2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing

A Secure JXTA-Overlay Platform for Robot Control

Evjola Spaho*, Keita Matsuo†, Leonard Barolli‡, Joan Arnedo-Moreno§, Fatos Xhafa¶ and Vladi Kolici||

*Graduate School of Engineering, Fukuoka Institute of Technology (FIT)
3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan
E-mail: evjolaspaho@hotmail.com

†Fukuoka Prefectural Kaho-Sogo High School
1117-1 Haji, Keisen-Machi, Kaho-Gun, Fukuoka 820-0607, Japan
E-mail: matuo-k7@fku.ed.jp

‡Department of Information and Communication Engineering, Fukuoka Institute of Technology (FIT)
3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan
E-mail: barolli@fit.ac.jp

§Department of Computer Science, Multimedia and Telecommunication, Open University of Catalonia
Rambla del Poblenou, 156 08018 Barcelona, Spain
E-mail: jarnedo@uoc.edu

¶Department of Languages and Informatics Systems, Technical University of Catalonia
Jordi Girona 1-3, 08034 Barcelona, Spain
E-mail: fatos@lsi.upc.edu

||Department of Electronics and Telecommunication, Polytechnic University of Tirana
Mother Teresa Square, No.4, Tirana, Albania
E-mail: ladi@istitech.net

***Abstract*—Due to the improvement of connections capabilities of end-devices there is an increasing interest to design and implement full featured P2P networks that integrate end-devices. In this paper, we use JXTA-Overlay for the secure control of the robot. We considered the robot as end-device. We investigate the time of secure robot control when in the network were connected different number of peers and compared the results with the scenarios where no security exists at all. The experimental results indicate that in the proposed system security has a cost at the application efficency by adding some overhead in the time of robot control, but JXTA-Overlay can be used successfully to control robots in real time.**

I. Introduction

Today research is focused on the design, implementation and deployment of full featured P2P networks that integrate end devices. P2P platforms are a good approach for e-learning and robot control [1]. Robots are expected to play a key role in the near future. Specifically, humanoid robots will not only be able to socialise with the human-beings but also to replace them in tedious and dangerous tasks [2]. The successfull introduction of robots in human environments will relay on competent and practical systems that are safe and easy to use.

Security is one of the key issues when evaluating a P2P system and a security baseline must be kept in any P2P system in order to ensure some degree of correctness even when some system components will not act properly.

In this paper, we present a secure JXTA-Overlay platform for robot control. This platform is implemented in JAVA language thus, it does not depend on the Operating System (OS). Secure primitives and functions of JXTA-Overlay are used for the secure control of the robot motors. We observed the time of robot control when secure and unsecure primitives are used and for different number of peers connected in the JXTA-Overaly network. All the experiments are done in a 100 Mbps LAN environment. The experimental results show that with the join of other peers in the network, the avarage time of robot control is increased, but the difference time between the secure and unsecure robot control avarage time is nearly the same.

The structure of this paper is as follows. In Section II, we introduce the related work. In section III, we introduce JXTA Technology. Section IV provides a description of JXTA-Overlay. In Section V, we present JXTA-Overlay security. In section VI, we present application of secure JXTA-Overlay for robot control. In Section VII we discuss the experimental result. Finally, conclusions and future work are given in Section VIII.

978-0-7695-4237-9/10 $26.00 © 2010 IEEE
DOI 10.1109/3PGCIC.2010.16

II. Related Work

In this section, we discuss the related work for robot control and security on P2P systems. There are some research works that deal with the autonomous distributed robot systems. In these systems, many autonomous mobile robots cooperate together to carry out complicated tasks [3], [4], [5], [6], [7], [8]. By using many robots in a distributed way, the robot functions can be simplified and the system cost, fault tolerance and flexibility can be improved. In the case, where there are many robots and if one robot has a problem, the other robots can cooperate to finish a requested task. Therefore, the distributed robot systems are better than single robots. However, in distributed robot systems, to realise a common task, the robots should cooperate together and consider not only their own task but also the complete task.

