SIMULATION RESULTS FOR SECURE AUTOMATIC ENERGY METER READING USING MOBILE AGENTS. PART 2

RADWAN TAHBOUB, RODICA CONSTANTINESCU, VASILE LĂZĂRESCU, CONSTANTIN RĂDOI

Key words: Automatic Meter Reading (AMR), Mobile agents, Client server, JADE, OPNET.

Intelligent and secure automatic energy meters reading and management using Mobile agents can be of great importance for municipalities and energy distribution companies so as to efficiently read these meters and to decrease the number of traditional visits required by the energy company, and decreasing the number of employees used in performing this traditional time consuming and high cost work [1, 3, 5]. In this work we will start by reviewing the current technologies and techniques used in securely handling remote meter reading and management using mobile agents. A more detailed simulation results for different configurations and techniques of secure automatic energy meter reading and management systems using mobile agents will be presented and compared to traditional client-server techniques. These results will be presented in two parts, the first part consisted of OPNET simulation results for our system, and this part will present the results of the secure system using java agent development environment (JADE).

1. **INTRODUCTION**

As we have presented in part one of this article: Mobile agents (MA) consists of a self-contained piece of software that can migrate and execute on different machines in a dynamic networked environment, and acts autonomously and proactively in this environment to realize a set of goals or tasks as described in [9]. Mobile agents are commonly used in distributed information-retrieval applications. Mobile agent automatic meter reader and management (MAMR) is a network-based application that can be used by electric utilities to increase performance and reliability of meter readings process [1, 3]. There exist many security issues that need to be addressed to protect the MAMR [2]. The trusted next move protocol (TNMP) can be used for protecting a MAMR who collects data on behalf of its originator, the energy company, as well as protecting the power meter platform

"Politehnica" University of Bucharest, E-mail: Radwan Tahboub: radwanrt@yahoo.com

(PMP) code from a malicious MAMR. In this work (part two), implementation and simulation details will be presented to show the workability of such approach.

2. **BACKGROUND**

There are many ways of implementing the algorithms used in fulfilling these security services. In general operating systems perform software functions and procedures to execute such security algorithms. Hardware related implementations are much more efficient in terms of speed and memory usage and usually supersedes the software implementations of these algorithms [4 – 7]. Some other systems use a combination of both hardware and software implementations to achieve the required functionality. Field programmable arrays (FPGA) implementations of ciphers protocols may show a throughput increase for real-time applications [10, 11].

3. **SECURE MAMR USING TRUSTED NEXT MOVE PROTOCOL (TNMP)**

The already existing MA data collection protocols solves problems like protecting data or agent code and most of them wait until the end to discover if the agent or data has been altered or tampered with, others discover this earlier but all do destroy both the data and MA when they discover this tampering. This may cause low data collection utilization since the collected data will be rejected even if 1 bit of huge collected data is tampered and a new MA is sent again to collect the data, which may return back with the same problem. Our protocol depends basically on not moving the MAMR to another host unless it is sure that the new host is not malicious and is a trusted one [7–9]. This is achieved by sending an inspection agent IMA to check the status of the next Power Meter Platform, PMP, before moving to it [1, 2, 4]. Since this step is taken from a current MAMR running on a trusted host, it will not depend on a third party check and this decision is taken by the collecting MAMR before moving to the next PMP. We applied the TNMP to the Power Meter Readings collection Problem using MAMR running at the current trusted Power meter PMP_i and checking the next PMP_{i+1} for safety conditions before moving to it.

3.1. MAMR SECURITY ISSUES

There are many security issues that should be considered in designing MAMR security protocols [6–9]. The main threats in such a system consist of the power meter platform (PMP) to MAMR threat, the MAMR to PMP threat and the MAMR to MAMR threat. The main security tools used in a PMP and MAMR

system are encryption, signatures, and hashing. The system will use both asymmetric public key infrastructure and a symmetric private key infrastructure. These security tools will be used to attain confidentiality, integrity, availability and accountability (logging) of the power meter readings collection process.

