International Journal of Computer Science and Soft Computing
Volume 1 Issue 1

Security Measures in Cloud Networks Robotics: A Study

Umang Kant
Assistant Professor
Dept. of Computer Science and Engineering
Krishna Engineering College, Ghaziabad, India
Corresponding Author's email id: *umang.kant@gmail.com*

Abstract

The aim of this paper is to explore the security measures that can be taken while accessing the data and resources from a cloud by a network of robots. This work is an extension of the prior work done in the field of cloud networked robotics. Cloud networked robotics has taken its inspiration from cloud computing and networked robotics. Since this paper deals with accessing services (data and resources) from the cloud by the robot network, hence great deal of attempts should be made in the security during the whole process of data transfer. An introduction to cloud computing, networked robotics and cloud robotics is given in Section I. Section II explores the objective of this study i.e. the need for security and measures taken by the cloud when it gets the request from the robot network for a service. A detailed overview of the various cases to provide security to the data transfer from cloud to robot network, their applications, advantages and limitations are given in Section III. Section IV consists of the conclusion and future work of this research.

Keywords: *Cloud; Networked Robotics; Cloud Robotics; Robot; Robot Network; Time Slice; Registry File; Registry; Complexity; Queuing; Deadlock; Starvation; Master Robot; Slave Robot; Hit.*

INTRODUCTION

In today's scenario, the scope of uses where robot technology frameworks are utilized is astonishingly wide and has brought noteworthy effects throughout the decades. Robots are machines modified for committed purposes and are fruitful in modern applications because of their alluring properties like pace, exactness and perseverance. They are broadly used for tedious, unsafe or monotonous undertakings [1]. Network technologies have collaborated with robotic technologies to expand the functional range of robots [1]. In network robotics, a team of robots are interconnected with a communication network to collaboratively accomplish tasks [2]. This interconnection can be wired or wireless. Fig 01 illustrates the concept of network robotics [1]. To enhance the working functionalities of robots and to deal with huge data and resources, the robotic technologies have amalgamated with cloud technologies. This further collaboration has resulted in cloud robotics also termed as cloud network robotics.

Fig 02 illustrates the concept of cloud robotics. The cloud permits robots to work in an autonomous framework for sharing information from physical trials in a diversity of scenarios. Hence cloud robotics is a concept that integrates robots and mechanized frameworks which depend on information or code from a network to bolster their maneuver [3]. It makes utilization of disseminated assets, consolidates them to accomplish higher throughput and takes care of expansive scale calculation issues.

In cloud robot technology, distributed computing amplifies the capacities of interconnected robots by disposing of a few confinements and giving adaptable assets to the system of robots. Cloud robotics structural planning is influenced by a blend of an expedient cloud framed by machine-to-machine (M2M) interchanges among the colluding robots, along with infrastructure cloud empowered by machine-to-cloud (M2C) correspondences [2] [1]. Numerous models have been proposed so far for establishing an efficient communication between robot and cloud. Cloud helps in achieving the goal of utilization of resources in the form of an elastic model. This model enables the allotment of assets from a common asset pool in the pervasive cloud.

There are numerous wellsprings of instability and security breaches possible in a framework like this - where robots take and store data from/on a cloud. Cloud computing can exhibit a highly unstable behavior at the time of ongoing requirements in a real time environment. Thus, the paper provides an outlook on the specialized difficulties related to computational complexities, data sharing and security hacking, along with the potential advantages of cloud robotics technology in distinctive applications.

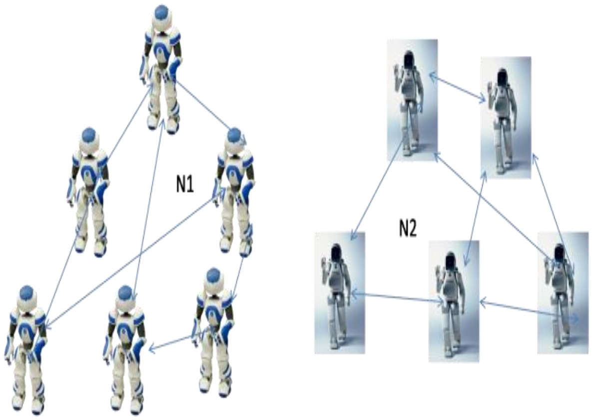

Fig 1. This figure shows two different networks of robots namely N1 and N2. M2M Communication has been show with the arrows [1].