Security is one of the key issues when evaluating a P2P system. Even of the cost of some impact on performance, a security baseline must be kept in any P2P system in order to ensure some degree of correctness even when some system components will not act properly. Some surveys regarding security in P2P systems can be found in [9], [10].

For message security, the JXTA reference implementation [11] provides two mechanisms: TLS (Transport Layer Security) and CBJX [12] (Crypto-Based JXTA Transfer). The former provides private, mutually authenticated, reliable streaming communications, whereas the latter provides lightweight secure message source verification (but not privacy).

JXTA provides its own definition of standard TLS as a transport protocol. The JXTA definition of TLS is composed of two subprotocols: the TLS Record Protocol and the TLS Handshake Protocol. The TLS Record Protocol provides connection security using symmetric cryptography for data encryption.

On the other hand, CBJX is a JXTA-specific security layer which pre-processes messages to provide an additional secure encapsulation, creating a new message that is then relayed to an underlying transport protocol. The original message is signed, and an additional information block is also added to the secured message. This information block contains the source peer credential, both the source and destination addresses, and the source peer ID.

In order to use both mechanisms, TLS and CBJX, a specific group membership service is required: the Personal Security Environment (PSE). The membership service is one of the JXTA core services, taking care of group membership and identity management by providing each group member with a credential. Peers may include credentials in messages exchanged within a group in order to prove membership and provide a means for implementing access control in offered services. However, PSE solely supports on X509 certificates as credentials and Java keystores [13] as a cryptographic module.

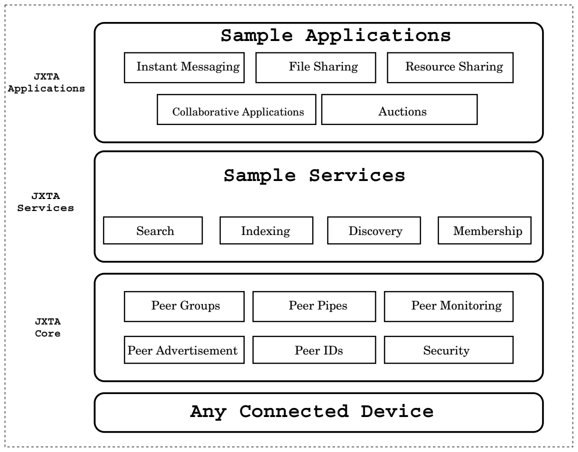


Figure 1. JXTA architecture.

III. JXTA Technology

JXTA technology is an open and innovative collaboration platform. It consist of six protocols that allow different types of peers to communicate and collaborate among them. The main purpose of JXTA is to built P2P systems that offers the basic functions for P2P communication [14]. JXTA realize this issue working and resolving three main problems of the existing P2P networks:

1) OS independence,
2) Language independence,
3) Providing services and infrastructure for P2P applications.

A. JXTA Architecture

JXTA has a structure with three layers, as shown in Fig. 1. The system is designed in a modular way to let the developers to choose the set of services that satisfy their needs [15].

1) **The Core Layer:** This layer implements the essential set of primitives that are common to P2P networking. These primitives includes creation of peers, discovery, transport, peer groups and communication even behind firewalls or NAT. The JXTA core includes also basic security services.
2) **The Services Layer:** This layer implements some services that are integrated to JXTA. These services include searching and indexing, file sharing, protocol translation, authentication and Public Key Infrastructure (PKI) services, as well resource search.
3) **Applications Layer:** This layer implements applications that are integrated to JXTA, such as P2P instant messaging, file sharing and content management.

The distinction between services and final applications may not always be clear, since what a client may consider an application may be considered a service by another peer. For this reason, the system is designed in a modular way, letting developers choose the set of services and applications which most satisfy their needs [16].

B. JXTA Protocols

JXTA comprises a set of six open protocols that allow devices with different software to collaborate. A peer can implement one of these protocols and they can ask the other peers for supplement functionalities. In following we give a brief description of these protocols.

Peer Resolver Protocol (PRP) - Allows a peer to send a search query to another peer. This protocol is a basic communications protocol that follows a request/response format.