3.2. BASIC TNMP SYSTEM STRUCTURE

For the purpose of secure data collection, we assume that the power company facility is by itself trusted, secure and protected. And it is responsible for creation of the MAMR, and IMA agents and receiving the MAMR with the final collected encrypted data (Fig. 1). This facility is also responsible of certifying each power meter and the server software installed on each power meter and issues a certificate for each meter using the power company's private key and storing this certificate in the FPGA card attached to each power meter.

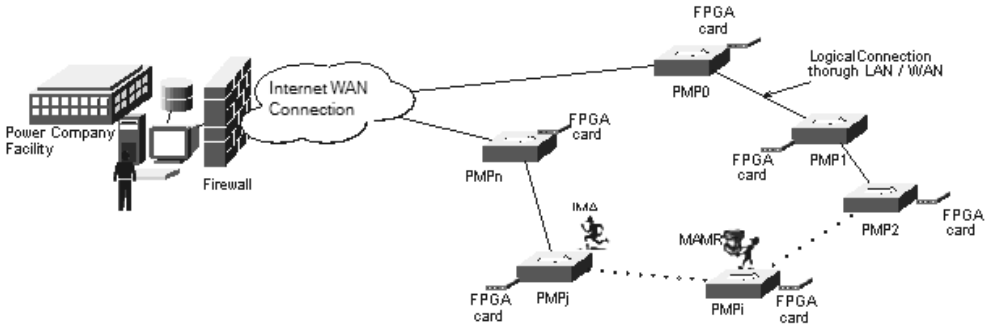

Fig. 1 – Secure collection PMP, MAMR and IMA.

Each power meter platform PMP_i contains the following components as shown in Fig. 2. The FPGA card forms an interface point that is capable of executing all required security and compression functions like encryption, decryption, hashing, digital signing and random number generation. It also stores a certificate $cert_i$ that is generated by the power company facility. This certificate includes the power meter identity and the hashing of the basic software functions code of each PMP server. That is the server code in each PMP is hashed and signed using the power company private key.

$$cert_i = S_{e_{pc}}\left(H\left(Code_{PMP_i}, ID_{PMP_i}\right)\right). \tag{1}$$

$S_{e_{pc}}$ is a signing function computed using the private key of the power company facility. $H()$ is a collision free one-way hashing function, $Code_{PMP_i}$ is the

basic code of the functions stored at the power meter platform with identity ID_{PMP_i}.

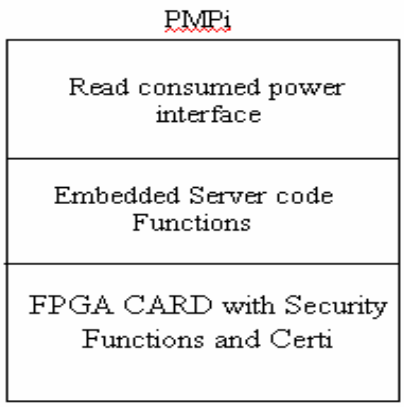

Fig. 2 – PMP components

3.3. MAMR STRUCTURE

MAMR is used in collecting the power meter readings from each trusted PMP. The agent can be represented as a $(C_{MAMR}, R, cert_{MAMR})$, where R is split into platform-specific data chunks mainly composed of the power meter readings:

$$\left.\begin{aligned} cert_{MAMR} &= S_{e_{pc}}\left(H\left(C_{MAMR}\right)\right), \\ R &= R_0,...,R_n; \\ R_i &= E_{d_{pc}}\left(r_i, T_i, ID_{PMP_i}, S_{e_{session}}\left(H\left(r_i, T_i, ID_{PMP_i}\right)\right)\right) \end{aligned}\right\}, \tag{2}$$

where $cert_{MAMR}$ denotes the MAMR code certificate signed by the power company private key e_{pc} since it is the creator of this MAMR. $S_{e_{session}}\left(H\left(r_i, T_i, ID_{PMP_i}\right)\right)$ denotes the meter-reading certificate signed by the session private key together with the meter identification number and the time stamp T_i at which the reading was taken. This session private and public keys pair $(S_{e_{session}}, S_{d_{session}})$ will be generated before moving the MAMR to the next meter as will be shown soon.

