ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

Secure and Migration Controlled SCADA System Using Mobile Agent

Aswathy A.S., J.Stanly Jaya Prakash

Abstract— Supervisory Control and Data Acquisition (SCADA) systems are established as means to exercise supervisory control over an industrial process. It describes how Software Agents technology can be used to increase the reliability of a control system. Software Agent technology can improve SCADA systems as it allows distribution, which inherently promotes redundancy, and modularity, which promotes versatility. Securing SCADA systems is a critical aspect of industrial systems. Industrial systems have installations which actively using the public network in order to provide new features and services which make the system unsecured .By introducing a filtering system ,we can analyze the critical state of the system which can be monitored and secure SCADA network protocols. But in this approach, there is no mathematical method for calculating filter parameters for DDOS, R2L, U2R attacks. In this project, we present a new Secured Agent Based System based approach for calculating those parameters to make the system more secure.

***Index Terms*— Distributed generators, mobile agent, Supervisory control and Data Acquisition (SCADA) systems.**

I. INTRODUCTION

SCADA[1] is a type of industrial control system that are controlled by a computer which monitor and control large distances. These processes include industrial, infrastructure, and facility-based processes. Industrial processes include those of manufacturing, production, power generation, fabrication, and refining, and may run in continuous, batch, repetitive, or discrete modes. Industrial processes that exist in the physical world. SCADA systems[2] are different from other ICS systems by being large scale processes that can include multiple sites, Infrastructure processes may be public or private, and include water treatment and distribution, wastewater collection and treatment, oil and gas pipelines, electrical power transmission and distribution, wind farms, civil defense siren systems, and large communication systems. Facility processes occur both in public facilities and private ones, including buildings, airports, ships, and space stations. They monitor and control heating, ventilation, and air conditioning systems (HVAC), access, and energy consumption.

The term SCADA usually refers to centralized systems which monitor and control entire sites, or complexes of systems spread out over large areas (anything from an industrial plant to a nation). Most control actions are performed automatically by RTUs or by PLCs. Host control functions are usually restricted to basic overriding or supervisory level intervention. For example, a PLC may control the flow of cooling water through part of an industrial process, but the SCADA system may allow operators to change the set points for the flow, and enable alarm conditions, such as loss of flow and high temperature, to be displayed and recorded. The feedback control loop passes. In the last few years, distributed computation has been the topic of many dissertations and research papers. The interest in distributed computation has resulted in a number of enhancements in networking, leading to the construction of internetworking protocols and the creation of a large geographic network (Internet) to implement them. This paper presents how the development of a directory service protocol using Software Agent technology increased the theoretical reliability of Supervisory Control and Data Acquisition Systems (SCADA). Software agents are autonomous program units supported by an execution environment; software agents can use the network to send themselves to other processors, thus “moving” among computers. Moreover, software agents can talk to each other on the network. This allows them to be used as elementary building blocks for complex systems.

Control systems[3] can benefit from agent technology in many ways. First of all, modularity is the key requirement of a control system and a key property of an agent system, so a control system can benefit from the already existing agent structures. Second, autonomy is another characteristic of agents that allows the user to accomplish the tasks in a collaborative manner with the computer. Thus, instead of exercising complete control and taking responsibility for every move the computer makes, people can engage in a cooperative process in which both human and computer agents initiate communication, monitor events, and perform tasks to meet a user’s goals Third, through

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

pro-activity agents take initiative and change their environment. For example, once an agent has completed its task on a machine or is unable to do it, the agent migrates.

A. Reliability analysis

The reliability of a SCADA system depends on its components. It will be assumed that n different components are needed to make the system work. Moreover, each component's reliability can be considered independent of every other's. A common reliability model for this kind of system is based on the joint distribution of n random variables. Since they are independent, their probability distribution can be calculated as the integral of the combination of the individual distribution functions. If the system fails if any one of the components fails, then its reliability is given by the following expression:

Rz[t < t0] = 1 – Fz[t < t0] (1)

Where Fz[t < t0] is the probability that the system fails before the time t0. Furthermore, since this probability results from a series of independent probabilities, it is admissible to write:

Fz[t < t0] = fx1[t < t0]ñ ñ ñ fxn[t < t0] (2)

In the most trivial case, one can assume a binomial distribution for the cumulative probability of failure at any given time, thus the probability of failure will be higher than the probability of any one system failing.