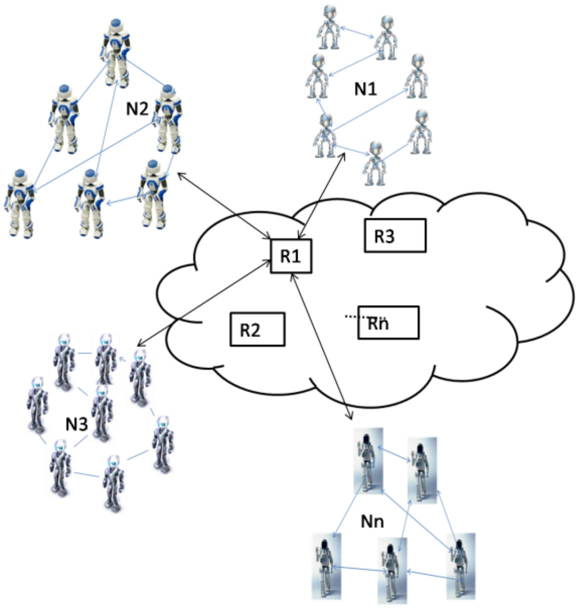

Fig 2. This figure illustrates the concept of cloud robotics. Here there are n number of robot networks namely N1, N2, N3 & Nn. This figure shows both M2M and M2C communication. Here all the robot networks send requests to cloud for the same resource R1 at the same time [1].

OBJECTIVES OF THE STUDY

As discussed, a cloud robotics network consists of a number of robot networks which request services from the cloud to meet their requirements. During this service handling, there might be some security threats which can have catastrophic effect on the whole environment. There is need to find a solution to this problem. Hence the following objectives have been identified during this study;

a) To find out the need for security while data/resource transfer between cloud and robot network and its robots.

b) To find out the various cases in which security lapses can occur during the data transfer.

c) To find out the different cases in which the robot network and robot can get registered with the cloud.

d) Application, advantages and limitations of each case.

DETAILED ANALYSIS OF SECURITY MEASURES TAKEN DURING DATA/RESOURCE TRANSFER

In this scenario, there is an environment which consists of a cloud and one or more robot networks; we assume that there are 'N' number of robot network and each robot network consists of 'M' number of robots. There can be two types of service requests to the cloud; first, the whole robot network can request for service, second; a particular robot from the robot network can request for the service [1]. In either case, before the robot or the whole robot network even begin to access the service from the cloud, they need to get themselves registered with the cloud i.e. the robot network and its robots cannot interact with the cloud until and unless they are registered with the cloud. The cloud maintains a Registry File which maintains the necessary information about the robot network and its robots. The robot network and robots have to validate their existence to the cloud before accessing the service from the cloud. It is the cloud's decision to whether grant permission to a particular robot network or robot; if granted, the cloud will register the robot network and robot in the registry file; else the request will be discarded. Fig 03 shows the registration process. The cloud's decision can be based on various factors such as availability of resource, priority of the robot network, latency, congestion etc. In the remaining part of this section, we will discuss the various cases of registering the robot network and the robots with the cloud.

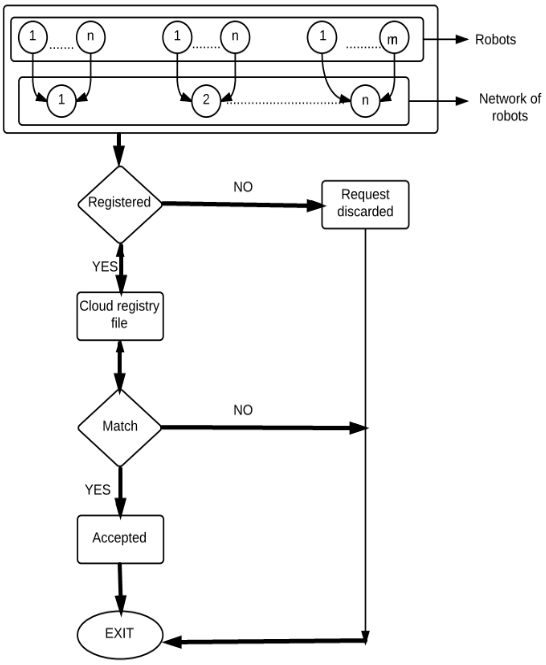

Fig 3: flow diagram of cloud robotics

CASE I: *Distributed System where the Cloud has complete control over the robot network*

In this case, the cloud grants permission to all the robots networks and their constituent robots which come for registry; which implies that the total number of entries in the registry file will be 'NxM'. 'NxM' also denotes the complexity of the registry table. In this case, the cloud has complete control over every robot of every network which is registered with it. By complete control it is meant that the robot network and its robots do not have the permission to interfere in the cloud's processing. Hence if the cloud discovers any security threat from any robot for a particular robot network, it can discard its entry from the registry file for that particular robot. This deletion of the robot's registry from the cloud's registry file would not affect the working of the robot network in whole, only the corrupt robot would face the consequences. Similarly, if the problem is coming from many robots of a single robot network, instead of discarding each robot, the entire robot network can be discarded. Fig 04 shows the working of this case I. This would save the time of the system, which can be used in some other processing. But due to the number of entries, the size of the registry file becomes large which increases the complexity of the whole system. Also, since a large number of robots and robot networks are registered with the cloud, and if they request for same resource for their processing, this will lead to the increase in queuing, which in turn can lead to deadlock and starvation.