It is clear that IMA and MAMR are capable of executing any of the library functions stored in the FPGA card which is a secure hardware electronic library.

3.4. INSPECTION MA STRUCTURE (IMA)

IMA is a mobile agent used in inspecting the next PMP before moving the MAMR to it. IMA can be represented as (C_{IMA}, D_{IMA}, $cert_{IMA}$, $d_{session}$), where $cert_{IMA}$ is the IMA certificate signed by the power company private key e_{pc}. D_{IMA} are the encrypted session parameters using the session private key. And ($S_{e\ session}$, $S_{d\ session}$) are the session private and public keys pair that will be generated before moving the MAMR to the next meter PMP, That is:

$$\left.\begin{aligned} cert_{IMA} &= S_{e_{pc}}\left(H\left(C_{IMA}\right)\right), \\ D_{IMA} &= E_{e_{session}}\left(T_i, ID_{PMP_i}, S_{e_{session}}\left(H\left(T_{session}, ID_{PMP_i}\right)\right)\right) \end{aligned}\right\}. \tag{3}$$

In [1, 2], we presented a detailed mathematical model for each part or component of the TNMP using all required data encryption, digital signature functiones, hashing, and other security related functions. When the MAMR finishes collecting the reading and management processes it returns back to the power company facility with a set of collected readings encrypted using the Power Company's public key. The power company is the only one who can get this data after checking the MAMR authenticity and integrity. Also it gets the report of malfunctioning PMPs and takes the required actions to repair and recertify them.

4. **SECURE AMR AND MAMR NETWORK SIMULATIONS USING TNMP**

In this subsection, we will present the implementation and simulation results using of a virtual AMR system using JAVA and JADE MA development environments.

4.1. SIMULATION RESULTS USING JAVA AND JADE

JADE/JAVA (Java Agent Development Framework) to simulate the client-server AMR and the MAMR behaviours is used. JADE is a software development framework aimed at developing multi-agent systems and applications conforming to FIPA standards for the agents. It includes two main products: a FIPA-compliant agent platform and a package to develop Java agents [13].

4.2. SYSTEM STRUCTURE FOR AMR SIMULATION

The energy/power company facility consists of a standard database server that is capable of receiving TCP/IP connection from energy meters using HTTP, FTP or E-mail protocols.

The energy/power meter platforms (PMPs) are modelled using normal computers running a process that generates energy/power load consumption and stores it in a local database. A timer is designed to start sending the stored data (adjustable parameter). While sending its data, a client (meter) opens a socket with the energy company server and sends the readings. The energy company receives all connections simultaneously by opening a thread for each connection and stores the received data in its database management system. A JAVA application program is built to control all the energy meters and the power company activities. Fig. 3 shows the system data flow for this program. The resulting system is tested in a computer lab with 12 computers and the systems worked 100% correctly. Fig. 5 shows the network activity shots taken from the network task manager during the AMR reading processes.

4.3. SYSTEM STRUCTURE FOR MAMR SIMULATION

The energy/power company facility consists of a standard database server that is capable of storing the meters readings collected by a MAMR. The MAMR instant is initiated by the energy/power company platform which consists of a JADE environment installed on one of the companies' computers (Fig. 4).