This is a direct derivation of the fact that the components are connected in series. Moreover, any redundancy can be translated in a parallel for each component, thus reducing the probability of failure for the component (ideally halving it) but unfortunately any such redundancy can only be local. So to make the full system redundant it would be necessary to double the amount of components mount of components. Of course, the ideal solution to this problem would be a "joker" component, capable of assuming the form of any other component, and thus effectively being able to replace any of them. This would double the system's reliability with the minimum expense. This project shows how an agent-based architecture succeeds in this task

B. SCADA infrastructure

A conventional SCADA system is based upon a collection of functional block that realize dedicated functionality such as data acquisition, decision support, database updates, etc .This realization of a SCADA system suffers from the shortcomings presented in the previous section, as its reliability is fundamentally connected. to the well being of every component it is comprised of. Although for low level components (e.g., PLCs) it might be possible to conceive redundancies, it is costly to replace dedicated servers as each needs different treatments.

II. PROPOSED SYSTEM

The mobile agent [4] has been seen as a promising distributed computing technology. The mobility characteristic of mobile agent makes it to travel often in open network. In this scenario, it is obvious that the mobile agents are vulnerable to various security threats. Protecting free-roaming mobile agents from malicious host and from other mobile agents has drawn much attention in recent years. Most of the existing work deals with the protection of mobile agent from malicious host or protecting the host from malicious agents. The issue of protecting a mobile agent from a malicious agent in the host's agent execution environment is not given much attention. Further, for applications like secure data transaction, it is an essential criterion to protect the retrieved data and the address of dynamically selected remote servers. This paper provides an environment that protects the legitimate mobile agent from the malicious mobile agent that performs passive attacks like eavesdropping. To ensure the confidentiality of the data that are retrieved from each remote server a serial encryption technique is implemented. The remote server's address that is selected dynamically is also secured to face the possible eavesdropping threat. The three dimensional security scheme presented in this project for free-roaming mobile agent addresses the code, data and itinerary security issues.

A.THE MODEL

Applying an agent execution environment (AEE) to real-time distributed control systems means to develop a new protocol for a distributed control system that utilizes mobile agents. An agent execution environment is a software system that provides a runtime environment for agents to execute, a standard interface for interactions, services for creation, migration and termination of mobile agents, supports agent mobility and communication while providing security for both hosts and agents.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