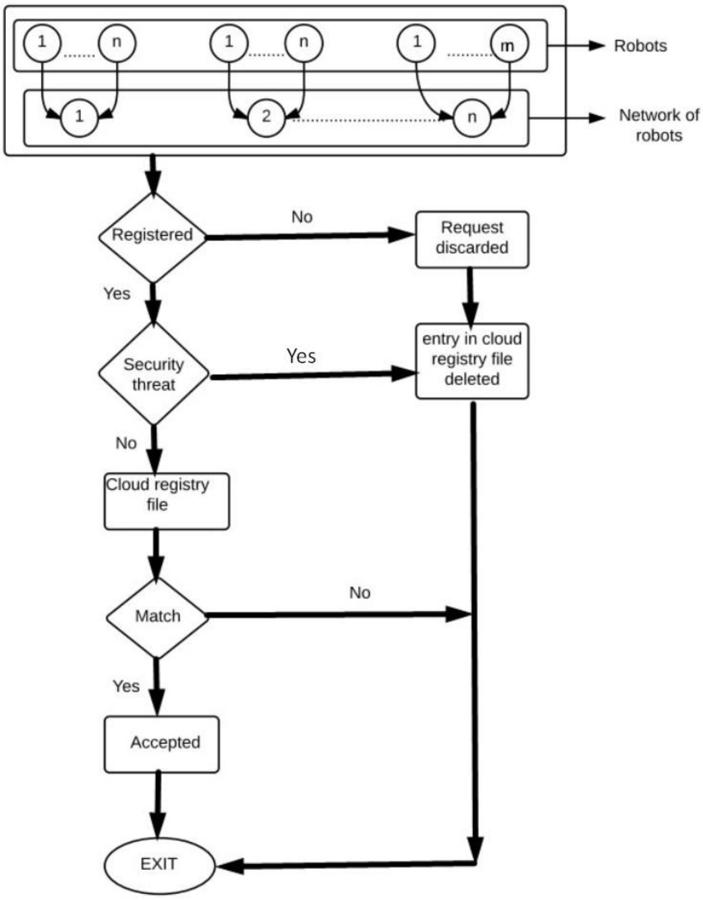

Fig 4: Distributed system in cloud robotics

CASE II: *Centralized System with a Master Robot*

In this case, every robot network will appoint a master robot for itself. Therefore, instead of keeping the entry for each robot of the robot network, the cloud will keep the entry for the robot network and its corresponding master robot. Hence, the number of entries in the table will be 'N' in this case which is nothing but the complexity of the system. Here the complexity is ‘N’ because the slave robots register through the master robot and there is a single master robot for one robot network. The robot appointed as master in a robot network, responsible for keeping track of all the recently used information and resources for that particular network. The rest of the robots are made slaves. If any of the slave robots want to access any resource from the cloud, they would first consult the master robot. If the resource is found with the Master, it's a Hit. Else it is a Miss. In the case of Miss, the slave robot informs the master about the non-availability of the needed resource. The master robot will then access the cloud for that resource and eventually pass down the resource to the particular slave robot. This whole scheme would work only if the master robot is registered with the cloud. Else, the whole request will be discarded. The robot which is made the master for the robot network would exhibit some extra functionality and hence would have extra responsibilities as compared to the slave robots.

As discussed before, the complexity in this case is 'N', therefore this scheme would reduce the traffic of the cloud and the access to resource in the cloud would be faster.

The downfall of this scheme is that the cloud would have to discard the entire robot network, even it has problem with just a single robot of that robot network. This is because, in this case, the robot network consists of a master robot which would hold almost all the important information about the cloud which can leak that information to other enterprises. This is a serious security threat. So apart from discarding the suspicious robot, the master robot would also have to be discarded. And with the master robot gone, the existence of whole robot network would have no meaning. So, it would be sensible to discard the whole robot network.

CASE III: *Adaptive Architecture with Time Slice*

This case applies constraints to both the cases discussed above. In this case, each robot network is given a time slice T i.e. the cloud registers the robot network and its robots for a limited time period only. Here, T represents the time limit up to which a robot network can access the cloud before another robot network gets control of the cloud. Here the number of entries in the registry table would be 'NxM', but for a limited time period. This case overcomes the shortcomings of both the cases discussed above. Incorporating the concept of time slice in cloud robotics increases its security and resources of the cloud become available to other robots and networks in a systematic manner i.e. no robot network can catch hold of the resource for large amount of time, the control of the resource will be passed over to the cloud once the time exhausts.