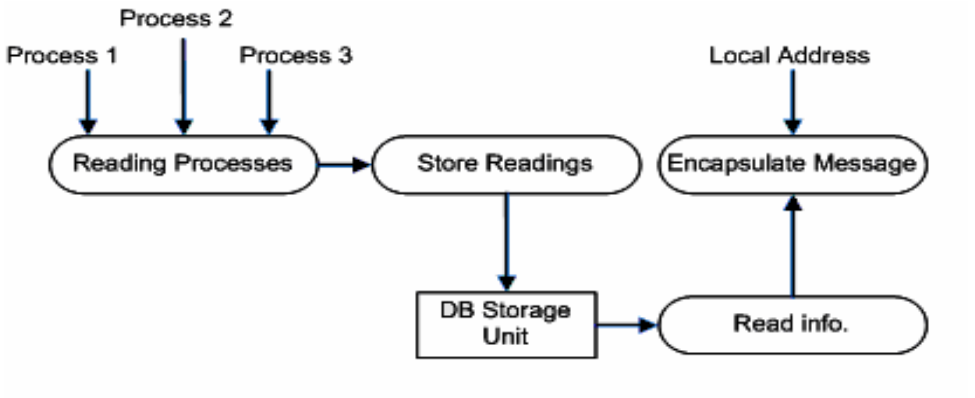

Fig. 3 – AMR system data flow.

The energy/power meter platform PMP is assumed to be a power meter running an agent platform that is capable of hosting MAMRs. The energy/power meters are simulated using computers running only java run time machine (JVM) with a small library of functions that can be installed on each computer [1, 13]. The size of the JVM and the JAVA library does not exceed 1 Mbyte which is acceptable for practical purposes if the energy meters will be running such

environment in the future. Each energy meter is running a process that simulates reading a multiple power load and storing the consumption value of each load in a local database. The computer simulating the energy company creates the MAMR and sends it to the first power meter; the MAMR reads the local database and moves to the next computer and so on. The list of computers that should be visited are stored at the energy company database, in the future and when simulating intelligent agents this list should be created dynamically by each MAMR according to a certain planning criterion. The last computer send back the MAMR to the energy company computer with all the collected readings from each meter and hence the MAMR updates the energy company database with the read data through its tour. Fig. 7 shows the MAMR basic flowchart design [1].

The testing environment consisted of a computer lab with 12 computers using one of them as the energy company platform and the other 11 computers as the energy meters to be visited. A JAVA application program through the JADE development environment is built to control all the energy meters activities, and the systems worked 100% correctly. Fig. 6 shows the network activity shots taken from the network task manager during the MAMR reading processes.

The stored data of both AMR and MAMR methods were compared and checked for 100% correctness. In the future we will test this system with the company platform being in another network using a routed WAN link while increasing the number of computers representing the energy meters to be in different labs and networks. Hence performance parameters can be tested and measured for this enlarged environment. This is very difficult to arrange currently since it is very difficult to arrange for large system while the university labs and networks are in use.

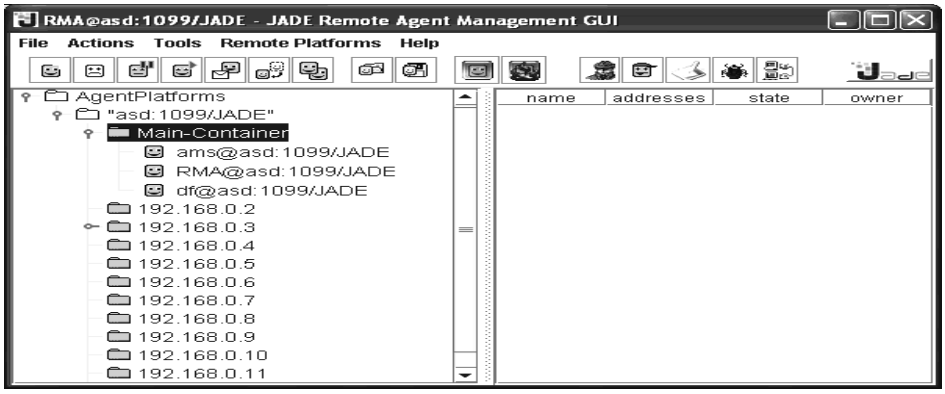

Fig. 4 – JADE Environment used in moving the MAMR and IMA.