Peer Information Protocol (PIP) - Allows a peer to learn about the status of another peer. The information protocol is used partially like ping and partially to obtain basic information about a peer's status.

Pipe Binding Protocol (PBP) - It is used to create a communications path between one or more peers. The protocol is primarily concerned with connecting peers via the route(s) supplied by the peer endpoint protocol.

Rendezvous Protocol (RVP) - The Rendezvous Protocol is responsible for the propagation of the messages This protocol controls how the messages are propagated.

Endpoint Routing Protocol (ERP) - Is used to find available routes to route messages to destination peers. The protocol uses gateways between peers to create a path that consists of one or more pipes.

Peer Discovery Protocol (PDP) - Allows a peer to discover other peer advertisements (peer, group, service, or pipe). This protocol uses a searching mechanism to locate the information.

IV. JXTA-Overlay

JXTA-Overlay is a middle-ware built on top of the JXTA specification, which defines a set of protocols that standardize how different devices may communicate and collaborate among them. It abstracts a new layer on the top of JXTA through a set of primitive operations and services that are commonly used in JXTA-based applications and provides a set of primitives that can be used by other applications, which will be built on top of the overlay, with complete independence. JXTA-Overlay extends JXTA protocols with the goal of overcoming some of its limitations: the need for the developer to manage the presence mechanism, peer group publication and message exchange. To achieve this, JXTA-Overlay provides a set of basic functionalities, primitives, intended to be as complete as possible to satisfy the needs of most JXTA-based applications.

A. Architecture of JXTA-Overlay

The architecture of the JXTA-Overlay middle-ware defines three modules, which let the different entities communicate: the Client Module, the Broker Module and the Control Module. Altogether, they form an abstraction layer on top of JXTA. JXTA-Overlay architecture is shown in Fig. 2.

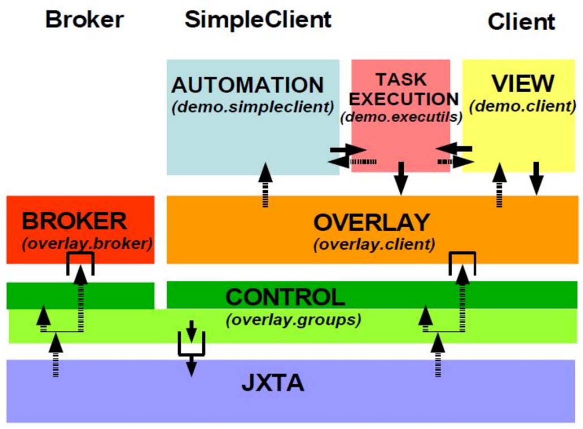


Figure 2. JXTA-Overlay architecture.

- **The Client Module** defines all necessary primitives for peer clients to join a JXTA-Overlay network and interact with other peers and the broker. Primitives are comprised for: peer discovery, peer resources discovery, resource allocation, task submission and execution, file/data sharing discovery and transmission, instant communication, peer group functionalities and, monitoring of peers, groups, and tasks.
- **The Broker Module** defines all the functions that client peers may call upon a broker in order to be granted access to the network, create and publish groups or retrieve other client-peers information.
- **The Control Module** acts as an intermediate layer between the Broker and Client Modules, providing the generic functionalities for group management and messaging.

V. JXTA-Overlay security

Some of the security concerns in JXTA-Overlay are shown in following.

- **No Privacy:** Transmitted data may be easily eavesdropped, since no data privacy is provided. Even though it may be argued that data privacy in message exchanges between end-users is just an optional feature, there are some cases where privacy should not be optional, namely the initial authentication user-name and password. Currently, both fields are sent as plain text. Therefore, any device at broadcast range may read this information with a network protocol analyser (such as Wireshark).
- **Advertisement Spoofing:** Any legitimate user may forge advertisements with no fear of reprisal. No integrity or source authenticity is maintained. False fields, such as the source client peer identifier or any statistics information, may be added into the advertisement, which will be automatically distributed by the broker and accepted by all group members, unaware of the false data it contains [17].
- **Broker Impersonation:** Client peers connect to a self-proclaimed broker, but never check if it is a legitimate