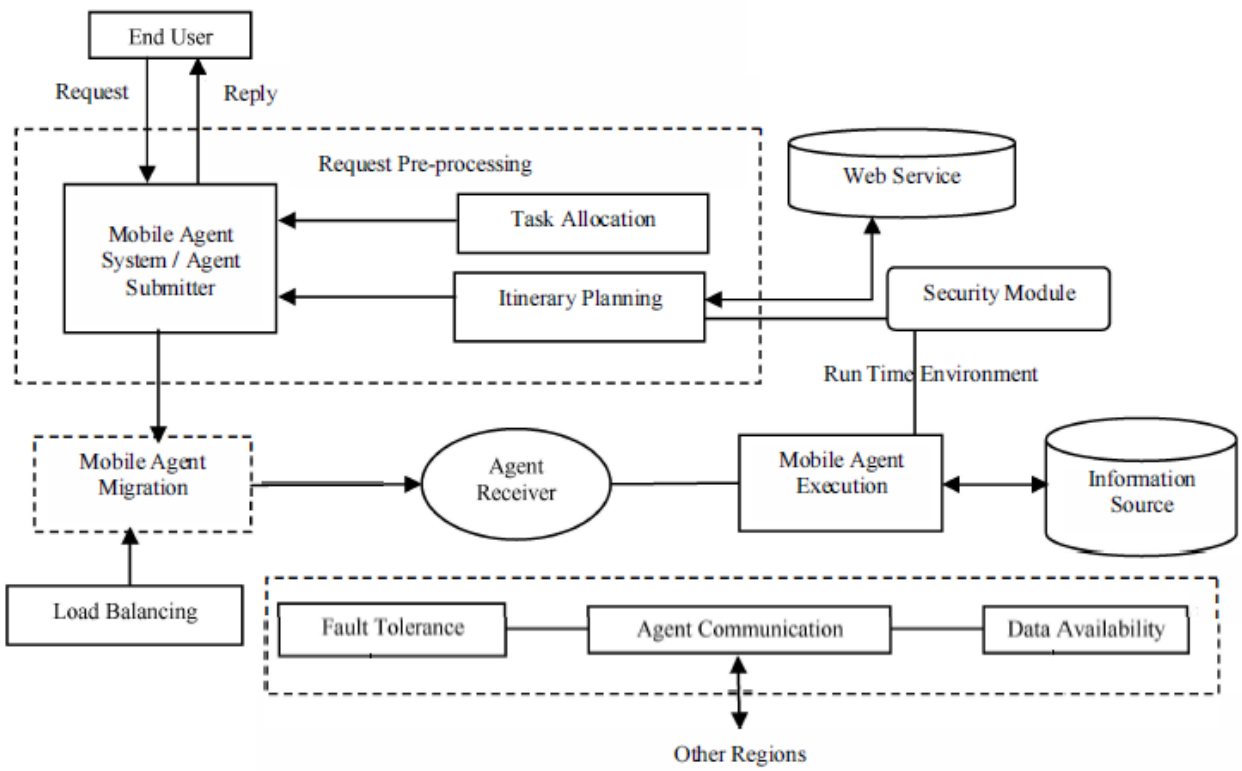

Fig 1: System model

The main purpose of the AEE is to host agents and allow them to run. Primarily, an AEE is composed of two main types of agents: system agents and user defined agents. System agents are created by the AEE and are used to help the AEE operate through the use of their services. On the other hand, user defined agents are agents which use the services of the AEE in order to operate. A mobile agent system consists of one of more AEE linked together. An agent execution environment will consist of many services that facilitate and control mobile agents. The application of an agent-based infrastructure and architecture to a SCADA system is relatively straight forward in principle. However, some design issues need to be addressed at the very beginning of the characterization. First of all, it is necessary to identify the variation points of the SCADA system architecture, given an agent-based environment. There are three main features of an agent-based system that are advantageous for a SCADA system: modularity, communications, and mobility. The protocol of the directory service system is mainly used to locate agents in the linked agent execution environments, or to give advice for the existing agents to other agents. There are two types of directory service: the local directory service agent and the central directory service agent. Each directory service is given a unique name in the agent execution environment during its initialization. This ensures location-transparent names at the application level. Thus, every service knows how to find the directory service itself.

On the other hand, agents must be identified uniquely in the environment in which they operate. Proper identification allows control, communication, cooperation, and coordination of agents to take place. A SCADA system takes advantage of many features provided by an agent execution environment. Above all, the most important feature is location independence. With it, an agent is no longer limited on a single machine, but it can be accessed wherever it is located. This makes it possible to implement a failsafe mechanism, in which multiple copies of an agent may be activated in different locations, thus effectively improving the reliability of the system. Another important advantage of agent systems is platform independence. While this comes at a cost (mostly speed) it is of great help for components that now can work on any workstation equipped with the agent execution environment.

