very Trust
Proposed Model based on Triple Dynamic window	Bad User	0	250	0	0	0
	Random	168	0	23	0	0
	Good User	0	0	0	56	168
Simple Storage Service (S3)	Bad User	0	220	0	0	0
	Random	69	139	0	0	0
	Good User	0	0	0	22	158

In the table 1, the comparison is shown between the proposed model with the S3 model and it has shown that the proposed model has performed better than the existing S3 model.

5. Conclusion and Future Work

Based on simulation results, this study created and tested a new UBT-based access control model that has the following advantages over the present state-of-the-art UBT model:

- Higher levels of trust and security
- Less reliance on the manual intervention
- Greater scalability

Using the "slow rise" approach to prevent trust fraud effectively and using a "rapid decline" penalty technique, the company can quickly respond to malicious activity and efficiently deter malicious behaviour and hostile users. Using an expired trust update approach and the most current trust calculation, it is possible to accurately reflect the recent credibility of the user accessing the site. Simple and configurable data structures and a trust evaluation mechanism that can be easily scaled are all features of this system.

References

[1] A. Tchernykh, U. Schwiegelsohn, E. ghazali Talbi, and M. Babenko, “Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability,” *Journal of Computational Science*, vol. 36, 2019, doi: 10.1016/j.jocs.2016.11.011.

[2] K. S. Chaudhury, S. Pattnaik, H. S. Moharana, and S. Pradhan, "Static Load Balancing Algorithms in Cloud Computing: Challenges and Solutions," in *Advances in Intelligent Systems and Computing*, 2020, vol. 1118. doi: 10.1007/978-981-15-2475-2_24.

[3] R. Jia, Y. Yang, J. Grundy, J. Keung, and L. Hao, "A systematic review of scheduling approaches on multi-tenancy cloud platforms," *Information and Software Technology*, vol. 132. 2021. doi: 10.1016/j.infsof.2020.106478.

[4] R. Kumar and R. Nair, "Multi-Cryptosystem based Privacy-Preserving Public Auditing for Regenerating Code based Cloud Storage," *International Journal of Computer Applications*, vol. 155, no. 10, pp. 16–21, 2016, doi: 10.5120/ijca2016912442.

[5] Y. Maher and B. Danouj, "Survey on deep learning applied to predictive maintenance," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 6. 2020. doi: 10.11591/ijece.v10i6.pp5592-5598.

[6] R. Nair, P. Sharma, A. Bhagat, and V. K. Dwivedi, "A Survey on IoT (Internet of Things) Emerging Technologies and Its Application," *International Journal of End-User Computing and Development*, 2019, doi: 10.4018/ijeucd.2018070101.

[7] P. Sharma, R. Nair, and V. K. Dwivedi, "Power consumption reduction in iot devices through field-programmable gate array with nanobridge switch," 2021. doi: 10.1007/978-981-15-7130-5_54.

[8] S. Almutairi, N. Alghanmi, and M. M. Monowar, "Survey of Centralized and Decentralized Access Control Models in Cloud Computing," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 2, 2021, doi: 10.14569/IJACSA.2021.0120243.

[9] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Computer role-based access control models," *Computer*, vol. 29, no. 2. 1996. doi: 10.1109/2.485845.

[10] S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A role-based access control model in modbus SCADA systems. A centralized model approach," *Sensors (Switzerland)*, vol. 19, no. 20, 2019, doi: 10.3390/s19204455.

[11] Z. Tang, J. Wei, A. Sallam, K. Li, and R. Li, "A new RBAC based access control model for cloud computing," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7296 LNCS. doi: 10.1007/978-3-642-30767-6_24.

[12] P. Mell and T. Grance, "The NIST definition of cloud computing," in *Cloud Computing and Government: Background, Benefits, Risks*, 2011. doi: 10.1016/b978-0-12-804018-8.15003-x.

[13] Z. Tan, Z. Tang, R. Li, A. Sallam, and L. Yang, "Research on trust-based access control model in cloud computing," in *Proceedings - 2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference, ITAIC 2011*, 2011, vol. 2. doi: 10.1109/ITAIC.2011.6030345.

[14] W. Wang, J. Han, M. Song, and X. Wang, "The design of a trust and role based access control model in cloud computing," 2011. doi: 10.1109/ICPCA.2011.6106526.

[15] J. Kantert, S. Tomforde, R. Scharrer, S. Weber, S. Edenhofer, and C. Müller-Schloer, "Identification and classification of agent behaviour at runtime in open, trust-based organic computing systems," *Journal of Systems Architecture*, vol. 75, 2017, doi: 10.1016/j.sysarc.2017.02.003.

[16] A. Duering and J. Wahl, "A massacred village community? Agent-based modelling sheds new light on the demography of the Neolithic mass grave of Talheim," *Anthropologischer Anzeiger*, vol. 71, no. 4, 2014, doi: 10.1127/anthranz/2014/0450.