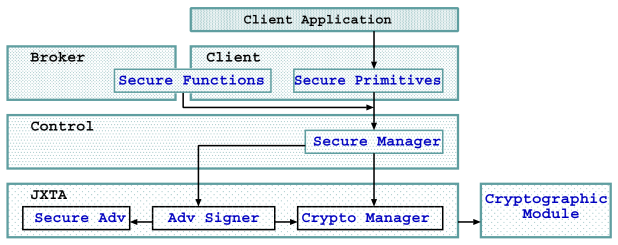


Figure 3. Security architecture for JXTA-Overlay.

one. Even in the case that client peers are connecting to the proper broker address, there are no guarantees that the broker is a legitimate one, since it may be that traffic is being redirected to a fake one via methods such as DNS spoofing.

To solve these problems is used a secure architecture.

A. Secure Architecture Overview

The main concepts of security for JXTA-Overlay are presented in Fig. 3.

The following modules are specified as an extension of JXTA core protocols and are not JXTA-Overlay specific.

- *Crypto Manager* provides an abstraction layer for the cryptographic module and key management method. Crypto Manager is completely modular and accepts different implementations, according to the needs of the specific cryptographic module being used for any particular deployment of JXTA-based application.
- *Secure Adv* defines the secure advertisement format, and provides a method for key transport and additional fields related to its signature.
- *Adv Signer* manages JXTA advertisement signature and validation by interacting with Crypto Manager.

The JXTA-Overlay specific modules are: *Secure Manager* and *Secure Functions and Primitives*.

- *Secure Manager* is the common to all interface additional security capabilities within the JXTA-Overlay Control Module, operating as a single entry point for all secure services.
- *Secure Functions and Primitives* modules just extend the base Broker and Client Modules, providing a set of additional primitives and functions which take into account security considerations.

B. Secure Messenger Primitives

JXTA-Overlay has 122 primitives and 84 events related to different functions. We will focus on *messenger primitives* which send simple messages between end-users. Messenger primitives define how to directly exchange simple text messages between end-users. For instance, in Robot Control, commands are sent using these primitives. Once the cryptographic setup has established key and credential information for all participating entities, it is finally possible to secure data exchanges by providing a secure version of each primitive. The two main messenger primitives are:

1) *sendMsgPeer*: Sends a simple message to some other client peers.
2) *sendMsgPeerGroup*: Sends a simple message to all members of a group. It is actually resolved by iteratively calling *sendMsgPeer*.

The secure versions of both primitives provide lightweight privacy, data integrity and message source authentication in a stateless, best effort method. This is in contrast, for example, with JXTA's secure pipes, which rely on TLS and require some previous negotiation between endpoints each time a message exchange is initiated [18]. The necessary steps for some end-user connected to client peer A to send a simple text message to another one connected to B are as follows.

1) A retrieves B's Pipe Advertisement. This step also happens in the original, insecure primitive, and thus means no additional burden.
2) A validates the advertisement signature in order to ensure that it has not been compromised. If the signature does not validate, the advertisement has been tampered, and is deemed invalid. If the message is sent, no guarantees can be made on regards to its security.
3) A retrieves PK_B from the signed advertisement's enclosed credential, $C^B_{red_{Br}}$.
4) $A \longrightarrow B : \{E_{PK_B}(m, S_{Sk_A}(m))\}$.
5) B decrypts the message using SK_B.
6) B retrieves A Pipe Advertisement and repeats steps 2, 3.
7) B validates the message signature using PK_A obtained via B's signed advertisement.

The *secureMsgPeerGroup* primitive just iteratively uses the *secureMsgPeer* to send the same message to a group of peers. As a standard message exchange between end-user endpoints, these primitives are mainly subject to data eavesdropping and end-user impersonation. The former is avoided by encrypting data using a wrapped key approach, whereas the latter, as well as data integrity, is provided by digitally signing the data. In both cases, each other endpoint's public key is necessary in steps 1-2 (encryption) and 6 (signature validation). However, both public keys are contained in the credential provided by the Broker in the secureLogin primitive call, and thus, can be considered that belongs to a legitimate owner, and not an attacker.