Implementation The general architecture of a typical mobile agent system (also referred as agent submitter) for an information retrieval application with fault tolerant and security model is shown in figure 1. Request preprocessing module process the request of client to define the state of mobile agent before migration.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

Run time environment module composes the actual execution of mobile agent in remote servers. Itinerary planning module decides the nature of itinerary for mobile agents. Web service is requested to plan and select dynamically, the next host to visit, based on parameters like size of data, aliveness, and width. Security module provides security for the dynamically selected itinerary and more importantly the mobile code and the state of the data collected from each server. From figure 1, the end user / client submits his request, for example, an information retrieval request, to the mobile agent system. If the number of hosts to be visited is very large, the hosts are divided into regions, based on its geographical location. This process is followed by dispatching agents into network to perform its task. During run time, the next host to visit is selected dynamically. On arrival at each host, the mobile agent undergoes security check and on success the mobile agent is allowed to execute in the environment.

B. Mobile Agent Security for Code, Data and itinerary (MASe-CDI)

The scenario that indicates the possible threat to mobile agent code and corresponding necessity of protection is discussed in this section. In a secure data transaction scenario, the legitimate mobile agent, on behalf of a user, carries the secret code and corresponding information within it. The attacker desires to access or tap some part of code from the executing host to his server. So the attacker designs a mobile agent which would append with the legitimate mobile agent without disturbing the internal functionalities or the data within the agent. The new look mobile agent reaches the entrance of a server. The malicious mobile agent, as soon as it enters the system detaches itself from the legitimate mobile agent and starts functioning autonomously and gains access over the system. Further, it starts performing its malicious functionalities in the system which may affect the normal operation of the entire system. The category of this threat is malicious mobile agent attacking other agents and agent platform.

C. Mobile Agent Security for Code Integrity (MASe-C)

The system architecture that reflects the functionalities in presenting the code security for mobile agent. The major components involved in the architecture are agent observer, critical section, agent verification, agent sizing and agent execution. The agent observer is a server that is distributed over different regions and its function is to monitor the movement of mobile agent in its corresponding region. It also acts as a service provider that services the requests related to those mobile agents. The agents generated at each host must be registered with agent observer and only the registered mobile agents are allowed to access the critical section. The security of the agent observer is assumed. The critical section in MASe-C model contains a buffer of definite size in which the mobile agent undergoes agent verification process by the host. The verification module performs verification checking at the external as well as internal entries of the critical section. Agent sizing is a process of sizing the agent to match receiver's buffer size. On clearing the agent size verification level, the mobile agent is permitted to execute in its environment; otherwise the mobile agent is disposed by default.

In agent execution environment, the mobile agent is executing its task. On successful completion of the task, the mobile agent is dispatched to next host after finding the proper host to migrate. The client creates a mobile agent with the code it needs to carry. Before migrating to next server, the current host dispatches a query to agent observer, requesting the list of servers that holds related information, bandwidth, corresponding buffer size and public keys. On receipt of the reply, the current host selects the optimal server to visit based on given parameters. The size of the mobile agent along with the data is expected to match the size of buffer in critical section of the receiver. If the size exceeds, the data is further segmented in order to make it attains the required size. Further, to carry these additional partitions, temporary mobile agents are created by the sender. Now the client registers the created mobile agent along with the identification of agent, its size and sender's identification with agent observer. The agent observer acts as a lookup record which could be referred by any platform for agent verification which has registered with it. The MA reaches the receiver and migrates to the entry point of critical section. At this point external level verification will be done with the help of agent observer. Agent observer is requested to verify the incoming mobile agent and in turn it replies with the matching result. If the mobile agent does not pass this authentication check, it is not allowed inside the critical section; otherwise it is allowed for internal authentication verification process. The entry point is closed as soon as the mobile agent enters the critical section. No other mobile agent is allowed to enter the critical section unless the verification process of the agent available inside the critical section is completed and the mobile agent is let inside the platform. Internal level verification is done with the help of agent sizing. If the size of mobile along with the data is larger than the critical section buffer, it is expelled out of the system. The mobile agent is kept in 'deactivated' state while this check is performed. This restricts the mobile agent not to perform any malicious action inside the host and thus protects the host platform