The main disadvantage of this case is that overhead will and increase and constant updation into the registry file would cost a lot of battery life.

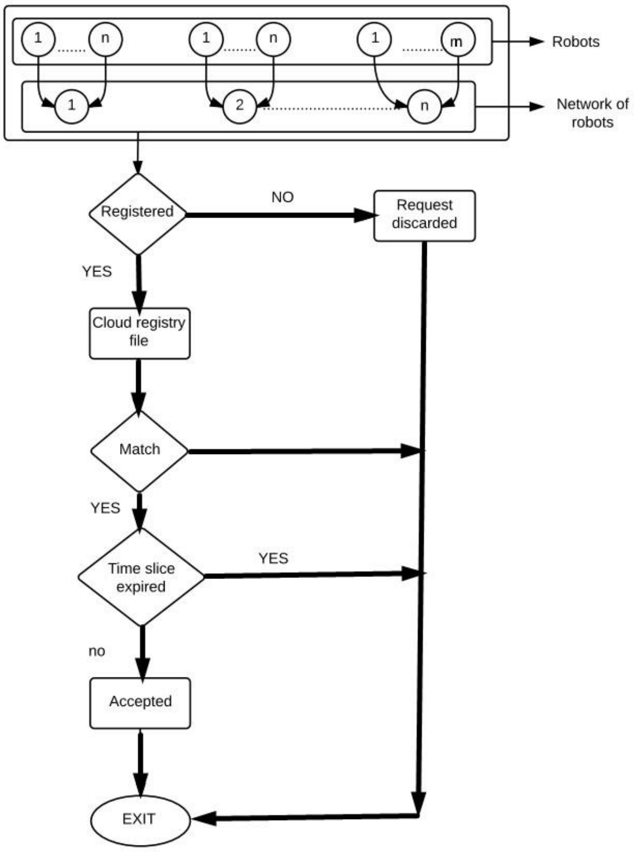

Fig 5: Adaptive architecture with time slice

CONCLUSION

Seeping autonomy and robotics procedures into the Cloud requires a secured infrastructure that encourages this move. New procedures and strategies are required to adapt to time-changing system dormancy and Quality-of-Service. Cloud assets may or may not be sedate, corrupted, or occupied. Hence the paper presents a methodology for cloud robotics autonomy to model the requirements confronted by current robots in the network in a secured manner. This methodology aims to permit robots to share assets, data and information with one another without the fear of a security hack. Thus, the paper provides a chance to get to new learnings and aptitudes not learned before. This opens another worldview in robotics autonomy that prompts energizing future advancements.

REFERENCES

I. Dipti Trivedi, Umang Kant, “Scheduling Algorithms and their Applications for Cloud Networked Robotics: A Study”, International Conference on Technical and Innovative Advancements in Computing and Communication Engineering (TIACCE-2013), ISBN No. 81-89278-18-5.

II. Hu, Guoqiang, Wee Peng Tay, and Yonggang Wen, "Cloud robotics: architecture, challenges and applications." Network, IEEE 26.3 (2012): 21-28.

III. Kehoe, Ben, et al. "A survey of research on cloud robotics and automation." Automation Science and Engineering, IEEE Transactions on 12.2 (2015): 398-409.

IV. Agostinho, Lucio, et al. "A cloud computing environment for supporting networked robotics applications." Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on. IEEE, 2011.

V. Liu, Bingwei, et al. "A holistic cloud-enabled robotics system for real-time video tracking application." Future Information Technology. Springer Berlin Heidelberg, 2014. 455-468.

VI. Furrer, Jonas, et al. "Unr-pf: An open-source platform for cloud networked robotic services." System Integration (SII), 2012 IEEE/SICE International Symposium on. IEEE, 2012.

VII. Sibley, Gabriel T., Mohammad H. Rahimi, and Gaurav S. Sukhatme. "Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks." Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on. Vol. 2. IEEE, 2002.

VIII. Wei-Fu Hsien, Chou-Chen Yang, and Min-Shiang Hwang, “A Survey of Public Auditing for Secure Data Storage in Cloud Computing.” International Journal of Network Security, Vol.18, No.1, PP.133-142, Jan 2016.

IX. Ch. Chakradhara Rao, A. V. Ramana, “Data Security in Cloud Computing.” International Journal of Current Trends in

Engineering & Research (IJCTER), e-ISSN 2455-1392 Volume 2 Issue 4, April 2016 pp. 84-92

X. Shepard Chifamba, “A Study on Cloud Robotics: Ad-Hoc Cloud (Cloud Seeding)” International Journal of Innovative Research in Computer and Communication Engineering.