VI. Application of JXTA-Overlay for Secure Robot Control

For evaluation of our system, we constructed a network with three peer nodes as shown in Fig. 4. The robot is connected with the peer via RS232C connector, which is a legal interface and many devices have implemented it.

For the experiments is used KHR-1 robot with the following specifications: Heigh 340 mm, Width 180 mm, depth 80 mm and Weight 1.2 kg. We considered the robot as end-device. Peer nodes specifications are shown in Table I.

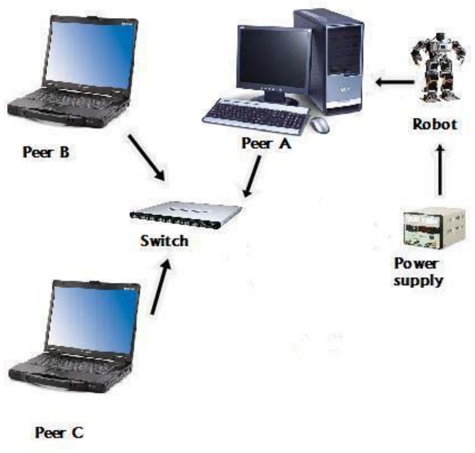
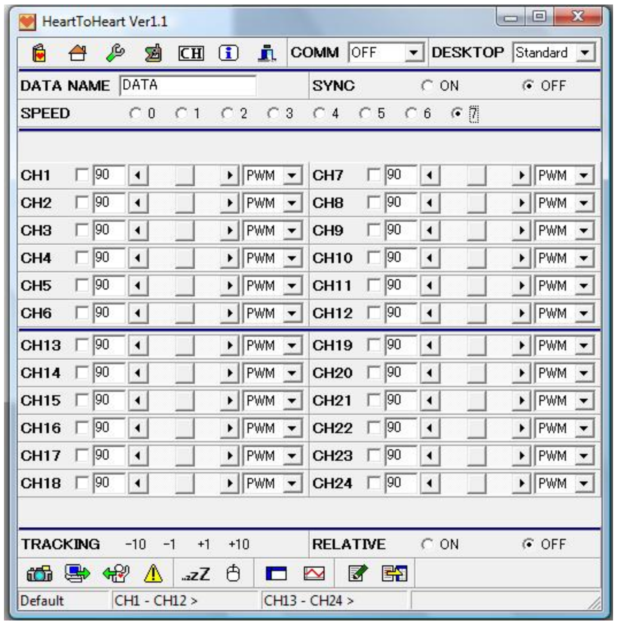


Figure 4. P2P robot control system.

Table I
PEER NODE SPECIFICATIONS.

Peer Nodes	Characteristics
Broker	AMD Athlon (tm) 3200+2.01 GHz, 100 GB RAM
Peer 1	Pentium M Centrino 1.2 GHz, 512 MB RAM
Peer 2	Intel Centrino Duo 1.3 GHz, 2GB RAM
Connection	100 Mbps


HeartToHeart Ver1.1

COMM OFF DESKTOP Standard

DATA NAME DATA SYNC ON [✓ OFF]

SPEED 0 1 2 3 4 5 6 [✓ 7]

CH1	90	PWM	CH7	90	PWM
CH2	90	PWM	CH8	90	PWM
CH3	90	PWM	CH9	90	PWM
CH4	90	PWM	CH10	90	PWM
CH5	90	PWM	CH11	90	PWM
CH6	90	PWM	CH12	90	PWM
CH13	90	PWM	CH19	90	PWM
CH14	90	PWM	CH20	90	PWM
CH15	90	PWM	CH21	90	PWM
CH16	90	PWM	CH22	90	PWM
CH17	90	PWM	CH23	90	PWM
CH18	90	PWM	CH24	90	PWM

TRACKING -10 -1 +1 +10 RELATIVE ON [✓ OFF]

Default CH1 - CH12 > CH13 - CH24 >

Figure 5. Snapshot of HeartToHeart interface used for robot control.