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

from malicious activities of mobile agent. In case, the mobile agent fails this test, it is killed immediately inside the critical section itself. Further, at this stage only, the append entry attack is detected. If there is an increase in size of the agent against the size fixed at the sender's side, illegal entry of mobile code is detected. On detecting this attack, the mobile agent is destroyed immediately based on malicious reasons. Otherwise the mobile agent is allowed to execute in its execution environment. Prior to the execution state, the agent is reorganized if it is already partitioned. Once all the data is reorganized, the data collected from current host is integrated with the existing data and appended to the actual mobile agent. Once the task is accomplished and the result obtained, the mobile agent repeat the process that starts with requesting agent observer for next server to migrate.

D. Mobile Agent Security for Data integrity (MASe-D)

The general idea behind this protection architecture is to construct a series of packets that contains the encrypted data retrieved from each server. The data collected from one server must not be known to others. To implement this constraint, every server will encrypt its data using the owner's public key. The owner's public key is obtained by decrypting the segment received from previous server using receiver's private key. Thus, each data packet can be encrypted using owner's public key. At the same time, the data packet can be decrypted only by the owner.
Additionally, these packets can only be opened in the correct order, since the symmetric key used to decrypt an envelope is protected inside the previous packet. The scheme that is used to secure the data is given below.

$$E_{KU}^{OWN}\ [\ D_{CURR} // E_{KU}^{OWN}[D_{PREV}\ ]\ ]$$

OWN - Owner of mobile agent; E - Encryption; PREY - All previous hosts; CURR - Current host; D – Data collected; KU - Public Key. The data packet received by any receiver is encrypted using owner's public and takes the form,

$$E_{KU}^{OWN}[D_{PREV}\ ].$$

On successful completion of task, the mobile agent appends the result with this packet and the new look packet is encrypted again using owner's public key.

F. Mobile Agent Security for itinerary Confidentiality (MASe-i)

The itinerary of mobile agent is protected to save it from eavesdropping attack. For secure data transaction application, the address of the host, the mobile agent is going to migrate, must be protected. In addition, the receiver must be ensured that the mobile agent is meant for intended receiver only. For confidentiality, sender's private key is used. Before dispatching a mobile agent, the sender protects the address of the receiver using its private key and the scenario takes the form,

$$E_{KR}^{SEN}[\ Addr_{REC},\ ID_A,\ KU^{OWN}]$$

Addr - Address; iDA - Agent identification; SEN - Sender; REC - Receiver. The receiver decrypts this packet using sender's public key. Sender's public key can be obtained by requesting corresponding web service. On decryption, the details revealed are, agent identification that can be verified through agent observer, owner's public key that is used to encrypt the data and receiver's address to confirm the correct receiver. To ensure the confidentiality between the sender and receiver, the entire details that holds the encrypted data and itinerary is encrypted using receiver's public key that can be obtained from agent observer request. In overall, this packet is decrypted only be intended receiver using its private key. On decryption, the receiver will get two parts; encrypted data and encrypted itinerary. The encrypted itinerary is decrypted by sender's public key. On decryption the receiver can verify its address and agent and get the owner's public key. This owner's public key is used to encrypt the data forwarded by the current host. The possible migration from a sender to receiver is denoted by,

$$SEN \rightarrow REC : E_{KU}^{REC}\left\{E_{KU}^{OWN}\left\{D_{CURR} // E_{KU}^{OWN}[D_{PREV}]\right\} // E_{KR}^{SEN}[AgentID,\ KU^{OWN},\ Addr_{REC}]\right\}$$

In overall, this packet is decrypted only be intended receiver using its private key. On decryption, the receiver will get two parts; encrypted data and encrypted itinerary. The encrypted itinerary is decrypted by sender's public key.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

On decryption the receiver can verify its address and agent and get the owner's public key. This owner's public key is used to encrypt the data forwarded by the current host.