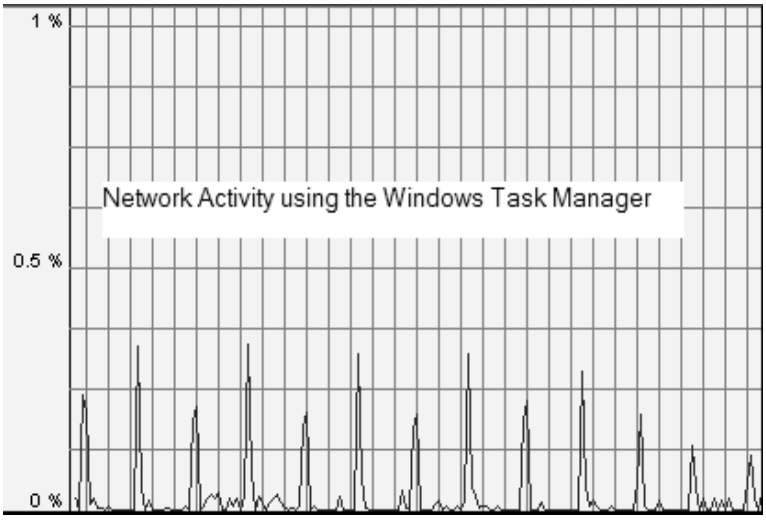

Fig. 5 – Network activity of AMR process.

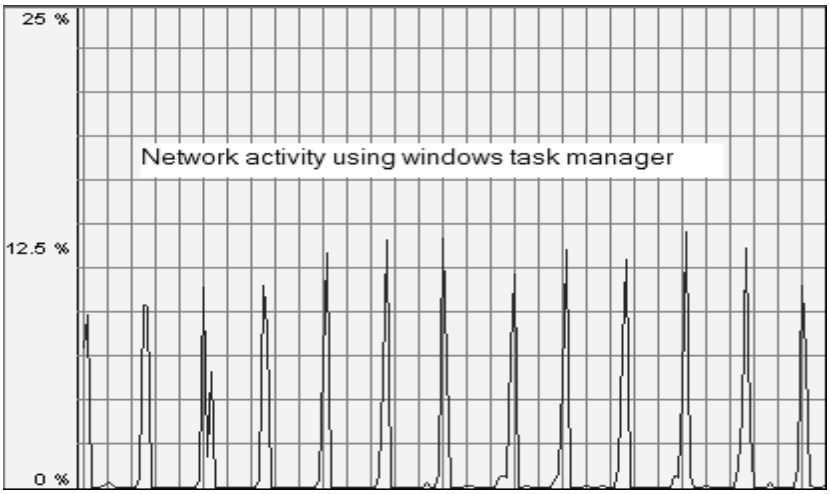

Fig. 6 – Network activity of MAMR process.

4.4. TNMP IMPLEMENTATION USING JADE/JAVA

In this simulation we used the testing environment described in the previous section. PMPs are simulated using computers running only Java Run Time Machine (JVM) with a library of required security functions that is installed on each computer. This library is assumed to be secure and can't be changed during

running the protocol to simulate the FPGA functionality. Also we assume that each PMP stores a digital certificate that identifies each computer (power meter) from another (usually this certificate is also stored on the FPGA card).

5. **DISCUSSIONS AND CONCLUSIONS**

Through the simulation results and different scenarios we have noticed the following observations:

Increasing the number of power meters in a region, leads to high throughput, utilization and queuing delay which requires higher data rates on all links including xDSL and DS1 links. This is true for both wired and wireless energy meter configurations, as shown in Fig. 16 of part one. The MAMR shows better performance parameters (link utilization, queuing delay, and throughput) over the case of client-server in the xDSL link. This is true for both wired and wireless networked energy meters. Results for one or two parts show the superiority of MAMR over client-server. This may not be true if the whole city is connected!

For the Multiple MAMR, we could simulate two MAMRs only and not very clear or minor enhancement in performance parameters was observed over the single MAMR case. This may not be the case for more MAMRs, but more RAM is needed for conducting the simulation for multiple MAMR networks.