To create the motions, we used HeartToHeart (motion editor for RCB-1). A snapshot of this interface is shown in Fig. 5.

Several experiments are done with different number of peers connected to the network. During the experiments are observed only the messenger primitives because the

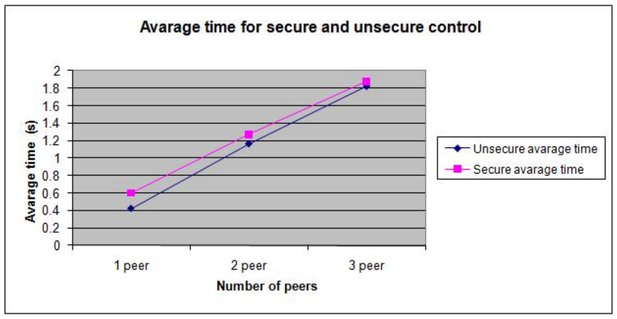


Figure 6. Avarage time for secure and unsecure robot control.

login and connect primitives just happens once for the full session. The command of the robot control is sent as simple message, using chat service. We experimentally studied the time for Robot Control for two different cases:

- Using *sendMsgPeer* primitive.
- Using *secureMsgPeer* primitive.

The messenger primitives define how to directly exchange simple text messages between end-users, such as a chat service.

After the synchronisation of the time between peer nodes, the time of robot control is measured using both primitives. First the experiments are done with only one peer in the network. Then, other scenarios were realised adding two other peers to the network and the times of robot control were measured again.

VII. EXPERIMENTAL RESULTS

In Fig. 6 is shown the average time for robot control when using and not using security. As can be observed, the difference between the average time of unsecure robot control and secure robot control is small and with increase of number of peers the difference time decreases.

The average time for robot control when in the network is only one peer is 0.42 s for unsecure and 0.6 for secure primitive and for two peers 1.17 s and 1.27 s, respectively. When in the network are three peers the time of robot control using unsecure primitive is 1.82 s and using secure primitive is 1.88 s.

When another peer joins the network, the time for robot control for both primitives increases. The join of other peers in the network increase the number of requests that the client primitives send to the broker. In this situation, the broker should manage all requests of the client primitives, and this cause an overhead in the time of robot control.

From this results we can conclude that the primitives of JXTA-Overlay *sendMsgPeer* and *secureMsgPeer* can be successfully used for robot control. We applied secured JXTA-Overlay architecture for the control of KHR-1 robot. In Fig. 7 from the left to the right, we show the scenario when the robot moves the left hand. In Fig. 8 from the left to the right, we show the steps of robot during walking. The experimental results show that the implemented JXTA-Overlay platform controls the robot in a smoothly way.

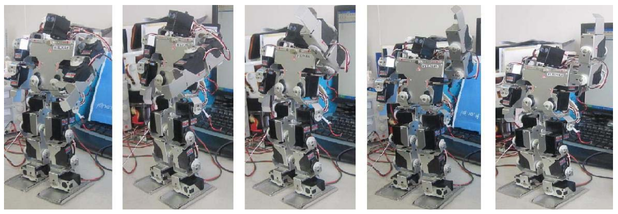


Figure 7. Robot hand movement.

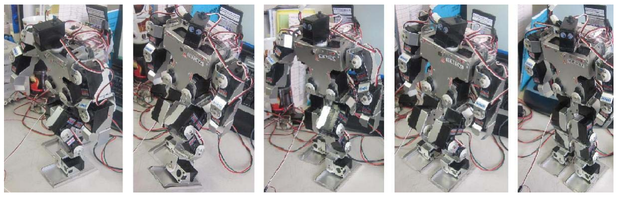


Figure 8. Robot walking.

VIII. Conclusions and Future Work

In this paper is presented the application of JXTA-Overlay P2P system for robot control. Specifically, we have experimentally measured the time for robot control, using two primitives: *sendMsgPeer* and *secureMsgPeer*. The objective was to see the performance as well as limitations of the P2P robot control when it is used as an end-device and the cost of using security in robot control.