E. Advanced Encryption Standard (AES)

The paper illustrates a novel distributed protocol for multi agent environments to improve the communication security in packet-switched networks. To enrich the overall system security the approach makes use of distribution and double encryption and some other traditional methods such as digital signature. In this approach the encrypted private key and the message are broken into different parts carrying by different agents which makes it difficult for malicious entities to mine the private key for message encryption, while the private key for the encrypted key is allocated on the predetermined destination nodes**.**

III. DISCUSSION

In the last few years, distributed computation has been the topic of many dissertations and research papers. The interest in distributed computation has resulted in a number of enhancements in networking, leading to the construction of internetworking protocols and the creation of a large geographic network (Internet) to implement them. This paper presents how the development of a directory service protocol using Software Agent technology increased the theoretical reliability of Supervisory Control and Data Acquisition Systems (SCADA). Software agents are autonomous program units supported by an execution environment; software agents can use the network to send themselves to other processors, thus "moving" among computers. Moreover, software agents can talk to each other on the network. This allows them to be used as elementary building blocks for complex systems.

IV. CONCLUSION

A SCADA system for the ADAPS using mobile agents and a wide-area Ethernet has been proposed. It could be made economical and flexible due to the characteristics of these two technologies. In addition, the agents have QoS control methods; via which processing is scheduled according to three priorities, and communication congestion would be avoided. Two types of protocols for agent migration are provided in the SCADA system, one of which iteratively sends a datagram packet to a destination to contribute to real-time and reliable processing in this system. Conversely, the wide-area Ethernet is composed of ADS and PON. The Ethernet-ADS provides RSTP that is capable of establishing an alternative route in case of communication failure. We performed experiments to evaluate the real-time performance and reliability of the SCADA system we propose. The system completed earth fault protection within the required time (1 s) in cases where an alternative communication route was established within 550 ms by RSTP in the Ethernet-ADS. We considered the feasibility of the SCADA system based on the experimental results and state-of-the-art communication technologies. In terms of real-time performance with fault tolerance, the current RSTP performance should be rapid enough to complete earth fault protection within the required time. Also boosting the feasibility of the SCADA system is the use of the real-time specification for Java in the implementation of mobile agents. The expandability of the SCADA system should be ensured by multicast used in the iterative transmission protocol and the roundtrip of rapid and best-effort agents. According to these results and considerations, we conclude that the SCADA system we propose should satisfy the requirements.

REFERENCES

[1] H. Kobayashi and M. Takasaki, "Demonstration study of autonomous demand area power system," in Proc. IEEE Power Eng. Soc. Transm.Distrib. Conf. Expo., Aug. 2006, pp. 548–555.

[2] N. Okada, "Development of loop power flow controller," presented at the Inst. Elect, Eng. Jpn. ICEMS, Nagasaki, Japan, Nov. 2006.

[3] M. Asari, "Ancillary service for low voltage distribution network with demand supply interface," CRIEPI Rep. R04021, 2005.

[4] C. H. Hauser, D. E. Bakken, and A. Bose, "A failure to communicate," IEEE Power Energy Mag., vol. 3, no. 2, pp. 47–55, Mar./Apr. 2005.

[5] S. K. Mazumber, K. Acharya, and M. Tahir, "Towards realization of a control-communication framework for interactive power networks," in Proc. IEEE Energy 2030 Conf., 2008, pp. 1–8.

[6] B. N. Ha, S. W. Lee, C. H. Shin, S. C. Kwon, S. Y. Park, and M. H. Park, "Development of intelligent distribution automation system," in Proc. IEEE T&D Asia, 2009, pp. 1–4.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

[7] C. Lin, H. Chuang, C. Chen, C. Li, and C. Ho, "Fault detection, isolation and restoration using a multiagent-based distribution automation system," in Proc. IEEE ICIEA, 2009, pp. 2528–2533.

[8] S. Mohagheghi, M. Mousavi, J. Stoupis, and Z. Wang, "Modeling distribution automation system components using IEC 61850," in Proc. IEEE Power Energy Soc. Gen. Meeting, 2009, pp. 1–6.