For a typical xDSL communication link configuration between power meters and Power Company, we observed that maximum number of energy meters in each site is preferred to be limited to 50 meters; otherwise a higher xDSL bandwidth should be used. Other simulation results and comparisons for both LANs and WLANs were also conducted. Simulating the same system using the JADE environment showed 100% workability of the MAMR/AMR systems.

Results showed the 100% workability of the TNMP protocol in securely collecting the meters readings using the JADE environment. If one of the meters (computers) is missed or disconnected, the IMA skips it and moves to the next meter. If the IMA code is changed from what it should contain, the meter does not authenticate it and kills that IMA directly. When one of the meter parameters is changed, then the IMA/MAMR declares it as not trusted meter and the MAMR does not move to this meter and sends a new IMA to the next meter. If one of the stored certificates characters is changed in a meter, then the IMA/MAMR declares that meter as not trusted meter and the MAMR does not move to this meter and sends a new IMA to the next meter. So in short:

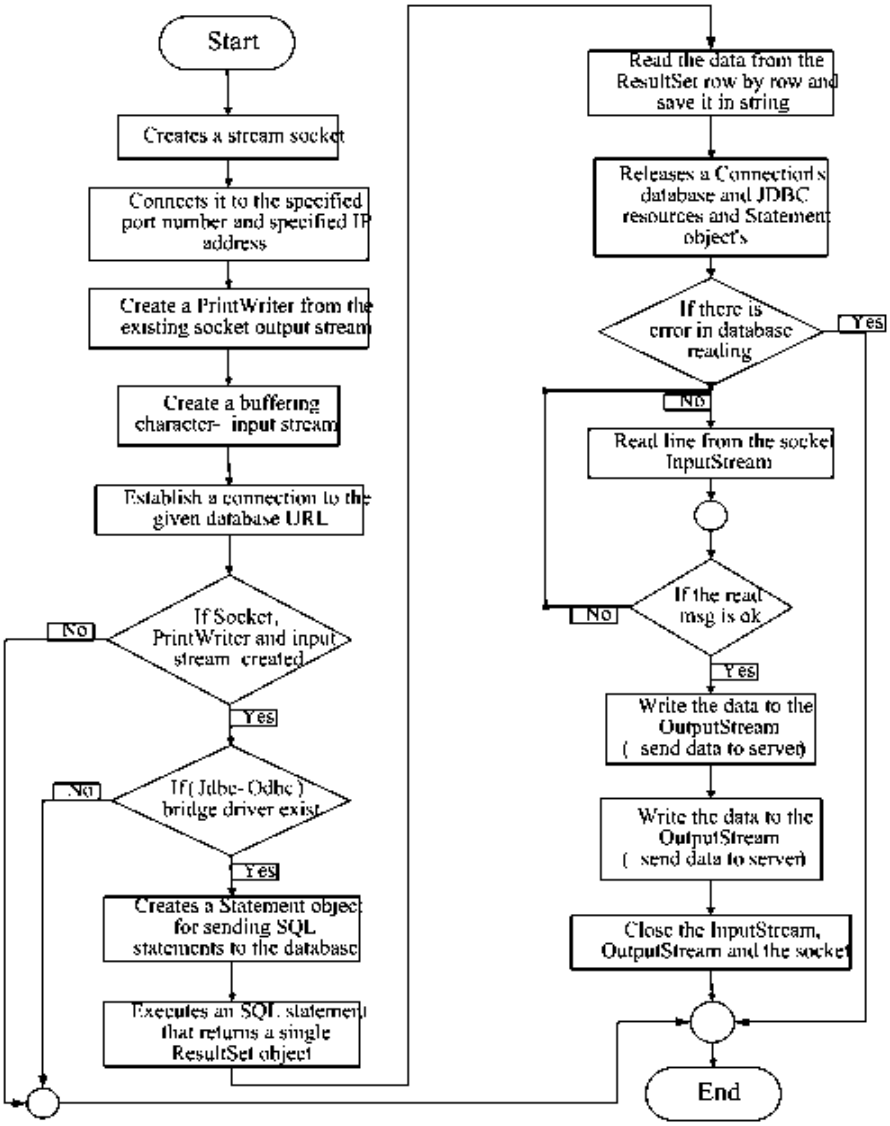

Fig. 7 – Basic flowchart of MAMR activity.