The experimental results have shown that the join of other peers in the network increase the number of requests that the client primitives send to the broker. In this situation, the broker should manage all requests of the client primitives, and this cause an overhead in the time of robot control. But, with the increasing of the number of peers in the network, the difference between the secure and unsecure average time for robot control is nearly the same. From the experimental results, we conclude that the primitives of JXTA-Overlay *sendMsgPeer* and *secureMsgPeer* can be successfully used to control the robot in a smoothly way.

In the future, we want to implement new secure functions and primitives for JXTA-Overlay, which can offer several improvements of the existing JXTA-Overlay protocols and services and increase the reliability of JXTA-based distribution applications and support group management of file sharing.

Acknowledgement

The authors would like to thank Japanese Society for the Promotion of Science (JSPS) for supporting this work.

References

[1] F. Xhafa, R. Fernandez, T. Daradoumis, L. Barolli, S. Caballe, ”Improvement of JXTA Protocols for Supporting Reliable Distributed Applications in P2P Systems”, Proc. of NBiS-2007 (Regensburg, Germany), LNCS 4658, pp. 345-354, September 2007.

[2] P. Caloud, W. Choi, J. C. Latombe, C. L. Pape, and M. Yim, ”Indoor Automation with Many Mobile Robots”, Proc. of the IEEE International Workshop on Intelligent Robots and Systems, pp. 67-72, July 1990.

[3] H. Asama, K. Ozaki, A. Matsumoto, Y. Ishida, and I. Endo, ”Development of Task Assignment System Using Communication for Multiple Autonomous Robots”, Journal of Robotics and Mechatronics, pp. 122-127, 1992.

[4] L. Chaimowicz, T. Sugar, V. Kumar, and M.F.M. Campos, ”An Architecture for Tightly Coupled Multi-robot Cooperation”, Proc. of IEEE International Conference on Robotics and Automation, pp. 2992-2997, 2001.

[5] L. Inoue, and T. Nakajima, ”Cooperative Object Transportation by Multiple Robots with Their Own Objective Tasks”, Journal of the Robotics Society of Japan, pp. 888-896, 2001.

[6] K. Ozaki, H. Asama, Y. Ishida, A. Matsumoto, I. Endo, ”Collision Avoidance Using Communication Between Autonomous Mobile Robot”, Journal of the Robotics Society of Japan, pp. 961-967, 1996.

[7] M. Parnichkun, S. Ozono, ”CDCSMA-CD Communication Method for Cooperative Robot Systems”, Advanced Robotics, pp. 669-694, 1998.

[8] P. E. Rybski, S. A. Stoeter, M. Gini, D. F. Hougen, and N. P. Papanikolopoulos, ”Performance of a Distributed Robotic System Using Shared Communications Channels”, IEEE Transactions on Robotics and Automation, pp. 713-727, 2002.

[9] S.Wallach, ”A Survey of Peer-to-Peer Security Issues”, Theories and Systems, Springer, pp. 253-258, 2003.

[10] V. Vroonhoven, ”Peer-to-Peer Security”, Proc. of the 4-th Twente Student Conference on IT, 2006.

[11] JXTA 2.5 RC1, http://download.java.net/jxta/build, June 2007.

[12] D. Bailly, ”CBJX: Crypto-based JXTA (An Internship Report)”, July 2002.

[13] CCITT, ”The Directory Authentication Framework. Recommendation”, 1988.

[14] D. Brookshier, D. Govoni, N. Krishnan, J. C. Soto, ”JXTA: Java P2P Programming”, Sams Publishing, 2002.

[15] SUN Microsystem, ”Project JXTA V2.0: Java Programmer’s Guide”, May 2003.

[16] SUN Microsystem, ”Project JXTA”, http://www.jxta.org, 2001.

[17] D. Sax, ”DNS Spoofing (Malicious Cache Poisoning)”, http://www.sans.org, 2003.

[18] T. Dierks, C. Allen, ”The TLS Protocol Version 1.0”, http://www.ietf.org/rfc/rfc2246.txt, September 1999.