[9] S. Mohagheghi, J. Tournier, J. Stoupis, L. Guise, T. Coste, C. A. Andersen, and J. Dall, "Applications of IEC 61850 in distribution automation," in Proc. Power Eng. Soc. PSCE , 2011, pp. 1–9.

[10] S. A. M. Javadian and M.-R. Haghifam, "Protection of distribution networks in presence of dg using distribution automation system capabilities," in Proc. IEEE Power Energy Soc. Gen. Meeting, 2008, pp. 1–6.

[11] T. Kasajima, R. Endo, Y. Wada, Y. Kudo, and H. Kanawa, "The development of the advanced distribution automation system with optical fiber network of Tokyo Electric Power Co., Inc.," in Proc. Power Eng Soc. Gen. Meeting, 2004, pp. 1441–1445.

[12] A. Vojdani, "Smart integration," IEEE Power Energy Mag., vol. 6, pp. 71–79, Nov./Dec. 2008.

[13] Communication Networks and Systems for Power Utility Automation— Part 7-4: Basic Communication Structure—Compatible Logical Node Classes and Data Object Classes, IEC 61850-7,

[14] Communication networks and systems for power utility automation— Part 7-2: Basic information and communication structure—Abstract communication service interface (ACSI), IEC 61850-7-2 Ed.2, IEC, 2010.

[15] D. Lange and M. Oshima, "Seven good reasons for mobile agents," Commun. ACM, vol. 42, no. 3, pp. 88–89, 1999.

[16] T. Otani and H. Kobayashi, "A SCADA system using mobile agents for a next-generation distribution system," in Proc. IEEE PowerTech, 2009, pp. 1–8.

[17] T. Otani, "System configuration and QoS control of mobile agents for a next-generation of distribution system," Inst. Elect. Eng. Jpn. Trans.Electron., Inf. Syst., vol. 124-C, no. 3, pp. 921–928, 2004.

[18] "OMG Unified Modeling Language (OMG UML) Superstructure,"ver. V 2.1.2, Object Manage. Group, 2007.

[19] J. Postel, User datagram protocol, IETF STD0006, 1980.

[20] M. Allman, V. Paxson, andW. Stevens, TCP congestion control, IETF RFC2581, 1999.

[21] L. Monostori1,2 (1), J. Váncza1,2 (2), S.R.T. Kumara3,(2006)," Agent-Based Systems for Manufacturing 3The Pennsylvania State University, Industrial and Manufacturing Engineering, University Park, PA 16802, USA.

[22] Ibharalu Friday Thomas, Sofoluwe Adetokunbo Babatunde, Akinwale Adio Taofiki ,(2011), "A Reliable Protection Architecture for Mobile Agents in Open Network Systems". International Journal of Computer Applications (0975 – 8887) Volume 17– No.7.

[23] Tarig Mohamed Ahmed,(2009), "Using Secure-Image Mechanism to Protect Mobile Agent against malicious Hosts", World Academy of Science, Engineering and Technology 35'.

[24] Qiuming zhu,(2005),"Topologies of agents interactions in knowledge intensive multiagent systems for networked information services", Department of computer science, University of Nebraska at Omaha,Omaha,NE68182,USA.

[25] Xiaolong Jin, Jiming Liu,(2007)," Mobile agent for Adaptive routing", Department of Computer Science Hong Kong Baptist University.

[26] Christian Cachin, Jan Camenisch, Gu¨nter Karjoth,(2001)," Securing mobile agent and its platform from passive attack of malicious mobile agent", IBM Research Zurich Research Laboratory CH-8803 Ru¨schlikon, Switzerland.

AUTHOR BIOGRAPHY

Aswathy A.S, IInd year ME, Department of Computer Science, Mahindra Institute of Technology, Tiruchengode, Tamil Nadu, India.

J.Stanly Jayaprakash, Assistant Professor, Department of Computer Science, Mahindra Institute of Technology, Tiruchengode, Tamil Nadu, India.