TNMP supports confidentiality, authenticity, data integrity, truncation resilience, and insertion resilience attacks. It is also self protecting, that is code and readings of IMA and MAMR are protected while migration. TNMP collects only trusted data so no need for destruction or ignorance of collected data. And reports of malfunctioning PMPs and/or malicious hosts to be fixed by the power company.

All data transfer between PMPs can be done using VPN, SSL, and IPSEC techniques so attacks can be minimized and detected and collected data readings should be higher than older collected data so proof of not tampering is increasing the trust of read data, hence no need for updates.

Received on 7 March 2007

REFERENCES

1. R. Tahboub, D. Lăzărescu, V. Lăzărescu, *Modeling and Simulation of Secure Automatic Energy Meter Reading and Management Systems using Mobile Agents*, International Journal of Computer Science and Network Security, **7**, *1*, pp. 244–253 (2007).
2. R. Tahboub, D. Lăzărescu, V. Lăzărescu, *Trusted Next Move Protocol for Secure Automatic Energy Meter Reading Using Mobile Agents*, International Conference "Communications 2006", Bucharest, Romania, June 8–10, 2006.
3. R. Tahboub, V. Lăzărescu, *Novel Approach for Remote Energy Meter Reading Using Mobile Agents,* ITNG, pp. 84–89, Third International Conference on Information Technology: New Generations (ITNG'06), IEEE Computer Society, LV, USA, April 2006.
4. R. Tahboub, D. Lăzărescu, V. Lăzărescu, I.A. Saroit, *Intelligent Secure Management of Electric Power Organizations: Data Collection Using Mobile Agents*, 5th IBIMA, Egypt, Dec, 2005.
5. R. Tahboub, V. Lăzărescu, *Mobile Agents in Remote Energy Meter Reading and Management Systems*, ECAI 2005 – International Conference, Electronics, Computers and Artificial Intelligence, Piteşti, Romania, July 1–2, 2005.
6. Carlos A. Osorio Urzúa, *Bits of Power: The Involvement of Municipal Electric Utilities in Broadband Services*, Massachusetts Institute of Technology, June 2004.
7. J. Ametller, S. Robles, J.A. Ortega-Ruiz, *Self-Protected Mobile Agents*, AAMAS'04, July, 2004.
8. Sergio Loureiro, Refik Molva, Alain Pannetrat, *Secure Data Collection with Updates.*, Electronic Commerce Research Journal, **1**, *2*, 2001.
9. Robert Gray, David Kotz, George Cybenko, Daniela Rus, *Mobile agents: Motivations and state-of-the-art systems*, Thayer School of Engineering / Department of Computer Science-Dartmouth College, Hanover, April 19, 2000.
10. M. Bakhouya, J. Gaber, A. Koukam, *Observations on Client-Server and Mobile Agent Paradigms for Resource Allocation*, Proceedings of the International Parallel and Distributed Processing Symposium IEEE (IPDPS.02), 2002.
11. Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko, D. Rus, *Mobile agents for distributed information retrieval*, in Mathias Klusch (editor), *Intelligent Information Agents*, Chapter 15, pp. 355–395, Springer-Verlag, 1999.
12. Katsuhiro Moizumi. Thayer, *Mobile Agent Planning Problems*, A Thesis Submitted to the Faculty in partial fulfillment of the requirements for the degree of Doctor of Philosophy, School of Engineering Dartmouth College, Hanover, New Hampshire, October 23, 1998.
13. * * *, www.opnet.com.
14. * * *, www.fipa.org.
15. * * *, www.sun.com.